Lecture Notes in Computer Science 2247
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Tokyo

C. Pandu Rangan Cunsheng Ding (Eds.)

Progress in Cryptology – INDOCRYPT 2001

Second International Conference on Cryptology in India
Chennai, India, December 16-20, 2001
Proceedings

Springer

Volume Editors

C. Pandu Rangan
Indian Institute of Technology, Madras
Department of Computer Science and Engineering
Chennai, India
E-mail: rangan@iitm.ernet.in

Cunsheng Ding
Hong Kong University of Science and Technology
Department of Computer Science
Hong Kong
E-mail: cding@cs.ust.hk

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Progress in cryptology : proceedings / INDOCRYPT 2001, Second International
Conference on Cryptology in India, Chennai, India, December 16 - 20, 2001.
C. Pandu Rangan ; Cunsheng Ding (ed.). - Berlin ; Heidelberg ; New York ;
Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2247)
ISBN 3-540-43010-5

CR Subject Classification (1998): E.3, G.2.1, D.4.6, K.6.5, F.2.1-2, C.2

ISSN 0302-9743
ISBN 3-540-43010-5 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by Steingräber Satztechnik GmbH Heidelberg
Printed on acid-free paper SPIN 10845931 06/3142 5 4 3 2 1 0

Preface

INDOCRYPT 2001, the Second Annual Crypto Conference, is proof of the significant amount of enthusiasm generated among Indian as well as International crypto communities. INDOCRYPT 2001 was organized by the Indian Institute of Technology, Madras and the Institute of Mathematical Sciences, also located in Madras (now Chennai). This event was enthusiastically co-sponsored by eAlcatraz Consulting Private Ltd, Chennai, Odyssey Technologies Ltd, Chennai, and Shanmuga Arts Science Technology and Research Academy (SASTRA), Thanjavur. The Program Committee Co-chair, Prof.C.Pandu Rangan was responsible for local organization and registration.

The Program Committee considered 77 papers and selected 31 papers for presentation. These papers were selected on the basis of perceived originality, quality, and relevance to the field of cryptography. The proceedings include the revised version of the accepted papers. Revisions were not checked as to their contents and authors bear full responsibility for the contents of their submissions.

The selection of papers is a very challenging and demanding task. We wish to thank the Program Committee members who did an excellent job in reviewing the submissions in spite of severe time constraints imposed by the tight processing schedule. Each submission was reviewed by at least three referees (only a few by two). The Program Committee was ably assisted by a large number of reviewers in their area of expertise. The list of reviewers has been provided separately. Our thanks go to all of them.

The conference program included three invited lectures by Prof. Andrew Klapper, University of Kentucky, USA, Dr. Anne Canteaut, INRIA, France, and Dr. Tatsuaki Okamoto, NTT Labs, Japan. In addition to these three invited lectures, pre-conference and post-conference tutorials were conducted by Ramarathnam Venkatesan, Microsoft, Redmond, USA on Random Number Generators: Theory and Practice and by Dipankar Dasgupta, The University of Memphis, USA on a Bio-Inspired Approach to Computer Security. Industrial presentations on the best practices were also scheduled during these days.

Our sincere thanks goes to Springer-Verlag, in particular to Mr. Alfred Hofmann, for publishing the proceedings of INDOCRYPT 2001 as a volume in their prestigious LNCS series. We are also indebted to Prof. Bimal Roy and Prof. C.E.Veni Madhavan and to all the members of the Steering Committee for their valuable advice and suggestions. We gratefully acknowledge the financial support extended by our co-sponsors and 'Golden' sponsors. We wish to make a special mention of the enthusiastic financial support extended by IIT Madras Alumni Association in North America, (IITMAANA) enabling a large number of students and faculty members from various universities in India to attend the conference.

This conference handled all the submissions as well as refereeing in electronic form. The ERNET centre located at IIT Madras, coordinated by Prof. S.V. Raghavan, provided excellent internet services at every stage of this conference.

We wish to place on record our sincere thanks to Prof. R. Natarajan, Director, IIT, Madras, Prof. C.R. Muthukrishnan, Deputy Director, IIT, Madras and Prof. Srinivasa Murthy, Dean, IC&SR, IIT, Madras for encouraging and supporting the conference in every possible way.

Finally we wish to thank all the authors who submitted papers, making this conference possible, and the authors of accpeted papers for updating their papers in a timely fashion, making the production of these proceedings possible.

December 2001

Pandu Rangan C
Cunsheng Ding

INDOCRYPT 2001

December 16-20, 2001, Indian Institute of Technology,
Madras, India

Organized by

Indian Institute of Technology, Madras, India
The Institute of Mathematical Sciences, Chennai, India

Co-sponsored by

eAlcatraz Consulting Private Ltd, Chennai
Odyssey Technologies Ltd, Chennai
Shanmuga Arts Science Technology and Research Academy (SASTRA),
Thanjavur

General Chair

Balasubramaniam R	The Institute of Mathematical Sciences, India

Program Co-chairs

Pandu Rangan C	Indian Institute of Technology, Madras, India
Cunsheng Ding	Hong Kong University of Science, Hong Kong

Steering Committee

Balakrishnan N	Indian Institute of Science, Bangalore, India
Balasubramaniam R	The Institute of Mathematical Sciences, India
Bimal Roy	Indian Statistical Institute, Calcutta, India
Gulati V P	IDRBT, Hyderabad, India
Kapil H Paranjape J	The Institute of Mathematical Sciences, India
Karandikar R L	Indian Statistical Institute, Delhi, India
Manindar Agrawal	Indian Institute of Technology, Kanpur, India
Palash Sarkar	University of Waterloo, Canada
Pandu Rangan C	Indian Institute of Technology, Madras
Saxena P K	SAG, New Delhi, India
Sitaram N	CAIR, Bangalore, India
Vidyasagar M	Tata Consultancy Services, Hyderabad, India

Program Committee

Alfred John Menezes	University of Waterloo, Canada
Arjen K Lenstra	Citibank, USA
Balasubramaniam R	The Institute of Mathematical Sciences, India
Bimal Roy	Indian Statistical Institute, Calcutta, India
Claude Carlet	University of Caen, France

Cunsheng Ding	Hong Kong University of Science, Hong Kong
Dingyi Pei	Academia Sinica, China
Eiji Okamoto	University of Wisconsin, USA
Harald Niederreiter	National University of Singapore, Singapore
Jennifer Seberry	University of Wollongong, Australia
Kwangjo Kim	Information and Communications University, Korea
Lam Kwok Yan	National University of Singapore, Singapore
Neal Koblitz	University of Washington, USA
Palash Sarkar	University of Waterloo, Canada
Pandu Rangan C	Indian Institute of Technology, Madras, India
Rei Safavi-Naini	University of Wollongong, Australia
Thomas Johansson	Lund University, Sweden
Tom Berson	Anagram Laboratories, USA
Tsutomu Matsumoto	Japan
Veni Madhavan C E	SAG, India

Organizing Committee

Boopal E	Indian Institute of Technology, Madras, India
Kamakoti V	Indian Institute of Technology, Madras, India
Veeraraghavan V	Indian Institute of Technology, Madras, India

List of External Reviewers

Alfred John Menezes
Amr Youssef
Andreas Westfeld
Antoine Valembois
Arash Reyhani-Masoleh
Arjen K Lenstra
Ashwin Kumar M V N
Bedi S S
Berry Schoenmakers
Bhatiga A K
Bimal Roy
Byoungcheon Lee
Caroline Fontaine
Claude Carlet
Cunsheng Ding
Ding Yi Pei
Eiji Okamoto
Enes Pasalic
Eric Filiol
Eugene P Xavier
Evelyne Lutton
Fredrik Jonsson

Gambhir R K
Guillaume Poupard
Harald Niederreiter
Huapeng Wu
Indivar Gupta
Kaisa Nyberg
Khanna R K
Kishan C. Gupta
Kwangjo Kim
Laxmi Narayan
Marc Girault
Martijn Stam
Masahiro Mambo
Meena Kumari
Miodrag Mihaljevic
Neal Koblitz
Nicolas Sendrier
Pabitra Pali Chowdhury
Palash Sarkar
Pandu Rangan C
Paul J. Schellenberg
Pierrick Gaudry

Prabhu B
Pranava Raja Goundan
Pratibha Yadav
Raghavan S V
Rana Barua
Reihanah Safavi-Naini
Samik Sengupta
Sandeepan Chowdhury
Sanjeev K. Mishra
Sarbani Palit
Sexena P K
Sikdar K
Srinathan K
Srivastava M C
Subhamoy Maitra
Supratik Mukhopadhyay
Tharani Rajan
Thomas Johansson
Veni Madhavan C E

Sponsors

Arunai Charitable Trust, Tiruvannamalai
Cyberspace, Chennai
Dharma Naidu Educational and Charitable Trust, Chennai
HSMAK Charitable Trust, Gulbarga
Jai Barath Charitable Trust, Vaniyambadi
Jaya Educational Trust, Chennai
IITMAANA, USA
Lalitha Educational Trust, Hyderabad
Mauritius Research Council, Mauritius
MESCO, Hyderabad
Microsoft Corporation India Pvt. Ltd, Chennai
Nalini Suresh, Chennai
Ponniamman Educational Society, Chennai
Prince Venkateswara Education Society, Chennai
Rajalakshmi Educational Trust, Chennai
Sapthagiri Engineering College, Dharmapuri
Satyabama Institute of Science and Technology (Deemed University)
Sri Nandanam Educational and Social Welfare Trust, Thiruppathur
SUN Microsystem, India
Vasista Education Soceity, Narsapur, AP
Velammal Engineering College, Chennai

Table of Contents

Invited Lecture

Hashing

Algebraic Schemes

Elliptic Curves

Coding Theory

Applications – I

Cryptanalysis

Distributed Cryptography

Boolean Functions

Digital Signatures

Shift Registers

Applications – II

Cryptographic Functions and Design Criteria for Block Ciphers

Anne Canteaut

INRIA – projet CODES,
BP 105, 78153 Le Chesnay, France
Anne.Canteaut@inria.fr

Abstract. Most last-round attacks on iterated block ciphers provide some design criteria for the round function. Here, we focus on the links between the underlying properties. Most notably, we investigate the relations between the functions which oppose a high resistance to linear cryptanalysis and to differential cryptanalysis.

1 Introduction

The development of cryptanalysis in the last ten years has led to the definition of some design criteria for block ciphers. These criteria correspond to some mathematical properties of the round function which is used in an iterated block cipher. They essentially concern the confusion part of the round function, usually named S-box. Most notably, the use of a highly nonlinear round function ensures a high resistance to linear attacks. Similarly, the resistance to differential attacks is related to some properties of the derivatives of the round function. The functions which are optimal regarding these criteria are respectively called almost bent and almost perfect nonlinear. For instance, such functions are used in the block cipher MISTY [26]. However, these functions present some particular properties which may introduce other weaknesses in the cipher (e.g. see [17]).

This paper describes the link between the design criteria related to differential attacks, linear attacks and higher order differential attacks. We provide some tools for establishing a general relationship between the nonlinearity of a function and its resistance to differential attacks. Most notably, we give a characterization of almost bent functions using some divisibility property of their Walsh coefficients. We also show that this structure is specific of optimal functions. Most results in this paper rely on a joined work with P. Charpin and H. Dobbertin [6,4,5].

The following section reviews the design criteria associated to some classical last-round attacks. Section 3 focuses on the functions which ensure the best resistance to differential attacks, to linear attacks and to higher order differential attacks. We show in Section 4 that these optimal functions are related to other optimal objects which appear in different areas of telecommunications. For example, almost bent functions correspond to particular error-correcting codes and to pairs of m-sequences with preferred crosscorrelation. Section 5 presents the links between the previous design criteria, especially for the case of optimal functions.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 1–16, 2001.

2 Last-Round Attacks on Iterated Block Ciphers

In an iterated block cipher, the ciphertext is obtained by iteratively applying a keyed round function F to the plaintext. In an r-round iterated cipher, we have

$$x_i = F(x_{i-1}, K_i) \text{ for } 1 \leq i \leq r ,$$

where x_0 is the plaintext, x_r is the ciphertext and the r-round keys $(K_1, \ldots, K_r)$ are usually derived from a unique secret key by a key schedule algorithm. For any fixed round key K, the round function $F_K : x \to F(x, K)$ is a permutation of the set of n-bit vectors, $\mathbb{F}_2^n$, where n is the block size.

Most attacks on iterated block ciphers consist in recovering the last round key K_r from the knowledge of some pairs of plaintexts and ciphertexts. For this purpose, we consider the *reduced cipher*, i.e., the cipher obtained by removing the final round of the original cipher. The reduced cipher corresponds to the function $G = F_{K_{r-1}} \circ \ldots \circ F_{K_1}$. The key point in a last-round attack is to be able to distinguish the reduced cipher from a random permutation for all round keys $K_1, \ldots, K_{r-1}$. If such a *discriminator* can be found, some information on K_r can be recovered by checking whether, for a given value k_r, the function

$$x_0 \to F_{k_r}^{-1}(x_r)$$

satisfies this property or not, where x_0 (resp. x_r) denotes the plaintext (resp. the ciphertext). The values of k_r for which the expected statistical bias is observed are candidates for the correct last-round key.

Different discriminators can be exploited. Most notably, a last-round attack can be performed when the reduced cipher satisfies one of the following properties:

- The reduced cipher G has a derivative, $D_aG : x \to G(x + a) + G(x)$, which is not uniformly distributed. This discriminator leads to a differential attack [1];
- There exists a linear combination of the n output bits of the reduced cipher which is close to an affine function. This leads to a linear attack [24,25];
- The reduced cipher has a constant k-th derivative for a small k. This leads to a higher order differential attack [20];
- The reduced cipher, seen as a univariate polynomial in $\mathbb{F}_{2^n}[X]$, is close to a low-degree polynomial. This leads to an interpolation attack [17] or to an improved version using Sudan's algorithm [16].

In most cases, such a property on the reduced cipher can be detected only if the round function presents a similar weakness. Therefore, a necessary condition for an iterated cipher to resist these attacks is to use a round function which does not present any of the previous characteristics. Then, the round function should satisfy the following properties for any round key K:

(i) For any $a \in \mathbb{F}_2^n$, $a \neq 0$, the output distribution of $D_aF_K : x \to F_K(x + a) + F_K(x)$ should be close to the uniform distribution;

(ii) For any $a \quad \mathbb{F}_2^n$, $a = 0$, the Boolean function $x \to a \cdot F_K(x)$ should be far away from all affine functions;
(iii) The Boolean functions $x \to a \cdot F_K(x)$ should have a high degree;
(iv) The function F_K, seen as a univariate polynomial in $\mathbb{F}_{2^n}[X]$, should be far away from all low-degree polynomials.

Some of these conditions may be sufficient in particular cases to guarantee that the iterated cipher resists the corresponding attack (e.g. see [31]).

Note that the first three properties are invariant under both right and left composition by a linear permutation of $\mathbb{F}_2^n$. Then, they only concern the *confusion part* of the round function. In the following, we only investigate the first three properties, since the mathematical nature of the last criterion is quite different.

3 Almost Perfect Round Functions

A *Boolean function f of n variables* is a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2$. It can be expressed as a polynomial in $x_1, \dots, x_n$, called its *algebraic normal form*. The *degree* of f, denoted by $deg(f)$, is the degree of its algebraic normal form.

3.1 Resistance against Differential Attacks

The resistance of an iterated cipher with round function F_K against differential cryptanalysis can be quantified by some properties of the derivatives (or differentials) of F_K.

Definition 1. *[22] Let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$. For any $a \quad \mathbb{F}_2^n$, the* derivative of F with respect to *a is the function*

$$D_aF(x) = F(x + a) + F(x) .$$

For any k-dimensional subspace V of $\mathbb{F}_2^n$, the k-th derivative of F with respect to *V is the function*

$$D_VF = D_{a_1}D_{a_2}\dots D_{a_k}F ,$$

where $(a_1, \dots, a_k)$ is any basis of V.

It is clear that an iterated cipher is vulnerable to a differential attack if there exists two nonzero elements a and b in $\mathbb{F}_2^n$ such that, for any round key K, the number of $x \quad \mathbb{F}_2^n$ satisfying

$$F_K(x + a) + F_K(x) = b \tag{1}$$

is high. Therefore, a necessary security condition is that, for any K,

$$\delta_{F_K} = \max_{a,b=0} \#\{x \quad \mathbb{F}_2^n,\ F_K(x + a) + F_K(x) = b\}$$

should be small. It clearly appears that the number of solutions of Equation (1) is even (because x_0 is a solution if and only if $x_0 + a$ is a solution). Then, we deduce

Proposition 1. *[31] For any function F from F_2^n into F_2^n, we have*

$$\delta_F \geq 2 .$$

In case of equality, F is said to be almost perfect nonlinear (APN).

Note that the terminology APN comes from the general bound

$$\delta_F \geq 2^{n-m}$$

for a function from F_2^n into F_2^m, where the functions achieving this bound are called *perfect nonlinear functions* [28]. Such functions only exist when n is even and $n \geq 2m$ [29].

The definition of APN functions can be expressed in terms of second derivatives:

Proposition 2. *A function F from F_2^n into F_2^n is APN if and only if, for any nonzero elements a and b in F_2^n, with $a = b$, we have*

$$D_a D_b F(x) = 0 \text{ for all } x \quad \mathrm{F}_2^n .$$

All known APN functions are functions of an odd number of variables. Actually, it is conjectured that, for any function F from F_2^n into F_2^n with n even, we have

$$\delta_F \geq 4 .$$

This statement is proved for some particular cases, most notably for power functions [2,10].

3.2 Resistance against Linear Attacks

The resistance against linear attacks involves the Walsh spectrum of the round function.

In the following, the usual dot product between two vectors x and y is denoted by $x \cdot y$. For any $\alpha \quad \mathrm{F}_2^n$, ϕ_α is the linear function of n variables: $x \to \alpha \cdot x$. For any Boolean function f of n variables, we denote by $F(f)$ the following value related to the Walsh (or Fourier) transform of f:

$$F(f) = \underset{x \quad \mathbf{F}_2^n}{□} (-1)^{f(x)} = 2^n - 2wt(f) ,$$

where $wt(f)$ is the Hamming weight of f, i.e., the number of $x \quad \mathrm{F}_2^n$ such that $f(x) = 1$.

Definition 2. *The* Walsh spectrum *of a Boolean function f of n variables f is the multiset*

$$\{F(f + \phi_\alpha), \alpha \quad \mathrm{F}_2^n\} .$$

The Walsh spectrum *of a vectorial function F from F_2^n into F_2^n consists of the Walsh spectra of all Boolean functions $\phi_\alpha \circ F : x \to \alpha \cdot F(x)$, $\alpha = 0$. Therefore, it corresponds to the multiset*

$$\{F(\phi_\alpha \circ F + \phi_\beta), \ \alpha \quad \mathrm{F}_2^n \setminus \{0\}, \beta \quad \mathrm{F}_2^n\} .$$

The security criterion corresponding to linear cryptanalysis is that all functions $\phi_\alpha \circ F_K$, $\alpha = 0$ should be far away from all affine functions. This requirement is related to the nonlinearity of the functions F_K.

Definition 3. *The* nonlinearity *of a function F from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ is the Hamming distance between all $\phi_\alpha \circ F$, $\alpha \quad \mathbf{F}_2^n$, $\alpha = 0$, and the set of affine functions. It is given by*

$$2^{n-1} - \frac{1}{2}L(F) \quad \textit{where} \quad L(F) = \max_{\alpha \quad \mathbf{F}_2^n} \max_{\beta \quad \mathbf{F}_2^n} |\mathcal{F}(\phi_\alpha \circ F + \phi_\beta)| \; .$$

Proposition 3. *[33,9] For any function $F : \mathbf{F}_2^n \to \mathbf{F}_2^n$,*

$$L(F) \geq 2^{\frac{n+1}{2}} \; .$$

In case of equality F is called almost bent (AB).

For a function F from $\mathbf{F}_2^n$ into $\mathbf{F}_2^m$, we have

$$L(F) \geq 2^{\frac{n}{2}}$$

where the functions achieving this bound are called *bent functions*. It was proved that a function is bent if and only if it is perfect nonlinear [28,29].

The minimum value of $L(F)$ where F is a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ can only be achieved when n is odd. For even n, some functions with $L(F) = 2^{\frac{n}{2}+1}$ are known and it is conjectured that this value is the minimum [32,12].

3.3 Resistance against Higher Order Differential Attacks

In a higher order differential attack, the attacker exploits the existence of a k-dimensional subspace $V \quad \mathbf{F}_2^n$ such that the reduced cipher G satisfies

$$D_V G(x) = c \quad \text{for all } x \quad \mathbf{F}_2^n$$

where c is a constant which does not depend on the round keys $K_1, \ldots K_{r-1}$. A natural candidate for V arises when the degree of the reduced cipher is known.

Definition 4. *The* degree *of a function F from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ is the maximum degree of its Boolean components:*

$$deg(F) = \max_{1 \leq i \leq n} deg(\phi_{e_i} \circ F)$$

where $(e_1, \ldots, e_n)$ denotes the canonical basis of $\mathbf{F}_2^n$.

Actually, we have

Proposition 4. *[22] Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ of degree d. Then, for any $(d+1)$-dimensional subspace $V \quad \mathbf{F}_2^n$, we have*

$$D_V F(x) = 0 \quad \textit{for all } x \quad \mathbf{F}_2^n \; .$$

Note that the dimension of the smallest subspace V satisfying $D_V F = 0$ may be smaller than $\deg(F) + 1$.

4 Related Objects

The results concerning almost perfect functions widely apply in several areas of telecommunications: almost perfect nonlinear and almost bent functions are related to metric properties of some linear codes, especially of binary cyclic codes with two zeros. Almost bent power functions also correspond to pairs of maximum-length sequences with preferred crosscorrelation.

4.1 Links with Error-Correcting Codes

Carlet, Charpin and Zinoviev have pointed out that both APN and AB properties can be expressed in terms of error-correcting codes [8].

Since both APN and AB properties are invariant under translation, we here only consider the functions F such that $F(0,\dots,0) = 0$. We use standard notation of the algebraic coding theory (see [23]). Any k-dimensional subspace of $\mathbf{F}_2^n$ is called a binary linear code of length n and dimension k and is denoted by $[n,k]$. Any $[n,k]$-linear code C is associated with its dual $[n,n-k]$-code, denoted by $C^\perp$:

$$C^\perp = \{x \in \mathbf{F}_2^n,\ x \cdot c = 0\ \ \forall c \in C\}\ .$$

Any $k \times n$ binary matrix G defines an $[n,k]$-binary linear code C:

$$C = \{xG, x \in \mathbf{F}_2^k\}$$

We then say that G is a generator matrix of C.

Let $(\alpha_i, 1 \le i \le 2^n)$ denote the set of all nonzero elements of $\mathbf{F}_2^n$. We consider the linear binary code C_F of length $(2^n - 1)$ and dimension $2n$ defined by the generator matrix

$$G_F = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \dots & \alpha_{2^n} \\ F(\alpha_1) & F(\alpha_2) & F(\alpha_3) & \dots & F(\alpha_{2^n}) \end{pmatrix}, \qquad (2)$$

where each entry in $\mathbf{F}_2^n$ is viewed as a binary column vector of length n. It clearly appears that any codeword in C_F corresponds to a vector $(a \cdot \alpha_i + b \cdot F(\alpha_i), 1 \le i \le 2^n)$. Therefore, its Hamming weight is given by

$$\#\{i, 1 \le i \le 2^n,\ a \cdot \alpha_i + b \cdot F(\alpha_i) = 1\} = 2^{n-1} - \frac{1}{2}F(\phi_b \circ F + \phi_a)\ .$$

Moreover, a vector $(c_1,\dots,c_{2^n})$ belongs to the dual code $C_F^\perp$ if and only if

$$\sum_{i=1}^{2^n} c_i\alpha_i = 0 \text{ and } \sum_{i=1}^{2^n} c_iF(\alpha_i) = 0\ .$$

Then, we obviously have that the minimum distance of $C_F^\perp$ is at least 3. Moreover, there exist three different indexes i_1, i_2, i_3 such that

$$F(\alpha_{i_1}) + F(\alpha_{i_2}) + F(\alpha_{i_3}) + F(\alpha_{i_1} + \alpha_{i_2} + \alpha_{i_3}) = 0$$

if and only if $C_F^\perp$ contains a codeword of Hamming weight 4 (or 3 if $\alpha_{i_1} + \alpha_{i_2} + \alpha_{i_3} = 0$).

Therefore, we obtain the following correspondence:

Theorem 1. *[8] Let F be a permutation from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$ with $F(0) = 0$. Let $\mathcal{C}_F$ be the linear binary code of length $2^n - 1$ and dimension $2n$ with generator matrix G_F described by (2). Then,*

(i)

$$L(F) = \max_{c \in \mathcal{C}_F, c \neq 0} |2^n - 2wt(c)| .$$

In particular, for odd n, F is AB if and only if for any non-zero codeword $c \in \mathcal{C}_F$,

$$2^{n-1} - 2^{\frac{n-1}{2}} \leq wt(c) \leq 2^{n-1} + 2^{\frac{n-1}{2}} .$$

(ii) *F is APN if and only if the code $\mathcal{C}_F^{\perp}$ has minimum distance 5.*

When the vector space $\mathbf{F}_2^n$ is identified with the finite field $\mathbf{F}_{2^n}$, the function F can be expressed as a unique polynomial of $\mathbf{F}_{2^n}[X]$. Now, we focus on power functions F, i.e., $F(x) = x^s$ over $\mathbf{F}_{2^n}$. In that case, the linear code $\mathcal{C}_F^{\perp}$ associated to $x \to x^s$ is a binary cyclic code of length $(2^n - 1)$ with two zeros.

Definition 5. *A linear binary code $\mathcal{C}$ of length N is cyclic if for any codeword $(c_0, \ldots, c_{N-1})$ in $\mathcal{C}$, the vector $(c_{N-1}, c_0, \ldots, c_{N-2})$ is also in $\mathcal{C}$.*

If each vector $(c_0, \ldots, c_{N-1}) \in \mathbf{F}_2^N$ is associated with the polynomial $c(X) = \sum_{i=0}^{N-1} c_i X^i$ in $R_N = \mathbf{F}_2^N[X]/(X^N - 1)$, any binary cyclic code of length N is an ideal of R_N. Since R_N is a principal domain, any cyclic code $\mathcal{C}$ of length N is generated by a unique monic polynomial g having minimal degree. This polynomial is called the generator polynomial of the code and its roots are the zeros of $\mathcal{C}$. For $N = 2^n - 1$, the defining set of $\mathcal{C}$ is then the set

$$I(\mathcal{C}) = \{i \in \{0, \cdots, 2^n - 2\} \mid \alpha^i \text{ is a zero of } \mathcal{C}\} .$$

where α is a primitive element of $\mathbf{F}_{2^n}$. Since $\mathcal{C}$ is a binary code, its defining set is a union of 2-cyclotomic cosets modulo $(2^n - 1)$, $Cl(a)$, where $Cl(a) = \{2^j a \bmod (2^n - 1)\}$. Therefore, the defining set of a binary cyclic code of length $(2^n - 1)$ is usually identified with the representatives of the corresponding 2-cyclotomic cosets modulo $(2^n - 1)$. In this context, the linear code $\mathcal{C}_F$ associated to the power function $F : x \to x^s$ on $\mathbf{F}_{2^n}$ is defined by the following generator matrix:

$$G_F = \begin{pmatrix} 1 & \alpha & \alpha^2 & \ldots & \alpha^{2^n-2} \\ 1 & \alpha^s & \alpha^{2s} & \ldots & \alpha^{(2^n-2)s} \end{pmatrix} .$$

Then, the dual code $\mathcal{C}_F^{\perp}$ consists of all binary vectors c of length $(2^n - 1)$ such that $c\, G_F^T = 0$. The code $\mathcal{C}_F^{\perp}$ is therefore the binary cyclic code of length $(2^n - 1)$ with defining set $\{1, s\}$.

4.2 Crosscorrelation of a Pair of Binary m-sequences

A binary sequence $(u_i)_{i \geq 0}$ generated by a linear feedback shift register (LFSR) of length n has maximal period when the feedback polynomial of the LFSR is primitive. Such a sequence is called an *m-sequence* of length $(2^n - 1)$. A binary m-sequence of length $(2^n - 1)$ is identified with the binary vector of length $(2^n - 1)$ consisting of its first $(2^n - 1)$ bits. A further property of m-sequences is that they are almost uncorrelated with their cyclic shifts. This property is important in many communication systems (as radar communications or transmissions using spread-spectrum techniques) since it is often required that a signal can be easily distinguished from any time-shifted version of itself. It is well-known that for any m-sequence u of length $(2^n - 1)$ there exists a unique $c \in \mathbf{F}_{2^n} \setminus \{0\}$ such that

$$\forall i,\ 0 \le i \le 2^n - 2,\ \ u_i = \mathrm{Tr}(c\alpha^i)$$

where α is a root of the feedback polynomial of the LFSR generating u (i.e., α is a primitive element of $\mathbf{F}_{2^n}$) and Tr denotes the trace function from $\mathbf{F}_{2^n}$ to $\mathbf{F}_2$.

When a communication system uses a set of several signals (usually corresponding to different users), it is also required that each of these signals can be easily distinguished from any other signal in the set and its time-shifted versions. This property is of great importance especially in code-division multiple access systems. The distance between a sequence u and all cyclic shifts of another sequence v can be computed with the crosscorrelation function:

Definition 6. *Let u and v be two different binary sequences of length N. The* crosscorrelation function between u and v, *denoted by $\theta_{u,v}$, is defined as*

$$\theta_{u,v}(\tau) = \sum_{i=0}^{N-1} (-1)^{u_i + v_{i+\tau}} .$$

The corresponding crosscorrelation spectrum *is the multiset*

$$\{\theta_{u,v}(\tau),\ \ 0 \le \tau \le N - 1\} .$$

Since $\theta_{u,v}(\tau) = N - 2wt(u + \sigma^\tau v)$ where σ denotes the cyclic shift operator, the above mentioned applications use pairs of sequences (u, v) such that $|\theta_{u,v}(\tau)|$ is small for all $\tau \in \{0, \dots, N - 1\}$.

If u and v are two different binary m-sequences of length $(2^n - 1)$, there exists an integer s in $\{0, \dots, 2^n - 2\}$ and a pair (c_1, c_2) of non-zero elements of $\mathbf{F}_{2^n}$ such that

$$\forall i,\ 0 \le i \le 2^n - 2,\ \ u_i = \mathrm{Tr}(c_1\alpha^i) \text{ and } v_i = \mathrm{Tr}(c_2\alpha^{si}) .$$

If $c_1 = c_2$, the sequence v is said to be a *decimation by s of u*. Writing $c_1 = \alpha^{j_1}$ and $c_2 = \alpha^{j_2}$, the crosscorrelation function for the pair (u, v) is given by:

$$\theta_{u,v}(\tau) = \sum_{i=0}^{2^n-2} (-1)^{\mathrm{Tr}(\alpha^{i+j_1} + \alpha^{si+j_2+\tau})} = \sum_{x \in \mathbf{F}_{2^m}^*} (-1)^{\mathrm{Tr}(\alpha^{\tau'}[\alpha^{j_1-\tau'}x + x^s])} ,$$

where $\tau' = j_2 + \tau$. It follows that the corresponding crosscorrelation spectrum does not depend on the choice of j_2. It is then sufficient to study the pairs (u, v) where v is a decimation by s of u.

Now, we show that the crosscorrelation spectrum of pairs of binary m-sequences is related to the Walsh spectrum of a power function.

Proposition 5. *Let n and s be two positive integers such that $\gcd(s, 2^n-1) = 1$ and s is not a power of 2. Let $\{\theta_s(\tau),\ 0 \leq \tau \leq 2^n-2\}$ be the crosscorrelation spectrum between an m-sequence of length (2^n-1) and its decimation by s. Let F be the power function $x \to x^s$ over $\mathbf{F}_{2^n}$. Then, for any $\alpha \in \mathbf{F}_2^n$, $\alpha \neq 0$, we have*

$$\{\theta_s(\tau),\ 0 \leq \tau \leq 2^n-2\} = \{\mathcal{F}(\phi_\alpha \circ F + \phi_\beta) - 1, \beta \in \mathbf{F}_2^n \setminus \{0\}\} .$$

Most notably,

$$\max_{0 \leq \tau \leq 2^n-2} |\theta_s(\tau) + 1| = \mathcal{L}(F) .$$

In particular when n is odd, the lowest possible value for $\max_\tau |\theta_s(\tau) + 1|$ is $2^{\frac{n+1}{2}}$.

Definition 7. *The crosscorrelation $\theta_{u,v}$ between two m-sequences u and v of length (2^n-1) is said to be* preferred *if it satisfies*

$$\max_\tau |\theta_{u,v}(\tau) + 1| = 2^{\frac{n+1}{2}} .$$

Therefore, the decimations s which lead to a preferred crosscorrelation exactly correspond to the exponents s such that $x \to x^s$ is an almost bent permutation over $\mathbf{F}_{2^n}$.

5 Relations between the Security Criteria

Now, we establish the links between both APN and AB properties. Chabaud and Vaudenay [9] proved that any AB function is APN. Here, we refine this result, since we give a necessary and sufficient condition for an APN function to be AB. We use the following relation involving the Walsh coefficients of a function.

Proposition 6. *Let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$. Then, we have*

$$\sum_{\alpha \in \mathbf{F}_2^n \setminus \{0\}} \sum_{\beta \in \mathbf{F}_2^n} \mathcal{F}^4(\phi_\alpha \circ F + \phi_\beta) = 2^{3n+1}(2^n-1) + 2^{2n}\Delta ,$$

where $\Delta = \#\{(x, a, b) \in (\mathbf{F}_2^n)^3,\ a \neq 0, b \neq 0, a \neq b,$ such that $D_aD_bF(x) = 0\}$.

Most notably, we have

$$\sum_{\alpha \in \mathbf{F}_2^n \setminus \{0\}} \sum_{\beta \in \mathbf{F}_2^n} \mathcal{F}^4(\phi_\alpha \circ F + \phi_\beta) \geq 2^{3n+1}(2^n-1),$$

with equality if and only if F is APN.

Proof. For any Boolean function f of n variables, we have [3, Prop. II.1]

$$\sum_{\beta \in \mathbf{F}_2^n} F^4(f + \phi_\beta) = 2^n \sum_{a,b \in \mathbf{F}_2^n} F(D_a D_b f)\ .$$

By applying this relation to all $\phi_\alpha \circ F$, we deduce

$$\begin{aligned} S &= \sum_{\alpha \in \mathbf{F}_2^n \setminus \{0\}} \sum_{\beta \in \mathbf{F}_2^n} F^4(\phi_\alpha \circ F + \phi_\beta) \\ &= 2^n \sum_{\alpha \in \mathbf{F}_2^n \setminus \{0\}} \sum_{a,b \in \mathbf{F}_2^n} F(D_a D_b(\phi_\alpha \circ F)) \\ &= 2^n \sum_{\alpha \in \mathbf{F}_2^n \setminus \{0\}} \sum_{a,b \in \mathbf{F}_2^n} F(\phi_\alpha \circ D_a D_b F) \\ &= 2^n \sum_{a,b \in \mathbf{F}_2^n} \sum_{\alpha \in \mathbf{F}_2^n} F(\phi_\alpha \circ D_a D_b F) - 2^{4n} \end{aligned}$$

where the last equality is obtained by adding the terms corresponding to $\alpha = 0$ in the sum. Now, for any $a, b \in \mathrm{F}_2^n$, we have

$$\sum_{\alpha \in \mathbf{F}_2^n} F(\phi_\alpha \circ D_a D_b F) = \sum_{\alpha \in \mathbf{F}_2^n} \sum_{x \in \mathbf{F}_2^n} (-1)^{\alpha \cdot D_a D_b F(x)}\ .$$

Using that

$$\sum_{\alpha \in \mathbf{F}_2^n} (-1)^{\alpha \cdot y} = 2^n \text{ if } y = 0 \text{ and } 0 \text{ otherwise,}$$

we obtain

$$\sum_{\alpha \in \mathbf{F}_2^n} F(\phi_\alpha \circ D_a D_b F) = 2^n \# \{x \in \mathrm{F}_2^n,\ D_a D_b F(x) = 0\}\ .$$

Therefore,

$$S = 2^{2n} \# \{x, a, b \in \mathrm{F}_2^n,\ D_a D_b F(x) = 0\} - 2^{4n}\ .$$

Since $D_a D_b F = 0$ when either $a = 0$ or $b = 0$ or $a = b$, we get

$$\begin{aligned} S &= 2^{2n} [2^n (3(2^n - 1) + 1) + \Delta] - 2^{4n} \\ &= 2^{3n+1}(2^n - 1) + 2^{2n} \Delta\ . \end{aligned}$$

Since $\Delta \geq 0$ with equality if and only if F is APN (see Proposition 2), we obtain the expected result.

We then derive the following theorem.

Theorem 2. *Let F be a function from F_2^n into F_2^n. Let*

$$\Delta = \# \{(x, a, b) \in (\mathrm{F}_2^n)^3,\ a = 0, b = 0, a = b,\ \text{such that } D_a D_b F(x) = 0\}\ .$$

Then, we have

(i)

$$\Delta \leq (2^n - 1)(L(F)^2 - 2^{n+1}) ,$$

where equality holds if and only if the values occurring in the Walsh spectrum of F belong to $\{0, \pm L(F)\}$.

(ii) *For any positive integer such that all nonzero Walsh coefficients of F satisfy*

$$|\mathcal{F}(\phi_\alpha \circ F + \phi_\beta)| \geq \ ,$$

we have

$$\Delta \geq (2^n - 1)(\ ^2 - 2^{n+1}) ,$$

where equality holds if and only if the values occurring in the Walsh spectrum of F belong to $\{0, \pm \ \}$.

Proof. Let be a positive integer. Let $I(\ )$ denote the following quantity

$$\begin{aligned} I(\ ) &= \sum_{\alpha\ \mathbf{F}_2^n\setminus\{0\}} \sum_{\beta\ \mathbf{F}_2^n} \left[\mathcal{F}^4(\phi_\alpha \circ F + \phi_\beta) - \ ^2\mathcal{F}^2(\phi_\alpha \circ F + \phi_\beta)\right] \\ &= \sum_{\alpha\ \mathbf{F}_2^n\setminus\{0\}} \sum_{\beta\ \mathbf{F}_2^n} \mathcal{F}^2(\phi_\alpha \circ F + \phi_\beta)\left[\mathcal{F}^2(\phi_\alpha \circ F + \phi_\beta) - \ ^2\right] . \end{aligned}$$

By combining Proposition 6 and Parseval's relation, we obtain that

$$\begin{aligned} I(\ ) &= 2^{3n+1}(2^n - 1) + 2^{2n}\Delta - 2^{2n}(2^n - 1)\ ^2 \\ &= 2^{2n}(2^n - 1)(2^{n+1} - \ ^2) + 2^{2n}\Delta . \end{aligned}$$

Now, any term in the sum defining $I(\ )$ satisfies

$$\mathcal{F}^2(\phi_\alpha \circ F + \phi_\beta)\left[\mathcal{F}^2(\phi_\alpha \circ F + \phi_\beta) - \ ^2\right] \begin{aligned} &< 0 \text{ if } 0 < |\mathcal{F}(\phi_\alpha \circ F + \phi_\beta)| < \\ &= 0 \text{ if } |\mathcal{F}(\phi_\alpha \circ F + \phi_\beta)| \quad \{0, \pm \ \} \\ &> 0 \text{ if } |\mathcal{F}(\phi_\alpha \circ F + \phi_\beta)| > \end{aligned}$$

This implies that all terms appearing in $I(L(F))$ are negative. Then, we have

$$\Delta \leq (2^n - 1)(L(F)^2 - 2^{n+1}) ,$$

with equality if and only if all terms in the sum are zero. This situation only occurs if the values occurring in the Walsh spectrum of F belong to $\{0, \pm L(F)\}$.

Similarly, if all nonzero Walsh coefficients of F satisfy

$$|\mathcal{F}(\phi_\alpha \circ F + \phi_\beta)| \geq \ ,$$

then all terms appearing in $I(\ )$ are positive. Therefore,

$$\Delta \geq (2^n - 1)(\ ^2 - 2^{n+1}) ,$$

with equality if and only if all terms in the sum are zero.

Another proof of this result can be obtained by using the error-correcting code corresponding to F [6]. In that case, the proof is based on Pless identities and on some techniques due to Kasami [18]. As a direct application of the previous theorem, we derive a characterization of almost bent functions.

Corollary 1. *Let n be an odd integer and let F be a function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$. Then, F is AB if and only if F is APN and all its Walsh coefficients are divisible by $2^{\frac{n+1}{2}}$.*

Proof. F is AB if and only if $\mathcal{L}(F) = 2^{(n+1)/2}$. Using Theorem 2 (i), we obtain that $\Delta \leq 0$. Since Δ is a non-negative integer, it follows that $\Delta = 0$, i.e., F is APN. Moreover, the upper bound given in Theorem 2 (i) is achieved. Therefore, the values occurring in the Walsh spectrum of F belong to $\{0, \pm 2^{(n+1)/2}\}$. This implies that all Walsh coefficients are divisible by $2^{(n+1)/2}$.

Conversely, if all Walsh coefficients are divisible by $2^{(n+1)/2}$, then all nonzero Walsh coefficients satisfy

$$|\mathcal{F}(\phi_\alpha \circ F + \phi_\beta)| \geq 2^{(n+1)/2} .$$

From Theorem 2 (ii) applied to = $2^{(n+1)/2}$, we obtain $\Delta \geq 0$. If F is APN, we have $\Delta = 0$ and the lower bound given in Theorem 2 (ii) is reached. Therefore, the values occurring in the Walsh spectrum of F belong to $\{0, \pm 2^{(n+1)/2}\}$. This implies that F is AB.

Note that both properties of AB functions derived from the sufficient condition in the previous corollary have been proved in [9].

A first consequence of the divisibility of the Walsh coefficients of an AB function is the following upper bound on its degree. This bound can be derived from [7, Lemma 3].

Corollary 2. *[8] Let n be an odd integer and F be an AB function from $\mathbf{F}_2^n$ into $\mathbf{F}_2^n$. Then,*

$$\deg(F) \leq \frac{n+1}{2} .$$

Therefore, there exists a trade-off between the security criteria involved by linear cryptanalysis and by higher order differential attacks.

When F is a power function, $F : x \to x^s$, the corresponding code C_F is the dual of the binary cyclic code of length $(2^n - 1)$ with defining set $\{1, s\}$ (see Section 4.1). The weight divisibility of a cyclic code can be obtained by applying McEliece's theorem:

Theorem 3. *[27] The weights of all codewords in a binary cyclic code C are exactly divisible by 2 if and only if is the smallest number such that (+ 1) nonzeros of C (with repetitions allowed) have product 1.*

This leads to the following characterization of AB power functions.

Corollary 3. *Let n be an odd integer and let $F : x \to x^s$ be a power function over $\mathbb{F}_{2^n}$. Then, F is AB if and only if F is APN and*

$$u, \quad 1 \leq u \leq 2^n - 1, \quad w_2(us \bmod (2^n - 1)) \leq \frac{n-1}{2} + w_2(u)$$

where $w_2(u)$ corresponds to the number of 1s in the 2-adic expansion of u.

Thanks to McEliece's theorem, the determination of the values of s such that $x \to x^s$ is almost bent on $\mathbb{F}_{2^n}$ is reduced to a combinatorial problem. Most notably, this technique was directly used to prove that some power functions are AB [5,15]. Moreover, it leads to a very efficient method for proving that a given power function is not AB. For example, the APN power function $x \to x^s$ over $\mathbb{F}_{2^{5g}}$ with $s = 2^{4g} + 2^{3g} + 2^{2g} + 2^g - 1$ does not satisfy the condition of Corollary 3 [6].

These recent results lead to the following list (up to equivalence) of known AB permutations (Table 1). All these functions are power functions. Here, we only give one exponent per cyclotomic coset modulo $(2^n - 1)$. We do not mention the exponent corresponding to the inverse permutation (which is AB too).

Table 1. Known AB power permutations x^s on $\mathbb{F}_{2^n}$

exponents s	condition on n	
$2^i + 1$ with $\gcd(i, n) = 1$ and $1 \leq i \leq (n-1)/2$		[13,30]
$2^{2i} - 2^i + 1$ with $\gcd(i, n) = 1$ and $2 \leq i \leq (n-1)/2$		[19]
$2^{\frac{n-1}{2}} + 3$		[5]
$2^{\frac{n-1}{2}} + 2^{\frac{n-1}{4}} - 1$	$n \equiv 1 \bmod 4$	[15]
$2^{\frac{n-1}{2}} + 2^{\frac{3n-1}{4}} - 1$	$n \equiv 3 \bmod 4$	[15]

When n is even, the smallest known value of $L(F)$ for a function F from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ is $L(F) = 2^{n/2+1}$. The only known functions (up to equivalence) achieving this bound are power functions. Since power permutations cannot be APN, it clearly appears that the security criteria corresponding to differential cryptanalysis and to linear cryptanalysis are not so strongly related. Moreover, the divisibility of the Walsh coefficients of these highly nonlinear functions varies. In particular, the degree of such a function is not upper-bounded since there is no requirement on the divisibility of the Walsh coefficients. Table 2 gives all known power functions achieving the highest known nonlinearity and the divisibility of their Walsh coefficients.

6 Conclusion

The functions which opposes the best resistance to linear cryptanalysis possess a very strong algebraic structure. The AB property appears very restrictive. In

Table 2. Known power permutations x^s on $\mathbb{F}_{2^n}$ with the highest nonlinearity and highest divisibility of their Walsh coefficients

exponents s	condition on n	divisibility	
$2^{n-1} - 1$		2^2	[21]
$2^i + 1$ with $\gcd(i, n) = 2$	$n \equiv 2 \bmod 4$	$2^{\frac{n}{2}+1}$	[13,30]
$2^{2i} - 2^i + 1$ with $\gcd(i, n) = 2$	$n \equiv 2 \bmod 4$	$2^{\frac{n}{2}+1}$	[19]
$\sum_{i=0}^{n/2} 2^{ik}$ with $\gcd(k, n) = 1$	$n \equiv 0 \bmod 4$	$2^{\frac{n}{2}}$	[12]
$2^{\frac{n}{2}} + 2^{\frac{n+2}{4}} + 1$	$n \equiv 2 \bmod 4$	$2^{\frac{n}{2}+1}$	[11]
$2^{\frac{n}{2}} + 2^{\frac{n}{2}-1} + 1$	$n \equiv 2 \bmod 4$	$2^{\frac{n}{2}+1}$	[11]
$2^{\frac{n}{2}} + 2^{\frac{n}{4}} + 1$	$n \equiv 4 \bmod 8$	$2^{\frac{n}{2}}$	[12]

particular, AB functions also guarantee the highest possible resistance against differential cryptanalysis. But, besides the APN property, they can be characterized by the divisibility of their Walsh coefficients. This particular structure leads to an upper-bound on their degree (it then limits their resistance against higher order differential attacks) and it may introduce some other weaknesses. Therefore, it seems preferable to use as round function a function whose nonlinearity is high but not optimal. Most notably, the functions of an even number of variables which have the highest known nonlinearity do not present any similar properties. As an example, the inverse function over a finite field $\mathbb{F}_{2^n}$ with n even (used in AES) offers a very high resistance against differential, linear and higher order differential attacks. Moreover, its Walsh coefficients are divisible by 4 only (which is the lowest possible divisibility).

References

1. E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.
2. A. Canteaut. Differential cryptanalysis of Feistel ciphers and differentially uniform mappings. In *Selected Areas on Cryptography, SAC'97*, pages 172–184, Ottawa, Canada, 1997.
3. A. Canteaut, C. Carlet, P. Charpin, and C. Fontaine. On cryptographic properties of the cosets of $R(1, m)$. *IEEE Trans. Inform. Theory*, 47(4):1494–1513, 2001.
4. A. Canteaut, P. Charpin, and H. Dobbertin. A new characterization of almost bent functions. In *Fast Software Encryption 99*, number 1636 in Lecture Notes in Computer Science, pages 186–200. Springer-Verlag, 1999.
5. A. Canteaut, P. Charpin, and H. Dobbertin. Binary m-sequences with three-valued crosscorrelation: A proof of Welch conjecture. *IEEE Trans. Inform. Theory*, 46(1):4–8, 2000.
6. A. Canteaut, P. Charpin, and H. Dobbertin. Weight divisibility of cyclic codes, highly nonlinear functions on $GF(2^m)$ and crosscorrelation of maximum-length sequences. *SIAM Journal on Discrete Mathematics*, 13(1):105–138, 2000.

7. C. Carlet. Two new classes of bent functions. In *Advances in Cryptology - EUROCRYPT'93*, number 765 in Lecture Notes in Computer Science, pages 77–101. Springer-Verlag, 1994.
8. C. Carlet, P. Charpin, and V. Zinoviev. Codes, bent functions and permutations suitable for DES-like cryptosystems. *Designs, Codes and Cryptography*, 15:125–156, 1998.
9. F. Chabaud and S. Vaudenay. Links between differential and linear cryptanalysis. In *Advances in Cryptology - EUROCRYPT'94*, number 950 in Lecture Notes in Computer Science, pages 356–365. Springer-Verlag, 1995.
10. P. Charpin, A. Tietäväinen, and V. Zinoviev. On binary cyclic codes with minimum distance $d = 3$. *Problems of Information Transmission*, 33(4):287–296, 1997.
11. T. Cusick and H. Dobbertin. Some new 3-valued crosscorrelation functions of binary m-sequences. *IEEE Transactions on Information Theory*, 42:1238–1240, 1996.
12. H. Dobbertin. One-to-one highly nonlinear power functions on $GF(2^n)$. *Appl. Algebra Engrg. Comm. Comput.*, 9(2):139–152, 1998.
13. R. Gold. Maximal recursive sequences with 3-valued recursive crosscorrelation functions. *IEEE Transactions on Information Theory*, 14:154–156, 1968.
14. T. Helleseth and P. Vijay Kumar. *Handbook of Coding Theory*, volume II, chapter 21 - Sequences with low correlation, pages 1765–1853. Elsevier, 1998.
15. H. Hollman and Q. Xiang. A proof of the Welch and Niho conjectures on crosscorrelations of binary m-sequences. *Finite Fields and Their Applications*, 7(2):253–286, 2001.
16. T. Jakobsen. Cryptanalysis of block ciphers with probabilistic non-linear relations of low degree. In *Advances in Cryptology - CRYPTO'98*, number 1462 in Lecture Notes in Computer Science, pages 212–222. Springer-Verlag, 1998.
17. T. Jakobsen and L.R. Knudsen. The interpolation attack on block ciphers. In *Fast Software Encryption 97*, number 1267 in Lecture Notes in Computer Science. Springer-Verlag, 1997.
18. T. Kasami. Weight distributions of Bose-Chaudhuri-Hocquenghem codes. In *Proceedings of the conference on combinatorial mathematics and its applications*, pages 335–357. The Univ. of North Carolina Press, 1968.
19. T. Kasami. The weight enumerators for several classes of subcodes of the second order binary Reed-Muller codes. *Information and Control*, 18:369–394, 1971.
20. L. R. Knudsen. Truncated and higher order differentials. In *Fast Software Encryption - Second International Workshop*, number 1008 in Lecture Notes in Computer Science, pages 196–211. Springer-Verlag, 1995.
21. G. Lachaud and J. Wolfmann. The weights of the orthogonal of the extended quadratic binary Goppa codes. *IEEE Transactions on Information Theory*, 36(3):686–692, 1990.
22. X. Lai. Higher order derivatives and differential cryptanalysis. In *Proc. "Symposium on Communication, Coding and Cryptography", in honor of J. L. Massey on the occasion of his 60'th birthday*, 1994.
23. F.J. MacWilliams and N.J.A. Sloane. *The theory of error-correcting codes*. North-Holland, 1977.
24. M. Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology - EUROCRYPT'93*, number 765 in Lecture Notes in Computer Science. Springer-Verlag, 1994.
25. M. Matsui. The first experimental cryptanalysis of the Data Encryption Standard. In *Advances in Cryptology - CRYPTO'94*, number 839 in Lecture Notes in Computer Science. Springer-Verlag, 1995.

26. M. Matsui. New Block Encryption Algorithm MISTY. In *Proceedings of the Fourth International Workshop of Fast Software Encryption*, number 1267 in Lecture Notes in Computer Science, pages 54–68. Springer-Verlag, 1997.
27. R.J. McEliece. Weight congruence for *p*-ary cyclic codes. *Discrete Mathematics*, 3:177–192, 1972.
28. W. Meier and O. Staffelbach. Nonlinearity criteria for cryptographic functions. In *Advances in Cryptology - EUROCRYPT'89*, number 434 in Lecture Notes in Computer Science, pages 549–562. Springer-Verlag, 1990.
29. K. Nyberg. Perfect nonlinear S-boxes. In *Advances in Cryptology - EUROCRYPT'91*, number 547 in Lecture Notes in Computer Science, pages 378–385. Springer-Verlag, 1991.
30. K. Nyberg. Differentially uniform mappings for cryptography. In *Advances in Cryptology - EUROCRYPT'93*, number 765 in Lecture Notes in Computer Science, pages 55–64. Springer-Verlag, 1993.
31. K. Nyberg and L.R. Knudsen. Provable security against differential cryptanalysis. In *Advances in Cryptology - CRYPTO'92*, number 740 in Lecture Notes in Computer Science, pages 566–574. Springer-Verlag, 1993.
32. D.V. Sarwate and M.B. Pursley. Crosscorrelation properties of pseudorandom and related sequences. *Proceedings of the IEEE*, 68(5):593–619, 1980.
33. V.M. Sidelnikov. On mutual correlation of sequences. *Soviet Math. Dokl.*, 12:197–201, 1971.

Mobile Agent Route Protection through Hash-Based Mechanisms[⋆]

Josep Domingo-Ferrer

Universitat Rovira i Virgili, Dept. of Computer Engineering and Mathematics,
Av. Països Catalans 26, E-43007 Tarragona, Catalonia, Spain,
jdomingo@etse.urv.es

Abstract. One approach to secure mobile agent execution is restricting the agent route to trusted environments. A necessary condition for this approach to be practical is that the agent route be protected. Previous proposals for agent route protection either offer low security or suffer from high computational costs due to cryptographic operations. We present two fast, hash-based mechanisms for agent route protection. The first solution relies on hash collisions and focuses on minimizing the computational cost of route verification by hosts along the route; the cost is shifted to the stage of route protection by the agent owner. The second solution uses Merkle trees and minimizes the cost of route protection by the agent owner, so that a single digital signature suffices to protect the whole route; for hosts along the route, the verification cost is similar to the cost of previous schemes in the literature, namely one digital signature verification per route step. The first solution is especially suitable for agent routes which go through heavily loaded hosts (to avoid denial of service or long delay). The second solution is more adapted to mitigating the bottleneck at agent owners who are expected to launch a great deal of agents. Both solutions provide independent protection for each route step and can be extended to handle flexible itineraries.

Keywords: Mobile agent security, agent route protection, hash collisions, Merkle trees.

1 Introduction

It is increasingly difficult for individuals to take advantage of the exponentially growing wealth of information on the Internet. Mobile agents can be very helpful, as they are programs that roam the network searching for products that fit best buyer requirements. The mobility property raises important security issues: (i) it is important to protect network hosts against malicious agents; (ii) agents should also be protected against malicious hosts. The first problem is analogous to anti-viral protection, and has thus profusely been studied. The second problem consists of attacks to modify the route or the code of the mobile agent by a

[⋆] This work is partly supported by the Spanish CICYT under project no. TEL98-0699-C02-02.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 17–29, 2001.

malicious host which may or may not be in the initial agent route. Only a few (non-exclusive) approaches to protecting agents against malicious hosts have been proposed:

- *Encrypted functions*. In [16], the agent code is identified with the function it computes and a solution is proposed based on computing with encrypted functions (an extension of computing with encrypted data, [3]). This only works for a restricted class of functions.
- *A posteriori detection*. Attacks are detected after they have happened (which may be too late), and upon detection, information can be retrieved and used to accuse the malicious host. Examples are [1,10,21]. Code watermarking [18] would also fall in this category.
- *Obfuscation of agent code*. This is an alternative which reduces code readability and thus makes attacks unlikely. Examples can be found in [5,7,9,17].
- *Trusted environments*. Agents only visit trusted execution environments. A necessary condition to restrict mobility to trusted environments is that the agent route be protected.

This paper contributes to the last aforementioned approach by showing efficient ways to protect agent routes. Section 2 discusses previous work. Section 3 describes a mechanism based on hash collisions which has an extremely low cost in terms of verification by hosts along the route, but is more costly for the agent owner. Section 4 describes a solution based on Merkle trees which has the same verification cost for hosts along the route than previous schemes, but offers a lower computational cost for the agent owner. Section 5 contains some conclusions and shows how the proposed schemes can be used to protect flexible itineraries with alternative paths.

2 Previous Work on Agent Route Protection

In [19] a general concept of an agent route, called itinerary, is given. Flexible agent travel plans can be specified which allow dynamic adaptation and expansion during the execution of the agent. A shortcoming of this scheme is that it is vulnerable to corruption of hosts specified in the itinerary; a corrupted host can modify the itinerary or attack other hosts in the itinerary to cause denial of service to the agent. In [2], a partial solution to the above problems is outlined, but no countermeasures are described to prevent host addition or removal.

In [22] nested encryptions and signatures are used to provide increased security to the agent route. The basic idea is to sign and encrypt host addresses iteratively to attain a high level of security. Let PK_i be the public key of the i-th host and let $S_0(\cdot)$ be the signature function of the agent owner H_0. Then the whole route r is coded as

$$r = E_{PK_1}[H_2, S_0(H_1, m_1, H_2, t, E_{PK_2}[\cdots]), E_{PK_2}[\cdots]]$$

where, for $i = 1$ to $n - 1$,

$$E_{PK_i}[\cdots] = E_{PK_i}[H_{i+1}, S_0(H_i, m_i, H_{i+1}, t, E_{PK_{i+1}}[\cdots]), E_{PK_{i+1}}[\cdots]]$$

and $E_{PK_n}[\cdots] = E_{PK_n}[H_0, S_0(H_n, m_n, H_0, t)]$. In the above expressions, H_i is the IP-address of the i-th host, m_i is the method (code) to be run at H_i, H_0 is the IP-address of the agent owner (host which originates the agent) and t is a timestamp (IP stands for Internet Protocol).

The above coding allows the route to be enforced as follows. The first host H_1 receives r and decrypts it using its private key to obtain $S_0(H_1,m_1,H_2,t,E_{PK_2}[\cdots])$, H_2 and $E_{PK_2}[\cdots]$. Using the public key of the agent owner, H_1 can verify that the next host address H_2 and the value of the rest of the route $E_{PK_2}[\cdots]$ were included in the route by the agent owner. The inclusion of his own address H_1 and method m_1 in the signature allows H_1 to detect that he was also included in the route by the agent owner and that the method m_1 being enclosed with the route is what the agent owner meant to be run at H_1 (this is actually an enhancement over the original proposal [22], which did not include the methods in the signatures). The timestamp t is used to include an expiration date to prevent re-use of older routes of the same agent by malicious hosts. Beyond these validations, H_1 cannot obtain any additional knowledge on the rest of the route since all remaining information is encrypted under the public key PK_2 of H_2. Then H_1 sends the agent to H_2 together with

$$r_1 = E_{PK_2}[H_3, S_0(H_2, m_2, H_3, t, E_{PK_3}[\cdots]), E_{PK_3}[\cdots]]]$$

The above decryption and verification process is repeated by H_2 and so on up to H_n. After n steps, the agent is returned by H_n to H_0. The dark point of proposal [22] is the high processing cost of nested encryptions and signatures at each host along the route (one decryption and one signature verification are needed).

The approach described in [11] is similar to [22] in that it uses one encryption and one signature for each step of the route. In this case, encryptions are nested, whereas signatures are not. This scheme can be generalized to accomodate alternative itineraries.

In the next two sections, two mechanisms are described which substantially reduce the computational overhead of the above proposals, while still preserving the feature that route verification at H_i does not require any information on previous route steps (in particular methods m_j for $j < i$ can be discarded).[1]

3 Reducing the Computational Cost of Route Verification

In comparison with proposals recalled in Section 2, the mechanism proposed in this section focuses on reducing the cost of route verification at the expense of making route protection more costly for the agent owner. This is especially useful if some of the hosts in the route are usually overloaded with computation

[1] Note that a single signature on the concatenation of all route steps is very efficient from a computational viewpoint, but forces the agent to convey all route steps (with their methods m_i) until the route is finished.

(verifications have to be fast to avoid long delay or denial of service). A realistic case where this may happen is when the agent route goes through hosts containing massively accessed Internet or database search engines.

The scheme proposed here borrows from the MicroMint micropayment system [15] the idea of replacing digital signatures with collisions of a hash function; even if not explicitly mentioned, all hash functions used in what follows are assumed to be one-way and collision-free (see Appendix). For implementation purposes, one-way hash functions like SHA [13], MD5 [14] or DES [12] are reasonable choices. The main advantage of replacing digital signatures with hash functions is the speed gain. According to figures by Rivest, if computing an RSA signature takes time t, verifying an RSA signature with low public exponent can be done in $t/100$ and, more important, evaluating a hash function can be done in $t/10000$.

Based on the above ideas, we propose to extend the application of hash collisions from micropayments to agent route protection. Basically, the problem is the same: instead of fast verification of coins by a payee, we need fast verification of route steps by the hosts along the route.

We assume in what follows that, for each host H_i, the agent owner H_0 has set up a symmetric encryption key k_i for secret communication from H_0 to H_i. A way to do this is for H_0 to generate k_i and send $E_{PK_i}(k_i)$ to H_i, where PK_i is the public key of host H_i[2]. The key k_i can be used by H_0 for all routes she schedules through H_i. This allows public-key encryption to be replaced with symmetric encryption; an advantage is that, with the higher speed of symmetric encryption, we can afford to encrypt the code methods to be run at each host, which results in higher confidentiality.

Also, one-way hash functions F_m and F_t are used whose outputs are, respectively, strings of lengths m and t. Standard hash functions, such as SHA, MD5 or DES, may have more output bits than the required m and t; in that case, take the m, resp. t, lower bits as output.

Algorithm 1 (Route protection with hash collisions)

1. [Notation] *The agent owner chooses n hosts represented by their IP-addresses $H_1, \cdots H_n$. Let H_0 be the address of the agent owner. Let k_i be the symmetric encryption key set up for secret communication from H_0 to H_i. Let m_i be the code to be run at H_i.*
2. [Encryption] *The agent owner computes $U_i = E_{k_i}(H_{i-1}, H_i, H_{i+1}, m_i)$ for $i = 1, \cdots, n-1$. For $i = n$, compute $U_n = E_{k_n}(H_{n-1}, H_n, H_0, m_n)$.*
3. [Collision computation] *For $i = 1$ to n, H_0 computes a k-collision $(x_{i,1}, \cdots, x_{i,k})$ such that*

$$F_m(x_{i,1}) = F_m(x_{i,2}) = \cdots = F_m(x_{i,k}) = y_i$$

[2] We do not require that H_0 authenticate itself to H_i when setting up k_i. Although authenticating the origin of k_i would certainly render route step authentication straightforward, it would burden H_i with something like a signature verification, which is against the main goal of the scheme discussed here (minimizing computation at route hosts).

where y_i is an m-bit string such that its t high-order bits match $F_t(U_i)$.

Algorithm 2 (Route verification with hash collisions)

1. [Start of route] *The agent is sent by H_0 to the first host of the route, namely H_1, together with the n k-collisions (one for each route step) and U_i for $i = 1, \cdots, n$.*
2. [Operation at host H_i] *H_i takes the i-th k-collision and checks that all of its k values actually hash to the same y_i. Then H_i checks that $F_t(U_i)$ matches the t high-order bits of y_i. If the check is OK, H_i decrypts U_i and obtains $(H_{i-1}, H_i, H_{i+1}, m_i)$ for $i < n$ or (H_{n-1}, H_n, H_0, m_n) for $i = n$. At this moment, H_i runs m_i and, after that, forwards $U_{i+1}, \cdots, U_n$ along with the k-collisions corresponding to the remaining route steps to H_{i+1} (or to H_0 if $i = n$).*
3. [End of route] *The route ends at the agent owner H_0.*

3.1 Computational Cost

For a host H_i along the route, the computational cost of route verification following Algorithm 2 is reduced to k hash computations and one symmetric decryption.

For the agent owner, the cost is dominated by the computation of k-collisions. Objections have been raised against the high cost of k-collision computation in the case of large-scale MicroMint [20]. We next give a quantitative cost analysis and illustrate the practicality of using k-collisions for our application with a realistic example.

Lemma 1. *If N hash values are computed, the probability of obtaining at least a k-way collision of length m bits with the t high-order bits fixed is*

$$1 - [e^{-N2^{-m}} \sum_{i=0}^{k-1} \frac{(N2^{-m})^i}{i!}]^{2^{m-t}} \qquad (1)$$

Proof. Computing a hash value $y = F_m(x)$, where the length of y is m bits, is analogous to the problem of tossing a ball x at random into one of 2^m bins (the possible values of y). If we call a ball x "good" when the t high-order bits of y match a fixed pattern, then N hash computations will yield an expected number $N := N2^{-t}$ of good balls to be tossed at random into one of 2^{m-t} bins (the possible values of the $m - t$ low-order bits of y). The probability of obtaining at least one k-way collision is 1 minus the probability of all bins getting $k - 1$ or less balls. The probability of a given bin getting a ball in a given toss is $p := 2^{t-m}$. If $N p = N2^{-m} < 5$, $p < 0.1$ and $N > 30$ (equivalently, $N > 2^t 30$), the probability that a bin gets $k-1$ or less balls can be computed using a Poisson approximation:

$$P(k - 1 \text{ or less balls in a bin}) = e^{-N p} \sum_{i=0}^{k-1} \frac{(N p)^i}{i!} \qquad (2)$$

Now the probability of getting at least one k-collision is

$$P(\text{at least one } k\text{-collision}) = 1 - (P(k-1 \text{ or less balls in a bin}))^{2^{m-t}} \quad (3)$$

By substituting Expression (2) in Expression (3), we obtain Expression (1). □

Example 1. In Algorithm 1, let F_m be the low-order m bits of the output of the SHA hash function; formally:

$$F_m(x) = [SHA(x)]_{1\cdots m}$$

Similarly define F_t as $[SHA(x)]_{1\cdots t}$. According to recent figures given by [20], current custom chip technology allows 2^{23} hashes to be computed per second per dollar of chip cost. Assume, as [15], that the agent owner spends \$100,000 in custom chips, so that she can evaluate F_m around 2^{39} times per second (we approximate $2^{23} \times 100,000$ by 2^{39} for ease of calculation). Take $k = 4$ and assume the agent owner is ready to spend 2^8 seconds (4 minutes) to compute a 4-collision; in that time, $N = 2^{47}$ hash values can be computed. If $m = 52$ and $t = 21$ are used, Lemma 1 gives the probability of the agent owner getting at least one good 4-collision of F_m (with the t higher-order bits fixed) in four minutes:

$$1 - [e^{-2^{-5}}(1 + 2^{-5} + \frac{(2^{-5})^2}{2} + \frac{(2^{-5})^3}{6})]^{2^{31}} \approx 1 - 0.716 \cdot 10^{-36} \quad (4)$$

Thus, a good 4-collision will be obtained by the agent owner in four minutes with extremely high probability. □

Thus, it can be seen from Example 1 that coming up with a 4-collision with fixed t high-order bits is costly but by no means unaffordable for the agent owner [3]. This is compensated by the cost reduction in route verification (a relevant figure if the route goes through very busy hosts).

3.2 Security of the Scheme

We will show in this section that the following properties for agent route protection identified as relevant for agent route protection in [22] (see Section 2) are fulfilled by the above scheme:

P1. *Hosts should not be able to modify the agent route.*
P2. *Every host should be able to check it was included in the agent route.*
P3. *Every host should only see the previous host and the next host in the route.*
P4. *Every host should be able to authenticate the host the agent is coming from.*
P5. *A host should not be able to replace a route by older routes of the same agent.*

[3] Unlike for MicroMint, finding one good collision at a time is enough in our application, so the storage and sorting costs additional to the chip cost are much lower.

The basic security of the above scheme rests on two defense lines:

- The difficulty of computing hash collisions of F_m with standard hardware
- For fixed U_i, the unfeasibility of finding $U_i = U_i$ such that $F_t(U_i) = F_t(U_i)$ and such that decryption of U_i under k_i yields H_i as the second of the three IP addresses obtained, where k_i is the key shared between the agent owner and the host with IP address H_i.

With proper parameter choice, the difference between the custom hardware of the agent owner and the standard hardware of a typical user is enough to guarantee efficient computation of m-bit k-collisions for the former and difficult computation for the latter. The following example illustrates this point.

Example 2. In [15], it is assumed that a 1995 standard workstation could perform 2^{14} hash operations per second. Using Moore's law (computer hardware gets faster by a factor of 2 every eighteen months), a more realistic figure for a 2001 standard workstation is that it can perform $2^{14} \cdot 2^4 = 2^{18}$ hash operations per second.

With the above choice $m = 52$, $t = 21$ and $k = 4$, assume 2^{25} seconds (more than one year time) are devoted to compute a 4-collision; in that time, $N = 2^{43}$ hash values can be computed by an attacker owning a standard workstation. Lemma 1 gives the probability of the attacker getting at least one good 4-collision of F_m (with the t higher-order bits fixed) in 2^{25} seconds:

$$1 - [e^{-2^{-9}}(1 + 2^{-9} + \frac{(2^{-9})^2}{2} + \frac{(2^{-9})^3}{6})]^{2^{31}} \approx 1 - 0.9987008 = 0.0012992 \quad (5)$$

Thus, the probability of forging a good 4-collision in one year time is very low. As computer hardware gets faster, slight increases of k may be needed in order for k-collisions to be computable by the agent owner and not by typical users. □

Regarding the second defense line, for a fixed U_i, consider the feasibility of finding $U_i = U_i$ such that $F_t(U_i) = F_t(U_i)$ and such that U_i decrypts into a valid IP address H_i when being decrypted by a host H_i under its key k_i. Note that it does not make sense for H_i to try to forge a $U_i = U_i$ (this does not cause any deviation in the route); nor does it make sense for a host H_i to try to replace U_i with a different U_i which decrypts into H_i as second IP address when using the key k_i shared between H_i and the agent owner. What makes sense is for a host H_j to try to forge $U_i = U_i$, for $j = i$; this could alter the i-th step of the planned route. Thus, the attacker does not know the encryption key k_i that will be used by the host H_i at the i-th step of the altered route. In this case, the best strategy is to randomly look for a $U_i = U_i$ that satisfies $F_t(U_i) = F_t(U_i)$ and then hope that decryption of U_i under k_i will yield the 32 bits corresponding to the IP address H_i in the correct positions. An attempt to meeting this second condition with a random U_i will succeed only with probability 2^{-32}. Thus, a huge number of attempts are likely to be needed. On the other hand, each attempt requires finding a U_i colliding with U_i under F_t, and then decrypting U_i; with proper parameter choice, finding a colliding U_i takes a non-negligible computing time (see Note 1 below), which makes it impractical to perform a huge number of attempts.

Note 1 (On satisfying $F_t(U_i) = F_t(U_i)$). If F_t is one-way, a $U_i = U_i$ such that $F_t(U_i) = F_t(U_i)$ must be looked for at random. The probability of coming up with a good U_i is analogous to the probability of hitting a fixed bin when randomly tossing a ball into one of 2^t bins; thus this probability is 2^{-t}, which means that 2^t hash computations will be needed on average to find a suitable U_i. Assuming $t = 21$ and a processing power of 2^{18} hash values per second as above, this means 8 seconds for a standard user to find U_i.

Now let us check P1 through P5 stated above.

P1. To modify the agent route, at least one step U_i should be modified into a $U_i = U_i$ by some attacker who does not know how to decrypt U_i nor U_i (see discussion above). This has been shown to be computationally infeasible earlier in this section.

P2. Host H_i decrypts U_i using the key k_i and should obtain three IP N addresses, of which the second one should be its own address H_i. If this is not so, then H_i was not included in the route by H_0.

P3. Decryption of U_i allows H_i to learn the addresses of H_{i-1} and H_{i+1}. The remaining addresses of the route are encrypted in U_j, for $j = i$, with keys k_j unknown to H_i. Thus, the rest of hosts in the route remain undisclosed to H_i.

P4. This property can be satisfied only if IP communication between hosts is authenticated. In this case, every host H_i knows which is the *actual* host H_{i-1} the agent is coming from. On the other hand, decryption of U_i provides H_i with the IP address of the host H_{i-1} they agent *should* come from. In this way, H_i can detect whether $H_{i-1} = H_{i-1}$.

P5. To satisfy this property, a timestamp or an expiration date t should be appended by the agent owner to each tuple $(H_{i-1}, H_i, H_{i+1}, m_i)$ before encrypting the tuple into U_i.

4 Reducing the Computational Cost of Route Protection

The main thrust behind the proposal of Section 3 was to reduce the verification cost. If the agent owner is very busy launching a lot of agents each on a different route, the priority may be to reduce the cost of route protection with respect to previous proposals. This is what is achieved by the mechanism presented in this section: as compared to conventional schemes recalled in Section 2, route protection is faster and route verification requires the same amount of work (one signature verification per step).

The mechanism discussed here uses binary Merkle trees as basic tool. Binary Merkle trees are trees constructed as follows. Each leaf is a statement plus the hash of that statement. The hash values of pairs of siblings in the tree are hashed together to get a hash value for their parent node; this procedure iterates until the hash value RV of the root node of the tree has been obtained. We use Merkle trees to construct signatures for each step of the route in a similar way they are used in [6] and [4] to construct public-key certificates. The main advantage of Merkle trees is that one signature on the root node of the tree allows independent integrity verification for all leaves, provided that the hash function used is one-way and collision-free.

Suppose the agent owner has to sign the steps of a route or of several routes, where the i-th step of the route is $(H_{i-1}, H_i, H_{i+1}, m_i)$, that is, the IP addresses of three consecutive hosts plus the method to be run at host H_i (just like in the mechanism described in Section 3).

The algorithm below uses a one-way collision-free hash function F and allows the agent owner to sign all the steps corresponding to a route with a single digital signature.

Algorithm 3 (Route protection with Merkle trees)

1. [Notation] *Let the IP addresses of the hosts along the route be $H_1, \cdots, H_n$. Let k_i be a symmetric encryption key set up for secret communication from the agent owner H_0 to H_i* [4].
2. [Encryption] *The agent owner computes $U_i = E_{k_i}(H_{i-1}, H_i, H_{i+1}, m_i)$ for $i = 1, \cdots, n-1$. For $i = n$, compute $U_n = E_{k_n}(H_{n-1}, H_n, H_0, m_n)$.*
3. [Merkle tree computation] *H_0 computes a binary Merkle tree by taking as leaves the statements U_i and their hash values $F(U_i)$, for $i = 1$ to n.*
4. [Signature] *After creating the Merkle tree, its root value RV is digitally signed into $S_0(RV)$ by the agent owner by using her private key.*

Define the *ver-path* for a route step U_i to be the path from the leaf containing $(U_i, F(U_i))$ to the root RV, together with the hash values needed to verify that path (*i.e.*, the hash values of siblings of nodes along that path). Note that the length of the ver-path equals the height of the tree for a leaf and grows only logarithmically with the number of leaves.

Let us now detail the route verification algorithm:

Algorithm 4 (Route verification with Merkle trees)

1. [Start of route] *The agent is sent by H_0 to the first host of the route, that is H_1, together with all route steps U_i, for $i = 1, \cdots, n$, and the Merkle tree for the whole route with signed RV.*
2. [Operation at host H_i] *H_i takes the i-th route step U_i, extracts its ver-path from the Merkle tree and verifies this ver-path (by recomputing all intermediate hash values starting from $(U_i, F(U_i))$ down to the root). Then H_i checks whether the root value recomputed using U_i and its ver-path are the same RV signed by H_0. If the check is OK, H_i decrypts U_i and obtains $(H_{i-1}, H_i, H_{i+1}, m_i)$ for $i < n$ or (H_{n-1}, H_n, H_0, m_n) for $i = n$. At this*

[4] As in the previous scheme, we do not require here that H_0 authenticate itself to H_i when setting up k_i (even if this would make route step authentication nearly trivial). The reason is that this scheme aims at minimizing the computation at H_0.

moment, H_i runs m_i and, after that, forwards to H_{i+1} (or to H_0 if $i = n$) $U_{i+1}, \cdots, U_n$ along with the part of the Merkle tree needed to verify the remaining route steps (nodes which do not belong to any ver-path of remaining steps can be pruned).

3. [End of route] *The route ends at the agent owner H_0.*

4.1 Computational Cost

For a host H_i along the route, the computational cost of route verification using Algorithm 4 is a number of hash computations equal to the length of the ver-path for step i (typically a one-digit figure), plus one signature verification and one symmetric decryption. According to the figures by Rivest mentioned in Section 3, a signature verification takes as long as 100 hash computations, so the route verification cost is essentially the cost of one signature verification (just like for previous schemes described in Section 2).

For the agent owner, the cost consists of the hash computations needed to create the Merkle tree (or update it, if the same tree is shared by all routes), plus the signature on the root value RV. Since computing a digital signature typically takes as long as 10000 hash computations, the cost is essentially one digital signature for the whole route. This is much lower than the cost for schemes in Section 2, which required one digital signature per route step.

In addition to reducing the number of signatures for route protection, Merkle trees allow mobile agents to convey the protected route in a compact way. Independent protection of each route step would require the agent to initially convey one signature per step, that is $1024n$ bits for an n-step route (assuming 1024-bit RSA signatures). Storing the route as a binary Merkle tree with one leave per step requires one hash value per tree node and a single signature for the whole tree; this makes $1024 + 160 \quad (2n - 1)$ bits to be conveyed by the agent, assuming SHA is used as hash function. This is substantially less than $1024n$ bits.

4.2 Security of the Scheme

Using Merkle trees to extend a single digital signature to a collection of messages is not new [8,6]. Provided that the hash function used is one-way and collision-free, there is no loss of security with respect to signing messages individually. Let us check for this scheme the security properties P1 to P5 discussed in Section 3.2 for the hash collision scheme.

P1. To modify the i-th step, U_i should be modified into $U_i \; = U_i$. This would require finding a ver-path for U_i such that its verification yields the same root value obtained from verification of the ver-path of U_i. If the hash function used is one-way and collision-free, this is computationally infeasible.

P2,P3,P4,P5. Same comments as in Section 3.2.

5 Conclusions and Extension to Flexible Itineraries

One approach to secure mobile agent execution is restricting the agent route to trusted environments. A necessary condition for this solution to be practical is that the agent route be protected. We have proposed hash-based schemes which try to improve computational efficiency without degrading security:

- The mechanism based on hash collisions concentrates on making route verification very lightweight, while route protection stays somewhat costly. This mechanism is very appropriate for agents going through very busy hosts (these could deny service if verification was too time-consuming).
- The mechanism based on Merkle trees aims at reducing the computational work carried out by the agent owner to protect a route. This mechanism is especially suitable when the agent owner is the bottleneck, as it might happen for very busy agent owners who must launch large number of mobile agents each on a different route.

Both mechanisms presented here can be extended to flexible itineraries in the sense of [19,11]. Since each route step is independently coded (either as a hash collision or as a tree leaf), we could think of including several alternative paths going from a host H_i to another host H_j along the route. To do this, code all steps of the alternative paths. When choosing one particular path at a junction, the information (including methods) related to steps in alternative paths does not need to be conveyed by the agent to the next hosts in the route. Coding all alternative path steps substantially increases the route protection work in the scheme based on hash collisions (a new collision is needed for each route step). For the Merkle tree scheme, the computational overhead of including alternative paths is negligible: adding more leaves to the Merkle tree is fast and does not even significantly increase the length of ver-paths.

Acknowledgments

This work was partly supported by the Spanish CICYT under project no. TEL98-0699-C02-02. Thanks go to Francesc Sebé and Marc Alba for reading an earlier draft of this paper and helping to work out some examples.

References

1. S. Y. Bennet, "A sanctuary for mobile agents", in *Foundations for Secure Mobile Code Workshop*. Monterey CA: DARPA, 1997, pp. 21-27.
2. J. Borrell, S. Robles, J. Serra and A. Riera, "Securing the itinerary of mobile agents through a non-repudiation protocol", in *33rd Annual IEEE Intl. Carnahan Conference on Security Technology*. Piscataway NJ: IEEE, 1999, pp. 461-464.
3. J. Domingo-Ferrer, "A new privacy homomorphism and applications", *Information Processing Letters*, vol. 60, no. 5, Dec. 1996, pp. 277-282.
4. J. Domingo-Ferrer, M. Alba and F. Sebé, "Asynchronous large-scale certification based on certificate verification trees", in *IFIP Communications and Multimedia Security'2001*, Boston MA: Kluwer, 2000, pp. 185-196.

5. D. Dyer, "Java decompilers compared", June 1997.
`http://www.javaworld.com/javaworld/jw-07-1997/jw-decompilers.html`
6. I. Gassko, P. S. Gemmell and P. MacKenzie, "Efficient and fresh certification", in *Public Key Cryptography'2000*, LNCS 1751. Berlin: Springer-Verlag, 2000, pp. 342-353.
7. F. Hohl, "Time limited blackbox security: Protecting mobile agents from malicious hosts", in *Mobile Agents and Security*, LNCS 1419. Berlin: Springer-Verlag, 1998, pp. 92-113.
8. C. Jutla and M. Yung, "PayTree: " Amortized-signature" for flexible micropayments", in *Second USENIX Workshop on Electronic Commerce*, Oakland CA, Nov. 1996.
9. D. Libes, *Obfuscated* C *and Other Mysteries*, New York: Wiley, 1993.
10. C. Meadows, "Detecting attacks on mobile agents", in *Foundations for Secure Mobile Code Workshop*. Monterey CA: DARPA, 1997, pp. 50-56.
11. J. Mir, "Protecting flexible routes of mobile agents", private communication, 2001.
12. National Bureau of Standards, "Data Encryption Standard", FIPS Publication 46, Washington DC, 1977.
13. U. S. National Institute of Standards and Technology, *Secure Hash Standard*, FIPS PUB 180-1, 1995.
`http://csrc.ncsl.nist.gov/fips/fip180-1.txt`
14. R.L. Rivest and S. Dusse, "RFC 1321: The MD5 message-digest algorithm", Internet Activities Board, Apr. 1992.
15. R.L. Rivest and A. Shamir, "PayWord and MicroMint: Two simple micropayment schemes", Technical report, MIT Laboratory for Computer Science, Nov. 1995.
16. T. Sander and C.F. Tschudin, "Protecting mobile agent against malicious hosts", in *Mobile Agents and Security*, LNCS 1419. Berlin: Springer-Verlag, 1998, pp. 44-60.
17. K. B. Sriram, "Hashjava - a java applet obfuscator", July 1997.
`http://www.sbktech.org/hashjava.html`
18. J. P. Stern, G. Hachez, F. Koeune and J.-J. Quisquater, "Robust object watermarking: application to code", in *Information Hiding'99*, LNCS 1768. Berlin: Springer-Verlag, 2000, pp. 368-378.
19. M. Strasser, K. Rothermel and C. Maihöfer, "Providing reliable agents for electronic commerce", in *TREC'98*, LNCS 1402. Berlin: Springer-Verlag, 1998, pp. 241-253.
20. N. van Someren, "The practical problems of implementing MicroMint", in *Financial Cryptography'2001*, February 2001 (proceedings still to appear). Available from author `nicko@ncipher.com`.
21. G. Vigna, "Cryptographic traces for mobile agents", in *Mobile Agents and Security*, LNCS 1419. Berlin: Springer-Verlag, 1998, pp. 137-153.
22. D. Westhoff, M. Schneider, C. Unger and F. Kaderali, "Methods for protecting a mobile agent's route", in *Information Security-ISW'99*, LNCS 1729. Berlin: Springer-Verlag, 1999, pp. 57-71.

Appendix. Hash Functions and the MicroMint System

Hash functions are widely used in cryptography to perform digital signatures. A hash function, sometimes called message digest, is a function F that takes a variable-length input string x and converts it to a fixed-length output string y.

A hash function F is said to be computationally one-way if it is easy and fast to compute the hash $y = F(x)$ but, given y, it is hard to compute x such that $y = F(x)$. A hash function F is said to be collision-free if it is hard to find two x, x such that $x = x$ and $F(x) = F(x\ )$.

MicroMint [15] is a system for low-value payments (micropayments) where the coins are not digitally signed by the bank (or minting organization). Instead, the bank computes coins as k-way collisions, *i.e.* k values whose hash images collide for a prespecified one-way hash function F. More formally, a coin is represented by $(x_1, \cdots, x_k)$ such that

$$F(x_1) = F(x_2) = \cdots = F(x_k) = y$$

where y is an m-bit string. Increasing k has the dual effect of increasing the computation needed to find the first collision, and also accelerating the rate of collision generation once the first collision has been found; actually, $k = 4$ is recommended by the MicroMint authors.

The verifier of a coin (the payee) accepts it as valid if it is a k-way collision and the t high-order bits of y match a value z advertised by the bank at the start of the current validity period. Thus verification only requires computing k hash values.

A New Anonymous Fingerprinting Scheme with High Enciphering Rate

Minoru Kuribayashi[1] and Hatsukazu Tanaka[2]

[1] Division of Information and Media Science,
Graduate School of Science and Technology, Kobe University,
1-1 Rokkodai-cho, Nada-ku, Kobe, Japan 657-8501
minoru@es3.eedept.kobe-u.ac.jp

[2] Department of Electrical and Electronics Engineering,
Faculty of Engineering, Kobe University,
1-1 Rokkodai-cho, Nada-ku, Kobe, Japan 657-8501
tanaka@eedept.kobe-u.ac.jp

Abstract. We propose a new anonymous fingerprinting scheme using Okamoto-Uchiyama cryptosystem [1]. In the previous schemes [2]–[4] the enciphering rate is so small that it seems very difficult to implement for any applications. In order to improve the rate, we have applied the Okamoto-Uchiyama cryptosystem for our fingerprinting protocol. As the results, a buyer can commit a fingerprint to a merchant being embedded in a digital content anonymously and efficiently, and then the amount of encrypted data is controlled in a reasonable size. The security can also be protected for both of a buyer and a merchant in our scheme.

1 Introduction

According to the development of the Internet, multi-media become to treat digital contents on the network. It enables us to purchase digital contents via a net easily. However, it causes several problems such as violation of ownership and illegal distribution of the copy. Watermarking [5] is one of the effective schemes to solve these problems. It enables the owner to embed some information in the contents and to extract it, and the applications can be classified by a kind of embedded information as follows. When the information indicates a copyright owner, it can be applied for the ownership protection. A fingerprinting scheme embeds the information related to a buyer, and enables a merchant to trace the buyer from the redistributed copy. First a symmetric fingerprinting scheme has been proposed. In the scheme an original merchant embeds the buyer's identity in his/her contents by himself/herself. Therefore, the merchant can not prove the buyer's treachery to anyone. To solve the problem, some cryptographic methods were applied for an asymmetric fingerprinting scheme [6]. Furthermore, an anonymous fingerprinting scheme [2] was introduced to solve the condition that electronic market places should offer to the customers the same privacy as the real-world market places.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 30–39, 2001.

The concept of anonymous fingerprinting introduced in [2] has been presented only a scheme using general theorems. The explicit construction was shown in [3] and [4] which are based on digital coins. Since all operations are simple computations such as modular multiplications and exponentiations, it seems easy to implement for a real application. However, from the point of enciphering information rate, the efficiency is very bad. If one uses the fingerprinting scheme for music, movie, etc., the amount of data to be sent will become incredibly large. Therefore, the problem is how to embed a fingerprint in the digital content efficiently.

In this paper we propose a new construction scheme of anonymous fingerprinting that overcomes the above drawback by exploiting Okamoto-Uchiyama cryptosystem [1]. Since it has a homomorphic property, the multiplication of encrypted fingerprint and digital content is equivalent to embed a fingerprint in the digital content. The property can make a merchant embed an buyer's identity information in the ciphertext of his/her contents. If the buyer can convince the merchant that the sent ciphertext really includes his/her identity, the anonymity of the buyer can be established. The trade between a buyer and a merchant is executed as follows. The buyer encrypts a fingerprint and commits it to the merchant using zero-knowledge proofs. The merchant embeds the received data in his/her encrypted digital content and returns it to the buyer. Finally the buyer decrypts and gets the fingerprinted content without disclosing the fingerprint to the merchant. As the results, only the buyer gets the fingerprinted content unless he/she redistributes it. Our main contribution is the achievement of a better enciphering rate than the conventional ones [2]–[4].

2 Preliminary

In this section we introduce some basic techniques used in our scheme. First we review and classify the fingerprinting techniques. Then bit commitment schemes that are exploited in the conventional scheme are reviewed, and some inherent problems are disclosed. Finally the Okamoto-Uchiyama public-key cryptosystem is summarized in order to refer the encryption and decryption functions, and their properties.

2.1 Fingerprinting

Digital contents such as image, music, movie, etc. are easily copied without any degradation. Fingerprinting is a cryptographic scheme for the copyright protection of digital contents assisted by a watermarking technique. And the scheme can prevent people from executing illegal redistribution of digital contents by making it possible for the merchant to identify the original buyer of the redistributed copy, where we call him/her a "traitor". The fingerprinting schemes can be classified into the following three classes.

Symmetric: The operation to embed a fingerprint is performed only by a merchant. Therefore, he/she cannot convince any third party of the traitor's treachery even if he/she has found out the identity of a traitor in the content.

Asymmetric: Fingerprinting is a interactive protocol between a buyer and a merchant. After the sale, only the buyer obtains the data with a fingerprint. If the merchant has found the fingerprinted copy somewhere, he/she can identify the traitor and prove to the third party.

Anonymous: A buyer can purchase a fingerprinted content without informing his/her identity to a merchant, but the merchant can identify the traitor later. It also retains the asymmetric property.

Pfitzmann et al. [2] has constructed an anonymous fingerprinting system by seven protocols; *Registration center key distribution, Registration, Data initialization, Fingerprinting, Identification, Enforced identification* and *Trail*. Our result is contributed to the *Fingerprinting* protocol, namely it is how to embed a fingerprint in a digital data anonymously at two-party protocol.

2.2 Bit Commitment Scheme

In the anonymous fingerprinting scheme, a buyer and a merchant jointly embed a fingerprint. First, the buyer encrypts a fingerprint and sends it to the merchant. Then the merchant verifies that the received ciphertext is made from the real fingerprint, and embeds it in his/her encrypted content. Finally, the buyer receives the encrypted and fingerprinted content and decrypts it. After the protocol, only the buyer gets the fingerprinted content without disclosing his/her identity. Here, one of the most important things is how to embed the encrypted fingerprint in the encrypted content. To accomplish it, Pfitzmann et al. [3], [4] exploit two commitment schemes. One is applied for the verification that the commitment really includes the fingerprint to be embedded and the other is for the embedding of the fingerprint in the merchant's contents. The former is based on the discrete logarithm problem, and the latter is on the quadratic residues [7] of which security depends on the difficulty of factoring n. Though an encrypted fingerprint can be embedded in the encrypted content, the enciphering rate is very small because the commitment can contain only one-bit message in $\log n$-bit ciphertext. To improve the rate, we propose a new method based on the Okamoto-Uchiyama cryptosystem [1].

2.3 Okamoto-Uchiyama Cryptosystem

Let p and q be two large primes ($|p| = |q| = k$ bits) and $N = p^2q$. Choose $g \quad (Z/NZ)$ randomly such that the order of $g_p = g^{p-1} \bmod p^2$ is p, where $g.c.d.(p, q-1) = 1$ and $g.c.d.(q, p-1) = 1$. Let $h = g^N \bmod N$ and a function $L(x) = (x-1)/p$. Here a public key is (N, g, h, k) and a secret key is (p, q).

The cryptosystem, based on the exponentiation mod$N$, is constructed as follows.

Encryption: Let m $(0 < m < 2^{k-1})$ be a plaintext. Selecting a random number r $(\mathbb{Z}/N\mathbb{Z})$, a ciphertext is given by

$$C = g^m h^r \pmod{N}. \tag{1}$$

Decryption: Calculate first $C_p = C^{p-1} \bmod p^2$ and then

$$m = \frac{L(C_p)}{L(g_p)} \pmod{p}, \tag{2}$$

We denote the encryption function $E(m, r)$ and decryption function $D(C)$. Three important properties of the scheme are given by the following P1, P2 and P3.

P1. It has a homomorphic property : if $m_0 + m_1 < p$,

$$E(m_0, r_0) \cdot E(m_1, r_1) = E(m_0 + m_1, r_0 + r_1) \pmod{N}. \tag{3}$$

P2. It is semantically secure if the following assumption, *i.e.* p-subgroup assumption, is true: $E(0, r) = h^r \bmod N$ and $E(1, r\ ) = gh^{r}$ mod N is computationally indistinguishable, where r and r are uniformly and independently selected from $\mathbb{Z}/N\mathbb{Z}$.

P3. Anyone can change a ciphertext, $C = E(m, r)$, into another ciphertext, $C\ = Ch^{r}$ mod N, while preserving plaintext of C (*i.e.*, $C\ = E(m, r\ )$), and the relationship between C and C can be concealed.

The notation used here is applied for our proposed scheme in the following section.

3 Proposed Scheme I

The idea of our proposed scheme is to exploit the Okamoto-Uchiyama cryptosystem for anonymous fingerprinting. If we assume that a fingerprint is denoted by a number m_0 and a digital content is given by a number m_1, then a fingerprinted item becomes $m_0 + m_1$ from the property P1. In our scheme a buyer B can commit his/her identity to a merchant M as a fingerprint without informing the real value, and M can embed the fingerprint in the content at the enciphered form. After receiving the encrypted and fingerprinted content, B decrypts it, but can not remove the fingerprint.

3.1 Fingerprinting Protocol

The anonymous fingerprinting protocol is executed between a buyer B and a merchant M. B commits his/her identity, $id = \ w_j 2^j$ $(0 \le j \le \ - 1)$ to M the enciphered form, com_j, and M encrypts his/her content I_i $(0 \le i \le L - 1)$ and multiplies it to the received com_j. We assume that B has already registered at a center RC and sent M the registration proof and his/her identity proof $W = g^{id} \bmod N$. Under the assumption, the fingerprinting protocol is given as follows.

[*Fingerprinting*]

Step 1. M generates a random number $a(2 < a < N)$ and sends it to B.

Step 2. B decomposes a into random numbers a_j to satisfy the following equation.

$$a = \sum_{j=0}^{\square 1} a_j 2^j \tag{4}$$

A bit commitment of each w_j is calculated as

$$com_j = g^{w_j} h^{a_j} \pmod{N}, \tag{5}$$

and sent to M.

Step 3. To verify the commitment, M calculates

$$V = h^a \pmod{N}, \tag{6}$$

and makes sure that the following equation can be satisfied.

$$\prod_j {com_j}^{2^j} \overset{?}{=} W \cdot V \pmod{N} \tag{7}$$

Step 4. M generates L random numbers b_i (Z/NZ) and embedding intensity T of even number. Then, in order to get the encrypted and fingerprinted content, M calculates

$$Y_i = \begin{cases} g^{I_i} h^{b_i} \cdot com_j^T \cdot g^{-\frac{T}{2}} \pmod{N} & \text{marking position} \\ g^{I_i} h^{b_i} \pmod{N} & \text{elsewhere} \end{cases} \tag{8}$$

and sends it to B

Step 5. Since the received Y_i is rewritten as

$$Y_i = \begin{cases} g^{(I_i + Tw_j - \frac{T}{2})} h^{Ta_j + b_i} \pmod{N} & \text{marking position} \\ g^{I_i} h^{b_i} \pmod{N} & \text{elsewhere,} \end{cases} \tag{9}$$

B can decrypt Y_i to get the plaintext.

$$D(Y_i) = \begin{cases} I_i + Tw_j - \frac{T}{2} \pmod{p} & \text{marking position} \\ I_i \pmod{p} & \text{elsewhere} \end{cases} \tag{10}$$

On the deciphered message, if $w_j = 1$, then $T/2$ has been added to I_i, and if $w_j = 0$, then $T/2$ has been subtracted from I_i. As the characteristic is suitable for several watermarking schemes like [8], our scheme can be applied easily.

Remark 1. If we regard w_j as a message and a_j as a random number, then com_j can be shown by $E(w_j, a_j)$ and com_j^T by $E(Tw_j, Ta_j)$ because

$$\begin{aligned} com_j^T &= (g^{w_j} h^{a_j})^T \pmod{N} \\ &= g^{Tw_j} h^{Ta_j} \pmod{N} \\ &= E(Tw_j, Ta_j). \end{aligned} \tag{11}$$

In Eq.(8), $g^{I_i}h^{b_i}g^{-T/2} = E(I_i - T/2, b_i)$ can be regarded as M's enciphered content, and then from the property P1 Y_i at the marking position can be rewritten as

$$\begin{aligned} Y_i &= E(Tw_j, Ta_j) \cdot E(I_i - \tfrac{T}{2}, b_i) \\ &= E(I_i + Tw_j - \tfrac{T}{2}, Ta_j + b_i) \end{aligned} \tag{12}$$

Here from the subsection 2.3, the message $I_i - T/2$ must satisfy an inequality $0 < I_i - T/2 < 2^{k-1}$. If M use I_i as a pixel value directly, the suitable pixel that satisfies the above inequality can be easily selected to embed a fingerprint. However, if M applies the transformed coefficients, the message should be modified for the adaptive data structure.

3.2 Security for the Merchant

In order to check the security, we consider some possible attacks. B may be able to forge his/her identity as he/she has not proved that the values w_j $(0 \le j \le \;\; -1)$ are binary in the fingerprinting protocol. To solve the problem, the following additional protocol should be performed.

[*Binary Proof*]

Step 1. In order to check com_j, M generates random numbers t_j and c_j such that $t_j + c_j$ is less than 2^{k-1}, calculates

$$Q_j = com_j^{t_j} \cdot g^{c_j} \pmod N, \tag{13}$$

and sends Q_j to B.

Step 2. B decrypts the received Q_j as

$$D(Q_j) = w_j t_j + c_j \pmod N \tag{14}$$

and then he/she generates a random number r_j and calculates

$$\hat{com}_j = com_j^{t_j+c_j} \cdot h^{r_j} \pmod N \tag{15}$$

using the values c_j and Q_j or $t_j + c_j$. The detail is shown in the following Remark 2.

Step 3. After M receives $\hat{com}_j$, he/she sends t_j and c_j to prove that Q_j has been really produced using them.

Step 4. If Eq.(13) is satisfied for the received t_j and c_j, B sends r_j to M. If it is not satisfied, he/she can claim M's fraud.

Step 5. By verifying Eq.(15), M can certified that com_j contains only 1-bit information.

Remark 2. If $w_j = 0$ in the Step 2, then $D(Q_j) = c_j$ and $Q_j = g^{c_j}g^{a_j t_j} \bmod N$. Using Q_j and c_j, B can calculate

$$\begin{aligned} \hat{com}_j &= Q_j \cdot g^{-c_j}h^{a_j c_j + r_j} \pmod N \\ &= h^{a_j(t_j+c_j)+r_j} \pmod N \\ &= E\left(0, a_j(t_j + c_j) + r_j\right) \\ &= com_j^{t_j+c_j} \cdot h^{r_j} \end{aligned} \tag{16}$$

If $w_j = 1$, then $D(Q_j) = t_j + c_j$. Therefore $\widehat{com}_j$ is obtained by the following.

$$\begin{aligned}\widehat{com}_j &= g^{t_j+c_j} h^{a_j(t_j+c_j)+r_j} \pmod N\\ &= E\left(t_j + c_j, a_j(t_j + c_j) + r_j\right)\\ &= com_j^{t_j+c_j} \cdot h^{r_j} \qquad (17)\end{aligned}$$

Otherwise, B can not calculate $\widehat{com}_j$ using the decrypted Q_j because the knowledge of each t_j and c_j or $t_j + c_j$ is inevitable. Therefore the lack of information makes it impossible to calculate $\widehat{com}_j$ when w_j is not binary. From the above facts, the following lemma can be proved.

Lemma 1. *B can prove that w_j is binary using a zero-knowledge protocol.*

Proof. B can not obtain the values both t_j and c_j from Q_j, but only $w_j t_j + c_j$. Without the knowledge of the two values, B can not calculate $com_j^{t_j+c_j}$ except only two cases of $w_j = 0$ and $w_j = 1$. As B knows w_j, a_j and $w_j(t_j + c_j)$, $\widehat{com}_j$ can be calculated by following Eqs.(16) and (17) if w_j is binary. It is remarkable that from the property P3 random number r_j changes the ciphertext $com_j^{t_j+c_j}$ to $com_j^{t_j+c_j} \cdot h_j^r = E\left(w_j(t_j + c_j), a_j(t_j + c_j) + r_j\right)$ preserving the plaintext $w_j(t_j + c_j)$. It guarantees that no information about w_j leaks to M as he/she can not distinguish $E\left(0, a_j(t_j + c_j) + r_j\right)$ and $E\left(t_j + c_j, a_j(t_j + c_j) + r_j\right)$. When B reveals r_j, M can make sure that w_j is binary by verifying Eq.(15), but can not get information anymore. Furthermore, M can not deceive B in the Step 2 as he/she should reveal the values t_j and c_j later to receive r_j.

Using the above protocol, B can prove that w_j is binary from the Lemma 1 and hence M can embed B's identity properly and securely in his/her contents. Other possible attack is to remove or change the embedded his/her identity information directly from a fingerprinted content, but it is equivalent to attack the applied watermarking system. Then we can obtain the following theorem.

Theorem 1. *The security concerning to M is protected if the applied watermarking system is robust against attacks.*

3.3 Security for the Buyer

In order to certify the security concerning to B, we must prove that M can not obtain B's identity under the following three assumptions:

A1 The discrete logarithm problem is too difficult to solve.
A2 The Okamoto-Uchiyama cryptosystem is secure.
A3 B dose not redistribute a copy.

From these assumptions, the following theorem can be proved.

Theorem 2. *B can purchase contents from M anonymously if three assumptions A1, A2 and A3 are satisfied.*

Proof. As $W = g^{id} \bmod N$, to derive the identity id from W is equivalent to solve the discrete logarithm problem, but it is extremely difficult from the assumption A1. In Step 2, the bit commitment com_j has only two forms: one is $E(0, r)$ and the other is $E(1, r)$ as the values of w_j are binary. From the property P2, M can not obtain the w_j from the commitment com_j if the assumption A2 is satisfied. Enabling M to get a fingerprint from illegally redistributed copy, the identity id can be extracted from the decrypted Y_i. However, M never get it under the assumption A3. Hence the anonymity of B is preserved.

From the Theorem 2, M can not abuse the identity of B. Therefore, the security concerning to B is protected.

4 Proposed Scheme II

4.1 Modified Fingerprinting Protocol

In the proposed scheme I, each I_i is encrypted and fingerprinted independently. Since I_i and T are much smaller than $2^{k-1} (< p)$ and the ciphertext is much larger than p, the enciphering rate is small. To improve the drawback, the size of message to be encrypted should be modified as large as 2^{k-1}. Let m_i be

$$m_i = \begin{cases} I_i + Tw_j - \frac{T}{2} & \text{marking position} \\ I_i & \text{elsewhere,} \end{cases} \tag{18}$$

and s be the maximum bit-length of m_i. Since s is much smaller than k, the message can be replaced by

$$M_{i'} = \sum_{t=0}^{c-1} m_{i'c+t} 2^{st}, \qquad 0 \le i' \le L/c - 1, \quad c = \lfloor k/s \rfloor \tag{19}$$

After the modification, each $M_{i'}$ is encrypted to $E(M_{i'}, r)$, where r is a random number. Let y_i be the encrypted and fingerprinted I_i. The fingerprinting protocol of Step 4 and Step 5 proposed in the previous section is changed as follows.

[*Fingerprinting(modified)*]

Step 4. In order to get the encrypted and fingerprinted content y_i, M calculates

$$y_i = \begin{cases} g^{I_i} \cdot com_j^T \cdot g^{-\frac{T}{2}} \pmod N & \text{marking position} \\ g^{I_i} \pmod N & \text{elsewhere} \end{cases} \tag{20}$$

To synthesize some y_i in one ciphertext $Y_{i'}$, the following operation is performed using a random number $b_{i'} \in (\mathbb{Z}/N\mathbb{Z})$.

$$Y_{i'} = \left(\prod_t (y_{i'c+t})^{2^{st}} \right) \cdot h^{b_{i'}} \pmod N \tag{21}$$

Step 5. B decrypts the received $Y_{i'}$ to obtain $M_{i'}$. Since he/she knows the bit-length s of m_i, he/she can decompose $M_{i'}$ into the pieces. Finally he/she can get the fingerprinted contents.

Remark 3. From Eqs.(11),(18)-(20) and the property P3, Eq.(21) can be expressed by

$$
\begin{aligned}
Y_{i'} &= \left(\prod_{t} g^{m_{i'c+t} 2^{st}} \right) \cdot h^{r} \pmod{N} \\
&= g^{M_{i'}} h^{r} \pmod{N} \\
&= E(M_{i'}, r). \qquad (22)
\end{aligned}
$$

4.2 Security

On the security of the proposed scheme II, we should consider only on Step 4 and Step 5 as we have already discussed the other steps in the previous section. First, we show the relation between $Y_{i'}$ and its data structure. If the Okamoto-Uchiyama cryptosystem is secure and the bit-length of $M_{i'}$ is less than k, B can decrypt $Y_{i'} = E(M_{i'}, r)$. Here, in Eqs.(21) and (22) several pieces $m_{i'c+t}$ of fingerprinted content that compose $M_{i'}$ are encrypted in one ciphertext $E(M_{i'}, r)$, though each piece is encrypted in the proposed scheme I. Therefore, $M_{i'}$ should retain a special data structure described by Eq.(19). If M changes the data structure, B can not decompose it into the correct pieces $m_{i'c+t}$, and then he/she can claim the fact. Hence, with the knowledge of data structure B can decompose the decrypted message $M_{i'}$ into $m_{i'c+t}$ and finally get the fingerprinted content. Furthermore, as $M_{i'}$ is simply produced by composing several pieces of $m_{i'c+t}$, B can not derive any information about original content from the decrypted message.

5 Improvement of the Enciphering Rate

In this section, we discuss the efficiency of our scheme compared with the conventional one. Here, omitting the computational complexity, we only consider the enciphering rate, as every calculation is simple modular multiplication or exponentiation that is similar to the conventional one. We assume that a digital content consists of L pixels of x-bit scale image and B's identity is bits. As L is much larger than , we evaluate the rate only by the encrypted and fingerprinted content. In [3] and [4], the security is based on the difficulty of factoring n. When each bit of the content is encrypted, thus the total amount of encrypted data is $xL \log n$ bits. On the other hand, the security of our schemes is based on the difficulty of factoring $N(= p^2q,\ 3k$ bits). In the proposed scheme I, the amount of encrypted data is $L \log N (= 3kL)$ bits as each pixel is encrypted. In the proposed scheme II, it is $(L \log N)/c\ (\ \ 3xL)$ bits, because there are L/c messages $M_{i'}$ to be encrypted, where s is the bit-length of each message and $s\ \ \ x$. Here, if $\log n\ \ \ \log N = 3k$, the enciphering information rates are indicated in Table 1.

Furthermore, the rate can be increased by restricting the embedding positions because of the following. Some watermarking schemes are designed to embed in the spatial domain, but almost all schemes in the transformed domain such as DCT, DFT, wavelet transform, etc. Generally, a signal embedded in the

Table 1. Enciphering rate

conventional	scheme I	scheme II
$1/3k$	$x/3k$	$1/3$

transformed (frequency) domain is more robust against attacks than in the spatial (time) domain, and the high frequency components are easily and seriously affected by attacks [5]. Hence, it is desirable to select some suitable components for embedding a fingerprint. Then, avoiding high frequency component to be encrypted, the total amount of the data can be decreased. However, if the number of the encrypted components is very few, B may be able to derive the selected position and remove or change the embedded fingerprint. Therefore, the trade-off between the security and the rate should be considered.

6 Conclusion

We have proposed a new anonymous fingerprinting scheme based on the Okamoto-Uchiyama cryptosystem. The achievement of our proposed scheme is the improvement of enciphering rate that is too small in the conventional one. Using the Okamoto-Uchiyama cryptosystem, an encrypted fingerprint can be embedded in an encrypted content with high enciphering rate, and then the buyer's anonymity can be protected. Furthermore, the protocol can be performed between only two parties, a buyer and a merchant, which is similar to a real-world market.

References

1. T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," Proc. of EUROCRYPT'98, LNCS 1403, Springer-Verlag, pp.308-318, 1998.
2. B. Pfitzmann and M. Waidner, "Anonymous fingerprinting," Proc. of EUROCRYPT'97, LNCS 1233, Springer-Verlag, pp.88-102, 1997.
3. B. Pfitzmann and A. Sadeghi, "Coin-based anonymous fingerprinting," Proc. of EUROCRYPT'99, LNCS 1592, Springer-Verlag, pp.150-164, 1999.
4. B. Pfitzmann and A. Sadeghi, "Anonymous fingerprinting with direct non-repudiation," Proc. of ASIACRYPT'2000, LNCS 1976, Springer-Verlag, pp.401-414, 2000.
5. S. Katzenbeisser and F. A. P. Petitcolas, Information hiding techniques for steganography and digital watermarking, Artech house publishers, Jan. 2000.
6. B. Pfitzmann and M. Schunter, "Asymmetric fingerprinting," Proc. of EUROCRYPT'96, LNCS 1070, Springer-Verlag, pp.84-95, 1996.
7. G. Brassard, D. Chaum and C. Crepeau, "Minimum disclosure proofs of knowledge," Journal of Computer and System Sciences vol. 37, pp.156-189, 1988.
8. M. Kuribayashi and H. Tanaka, "A watermarking scheme based on the characteristic of addition among DCT coefficients," Proc. of ISW2000, LNCS 1975, Springer-Verlag, pp.1-14, 2000.

A Parallel Algorithm for Extending Cryptographic Hash Functions (Extended Abstract)*

Palash Sarkar and Paul J. Schellenberg

Department of Combinatorics and Optimization, University of Waterloo,
200, University Avenue West, Waterloo, Ontario, Canada N2L 3G1,
psarkar@cacr.math.uwaterloo.ca, pjschell@math.uwaterloo.ca

Abstract. We describe a parallel algorithm for extending a small domain hash function to a very large domain hash function. Our construction can handle messages of any practical length and preserves the security properties of the basic hash function. The construction can be viewed as a parallel version of the well known Merkle-Damgård construction, which is a sequential construction. Our parallel algorithm provides a significant reduction in the computation time of the message digest, which is a basic operation in digital signatures.

Keywords: cryptographic hash function, Merkle-Damgård construction, parallel algorithm, collision resistance.

1 Introduction

Hash functions are extensively used in cryptographic protocols. One of the main uses of hash functions is to generate a message digest from a message. This message digest is signed to get a digital signature. Due to the central importance of hash functions in cryptography, there has been a lot of work in this area. See [6] for a recent survey.

For a hash function to be used in cryptographic protocols, it must satisfy certain necessary conditions. In a recent paper [8], Stinson provides a comprehensive discussion of these conditions and also relations among them. The two most important properties that a cryptographic hash function must satisfy are the following: (a) finding a collision must be computationally infeasible and (b) finding a preimage of a given message digest must be computationally infeasible.

A hash function maps a set of longer messages into a set of shorter message digests. The range is finite, while the domain can possibly be (countably) infinite. Thus, theoretically, a hash function can map arbitrary length strings to finite length strings. However, hash functions with an infinite (or very large) domain can be difficult to construct directly. An alternative approach is to take a hash

* This research supported by a grant from the Mathematics of Information Technology and Complex Systems (MITACS) project.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 40–49, 2001.

function with a small finite domain and suitably extend it to tackle long strings. The extension must preserve the security properties (difficulty of finding collision and preimage) of the original hash function. An important construction for such extensions has been described by Merkle [3] and Damgård [2]. The construction is called the Merkle-Damgård (MD) construction.

The MD construction is a sequential construction. Suppose the basic hash function has domain $\{0,1\}^{512}$ and range $\{0,1\}^{128}$. Further, suppose that the message to be signed is long, say 1 Mbits ($=2^{20}$ bits). If the MD construction is applied to the message, then the time taken to generate the digest would be proportional to $2^{20}/(512-128)$. For many applications this can cause an undesirable delay.

In this paper we build on the basic MD construction. We introduce a parallel version of this construction which preserves the security features of the basic hash function. The parallel version uses 2^t processors for some t and produces a significant speed up in the computation of the message digest.

Related Work: The concept of tree hashing has appeared before in the literature. Wegman and Carter [10] used tree hashing techniques to build universal hash functions. This was followed up by Naor and Yung [5] and Bellare and Rogaway [1] in the context of universal one way hash functions. Damgard [2] briefly outlines a tree hashing approach for extending collision resistant hash functions.

In this paper we concentrate exclusively on developing a parallel tree based algorithm for extending cryptographic hash functions. The main difference between our model and previous models is that we consider the number of available processors to be fixed while the length of the message can be arbitrarily long. Thus we consider a fixed processor tree and use it to hash arbitrarily long messages. Each processor simply computes the base hash function. The resulting increase in speed of computation of the message digest is almost linear in the number of processors. As an example, it may not be very expensive to use a tree of 32 or 64 processors to reduce the time required for message digest computation by a factor of 32 or 64 respectively. We believe that our algorithm has potential practical applications in digital signature computation.

Due to lack of space, proofs and detailed discussions cannot be presented in this paper. For these we refer the reader to [7].

2 Basics

2.1 Hash Functions

Our description of hash functions closely parallels that of Stinson [8]. An (n,m) hash function h is a function $h : \{0,1\}^n \rightarrow \{0,1\}^m$. *Throughout this paper we require that* $n \geq 2m$. Consider the following problem as defined in [8].

Problem	: Collision $Col(n,m)$
Instance	: An (n,m) hash function h.
Find	: x, x $\{0,1\}^n$ such that $x = x$ and $h(x) = h(x)$.

By an $(, p)$ (randomized) algorithm for Collision we mean an algorithm which invokes the hash function h at most p times and solves Collision with probability of success at least .

The hash function h has a finite domain. We would like to extend it to an infinite domain. Our first step in doing this is the following. Given h and a positive integer L, we construct a hash function $h_L : \{0,1\}^L \rightarrow \{0,1\}^m$. The next step, in general, is to construct a hash function $h^{\infty} : {}_{L \geq n}\{0,1\}^L \rightarrow \{0,1\}^m$. However, instead of doing this, we actually construct a hash function $h\ :\ {}^{N}_{L=n}\{0,1\}^L \rightarrow \{0,1\}^m$, where $N = 2^{n-m} - 1$. Since we assume $n \geq 2m$, we have $n - m \geq m$. Practical message digests are at least 128 bits long meaning that $m = 128$. Hence our construction of h can handle any message with length $\leq 2^{128}$. This is sufficient for any conceivable application. (If we estimate that there are 32 billion computers, that is about 5 computers per man, woman and child, and each computer has 1024 gigabytes of disk storage, and each byte has eight bits, the number of bits that can be stored on all the these computer systems combined is a mere $2^5 \times 2^{30} \times 2^{10} \times 2^{30} \times 2^3 = 2^{78}$ bits. Our construction of h can be extended to construct h^{∞} and will be provided in the full version of the paper.

We would like to relate the difficulty of finding collision for h_L, h to that of finding collision for h. Thus we consider the following two problems (see [8]).

Problem	: Fixed length collision $FLC(n, m, L)$
Instance	: An (n, m) hash function h and an integer $L \geq n$.
Find	: $x, x \ \ \{0,1\}^L$ such that $x = x$ and $h_L(x) = h_L(x)$.

Problem	: Variable length collision $VLC(n, m, L)$
Instance	: An (n, m) hash function h and an integer L with $n \leq L \leq 2^{n-m}$.
Find	: $x, x \ \ {}^{L}_{i=n}\{0,1\}^i$ such that $x = x$ and $h\ (x) = h\ (x)$.

By an $(, p, L)$ (randomized) algorithm A for Fixed length collision (resp. Variable length collision) we will mean an algorithm that requires at most p invocations of the function h and solves Fixed length collision (resp. Variable length collision) with probability of success at least . The algorithm A will be given an oracle for the function h and p is the number of times A queries the oracle for h in attempting to find a collision for h_L (resp. h).

Later we show Turing reductions from Collision to Fixed length collision and Variable length collision. Informally this means that given oracle access to an algorithm for solving $FLC(n, m, L)$ for h_L or $VLC(n, m, L)$ for h it is possible to construct an algorithm to solve $Col(n, m)$ for h. These will show that our constructions preserve the intractibility of finding collisions.

2.2 Processor Tree

Our construction is a parallel algorithm requiring more than one processors. *The number of processors is* 2^t. Let the processors be $P_0, \ldots, P_{2^t-1}$. For $i = 0, \ldots, 2^{t-1} - 1$, processor P_i is connected to processors P_{2i} and P_{2i+1} by arcs pointing towards it. The processors $P_{2^{t-1}}, \ldots, P_{2^t-1}$ are the *leaf processors* and the processors $P_0, \ldots, P_{2^{t-1}-1}$ are the *internal processors*. We call the resulting tree the processor tree of depth t. For $1 \leq i \leq t$, there are 2^{i-1} processors at level i. Further, processor P_0 is considered to be at level 0.

Each of the processors gets an input which is a binary string. The action of the processor is to apply the hash function h on the input if the length of the input is n; otherwise, it simply returns the input -

$$P_i(y) = \begin{cases} h(y) & \text{if } |y| = n; \\ y & \text{otherwise.} \end{cases} \tag{1}$$

For $0 \leq i \leq 2^t - 1$, we have two sets of buffers u_i and z_i. We will identify these buffers with the binary strings they contain. The buffers are used by the processors in the following way. There is a formatting processor P_F which reads the message x, breaks it into proper length substrings, and writes to the buffers u_i. For $0 \leq i \leq 2^{t-1} - 1$, the input buffers of P_i are z_{2i}, z_{2i+1} and u_i and the input to P_i is formed by concatenating the contents of these buffers. For $2^{t-1} \leq i \leq 2^t - 1$, the input buffer of P_i is u_i. The output buffer of P_i is z_i for $0 \leq i \leq 2^t - 1$.

Our parallel algorithm goes through several parallel rounds. The contents of the buffers u_i and z_i are updated in each round. To avoid read/write conflicts we will assume the following sequence of operations in each parallel round.

1. The formatting processor P_F writes into the buffers u_i, for $0 \leq i \leq 2^t - 1$.
2. Each processor P_i reads its respective input buffers.
3. Each processor P_i performs the computation in (1).
4. Each processor P_i writes into its output buffer z_i.

Steps (2) to (4) are performed by the processors $P_0, \ldots, P_{2^t-1}$ in parallel after Step (1) is completed by processor P_F.

2.3 Parameters and Notation

Here we introduce some notation and define certain parameters which are going to be used throughout the paper.
Number of processors: 2^t.
Start-up length: $2^t n$.
Flushing length: $(2^{t-1} + 2^{t-2} + \cdots + 2^1 + 2^0)(n - 2m) = (2^t - 1)(n - 2m)$.
Start-up + flushing length: $\delta(t) = 2^t n + (2^t - 1)(n - 2m) = 2^t(2n - 2m) - (n - 2m)$.
Steady-state length: $\lambda(t) = 2^{t-1}n + 2^{t-1}(n - 2m) = 2^{t-1}(2n - 2m)$.
Message: a binary string x of length L.
Parameters q, b and r:

1. If $L > \delta(t)$, then q and r are defined by the following equation: $L - \delta(t) = q\lambda(t) + r$, where r is the unique integer from the set $\{1, \ldots, \lambda(t)\}$. Define $b = \frac{r}{2n-2m}$.

2. If $L = \delta(t)$, then $q = b = r = 0$.

Note that $0 \leq b \leq 2^{t-1}$. *We will denote the empty string by* $<>$ *and the length of a binary string* x *by* $|x|$.

3 Fixed Length Input

In this section we describe the construction of the function h_L. The construction is naturally divided into two cases depending on whether $L \geq \delta(t)$ or $L < \delta(t)$. We first show that the case $L < \delta(t)$ reduces to the case $L \geq \delta(t)$ for some $t < t$. Thus the case $L < \delta(t)$ is tackled using only a part of the processor tree.

3.1 Case $L < \delta(t)$

Let $t < t$ be such that $\delta(t) \leq L < \delta(t + 1)$. We use the processor tree only upto level t and use the parallel hashing algorithm of Section 3.2 with t replaced by t . Thus we are not utilizing all the available processors. It can be shown that this results in a cost of at most one additional parallel round. We will present this proof in the full version of the paper.

3.2 Case $L \geq \delta(t)$

We first describe the parallel hashing algorithm. This algorithm uses several other algorithms as subroutines. We describe these later.

The parameters b and q defined in Section 2.3 will be used in the algorithms that we describe next. More specifically, the parameter q will be used in algorithm PHA and the parameter b will be used in algorithms FEG and FF. These parameters are assumed to be global parameters and are available to all the subroutines. It is quite simple to modify the subroutines such that the parameters are computed as and when required.

Parallel Hashing Algorithm (PHA)
Input: message x of length $L \geq \delta(t)$.
Output: message digest $h_L(x)$ of length m.

```
if $L > \delta(t)$, then
        $x := x||0^{b(2n-2m)-r}$
        (ensures that the length of the message becomes
        $\delta(t) + q\lambda(t) + b(2n - 2m)$.)
endif.
Initialise buffers $z_i$ and $u_i$ to empty strings, $0 \leq i \leq 2^t - 1$.
Do FormatStartUp.
Do ParallelProcess.
for     $i = 1, 2, \ldots, q$ do
        Do FormatSteadyState.
        Do ParallelProcess.
```

```
endfor
Do FormatEndGame.
Do ParallelProcess.
for    $s = t - 1, t - 2, \ldots 2, 1$ do
       Do FormatFlushing($s$).
       Do ParallelProcess.
endfor
$z_0 = P_0(z_0 || z_1 || x)$.
return $z_0$.
end algorithm PHA
```

ParallelProcess (PP)
Action: Read buffers u_i and z_i, and update buffers z_i, $0 \le i \le 2^t - 1$.

```
for    $i = 0, \ldots, 2^t - 1$ do in parallel
       $z_i := P_i(z_{2i} || z_{2i+1} || u_i)$    if $0 \le i \le 2^{t-1} - 1$.
       $z_i := P_i(u_i)$                          if $2^{t-1} \le i \le 2^t - 1$.
endfor
end algorithm PP
```

Formatting Algorithms. There are four formatting subroutines which are invoked by PHA. Each of the formatting subroutines modifies the message x by removing prefixes which are written to the buffers u_i for $0 \le i \le 2^t - 1$. All the formatting subroutines are executed on the formatting processor P_F.

FormatStartUp (FSU)
Action: For $0 \le i \le 2^t - 1$, write a prefix of message x to buffer u_i and update the message x.

```
for    $i = 0, \ldots, 2^t - 1$ do
       Write $x = v || y$, where $|v| = n$.
       $u_i := v$.
       $x := y$.
endfor
end algorithm FSU
```

FormatSteadyState (FSS)
Action: For $0 \le i \le 2^t - 1$, write a prefix of message x to buffer u_i and update the message x.

```
for    $i = 0, \ldots, 2^{t-1} - 1$ do
       Write $x = v || y$, where $|v| = n - 2m$.
       $u_i := v$.
       $x := y$.
endfor
for    $i = 2^{t-1}, \ldots, 2^t - 1$ do
       Write $x = v || y$, where $|v| = n$.
```

```
        $u_i := v$.
        $x := y$.
endfor
end algorithm FSS
```

FormatEndGame (FEG)
Action: For $0 \leq i \leq 2^t - 1$, write a prefix of message x to buffer u_i and update the message x.

```
for   $i = 0, 1, 2, \ldots, 2^{t-1} - 1$ do
      Write $x = v||y$ where $|v| = n - 2m$.
      $u_i := v$.
      $x := y$.
endfor
for   $i = 2^{t-1}, 2^{t-1} + 1, \ldots, 2^{t-1} + b - 1$ do
      Write $x = v||y$ where $|v| = n$.
      $u_i := v$.
      $x := y$.
endfor
for   $i = 2^{t-1} + b, 2^{t-1} + b + 1, \ldots, 2^t - 1$ do
      $u_i := <>$.
endfor
end algorithm (FEG)
```

FormatFlushing(s) (FF(s))
Input: Integer s.
Action: For $0 \leq i \leq 2^t - 1$, write a prefix of message x to buffer u_i and update the message x.

```
$k = \frac{b+2^{t-s-1}-1}{2^{t-s}}$ .

for   $i = 0, 1, 2, \ldots, 2^{s-1} + k - 1$ do
      Write $x = v||y$ where $|v| = n - 2m$.
      $u_i := v$.
      $x := y$.
endfor
for   $i = 2^{s-1} + k, 2^{s-1} + k + 1, \ldots, 2^t - 1$,
      Write $u_i := <>$.
endfor
end algorithm FF
```

3.3 Correctness and Complexity

Here we state that algorithm PHA properly computes an m-bit message digest and state various properties of the algorithm. In Section 3.4 we will provide the security reduction of $Col(n, m)$ to $FLC(n, m, L)$. More detailed discussion and proofs can be found in [7]. Algorithm PHA executes the following sequence of parallel rounds.

1. Lines 5-6 of PHA execute one parallel round.
2. Lines 7-10 of PHA execute q parallel rounds.
3. Lines 11-12 of PHA execute one parallel round.
4. Lines 13-16 of PHA execute $t - 1$ parallel rounds.
5. We consider Line 17 of PHA to be a special parallel round.

From this we get the following result.

Theorem 1. *Algorithm PHA executes $q + t + 2$ parallel rounds.*

Each of the first $(q + t + 1)$ parallel rounds consist of a formatting phase and a hashing phase. In the formatting phase, the formatting processor P_F runs a formatting subroutine and in the hashing phase the processors P_i $(0 \le i \le 2^t - 1)$ are operated in parallel. Denote by $z_{i,j}$ the state of the buffer z_i at the end of round j, $0 \le i \le 2^t - 1$, $1 \le j \le q + t + 2$. Clearly, the state of the buffer z_i at the start of round j $(2 \le j \le q + t + 2)$ is $z_{i,j-1}$. Further, let $u_{i,j}$ be the string written to buffer u_i in round j by the processor P_F. For $0 \le i \le 2^{t-1} - 1$, the input to processor P_i in round j is $z_{2i,j-1} || z_{2i+1,j-1} || u_{i,j}$. For $2^{t-1} \le i \le 2^t - 1$, the input to processor P_i in round j is the string $u_{i,j}$.

Theorem 2 (Correctness of PHA).

1. *Algorithm PHA terminates and provides an m-bit message digest.*
2. *Algorithm PHA provides each bit of the message x as part of the input to precisely one invocation of the hash function h.*

Proposition 1. *Let $\psi(L)$ be the number of invocations of h by PHA on a padded message of length $L = \delta(t) + q\lambda(t) + b(2n - 2m)$. Then $\psi(L) = (q + 2)2^t + 2b - 1$. Moreover, $\psi(L)$ is also the number of invocations of h made by the sequential MD algorithm.*

Remark: The time taken by the MD algorithm is proportional to the number of invocations of h whereas the time required by PHA is proportional to the number of parallel rounds. This is the basis for the speed-up obtained by PHA.

Proposition 2. *The maximum amount of padding added to any message is less than $2n - 2m$.*

3.4 Security Reduction

In this section we provide a Turing reduction of $Col(n, m)$ to $FLC(n, m, L)$. This will show that if it is computationally difficult to find collisions for h, then it is also computationally difficult to find collisions for h_L. We provide only a sketch of the proof. The detailed proof can be found in [7].

Theorem 3. *Let h be an (n, m) hash function and for $L \ge n$ let h_L be the function defined by algorithm PHA. If there is an $(, p, L)$ algorithm A to solve $FLC(n, m, L)$ for the hash function h_L, then there is an $(, p + 2\psi(L))$ algorithm B to solve $Col(n, m)$ for the hash function h.*

Sketch of proof: The idea of the proof is to show that if A can find x and x such that $x = x$ but $h_L(x) = h_L(x\ )$, then one can find w and w such that $w = w$ but $h(w) = h(w\ )$. The proof proceeds in the following manner. The output of PHA on input x is $h_L(x) = z_{0,q+t+2}$ and the output of PHA on input x is $h_L(x\ ) = z_{0,q+t+2}$. Assume that $b > 0$ (the case $b = 0$ is similar). In this case the inputs to processor P_0 in round $q + t + 2$ are $z_{0,q+t+1}||z_{1,q+t+1}||u_{0,q+t+2}$ and $z_{0,q+t+1}||z_{1,q+t+1}||u_{0,q+t+2}$ corresponding to strings x and x respectively. If $z_{0,q+t+1}||z_{1,q+t+1}||u_{0,q+t+2} = z_{0,q+t+1}||z_{1,q+t+1}||u_{0,q+t+2}$, then we have a collision for h. Otherwise $z_{0,q+t+1} = z_{0,q+t+1}$, $z_{1,q+t+1} = z_{1,q+t+1}$ and $u_{0,q+t+2} = u_{0,q+t+2}$. Note that $u_{0,q+t+2}$ and $u_{0,q+t+2}$ are substrings of x and x respectively. Thus not obtaining a collision for h at this stage implies a certain portion of x and x are equal. At this point we use an reverse induction on round number to show that if there is no collision for h, then $x = x$. Since by assumption we have $x = x$, we must find a collision for h.

4 Variable Length Input

In the previous section we developed composition schemes which work for fixed input lengths. More precisely, given $h : \{0,1\}^n \rightarrow \{0,1\}^m$ and a positive integer L, we have shown how to construct $h_L : \{0,1\}^L \rightarrow \{0,1\}^m$. We now extend this to $h\ :\ {}_{L=n}^{N}\{0,1\}^L \rightarrow \{0,1\}^m$, where $N = 2^{n-m} - 1$. For $0 \leq i \leq 2^s - 1$, let $bin_s(i)$ be the s-bit binary expansion of i. We treat $bin_s(i)$ as a binary string of length s. Then $h\ (x)$ is defined as follows.

$$h\ (x) = h\left(bin_{n-m}(|x|)||(h_{|x|}(x))\right). \qquad (2)$$

In other words, we first apply $h_L(x)$ (where $|x| = L$) on x to obtain an m-bit message digest w. Let $v = bin_{n-m}(|x|)$. Then v is a bit string of length $n - m$. We apply h to the string $v||w$ to get the final message digest.

Remark: 1. We do not actually require the length of the message to be $< 2^{n-m}$. The construction can easily be modified to suit strings having length $< 2^c$ for some constant c. Since we are assuming $n \geq 2m$ and $m \geq 128$ for practical hash functions, choosing $c = n - m$ is convenient and sufficient for practical purposes.
2. The construction can be modified to tackle arbitrary length strings. For the usual Merkle-Damgård procedure this is described in [9]. We will provide the extension to arbitrary length strings for our construction in the full version of the paper.

Proposition 3. *Let τ_h and $\tau_h(L)$ be the time taken to compute h and h_L respectively. Then the time taken to compute h is $\tau_h(L) + \tau_h$ and the number of invocations of h is $1 + \psi(L)$.*

Theorem 4. *Let h be an (n,m) hash function and h be the function defined by Equation 2. If there is an $(\ ,p,L)$ algorithm A to solve $VLC(n,m,L)$ for the hash function h , then there is an $(\ ,p + 2 + 2\psi(L))$ algorithm B to solve $Col(n,m)$ for the hash function h.*

5 Concluding Remarks

We have considered only one property of hash functions - namely intractibility of finding collisions. There are other desirable properties that a hash function must satisfy. These are Zero Preimage and Preimage (see [8]). In [8], reductions between these properties have been studied. In our case, we are required to show that our constructions preserve the intractibility of these problems. In fact, these properties are indeed preserved and the proofs will be provided in the full version of the paper.

The second important point is that we have considered the processors to be organised as a binary tree. In fact, the same technique carries over to k-ary trees, with the condition that $n \geq km$. The computation can be made even faster by moving from binary to k-ary processor trees. However, the formatting processor will progressively become more complicated and will offset the advantage in speed up. Hence we have not explored this option further.

References

1. M. Bellare and P. Rogaway. Collision-resistant hashing: towards making UOWHFs practical. *Proceedings of CRYPTO 1997*, pp 470-484.
2. I. B. Damgård. A design principle for hash functions. *Lecture Notes in Computer Science*, 435 (1990), 416-427 (Advances in Cryptology - CRYPTO'89).
3. R. C. Merkle. One way hash functions and DES. *Lecture Notes in Computer Science*, 435 (1990), 428-226 (Advances in Cryptology - CRYPTO'89).
4. I. Mironov. Hash functions: from Merkle-Damgård to Shoup. *Lecture Notes in Computer Science*, 2045 (2001), 166-181 (Advances in Cryptology - EUROCRYPT'01).
5. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic aplications. *Proceedings of the 21st Annual Symposium on Theory of Computing*, ACM, 1989, pp. 33-43.
6. B. Preneel. The state of cryptographic hash functions. *Lecture Notes in Computer Science*, 1561 (1999), 158-182 (Lectures on Data Security: Modern Cryptology in Theory and Practice).
7. P. Sarkar and P. J. Schellenberg. A parallel algorithm for extending cryptographic hash functions. CACR Technical Report, University of Waterloo, http://www.cacr.math.uwaterloo.ca
8. D. R. Stinson. Some observations on the theory of cryptographic hash functions. IACR preprint server, http://eprint.iacr.org/2001/020/.
9. D. R. Stinson. *Cryptography: Theory and Practice*, CRC Press, 1995.
10. M. N. Wegman and J. L. Carter. New Hash Functions and Their Use in Authentication and Set Equality. *Journal of Computer and System Sciences*, 22(3): 265-279 (1981)

Incremental Hash Function Based on Pair Chaining & Modular Arithmetic Combining

Bok-Min Goi, M.U. Siddiqi, and Hean-Teik Chuah

Faculty of Engineering, Multimedia University,
Cyberjaya 63100, Malaysia
{bmgoi, umar, htchuah}@mmu.edu.my

Abstract. Most of the hash functions using iterative constructions, are inefficient for bulk hashing of documents with high similarity. In this paper, we present a new approach to construct a cryptographic hash function called Pair Chaining & Modular Arithmetic Combining Incremental Hash Function (PCIHF). PCIHF has some attractive properties, which are incrementality and parallelizability. The security of PCIHF has also been analyzed comprehensively. Finally, we show that PCIHF is not only universal one-way but also collision-free.

Key words. Cryptography, Hash Function, Incremental Cryptography.

1 Introduction

Hash functions take an input string of arbitrarily length and output a unique fixed length string called as hash value. Originally, hash functions were designed for the purpose of data integrity. The simplest protocol for ensuring data integrity is as follows. Firstly, it hashes a large message M to obtain its corresponding fixed size hash value μ. Then, μ with smaller size is protected by storing it in an expensive private space, such as a smart card. Meanwhile, M is kept in a public space. Later, we verify the integrity of M by hashing the message to obtain a new hash value μ . Then, we compare μ with the protected hash value μ. If they are equal, with high probability, we conclude that M has not been altered. By using hash functions, we can save the cost and speed up the verification process.

In order to ensure that the data integrity protection scheme works, the hash functions must fulfill some cryptographic properties, which are 1^{st} and 2^{nd}-preimage resistance and collision resistance (Menezes et. al. [8]). Most of the hash functions using iterative constructions, are inefficient for bulk hashing of documents with high similarity. This is because of the nature of the iterative process where the processing time is proportional to the total length of the message.

Here, we propose a new cryptographic hash function called Pair Chaining & Modular Arithmetic Combining Incremental Hash Function (PCIHF) by using the concept of incremental cryptography, which was first introduced by Bellare et. al. [1]. For the proposed PCIHF, the time taken for updating the hash value is proportional to the amount of modifications made to the message or constant for certain text modification functions. We have assumed that the amount of

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 50–61, 2001.

modification is small compared to the size of the message. Therefore, PCIHF is more efficient. Moreover, PCIHF is parallelizable which is very useful for hardware and software implementation.

Section 2 presents the notation and related works on incremental cryptography. We elaborate on the construction of PCIHF and analysis of its efficiency on some text modification functions in Section 3. In Section 4, we show that our proposed PCIHF is universal one-way and collision-free.

2 Preliminary

2.1 Standard Notation and Definition

Here, we briefly go over some standard notation. $\{0,1\}^k$ denotes a binary string with k bit. $[q]$ is the set of integers $\{x : 1 \le x \le q\}$. $|s|$ denotes the length (in bits) of s. $(R - S)$ denotes the difference of sets R and S. Symbol Ω denotes the asymptotic lower bound. $r||s$ and r s denote that string r is concatenated and exclusive-OR with string s respectively. Finally, s denotes the binary representation of s.

2.2 Incremental Hash Function and Related Works

Incremental cryptography has the special property that after applying it to a message, the cryptographic primitive can quickly be updated for the modified message. In details, by applying a message M, its hash value μ, and the text modification function F to an ideal incremental hash function H, will produce an updated hash value μ of the modified message M faster than recomputing it from scratch. The time taken must be independent of the size of the message; but dependent only on the number of changed blocks.

Theoretically, any collision-free compression function can be modified to become an incremental hash function. Bellare and Micciancio [4] presented the randomize-then-combine paradigm. They pointed out the reasons why conventional combining operators (such as exclusive-OR) are rejected. However, it can still be used in message authentication scheme (MAC) where a secret key is involved (Bellare et. al. [3] and Fischlin [5]). Besides that, incremental hash functions based on tree scheme and discrete logarithm problem have been proposed in Bellare et. al. [1] and [2]. Due to some disadvantages in the later two schemes, we will only concentrate on randomize-then-combine paradigm. The basic construction of incremental MAC is as below,

$$\mu = \bigwedge_{i \ [n]} R_a(M[i]) \qquad (1)$$

$\bigwedge$ is the group operation and R_a is a keyed pseudo-random function with secret key a. Unfortunately, (1) is not 2$^{\text{nd}}$ pre-image resistance. This is because the hash value remains unchanged by rearranging the message block. For example,

$H(M[1]||M[2]) = H(M[2]||M[1])$. In order to solve this, each block is added with a counter, and the construction becomes,

$$\mu = \bigwedge_{i\ \ [n]} R_a(\ i\ ||M[i]) \tag{2}$$

Now, the sequence of message block is essential. Although dramatic improvements in term of speed can be achieved in using the incremental settings, it posed some new security concerns, which are tamper-proof security and privacy. Since (2) is still a linear function, it is insecure against message substitution attack. For example, let $\mu_1 = H(A||B)$, $\mu_2 = H(A||D)$ and $\mu_3 = H(C||B)$ where $A, B, C, D \quad M[i]$. Without knowing a, an adversary will be able to obtain the valid μ for message $C||D$, because $\mu = \mu_1 \wedge \mu_2 \wedge \mu_3 = H(C||D)$. To counter this, a random number r has to be included. The final setting is,

$$\mu = R_a(\ 0\ ||\ r\ )\ \bigwedge_{i\ \ [n]} R_a(\ 1\ ||\ i\ ||M[i]) \tag{3}$$

In order to make sure the incremental hash function is collision free and secure against message substitution attack, some redundancy bits must be added. Furthermore, a user must keep record on the sequence of the counters. Micciancio [7] came out with an idea of oblivious data structure to solve the privacy problem. An incremental algorithm is said to be private or historically free if the final hash value yields no information on the previous text modification operations that have been applied to the final message. Therefore, (3) is partially historically free because the adversary knows which message blocks have been modified.

3 The Proposed **PCIHF**

In this section, we explain the construction of the proposed Pair Chaining & Modular Arithmetic Combining Incremental Hash Function (PCIHF). Then, we show how PCIHF performs incremental updating of hash values over some text modification functions. Its efficiency in term of total number of computation steps required for each incremental updating process will also be analyzed. For ease of discussion, we will only consider PCIHF as single pair block chaining throughout this paper.

3.1 The Construction of **PCIHF**

Initialisation. For the sake of simplicity, we ignore the nonce and the Merkle-Damgård strengthening (MD-strengthening), which are usually be added as the first and last block in the original message respectively. Eventually, these are essential in practical schemes to guard against prefix and postfix type of attacks. We apply a standard padding method which pad bit-'1' and minimum number of bit-'0' at the tail of the message M so that the message length, $|M|$ is a multiple of b-bits. Finally, an input message $M = M[1]M[2] \ldots M[n]$, with single message block $M[i] \in \{0,1\}^b$, for $i \in [n]$. In order to make our construction works, we make a stronger assumption which is that all the message blocks are distinct. This means that $M[i] = M[j]$ for $i, j \in [n]$ and $i = j$. Finally, $R : \{0,1\}^{2b} \rightarrow \{0,1\}^k$ is a standard collision-free compression function (or pseudo-random function) which has an input with $2b$-bits and output k-bits "random" string.

Randomize. We implement the concept of randomize-then-combine paradigm. Instead of using a counter for each block, we chain the block before performing the randomizing process. The randomizer must definitely be collision-free; otherwise, the entire construction fails to be collision-free. After going through R, we obtain a series of intermediate hash values for $i \in [n-1]$ as shown below,

$$h[i] = R(M[i]||M[i+1]) \tag{4}$$

Combining. Since hash function is public, we have to use some modular arithmetic operations, if not, it is not secure (Bellare et. al. [4]). Additionally, these operations must be associative, commutative and invertible in a particular group, so that PCIHF is parallelizable and incremental. Some of the suitable operations are addition and multiplication. Later, we will prove that the hardness of breaking additive operation is equal to breaking the weighted subset sum problem. In this paper, we choose modular summation as the combining operation. We fixed the length of final hash value to $k = 160$ bits. The final hash value μ for PCIHF could be obtained as follows:

$$\mu = \sum_{i \in [n-1]} h[i] (\bmod 2^{160} + 1) \tag{5}$$

Here we assume that Random Access Machine model (RAM machine) is chosen and the original (or previous) message is stored inside the memory, so time taken to access the corresponding block could be ignored. Hence, total computation steps taken by PCIHF for obtaining the first original hash value T_t is equal to $n(R_t + C_t) - (R_t + 2C_t)$, where C_t and R_t are the constant time taken for combining and randomizing process respectively. Its efficiency decrease dramatically if compared to standard hash function which only takes $n(R_t/2)$. Nevertheless, PCIHF is incremental and parallelizable.

3.2 Text Modification Function f and Efficiency

In addition to single block update operations, PCIHF can also handle multiple blocks and messages update operations. Let $\sigma_i \in \{0,1\}^b$ be a set of distinct message blocks (also with $M[i]$) for $i \in [n]$. A set of text modification function F, so that the updated message $M' = F(M, f)$, where f is the modification augment. The description of f is given below:

Block Replacement. f = replace$(p, i, \sigma_1, ..., \sigma_p)$ denotes the sequence of p-blocks starting from location i of previous M with hash value μ are replaced by

$\sigma_1, ..., \sigma_p$. The updated hash value is:

$$\mu \ = \mu + R(M[i-1]||M[\sigma_1]) + R(M[\sigma_p]||M[i+p]) \\ - \sum_{j=0,1,...,p} R(M[i-1+j]||M[i+j]) + \sum_{j\ \ [p-1]} R(\sigma_j||\sigma_{j+1}) \tag{6}$$

The total cost $T_t = 2(1+p)(R_t + C_t)$.

Block Deletion. $f = \text{delete}(p, i)$ denotes the sequence of p-blocks starting from location i of M with μ are deleted. The updated hash value is:

$$\mu \ = \mu - \sum_{j=0,1,...,p} R(M[i-1+j]||M[i+j]) + R(M[i-1]||M[i+p]) \tag{7}$$

The total cost $T_t = (2+p)(R_t + C_t)$.

Block Insertion. $f = \text{insert}(p, i, \sigma_1, ..., \sigma_p)$ denotes the p-blocks $(\sigma_1, ..., \sigma_p)$ are inserted after the location i of M with hash value μ. The updated hash value is:

$$\mu \ = \mu - R(M[i]||M[i+1]) + R(\sigma_p||M[i+p]) + R(M[i]||\sigma_1) + \sum_{j\ \ [p-1]} R(\sigma_j||\sigma_{j+1}) \tag{8}$$

The total cost $T_t = (2+p)(R_t + C_t)$ is the same as the deletion function.

Block Cut-&-Paste. $f = \text{cut} - \text{paste}(p, i, j)$ denotes the sequence of p-blocks from the location i are cut and pasted after the location j of M. The updated hash value is:

$$\mu \ = \mu - R(M[i-1]||M[i]) - R(M[i+p-1]||M[i+p]) \\ + R(M[j]||M[i]) + R(M[i-1]||M[i+p]) \\ - R(M[j]||M[j+1]) + R(M[i+p-1]||M[j+1]) \tag{9}$$

The total cost $T_t = 6(R_t + C_t)$, which is a constant value.

Multi-Messages Merging. $f = \text{merge}(M_1, M_2, \ldots, M_p \ )$ where p is the total number of messages, denotes the multiple documents $M_1, M_2, \ldots, M_p$ with respective hash value $\mu_1, \mu_2, \ldots, \mu_p$ and the number of message block $n_1, n_2, \ldots, n_p$ are then merged into one. The updated hash value is:

$$\mu \ = \sum_{j\ \ [p\ \ -1]} R(M_j[n_j]||M_{j+1}[1]) + \sum_{j\ \ [p\ \ ]} \mu_j \tag{10}$$

The total cost $T_t = p\ \ (R_t + 2C_t) - (R_t + 2C_t)$. It is only proportional to p .

3.3 Advantages of **PCIHF**

Omitting the Index. In previous proposed incremental schemes, each block is concatenated with a counter. Therefore, the size of the hash value increases proportionally to the message size and the number of updating process. Contrarily, our PCIHF outputs a fixed size of final hash value. This directly reduces the space storage required. Furthermore, no information is leaked during the updating process. The recipient needs only the updated message and the hash value for the verification procedure. Therefore, our proposed scheme is oblivious and total tamper-proof secure. In addition to that, we do not need to keep a record on the intermediate values or the sequences of the orientation of the block counter as *virtual message* (Bellare et. al. [2]). This is hard to be achieved in the incremental tree scheme, where all the nodal information is needed to be stored.

Using R Function as a 'Black Box'. Using an existing hash function for the randomized operation makes our algorithm more portable and simplifies our analysis. This is because the existing 'black box', such as SHA-1, have been studied in details and proven as a strong candidate.

Fast Combination Operation. We stick to the summation operation which is the simplest arithmetic operation as it is a very fast operation for current microprocessor compared to other operations (modular multiple operation, etc). Of course, using a combination operation with higher complexity will increase the hardness of attacking the scheme.

Further Improvement. Applying multi-blocks chaining instead of just two blocks per pair could further speed up the proposed scheme. Furthermore, the performance of PCIHF could also be improved by storing the original intermediate values $h[i]$ which are needed during the updating process, like in the tree scheme. If this is fully implemented, the updating process can be done in almost constant time, however we will end up using more space.

4 Analysis of **PCIHF**

In this section, we show that the proposed PCIHF is not only universal one-way but also collision-free. Without loss of generality, we take SHA-1 as the randomizer and the modular addition operator as the combining operator.

4.1 **PCIHF**: Universal One-Way

We show that the strength of PCIHF can be related to the standard modular knapsack problem (subset sum problem). This only suffices to show that it is a universal one-way hash function UOWHF, as defined in Impagliazzo & Naor [6] and Naor & Yung [9]. In order to show that it possesses the collision-free property, we have to make a stronger examination, which will be discussed in the next section.

According to Bellare et. al. [4], formally, we construct a 2nd *preimage-finder* (PF_1, PF_2). Firstly, algorithm PF_1 outputs a x. Then, the second PF_2 will base on the input x to get y. We fix the random oracle in the randomizing process, which only allows PF_2 to access but not PF_1. Finally, the finder is successful if PCIHF(x) = PCIHF(y) and $x = y$. This type of finder is a weaker notion of security, because the adversary is not allowed to choose x, as compared to collision-free where birthday attack is allowed. We conclude that a proven universal one-wayness cryptographic hash function is not necessarily collision-free, but it is always true vice versa.

Definition 1. *A 2^{nd} preimage-finder (t, q, ε)-breaks a hash family if given an oracle, it makes at most q oracle queries, runs in time t and finds a collision with probability at least ε. A hash family is (t, q, ε)-universal-one-way, iff there is no 2^{nd} preimage-finder which (t, q, ε)-breaks it.*

Standard Modular Subset-Sum Problem. Given a k-bit integer N and q distinct numbers $a_1, \ldots, a_q \quad Z_N$, we can construct a (k, q)-standard knapsack problem. For optimization purposes, we do assume that $N = 2^k + 1$. Then, we are asked to get a set of weights $w_1, \ldots, w_q \quad \{0, 1\}$, not all zero, where

$$\sum_{i \quad [q]} w_i a_i \equiv 0 (\bmod N) \qquad (11)$$

The task is hard, when $a_1, \ldots, a_q$ are chosen randomly in Z_N and N is sufficiently large.

Definition 2. *A (k, q)-standard knapsack problem is (t, ε)-hard, iff there is no algorithm which can find a solution to an instance $N, a_1, \ldots, a_q$ of the (k, q)-standard knapsack problem with probability more then ε in time t, provided that $a_1, \ldots, a_q$ are selected uniformly and independently in Z_N.*

After relating the universal one-way of PCIHF to the standard modular knapsack problem, we obtain *Theorem 1*.

Theorem 1. *Let k and q be integers such that the (k, q)-standard knapsack problem is (t , ε)-hard. Then PCIHF is (t, q, ε)-universal one-way family of hash functions where $t = t - \Omega(2bq)$ and $\varepsilon = 2\varepsilon$.*

Proof: The proof is given in *Appendix A*.

4.2 **PCIHF**: Collision-Free

A collision-finder, CF is an algorithm that tries to output a pair of collision. Here, we will examine the probability that it is successful.

Definition 3. *A collision-finder CF (t, q, ε)-breaks a hash family H if it runs in time t, makes at most q oracle queries, and finds a collision in H with probability at least ε. We say that H is (t, q, ε)-collision-free if there is no collision-finder which (t, q, ε)-breaks H.*

Weighted Modular Knapsack Problems. By giving a k-bit integer N and q distinct numbers $a_1, \ldots, a_q \quad Z_N$, we can construct a (k, q)-standard knapsack problem. We are asked to get a set of weights $w_1, \ldots, w_q \quad \{-1, 0, +1\}$, but not all zero, as in (11). In other words, we need to find two disjoint non-empty subsets I and J, where $I, J \quad [q]$, such that,

$$\underset{i \quad I}{\square} a_i - \underset{j \quad J}{\square} a_j \equiv 0 \qquad (12)$$

Note that set I and J are the set of indices i and j such that $w_i = +1$ and $w_j = -1$.

Definition 4. *A (k, q)-weighted knapsack problem is (t, ε)-hard, iff there is no algorithm which can find a solution to an instance $N, a_1, \ldots, a_q$ of the (k, q)-standard knapsack problem with probability more then ε in time t, provided that $a_1, \ldots, a_q$ are selected uniformly and independently in Z_N.*

Balance Problem. We identify a computational problem – *balance problem* (Bellare et.al [4]) that can be defined in an arbitrary group. It unifies the collision-free treatment of PCIHF based on the underlying arithmetic modular combining process. We only show how the hardness of the balance problem for the additive groups is the weighted subset sum problem, although it could be implemented in any algebraic group. The construction of balance problem is as follows. Let G be some family of groups and n is an integer. In the (G, n)-balance problem we are given a group $G \quad G$ and a sequence $a_1, \ldots, a_n \quad G$. We must find the weights $w_1, \ldots, w_n \quad \{-1, 0, +1\}$, not all zero, such that

$$\underset{i \quad [n]}{\wedge} a_i^{w_i} = e(\mathrm{mod}\, N) \qquad (13)$$

$\wedge$ is the group operation and e is the identity element in the group. The security of the proposed PCIHF paradigm can be related to the balance problem in the underlying class of groups, if the group is hard. Recall that q refers to the number of computations of oracle R. Since R is ideal, it maps $\{0, 1\}^{2b}$ to G. *Lemma 1* says that if the balance problem is hard then the corresponding family of hash functions is collision-free.

Lemma 1. *Let G and q be integers such that the (G, q)-balance problem is (t, ε)-hard. Then, PCIHF is (t, q, ε)-collision-free family of hash functions where $t = t - \Omega(2bq)$ and $\varepsilon = \varepsilon$.*

Proof: The proof is given in *Appendix B.*

Theorem 2. *Let k and q be integers such that the (k, q)-weighted knapsack problem is (t, ε)-hard. Then PCIHF is (t, q, ε)-collision-free family of hash functions where $t = t - \Omega(2bq)$ and $\varepsilon = \varepsilon$.*

Proof: We have related the security of the PCIHF to the balance problem, which will further be reduced to a conventional hard problem. Obviously, since additive operator is chosen, the corresponding balance problem of PCIHF turns into (12), which is essentially equivalent to weighted knapsack problem. Therefore, PCIHF is proven to be a collision-free hash function as long as the weighted modular knapsack problem is hard.

5 Conclusion

We have introduced a new hash function PCIHF which is incremental and parallelizable. Time taken for the Cut-&-Paste text modification function is constant, regardless of the amount of blocks involved. Also, for multiple message merging, the time taken only depends on the number of messages. Table 1 summarizes the time taken by PCIHF for various modification functions. Finally, we show that PCIHF is not only universal one-way but also collision-free.

Table 1. Complexity of computation

Text modification function, f	Total cost, T_t
Replace($p, i, \sigma_1, .., \sigma_p$)	$2(1 + p)(R_t + C_t)$
Delete/insert(p, i)	$(2 + p)(R_t + C_t)$
cut − paste(p, i, j)	$6(R_t + C_t)$
Merge($M_1, M_2, \ldots . M_p$)	$p^{\square}(R_t + 2C_t) - (R_t + 2C_t)$

References

1. Bellare, M., Goldreich, Oded., Goldwasser, S.(1994). Incremental Cryptography: The Case of Hashing and Signing. *Advances in Cryptology – Proceedings of Crypto '94*, 216-233
2. Bellare, M., Goldreich, Oded., Goldwasser, S.(1995). Incremental Cryptography and Application to Virus Protection. *Proceedings of the 27th ACM symposium on the Theory of Computing, May 1995*, 45-56
3. Bellare, M., Guerin, R., Rogaway,P. (1995). XOR MACs: New Methods for Message Authentication Using Finite Pseudorandom Functions. *Advances in Cryptology – Proceedings of Crypto '95*, 15-29
4. Bellare, M., Micciancio, D. (1997). A New Paradigm for Collision-free Hashing: Incrementality at Reduced Cost. *Advances in Cryptology – Proceedings of Eurocrypt '97*
5. Fischlin, M. (1997). Incremental Cryptography and Memory Checkers. *Advances in Cryptology – Proceedings of Eurocrypt '97*, 393-408
6. Impagliazzo, R., Naor, M. (1996). Efficient Cryptographic Schemes Provably as Secure as Subset Sum. *Journal of Cryptology, Vol. 9, No.4, Autumn 1996*
7. Micciancio, D.(1997). Oblivious Data Structures: Applications to Cryptography. *Proceedings of the 29th Annual symposium on the Theory of Computing, 1997*

8. Menezes, A. J., Oorschot, P. C. van, & Vanstone, S. A. (1997). *Handbook of Applied Cryptography*. Boca Raton: CRC Press
9. Naor, M., Yung, M. (1989). Universal One-Way Hash Functions and Their Cryptographic Applications. *Proceedings of the 21st Annual Symposium on Theory of Computing, ACM, 1989*

Appendix A: Proof of Theorem 1

A 2^{nd} *preimage-finder* (PF_1, PF_2) which (t, q, ε)-breaks PCIHF is given. Algorithm PF_1 outputs a string x with n-block. Then, PF_2 takes an oracle R, eventually outputs a string y with m-block where (x, y) is a pair of collision for PCIHF. Meanwhile, we construct algorithm P to solve the (k, q)-standard knapsack problem. N is an integer with k-bit and a list of $a_1, \ldots, a_q \in_R Z_N$. At first, P runs PF_1 to obtain x. Then it runs PF_2 with an input N, and calls R to answer its random oracle queries. There are two strategies for performing the task. We assume that those oracle queries are distinct and only be made on block replacement or appends to x. The answers are also distinct and randomly distributed. Since, we use pair-chaining technique and according to some restrictions that have been made in PF_2, a successful collision output y is as follows. For $i = [m]$, y_i must be either x_i or z_{k_i}. Note that $x_i, y_i, z_i, w_i \in \{0,1\}^b$ and k_i, $i_j \in [q]$. Let I be the set of indices i, where $y_i = z_{k_i}$ and $I \subseteq [m]$.

Strategy A: For $i = [n-1]$, answering $R(x_i||x_{i+1}) = b$, where $b_i \in Z_N$. For $j = [q - 2(2n-1)]$, answering $R(w_{k_j}||w_{k_{j+1}})$ as follows:

If $(w_{k_j} = x_j)$ and $(w_{k_{j+1}} = z_{k_{j+1}})$,

$$R(x_j||z_{k_{j+1}}) = \begin{cases} b_j + \frac{a_{k_{j+1}}}{2}, & \text{for } 1 \le j \le n-1 \\ a_{k_{n+1}}, & \text{for } j = n \end{cases}$$

else if $(w_{k_j} = z_{k_j})$ and $(w_{k_{j+1}} = x_{j+1})$,

$$R(z_{k_j}||x_{j+1}) = \begin{cases} b_1 + a_{k_1}, & \text{for } j = 1 \\ b_j + \frac{a_{k_j}}{2}, & \text{for } 1 < j < n \end{cases}$$

else,

$$R(z_{k_j}||z_{k_{j+1}}) = \begin{cases} b_1 + a_{k_1} + \frac{a_{k_2}}{2}, & \text{for } j = 1 \\ b_j + \frac{a_{k_j}}{2} + \frac{a_{k_{j+1}}}{2}, & \text{for } 1 < j \le n-1 \\ \frac{a_{k_n}}{2} + a_{k_{n+1}}, & \text{for } j = n \\ a_{k_{j+1}}, & \text{for } j > n \end{cases}$$

We choose this strategy, if we do assume $m \ge n$, meaning $|y| \ge |x|$. For $i = [m-1]$, let I_1 be the set of indices i, such that $(y_i||y_{i+1}) = (x_i||z_{k_{i+1}})$; I_2 be the set of indices i, such that $(y_i||y_{i+1}) = (z_{k_i}||x_{i+1})$; and I_3 be the set of indices i, such that $(y_i||y_{i+1}) = (z_{k_i}||z_{k_{i+1}})$ respectively. Practically, all these set I_1, I_2 and I_3 are totally disjoint. Therefore, $I = \{i : i \in I_2 \cup I_3\} \cup \{i+1 : i \in I_1 \cup I_3\}$.

$$\text{PCIHF}(x) = \sum_{i=1}^{n-1} R(x_i||x_{i+1}) = \sum_{i=1}^{n-1} b_i (\bmod N)$$

$$\mathrm{PCIHF}(y) = \sum_{i=1}^{m-1} R(y_i||y_{i+1}) = \sum_{i \in I_1 \cup I_2 \cup I_3} R(y_i||y_{i+1}) + \sum_{i \in [n-1]-I} R(x_i||x_{i+1})$$

$$= \sum_{i=1}^{n-1} b_i + \sum_{i \in I} a_{k_i} = \mathrm{PCIHF}(x) + \sum_{i \in I} a_{k_i} (mathop \mathrm{mod} N)$$

If the finder is successful, then (x, y) will be a collision of PCIHF. In other words, it is equivalent to finding a subset $I \subseteq [m]$ such that, $\sum_{i \in I} a_{k_i} \equiv 0 (\mathrm{mod}\, N)$. This is a solution to the given standard knapsack problem.

Strategy B. For $i = [n-2]$ answering $R(x_i||x_{i+1}) = a_i$, and answering $R(x_{n-1}||x_n) = \sum_{i=n-1}^{q} a_i (\mathrm{mod}\, N)$. For $j = [q-(n-1)]$, answering $R(w_{i_j}||w_{i_{j+1}})$, where $w_{i_j} = z_{k_j}$ and $w_{i_{j+1}} \in \{x_j, z_{k_{j+1}}\}$, then $R(z_{k_j}||w_{i_{j+1}}) = a_{j+(n-1)}$. We choose this strategy, if we assume $m < n$. Let J is the set of indices j, such that $y_j = z_{k_j}$, where $J \subseteq [m]$ and $k_j \in [q]$ for $j \in [m]$. Let $J = \{j + (n-1) : j \in J\} \subseteq [n-2] - J$ and $J = [q] - J$.

$$\mathrm{PCIHF}(y) = \sum_{j \in J} a_j (\mathrm{mod}\, N)$$

$$\mathrm{PCIHF}(x) = \sum_{j=1}^{q} a_j = \sum_{j \in J} a_j + \sum_{j \in J} a_j = \mathrm{PCIHF}(y) + \sum_{j \in J} a_j (\mathrm{mod}\, N)$$

Similiarly, if the finder is successful, then (x, y) is a collision of PCIHF, and also $J = \emptyset$. In other words, it is equivalent to asking us to find a subset $J \subseteq [q]$ such that,

$$\sum_{j \in J} a_j \equiv 0 (\mathrm{mod}\, N).$$

Once again, this is a solution to the given standard knapsack problem in *Definition 2*. From the view of PF_2, these two strategies are identical and totally independent. P can successfully finds a collision, if he makes a right choice which is only half of the probability of solving the given knapsack problem.

Appendix B: Proof of Lemma 1

From Bellare et. al. [4], collision-finder CF, which take $\langle G \rangle$ (description of particular group) and an oracle R, and eventually outputs a pair of distinct strings $x = x_1 \ldots x_n$ and $y = y_1 \ldots y_m$, but is collided to each other. We construct a algorithm P to solve the (G, q)-balance problem. P runs CF on input $\langle G \rangle$, answering its random oracle queries with the values $a_1, \ldots, a_q$ in order. We assume oracle queries are not repeated. Notice the answers to the oracle queries are uniformly and independently distributed over G, as they would be if $R : \{0, 1\}^{2b} \rightarrow G$ is a random function. We will let Q_i denote the i-th oracle query of CF, namely the one answered by a_i, so that $R(Q_i) = a_i$, and we let $Q = \{Q_1, \ldots, Q_q\}$. Finally, collision-finder outputs a pair of collisions (x, y). We know this means,

$$\bigwedge_{i=[n-1]} R(x_i||x_{i+1}) = \bigwedge_{j=[m-1]} R(y_j||y_{j+1})$$

Note that the operations are in G. Strings x and y are not neceesarily of the same size; that is, m may not be equal to n. We will construct a solution to the balance problem from x and y. Let $x_i = x_i||x_{i+1}$ for $i \quad [n-1]$ and for $y_i = y_i||y_{i+1}$ for $i \quad [m-1]$. We let $f_x(i)$ be the (unique) value $j \quad [q-1]$ such that $x_i = q_j$ and let $f_y(i)$ be the (unique) value $j \quad [q-1]$ such that $y_i = q_j$. We then let $I = \{f_x(i) : i \quad [n-1]\}$ and let $J = \{f_y(i) : i \quad [m-1]\}$ be, respectively, the indices of queries corresponding to x and y. We rewrite the above equation as,

$$\bigwedge_{i \quad I} a_i = \bigwedge_{j \quad J} a_j$$

We know that $x = y$ and so $I = J$. Now for $i = 1, \ldots, q$ let us define,

$$w_i = \begin{cases} -1 \text{ if } i \quad J - I \\ 0 \text{ if } i \quad I \cap J \\ +1 \text{ if } i \quad I - J \end{cases}$$

Then the fact that $I = J$ means that not all $w_1, \ldots, w_q$ are 0, this implies that

$$\bigwedge_{i=1}^{q} a_i^{w_i} = e$$

is equivalent to the defined balance problem. Therefore, the probability that we find a solution to the balance problem is exactly that with CF outputs a collision, and the time taken can be estimated.

Multiples of Primitive Polynomials over GF(2)

Kishan Chand Gupta[1] and Subhamoy Maitra[2]

[1] Applied Statistics Unit, Indian Statistical Institute,
203, B T Road, Calcutta 700 035, India
kishan_t@isical.ac.in

[2] Computer and Statistical Service Center, Indian Statistical Institute,
203, B T Road, Calcutta 700 035, India
subho@isical.ac.in

Abstract. In this paper we concentrate on finding out multiples of primitive polynomials over GF(2). Given any primitive polynomial $f(x)$ of degree d, we denote the number of t-nomial multiples ($t < 2^d - 1$) with degree less than $2^d - 1$ as $N_{d,t}$. We show that $(t-1)N_{d,t} = \binom{2^d-2}{t-2} - N_{d,t-1} - \frac{t-1}{t-2}(2^d - t + 1)N_{d,t-2}$, with the initial conditions $N_{d,2} = N_{d,1} = 0$. Moreover, we show that the sum of the degree of all the t-nomial multiples of any primitive polynomial is $\frac{t-1}{t}(2^d - 1)N_{d,t}$. More interestingly we show that, given any primitive polynomial of degree d, the average degree $\frac{t-1}{t}(2^d - 1)$ of its t-nomial multiples with degree $\leq 2^d - 2$ is equal to the average of maximum of all the distinct $(t-1)$ tuples from 1 to $2^d - 2$. In certain model of Linear Feedback Shift Register (LFSR) based cryptosystems, the security of the scheme is under threat if the connection polynomial corresponding to the LFSR has sparse multiples. We show here that given a primitive polynomial of degree d, it is almost guaranteed to get one t-nomial multiple with degree $\leq 2^{\frac{d}{t-1} + \log_2(t-1) + 1}$.

Keywords : Primitive Polynomials, Galois Field, Polynomial Multiples, Cryptanalysis, Stream Cipher.

1 Introduction

Linear Feedback Shift Register (LFSR) is used extensively as pseudorandom bit generator in different cryptographic schemes and the connection polynomial of the LFSRs are the polynomials over GF(2) (see [3,12,2] for more details). To get the maximum cycle length this connection polynomial need to be primitive [9]. To resist cryptanalytic attacks, it is important that these primitive polynomials should be of high weight and also they should not have sparse multiples [11,1] (see also [7] and the references in this paper for current research on cryptanalysis in this direction). With this motivation, finding out sparse multiples of primitive polynomials has received a lot of attention recently, as evident from [6,4]. We here concentrate on this problem and show that sparse multiples of primitive polynomials are not hard to find. Our observations raise serious questions on the safety of a certain class of LFSR based cryptosystems [12,2].

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 62–72, 2001.

First we concentrate on an enumeration problem. It is to find out the total number of t-nomial multiples of a d-degree primitive polynomial $f(x)$. We look into the multiples upto degree $2^d - 2$, since the exponent of any d-degree primitive polynomial is $2^d - 1$. In [4], it has been shown that given any primitive polynomial of degree d, it has $(2^{d-1} - 1)$ trinomial multiples. To generalize this, we here concentrate on a more involved counting problem. We show that given any primitive polynomial of degree d, it has exactly $N_{d,t} = \frac{\binom{2^d-2}{t-2} - N_{d,t-1} - \frac{t-1}{t-2}(2^d - t + 1)N_{d,t-2}}{t-1}$ many t-nomial multiples. Taking the initial conditions $N_{d,2} = N_{d,1} = 0$, it is easy to see that our formula provides $N_{d,3} = 2^{d-1} - 1$. Also we show that the sum of the degree of all the t-nomial multiples is $\frac{t-1}{t}(2^d - 1)N_{d,t}$. This gives that given any primitive polynomial of degree d, the average degree of its t-nomial multiples with degree $\leq 2^d - 2$ is $\frac{t-1}{t}(2^d - 1)$. This value is also equal to the average of maximum of all the distinct $(t-1)$ tuples from 1 to $2^d - 2$.

In the other direction, it is important to find out the sparse multiples. Thus, it is very clear that the main problem is to find out the least degree t-nomial multiple of $f(x)$. The simplest algorithm to find such a t-nomial multiple is to check all the t-nomials starting from degree d and then go on checking upwards until such a multiple is found. The run time of such an exhaustive search algorithm is output sensitive. In fact, in [6,4], it has been shown that it is possible to find a trinomial multiple of degree less than or equal to $\frac{2^d+2}{3}$. However, this result is not encouraging since for large d, it will take a long time to get the least degree trinomial multiple. On the other hand, if it can be shown that it is possible to get a multiple in a much lower range, then the exhaustive search algorithm seems reasonable. We use simple statistical assumptions to show that it is almost guaranteed to get a t-nomial multiple of degree less than or equal to $2^{\frac{d}{t-1}+\log_2(t-1)+1}$. In particular, it is expected to get a trinomial multiple of degree less than or equal to $2^{\frac{d}{2}+2}$ which is much better than the expression $\frac{2^d+2}{3}$ given in [6,4].

Let us now present an example for clarity. Consider a randomly chosen 31 degree primitive polynomial $f(x) = x^{31} + x^{30} + x^{29} + x^{28} + x^{27} + x^{25} + x^{24} + x^{23} + x^{22} + x^{21} + x^{16} + x^{14} + x^{13} + x^{11} + 1$. Note that this polynomial has 15 terms. We find the following sparse multiples of $f(x)$. Also in bracket we provide the time taken by a simple C language implementation (Unix operating system on SUN Enterprise Server E 3000, four 250 MHz CPU, 1 GB RAM). The sparse multiples are $x^{89498} + x^{8581} + 1$ (1546 sec), $x^{3286} + x^{2417} + x^{1001} + 1$ (2280 sec), $x^{536} + x^{497} + x^{292} + x^{199} + 1$ (1359 sec) and $x^{150} + x^{148} + x^{124} + x^{122} + x^{117} + 1$ (218 sec). Our bound of $2^{\frac{d}{t-1}+\log_2(t-1)+1}$ provides the values 185362, 7740, 1722, 734 for the value of $t = 3, 4, 5, 6$ respectively. Note that we get sparse multiples at low degree in small time. This identifies that the use of moderate degree primitive polynomials in cryptographic schemes is not very safe [11,1] as it is easy to get sparse multiples.

A polynomial with t non zero terms, one of them being the constant term is called t-nomial, or in other words a polynomial of weight t. By a sparse multiple we generally consider t-nomial multiples for $t \leq 9$. Note that the roots of a primitive polynomials are the primitive elements of $GF(2^d)$. Consider a primitive polynomial $f(x)$ of degree d and let α be a root of this, i.e., $f(\alpha) = 0$. Consider $g(x)$ is a t-nomial multiple of $f(x)$ having degree $\leq 2^d - 2$. Then it is very clear that $g(\alpha) = 0$. In other words, $g(x) = 1 + x^{i_1} + x^{i_2} + \ldots + x^{i_{t-1}}$ $(1 \leq i_1 < i_2 < \ldots < i_{t-1} \leq 2^d - 2)$ is a t-nomial multiple of $f(x)$, iff $1 + \alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-1}} = 0$. For more details on finite fields, the reader is referred to [10,9].

2 Enumeration of t-nomial Multiples

In this section first we prove the following important result on the number of t-nomial multiples of a degree d primitive polynomial.

Theorem 1. $N_{d,t} = \dfrac{\binom{2^d-2}{t-2} - N_{d,t-1} - \frac{t-1}{t-2}(2^d - t + 1)N_{d,t-2}}{t-1}$.

Proof. Let us consider a primitive polynomial $f(x)$ of degree d. Any t-nomial multiple of $f(x)$ can be written as $1 + x^{i_1} + x^{i_2} + \ldots + x^{i_{t-2}} + x^{i_{t-1}}$ for $1 \leq i_1 < i_2 < \ldots < i_{t-2} < i_{t-1} \leq 2^d - 2$. Hence, if we consider α be a root of $f(x)$, then $1 + \alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-2}} = \alpha^{i_{t-1}}$.

Now consider the expression $1 + \alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-2}}$. We can take any $(t-2)$ valued tuple $< i_1, i_2, \ldots, i_{t-2} >$ out of $2^d - 2$ possible values. This can be done in $\binom{2^d-2}{t-2}$ ways. For each such combination $< i_1, i_2, \ldots, i_{t-2} >$, we have $1 + \alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-2}}$ must be one of the elements $0, 1, \alpha^k, \alpha^l$, where, $k \in \{i_1, i_2, \ldots, i_{t-2}\}$, and $l \in \{1, \ldots, 2^d - 2\} - \{i_1, i_2, \ldots, i_{t-2}\}$. Let us consider the four cases separately.

1. We first consider $1 + \alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-2}} = 0$. This implies that this is a $(t-1)$-nomial multiple of $f(x)$. Such a situation will occur for each of the $(t-1)$-nomial multiples. This count is thus $N_{d,t-1}$. This need to be subtracted from $\binom{2^d-2}{t-2}$.
2. Next we consider, $1 + \alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-2}} = \alpha^k$, where $k \in \{i_1, i_2, \ldots, i_{t-2}\}$. Consider $k = i_r$. Then, $1 + \alpha^{i_1} + \ldots + \alpha^{i_{r-1}} + \alpha^{i_{r+1}} + \ldots + \alpha^{i_{t-2}} = 0$. This implies that this is a $(t-2)$-nomial multiple of $f(x)$. Such a situation will occur for each of the $(t-2)$-nomial multiples. This count is thus $N_{d,t-2}$. This again need to be subtracted from $\binom{2^d-2}{t-2}$.
3. Next we consider $1 + \alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-2}} = 1$. Then we get $\alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-2}} = 0$. The number of cases for which $\alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-2}} = 0$, where, $1 \leq i_1 < i_2 < \ldots < i_{t-2} < i_{t-1} \leq 2^d - 2$, must be subtracted from the expression $\binom{2^d-2}{t-2}$. So we need to count the cases for which $\alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-2}} = 0$.

   Now consider the expression $\alpha^i(1 + \alpha^{j_1} + \alpha^{j_2} + \ldots + \alpha^{j_{t-3}})$ for $i = 1$ to $2^d - 2$, where $1 + x^{j_1} + x^{j_2} + \ldots + x^{j_{t-3}}$ is a $(t-2)$-nomial multiple of $f(x)$. Out of these values of i, we will get $(t-3)$ values for which $j_p + i = 2^d - 1$, $1 \leq p \leq t - 3$. So these many values must be subtracted from the counting which gives us $((2^d - 2) - (t - 3)) = 2^d - t + 1$ different cases. This is because we like to count the cases where $\alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-2}} = 0$ and none of the α^{i_q} can be 1, $1 \leq q \leq t-2$.

The terms $\alpha^i(1+\alpha^{j_1}+\alpha^{j_2}+\ldots+\alpha^{j_{t-3}})$ for $i = 1$ to $2^d - 2$, will produce $(t-3)$ number of $(t-2)$-nomial multiples of the form $1 + x^{k_1} + x^{k_2} + \ldots + x^{k_{t-3}}$. Also $1 + x^{j_1} + x^{j_2} + \ldots + x^{j_{t-3}})$ need to be counted here as it by itself is a $(t-2)$-nomial multiple. Thus whenever we start from any $(t-2)$-nomial multiple, it includes the case of $((t-3)+1) = (t-2)$ number of $(t-2)$-nomial multiples. Hence, if we choose each of the $(t-2)$-nomial from the total $N_{d,t-2}$ choices, ultimately each will be repeated $(t-2)$ times. So total number of cases for which for which $\alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-2}} = 0$, where, $1 \leq i_1 < i_2 < \ldots < i_{t-2} < i_{t-1} \leq 2^d - 2$ is $\frac{1}{t-2}(2^d - t + 1)N_{d,t-2}$. This need to be subtracted from $\binom{2^d-2}{t-2}$.

4. Thus we get the t-nomial count $\binom{2^d-2}{t-2} - N_{d,t-1} - N_{d,t-2} - \frac{1}{t-2}(2^d - t + 1)N_{d,t-2}$. In all these cases, it is guaranteed that $1 + \alpha^{i_1} + \alpha^{i_2} + \ldots + \alpha^{i_{t-2}} = \alpha^{i_{t-1}} = \alpha^l$, where, $l \in \{1, \ldots, 2^d - 2\} - \{i_1, i_2, \ldots, i_{t-2}\}$. Now we can bring α^l in the left hand side by shifting any of the α^i's in the right hand side, $i \in \{i_1, i_2, \ldots, i_{t-2}\}$. This means that each of the t-nomials are counted $(t-1)$ times.

Thus we get that $N_{d,t} = \dfrac{\binom{2^d-2}{t-2} - N_{d,t-1} - \frac{t-1}{t-2}(2^d - t + 1)N_{d,t-2}}{t-1}$.

It is not clear how to solve the recurrence relation of Theorem 1 for any t. However, we solve it for some low values of t as it is important to analyse the sparse multiple. We list the values here.

$$\begin{aligned}
N_{d,3} &= \tfrac{1}{2!}(2^d-2)\\
N_{d,4} &= \tfrac{1}{3!}(2^d-2)(2^d-4)\\
N_{d,5} &= \tfrac{1}{4!}(2^d-2)(2^d-4)(2^d-8)\\
N_{d,6} &= \tfrac{1}{5!}(2^d-2)(2^d-4)(2^d-6)(2^d-8)\\
N_{d,7} &= \tfrac{1}{6!}(2^d-2)(2^d-4)(2^d-6)(2^{2d}-15\cdot 2^d+71)\\
N_{d,8} &= \tfrac{1}{7!}(2^d-2)(2^d-4)(2^d-6)(2^d-8)(2^{2d}-15\cdot 2^d+71)\\
N_{d,9} &= \tfrac{1}{8!}(2^d-2)(2^d-4)(2^d-6)(2^d-8)(2^d-12)(2^{2d}-12\cdot 2^d+62)\\
N_{d,10} &= \tfrac{1}{9!}(2^d-2)(2^d-4)(2^d-6)(2^d-8)(2^d-10)(2^d-12)(2^{2d}-12\cdot 2^d+62)
\end{aligned}$$

The above expressions show that it will not be easy to get a generalized solution for any t. It will be an interesting exercise to find this out. Next we present the following result.

Theorem 2. $\frac{N_{d,t}}{t} = \frac{N_{d,2^d-1-t}}{2^d-1-t}$.

Proof. It is known that for any primitive element α, $1 + \alpha + \alpha^2 + \ldots + \alpha^{2^d-2} = 0$, i.e., $1 + x + x^2 + \ldots + x^{2^d-2}$ is the only (2^d-1)-nomial multiple of any primitive polynomial $f(x)$ of degree d. Now we calculate the number of $(2^d - 1 - t)$-nomial multiples in terms of t-nomial multiples. Whenever $\alpha^{i_1} + \ldots + \alpha^{i_t} = 0$ for $1 \leq i_1 <$

$i_2 < \ldots < i_{t-2} < i_{t-1} \le 2^d - 2$, coupling this with $1 + \alpha + \alpha^2 + \ldots + \alpha^{2^d-2} = 0$, we will get a $(2^d - 1 - t)$-nomial multiple. Similar to item 3 of the proof of Theorem 1, we find that $\alpha^{i_1} + \ldots + \alpha^{i_t} = 0$ for $\frac{2^d-1-t}{t} N_{d,t}$ cases. (Note that in item 3 of the proof of Theorem 1, we have analysed the result for $(t-2)$ terms, which is t terms here). So, $N_{d,2^d-1-t} = \frac{2^d-1-t}{t} N_{d,t}$.

The above theorem can be used to get the number of $(2^d - 1 - t)$-nomial multiples from t-nomial multiples. We already know that $N_{d,1} = N_{d,2} = 0$. This immediately gives that $N_{d,2^d-2} = N_{d,2^d-3} = 0$. Also in the proof of the above theorem we have noted that $N_{d,2^d-1} = 1$. We have found by computer program that the value of $N_{d,t}$ increases strictly from $t = 3$ upto $t = 2^{d-1}$ and then it starts decreasing strictly till $t = 2^d - 4$. Also for $3 \le t \le 2^d - 4$, all the values of $N_{d,t}$ are positive integers. We have checked this for $d = 4, 5, 6$ and could not check further since the count values are extremely large. However, we get the following two results in this direction.

Corollary 1. *For* $t < \frac{2^d+2}{3}$, $N_{d,t} > N_{d,t-1}$.

Proof. We need to get the value of t, such that, $N_{d,t} = (\binom{2^d-2}{t-2} - N_{d,t-1} - \frac{t-1}{t-2}(2^d - t + 1)N_{d,t-2})/(t-1) > N_{d,t-1}$. This gives, $\binom{2^d-2}{t-2} > tN_{d,t-1} + \frac{t-1}{t-2}(2^d - t + 1)N_{d,t-2}$. Consider the case when we overestimate $N_{d,t-1}$ as $\frac{\binom{2^d-2}{t-3}}{t-2}$ and $N_{d,t-2}$ as $\frac{\binom{2^d-2}{t-4}}{t-3}$ and even then, $N_{d,t} > N_{d,t-1}$. From this we get, $1 > \frac{t}{2^d-t+1} + \frac{t-1}{2^d-t+2}$. This gives the bound for t.

Corollary 2. *For* $0 \le i \le 2^{d-1} - 4$, $N_{d,2^{d-1}+i} > N_{d,2^{d-1}-i-1}$.

Proof. From Theorem 2, we have $\frac{N_{d,2^{d-1}+i}}{2^{d-1}+i} = \frac{N_{d,2^d-1-2^{d-1}-i}}{2^d-1-2^{d-1}-i}$. Then the proof follows from $2^{d-1} + i > 2^d - 1 - 2^{d-1} - i$.

Next we prove an important result on the sum of the degree of all the t-nomial multiples for any primitive polynomial $f(x)$.

Theorem 3. *Given any primitive polynomial of degree d, the sum of the degree of all its t-nomial multiples is* $\frac{t-1}{t}(2^d - 1)N_{d,t}$.

Proof. Consider each t-nomial multiple of degree d_r, where $1 \le r \le N_{d,t}$. Now multiply each t-nomial by x^i for $1 \le i \le 2^d - 2 - d_r$. If we consider α as a primitive root of $f(x)$, then for each value of i, $1 \le i \le 2^d - 2 - d_r$, we will get expressions of the form $\alpha^{i_1} + \ldots + \alpha^{i_t} = 0$. Thus, each t-nomial will provide $2^d - 2 - d_r$ such expressions. Hence considering all the t-nomials we will get $\sum_{k=1}^{N_{d,t}}(2^d - 2 - d_r)$ such expressions and all such expressions are distinct. Similar to the proof of Theorem 2, this gives the count of all $(2^d - 1 - t)$-nomial multiples, which we denote as $N_{d,2^d-1-t}$. Moreover, from Theorem 2, $N_{d,2^d-1-t} = \frac{2^d-1-t}{t} N_{d,t}$, i.e., $\sum_{k=1}^{N_{d,t}}(2^d - 2 - d_r) = \frac{2^d-1-t}{t} N_{d,t}$. Hence $\sum_{k=1}^{N_{d,t}} d_r = (2^d - 2 - \frac{2^d-1-t}{t})N_{d,t} = \frac{t-1}{t}(2^d - 1)N_{d,t}$.

From the above theorem we get that the average degree of a t-nomial multiple is $\frac{t-1}{t}(2^d - 1)N_{d,t}$ divided by $N_{d,t}$, i.e., $\frac{t-1}{t}(2^d - 1)$. This gives that plenty of t-nomial multiples are available at higher degree, whereas there are very few at the lower part.

3 On Least Degree t-nomials

A very simple algorithm to find the least degree t-nomial multiple of a degree d primitive polynomial $f(x)$ is as follows.

Algorithm Find-t-Nomial-Multiple.

```
For $i = d$ to $2^d - 2$,
(a) Consider all possible $t$-nomial $g(x)$ of degree $i$.
(b) If $f(x)$ divides $g(x)$ then report this $t$-nomial and terminate.
```

The time complexity of this algorithm becomes reasonable only when we can expect that we will get such a t-nomial multiple of degree much less than $2^d - 2$ for small t. The requirement of small t is due to the fact that getting sparse multiple of a primitive polynomial can be used for cryptanalytic techniques [11]. If we consider that the least degree t nomial multiple has the value $c_{d,t}$, then the algorithm will run for $i = d$ to $i = c_{d,t}$. In each step we have to consider $\binom{i-1}{t-2}$ tuples. This is because we consider the t-nomial multiple $1 + x^{i_1} + \ldots + x^{i_{t-1}}$, where $1 \leq i_1 < i_2 < \ldots < i_{t-2} < i_{t-1} \leq 2^d - 2$. Now we have the value 1 and the value $i_{t-1} = i$ fixed for the i-th step. Thus we need to check whether $f(x)$ divides $g(x)$ for $\sum_{i=d}^{c_{d,t}} \binom{i-1}{t-2}$ different t-nomials in total. We like to estimate the value of $c_{d,t}$.

Once a primitive polynomial $f(x)$ of degree d is specified, it is very clear that $f(x)$ has $N_{d,t}$ many t-nomial multiples. Note that any t-nomial multiple $1 + x^{i_1} + x^{i_2} + \ldots + x^{i_{t-2}} + x^{i_{t-1}}$ can be interpreted as the $(t-1)$-tuple $< i_1, i_2, \ldots, i_{t-2}, i_{t-1} >$. We have observed that by fixing $f(x)$, if we enumerate all the $N_{d,t}$ different $(t-1)$ tuples, then the distribution of the tuples seems random. To analyse the degree of these t-nomial multiples, we consider the random variate X which is $\max(i_1, i_2, \ldots, i_{t-2}, i_{t-1})$, where $1 + x^{i_1} + x^{i_2} + \ldots + x^{i_{t-2}} + x^{i_{t-1}}$ is a t-nomial multiple of $f(x)$. Also the value of $\max(i_1, i_2, \ldots, i_{t-2}, i_{t-1})$ is i_{t-1}, since we consider the tuples as ordered ones. Let us look at the mean value of the distribution of X. From Theorem 3, it is clear that the average degree of a t-nomial multiple is $\frac{t-1}{t}(2^d - 1)N_{d,t}$ divided by $N_{d,t}$. Thus we get the mean value $\overline{X} = \frac{t-1}{t}(2^d - 1)$.

This mean value $\overline{X}$ clearly identifies that the t-nomials are dense at higher degree and there are very few at the lower degree. On the other hand, for cryptanalysis, we are not interested in getting all the t-nomial multiples. We only concentrate on the least degree t-nomial multiple $g(x)$ of $f(x)$. Thus our motivation is to get an estimate on the degree of $g(x)$. This is not clear from the distribution of X and that is why we like to look into another distribution which seems to be close to the distribution of X.

Let us consider all the $(t-1)$-tuples $< i_1, i_2, \ldots, i_{t-2}, i_{t-1} >$ in the range 1 to 2^d-2. There are $\binom{2^d-2}{t-1}$ such tuples. We consider the tuples in ordered form such that $1 \le i_1 < i_2 < \ldots < i_{t-2} < i_{t-1} \le 2^d-2$. Now consider the random variate Y which is $\max(i_1, i_2, \ldots, i_{t-2}, i_{t-1})$, where $< i_1, i_2, \ldots, i_{t-2}, i_{t-1} >$ is any $(t-1)$-tuple from the values 1 to 2^d-2. Also the value of $\max(i_1, i_2, \ldots, i_{t-2}, i_{t-1})$ is i_{t-1} as we consider the tuples as ordered ones. Note that there is only 1 tuple with maximum value $(t-1)$. There are $\binom{t-1}{t-2}$ tuples with maximum value t, $\binom{t}{t-2}$ tuples with maximum value $t+1$ and so on. Thus, the mean of this distribution is $\overline{Y} = \left(\sum_{i=t-1}^{2^d-2} i\binom{i-1}{t-2}\right) / \binom{2^d-2}{t-1}$. Now, $\sum_{i=t-1}^{2^d-2} i\binom{i-1}{t-2} = (t-1)\sum_{i=t-1}^{2^d-2}\binom{i}{t-1} = (t-1)\binom{2^d-1}{t}$. Thus, $\overline{Y} = \frac{t-1}{t}(2^d-1)$. Note that this is equal to the value of $\overline{X}$. Thus we have the following theorem.

Theorem 4. *Given any primitive polynomial $f(x)$ of degree d, the average degree of its t-nomial multiples with degree $\le 2^d-2$ is equal to the average of maximum of all the distinct $(t-1)$ tuples from 1 to 2^d-2.*

With the result of the above theorem, With the result of the above theorem, we assume that the distributions X, Y are indistinguishable. Consider $N_{d,t}$ tuples which represent the actual t-nomial multiples of $f(x)$. Since the distribution of these tuples seems random, if we select any tuple, the probability that the tuple will represent a genuine t-nomial multiple is $N_{d,t}/\binom{2^d-2}{t-1}$. Thus we can estimate the expected number of t-nomials with degree less than or equal to c as $\binom{c}{t-1} N_{d,t}/\binom{2^d-2}{t-1}$. At this point let us summarize our assumption for this estimate.

Assumption RandomEstimate: Let $f(x)$ be a primitive polynomial of degree d. Consider the set of all t-nomial multiples of $f(x)$ which are of the form $1 + x^{i_1} + x^{i_2} + \ldots + x^{i_{t-2}} + x^{i_{t-1}}$ for $1 \le i_1 < i_2 < \ldots < i_{t-2} < i_{t-1} \le 2^d-2$. Interpret each t-nomial multiple as an ordered $(t-1)$ tuple $< i_1, i_2, \ldots, i_{t-2}, i_{t-1} >$. Note that the degree of this t-nomial is i_{t-1}. $N_{d,t}(c)$ denotes the number of t-nomial multiples which have the degree at most c. Now we expect that $N_{d,t}(c)/N_{d,t} \approx \binom{c}{t-1}/\binom{2^d-2}{t-1}$. Given some t we like to get an estimate of c, such that $\binom{c}{t-1} N_{d,t}/\binom{2^d-2}{t-1} \approx 1$. This value of c will give an expected value of $c_{d,t}$, the degree of the least degree t-nomial multiple of $f(x)$.

Next we present some experimental results in support of our assumption. We consider the trinomial multiples for this. In the following three tables we consider the case for degree 8, 9 and 10. In the first row A we provide some intervals. These intervals represent the degree of the trinomial multiples. In the second row B we provide the expected number of trinomial multiples less than or equal to the degree given in row A. As example, from the Table 1 we get that there are estimated 2.05 trinomial multiples at degree less than or equal to 32, 4.1 trinomial multiples in the range of degree $32 < d \le 57$, 6.15 trinomial multiples in the range of degree $57 < d \le 82$, etc. Note that these values are calculated from our assumption RandomEstimate and that is why these values are fractional. In the third row C, we present the result corresponding to a randomly chosen primitive polynomial. As example, from the Table 1 we get that there are 2 trinomial multiples at degree less than or equal to 32, 5 trinomial multiples in the range of degree $32 < d \le 57$, 5 trinomial multiples in the range of degree $57 < d \le 82$, etc. In the fourth row D, we present the result corresponding to all the primitive polynomials. That is for degree 8, we consider all the 16 primitive polynomials and check the result in aggregate. As example, from the Table 1 we get that there are 32 trinomial multiples at degree less than or equal to 32, 66 trinomial multiples in the range of degree $32 < d \le 57$, 116 trinomial multiples in the range of degree $57 < d \le 82$, etc corresponding to all the primitive polynomials of degree 8. We normalize the result of the fourth row D in the fifth row E. That is in Table 1, we divide the entries of the fourth row by 16 (total number of primitive polynomials of degree 8) to get the values in the fifth row E.

From the data in these three tables for the degree 8, 9 and 10, it is clear that our assumption is supported by the empirical results. With this observation we land into the following result.

Theorem 5. *Given a primitive polynomial $f(x)$ of degree d, under the assumption RandomEstimate, there exists a t-nomial multiple $g(x)$ of $f(x)$ such that degree of $g(x)$ is less than or equal to $2^{\frac{d}{t-1}+\log_2(t-1)+1}$.*

Table 1. Results for degree 8 primitive polynomials.

A	32	57	82	107	132	157	182	207	232	254	Total
B	2.05	4.1	6.15	9.22	11.25	14.35	17.85	18.48	21.6	21.95	127
C	2	5	5	11	11	12	20	20	20	21	127
D	32	66	116	146	182	228	284	288	348	342	2032
E	2	4.12	7.25	9.12	11.38	14.25	17.75	18	21.75	21.38	127

Table 2. Results for degree 9 primitive polynomials.

A	60	110	160	210	260	310	360	410	460	510	Total
B	3.05	9.08	13.1	18.15	23.19	27.22	32.26	38.3	43.07	47.58	255
C	3	8	12	23	24	25	32	38	43	47	255
D	166	398	629	880	1116	1337	1566	1818	2032	2298	12240
E	3.46	8.29	13.1	18.33	23.27	27.85	32.62	37.87	42.34	47.87	255

Table 3. Results for degree 10 primitive polynomials.

A	111	212	313	414	515	616	717	818	919	1022	Total
B	6.02	15.05	26.1	36.14	46.18	55.22	66.26	76.3	85.34	98.39	511
C	5	16	26	35	49	54	65	77	86	98	511
D	360	938	1566	2142	2732	3386	3962	4544	5168	5862	30660
E	6	15.63	26.12	35.7	45.53	56.43	66.03	75.73	86.13	97.7	511

Proof. From the assumption RandomEstimate, we need $\binom{c}{t-1} N_{d,t}/\binom{2^d-2}{t-1}$ approximately equal to 1. Let us consider the approximation as follows.

$\binom{c}{t-1} N_{d,t}/\binom{2^d-2}{t-1} \approx \binom{c}{t-1}\binom{2^d-2}{t-2}/(2\binom{2^d-2}{t-1}(t-1))$. In this step we have approximated $N_{d,t}$ as $\binom{2^d-2}{t-2}/(2(t-1))$. Note that $\binom{c}{t-1}\binom{2^d-2}{t-2}/(2\binom{2^d-2}{t-1}(t-1)) = \frac{1}{2}\frac{\frac{c!}{(t-1)!(c-t+1)!}}{\frac{(2^d-2)!}{(t-1)!(2^d-t-1)!}}\frac{\frac{(2^d-2)!}{(t-2)!(2^d-t)!}}{t-1} = \frac{1}{2}\frac{(c!)}{(c-t+1)!(t-1)!(2^d-t)} = \frac{1}{2}\frac{c(c-1)\ldots(c-t+1)}{(t-1)(t-2)\ldots 1}\frac{1}{2^d-t} \approx \frac{1}{2}(\frac{c}{t-1})^{t-1}\frac{1}{2^d}$. Here we underestimate the expression. Now we need the expression $\frac{1}{2}(\frac{c}{t-1})^{t-1}\frac{1}{2^d}$ to be approximately equal to 1. This will give the estimate of $c_{d,t}$. Thus, $c_{d,t} \approx 2(t-1)2^{\frac{d}{t-1}} = 2^{\frac{d}{t-1}+\log_2(t-1)+1}$.

We now discuss the significance of the above theorem. In [6,4] it has been shown that given any primitive polynomial $f(x)$, there exists a trinomial multiple of $f(x)$ with degree $\leq \frac{2^d+2}{3}$. Our result states that it is almost guaranteed to get a trinomial multiple with degree $\leq 2^{\frac{d}{2}+2}$. Our result is much sharper than the result of [6,4].

Let us also refer to a result on 4-nomial multiples of a primitive polynomial [11, Page 174]. It states that given a primitive polynomial $f(x)$ of degree d, it is possible to get a 4-nomial multiple of $f(x)$ having degree less than $2^{\frac{d}{4}}$ with high probability. This result is not exactly true. By computer experiment we observe that for a randomly chosen primitive polynomial $f(x)$, in most of the times $f(x)$ does not have a 4-nomial multiple with degree less than $2^{\frac{d}{4}}$. As an example we once again repeat our result presented in the introduction. Given $f(x) = x^{31}+x^{30}+x^{29}+x^{28}+x^{27}+x^{25}+x^{24}+x^{23}+x^{22}+x^{21}+x^{16}+x^{14}+x^{13}+x^{11}+1$, it has the minimum degree 4-nomial multiple $x^{3286}+x^{2417}+x^{1001}+1$. Note that 3286 is much larger than $2^{\frac{d}{4}} = 2^{\frac{31}{4}} = 215$ for $d = 31$. On the other hand, our estimate $2^{\frac{d}{t-1}+\log_2(t-1)+1} = 2^{\frac{d}{3}+\log_2 3+1} = 2^{\frac{d}{3}+\log_2 3+1} = 2^{\frac{d}{3}+2.585}$ is much more reasonable. Our estimate gives the value 7740 for $d = 31$.

We are generally interested about the sparse multiples. So even if we consider the value of t upto 9, it is clear that the estimate for the minimum degree of t-nomial multiple is $2^{\frac{d}{t-1}+3+1}$ as $\log_2(t-1) = \log_2 8 = 3$. Thus we propose that in practical systems the primitive polynomials of degree at least 128 should be used. Even in such a case, it is expected to get a 9-nomial multiple of degree 2^{20} and a 5-nomial multiple of degree 2^{36}.

The existing systems where LFSRs of lower size, i.e., say 64 are being used, the systems are susceptible to cryptanalytic attacks. For $d = 64$, we can expect to get a 9-nomial multiple at degree as low as $2^{12} = 4096$. It is known that if there is a primitive polynomial $f(x)$ of degree d which has a moderate degree ($> d$) t-nomial multiple $g(x)$, then the recurrence relation satisfied by $f(x)$ will also be satisfied by $g(x)$. Thus we can very well exploit the attack proposed in [11] by choosing the recurrence relation induced by $g(x)$. Given our observation, whatever be the weight of the primitive polynomial $f(x)$ (it does not matter whether it is of high or low weight as we have a low weight multiple), we can attack the system using $g(x)$. For a 64-degree primitive polynomial $f(x)$ we can expect to get a 9-nomial multiple of degree 4096. Now if it is possible to get around 2 × 4096 = 8192 ciphertext bits, it is feasible to estimate the key from the recurrence relation induced by $g(x)$ [11]. Getting 8192 ciphertext bits is a feasible proposition. Thus the current systems using small length LFSRs should be under scrutiny with our results.

Our result in Theorem 5 can be used to calculate the expected running time of the Algorithm Find-t-Nomial-Multiple at the beginning of this section. Considering our estimate of Theorem 5, we find that the value of $c_{d,t}$, in the discussion for complexity, should be estimated as $2^{\frac{d}{t-1}+\log_2(t-1)+1}$. Thus we need to check whether $f(x)$ divides $g(x)$ for $\sum_{i=d}^{c_{d,t}} \binom{i-1}{t-2} \approx \sum_{i=d}^{2^{\frac{d}{t-1}+\log_2(t-1)+1}} \binom{i-1}{t-2}$ different t-nomials in total. Note that the algorithm can be parallelized easily using more than one machines for faster solution.

Next we present some more experimental results to support Theorem 5. We consider the primitive polynomials of degree 8 to 16 and present the results as follows. For each degree d we provide how many primitive polynomials of that degree does not have a t-nomial multiple having degree $\leq 2^{\frac{d}{t-1}+\log_2(t-1)+1}$ given in Theorem 5. We consider trinomials and 4-nomials. In the first column we present the degree of the primitive polynomial. In the second column we present the total number of primitive polynomials of degree d, which is $\frac{\varphi(2^d-1)}{d}$ [9]. In the third column we provide the estimated value of $c_{d,3}$ from Theorem 5. The fourth column A provides the number of primitive polynomials for which the least degree trinomial multiples have degree $> c_{d,3}$. Similarly in the fifth column we provide the estimated value of $c_{d,4}$ and the sixth column B provides the number of primitive polynomials for which the least degree 4-nomial multiples have degree $> c_{d,4}$.

Table 4 strongly supports the estimation of Theorem 5. However, it is interesting to see that there are indeed a few primitive polynomials which do not have minimum degree t-nomials in the range of estimated degree in Theorem 5. These primitive polynomials are more suitable for cryptographic purposes. In fact this motivates us to present the following criteria in selection of primitive polynomials to be used as LFSR connection polynomials. *Given a set of primitive polynomials of degree d and weight w, we need to choose the one out of those whose least degree t-nomial multiple has maximum degree for low values of t.* Currently the only available option to find out such a primitive polynomial is exhaustive search technique.

As a passing remark, we also like to mention the problem of finding Zech's logarithm. Given a primitive element $\alpha \in GF(2^d)$, we can write $1 + \alpha^n = \alpha^{Z(n)}$. Given n, calculation of $Z(n)$ is called the problem of finding Zech's logarithm [10, Page 91, Volume 1]. This problem (see [5,8] and the references in these papers) is related with the problem of getting the trinomial multiples of a primitive polynomial. Note that we have the result that it is expected to get a trinomial multiple of any primitive polynomial having degree $\leq 2^{\frac{d}{2}+2}$. This gives that given a primitive element α, it is expected to get an $< n, Z(n) >$ pair with $\max(n, Z(n)) \leq 2^{\frac{d}{2}+2}$.

Table 4. Experimental results with respect to Theorem 5.

degree d	$\frac{\varphi(2^d-1)}{d}$	Estimated $c_{d,3}$	A	Estimated $c_{d,4}$	B
8	16	64	0	38	0
9	48	90	0	48	0
10	60	128	0	60	0
11	176	181	0	76	0
12	144	256	0	96	0
13	630	362	0	120	0
14	756	512	0	153	0
15	1800	724	6	192	0
16	2048	1024	13	241	0

References

1. A. Canteaut and M. Trabbia. Improved fast correlation attacks using parity-check equations of weight 4 and 5. In *Advances in Cryptology - EUROCRYPT 2000*, number 1807 in Lecture Notes in Computer Science, pages 573–588. Springer Verlag, 2000.
2. C. Ding, G. Xiao, and W. Shan. *The Stability Theory of Stream Ciphers*. Number 561 in Lecture Notes in Computer Science. Springer-Verlag, 1991.
3. S. W. Golomb. *Shift Register Sequences*. Aegean Park Press, 1982.
4. K. C. Gupta and S. Maitra. Primitive polynomials over GF(2) – A cryptologic approach. In *ICICS 2001*, Lecture Notes in Computer Science, Springer Verlag (to appear), 2001.
5. K. Huber. Some comments on Zech's logarithms. *IEEE Transactions on Information Theory*, IT-36(4):946–950, July 1990.
6. K. Jambunathan. On choice of connection polynomials for LFSR based stream ciphers. In *Progress in Cryptology - INDOCRYPT 2000*, number 1977 in Lecture Notes in Computer Science, pages 9–18. Springer Verlag, 2000.
7. T. Johansson and F. Jonsson. Fast correlation attacks through reconstruction of linear polynomials. In *Advances in Cryptology - CRYPTO 2000*, number 1880 in Lecture Notes in Computer Science, pages 300–315. Springer Verlag, 2000.
8. F. M. Assis and C. E. Pedreira. An architecture for computing Zech's logarithms in GF(2^n). *IEEE Transactions on Computers*, volume 49(5):519–524, May 2000.
9. R. Lidl and H. Niederreiter. *Finite Fields*. Addison Wesley, 1983.
10. F. J. MacWillams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. North Holland, 1977.
11. W. Meier and O. Staffelbach. Fast correlation attacks on certain stream ciphers. *Journal of Cryptology*, 1:159–176, 1989.
12. T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, IT-30(5):776–780, September 1984.

Fast Generation of Cubic Irreducible Polynomials for XTR

Jae Moon Kim, Ikkwon Yie, Seung Ik Oh, Hyung Don Kim, and Jado Ryu

Department of Mathematics, Inha University, Inchon, Korea
{jmkim,ikyie,sioh,hdkim,jdryu}@math.inha.ac.kr

Abstract. XTR cryptosystem makes use of an irreducible polynomial $F(c,x) = x^3 - cx^2 + c^p x - 1$ over a finite field $\mathbb{F}_{p^2}$. In this paper, we develop a new method to generate such an irreducible polynomial. Our method requires only computations of Jacobi symbols and thus improves those given [1], [2] and [3].

1 Introduction

In a series of papers [1], [2], [3] and [4], A.K. Lenstra and E.R. Verheul introduced and developed a new cryptosystem so called XTR public key system. Let $F(c,x) = x^3 - cx^2 + c^p x - 1$ be an irreducible polynomial in $\mathbb{F}_{p^2}[x]$ for some element $c \in \mathbb{F}_{p^2}$. Then a root h of $F(c,x)$ is in $\mathbb{F}_{p^6}$ and satisfies $h^{p^2-p+1} = 1$ ([1]). Thus h generates a subgroup of $\mathbb{F}_{p^6}$ of order dividing $p^2 - p + 1$. For each integer k, put $c_k = Tr(h^k)$, where Tr is the trace from $\mathbb{F}_{p^6}$ to $\mathbb{F}_{p^2}$. The idea of XTR (Efficient and Compact Subgroup Trace Representation) is that one can make use of $\{c_k\}$ instead of the subgroup $< h >= \{h^k\}$ in implementing various cryptosystems such as Diffie-Hellman key agreement protocol and Elgamal system. Note that $\{c_k\}$ is in $\mathbb{F}_{p^2}$, while $< h >= \{h^k\}$ is in $\mathbb{F}_{p^6}$. Thus XTR system has the obvious advantages in both computation and communication (XTR reduces the cost to $\frac{1}{3}$) with maintaining the same security level as one works with $\{h^k\}$ ([1]). Considering the size of p which is supposed to be as large as $\frac{1}{6} \times 1024 \approx 170$ bits, saving $\frac{2}{3}$ is huge.

One of the problems in running XTR system is the generation of c which guarantees the irreducibility of $F(c,x) = x^3 - cx^2 + c^p x - 1$. For a randomly chosen c in $\mathbb{F}_{p^2}$, the probability for $F(c,x)$ to be irreducible is $\frac{1}{3}$. In [1], [2] and [3], several algorithms of irreducibility test of $F(c,x)$ are given. The best algorithm for irreducibility test known so far requires about $1.8 log_2 p$ multiplications in $\mathbb{F}_p$ ([3]). So one needs expectedly about $2.7 log_2 p$ multiplications to initiate XTR system. There is another improvement in finding irreducible polynomials $F(c,x)$. Namely, in [2], it is proved that $F(c,x)$ is always irreducible for a certain special value c when $p \equiv 2$ or 5 mod 9 and 3 mod 4.

The purpose of this paper is to generate irreducible polynomials $F(c,x)$ in $\mathbb{F}_{p^2}[x]$ more efficiently. In section 2, we will show that they can be derived from

This paper was supported by the Basic Research Institute Program, Korea Research Foundation, 2000-015-DP0006.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 73–78, 2001.

irreducible polynomials in $\mathbb{F}_p[x]$. Fix a quadratic nonresidue t in $\mathbb{F}_p$ and suppose an irreducible polynomial in $\mathbb{F}_p[x]$ of the form $x^3 - tax^2 + bx + a$ is given. Then we will show that $F(c, x) = x^3 - cx^2 + c^p x - 1$ is always irreducible, where $c = \frac{-tb+3}{tb+1} + \frac{4ta}{tb+1}\alpha$ with $\alpha^2 = t$. Therefore to generate an irreducible polynomial $x^3 - cx^2 + c^p x - 1$ in $\mathbb{F}_{p^2}[x]$, start with an irreducible polynomial $x^3 - tax^2 + bx + a$ in $\mathbb{F}_p[x]$ and get the corresponding one in $\mathbb{F}_{p^2}[x]$. Of course, not every irreducible polynomial in $\mathbb{F}_p[x]$ is of the form $x^3 - atx^2 + bx + a$. However, note that for a given irreducible polynomial $H(x) = x^3 + lx^2 + mx + n$ in $\mathbb{F}_p[x]$ with $lmn = 0$, $H(x)$ is of the form $x^3 - tax^2 + bx + a$ for some quadratic nonresidue t if and only if $-\frac{l}{n}$ is a quadratic nonresidue. So the probability for $H(x)$ to be of the form $x^3 - tax^2 + bx + a$ is $\frac{1}{2}$. Therefore once we have an irreducible polynomial $H(x)$ in $\mathbb{F}_p[x]$, by considering $H(x + k)$ for $k = 0, \pm 1, \pm 2, \cdots$, we can find an irreducible polynomial of the form $x^3 - tax^2 + bx + a$ in $\mathbb{F}_p[x]$ (and thus $F(c, x)$ also) before long since the probability that $H(x + k)$ cannot produce $F(c, x)$ is $\frac{1}{2}$ for each k. In this way, we can find plenty of irreducible polynomials $F(c, x)$.

Thus our problem is narrowed down how to create irreducible cubic polynomials in $\mathbb{F}_p[x]$. This is handled in section 3. In section 3, we explain how to find irreducible cubic polynomials in $\mathbb{F}_p[x]$ by examining several examples. In example 1, we find an irreducible polynomial $F(c, x)$ when $p \equiv 2$ or 5 *mod* 9. This case was studied in [2], where a different irreducible polynomial is constructed by a different method. In the next two examples, we produce irreducible polynomials $F(c, x)$ for different moduli, namely the case $p \equiv \pm 1$ *mod* 7 in example 2 and the case $p \equiv \pm 2$ or ± 3 or ± 4 or ± 6 *mod* 13 in example 3. Thus by looking at p *mod* 9, or *mod* 7, or *mod* 13, we get an irreducible polynomial $F(c, x)$ immediately. If we are in such an unfortunate case that $p \equiv 2$ or 5 *mod* 9 and $p \equiv \pm 1$ *mod* 7 and $p \equiv \pm 1$ or ± 5 *mod* 13, another appropriate modulus, for instance *mod* 19, would yield an irreducible polynomial $F(c, x)$.

2 Generation of $F(c, x)$

For a given prime p, which is supposed to be as large as $\frac{1}{6} \times 1024 \approx 170$ bits for applications to cryptosystem, fix a quadratic nonresidue t in $\mathbb{F}_p$. Let $G(x) = x^3 - tax^2 + bx + a$ be a polynomial in $\mathbb{F}_p[x]$ with $a, b \quad \mathbb{F}_p$. Let β_1, β_2 and β_3 be the roots of $G(x)$, so that

$$G(x) = x^3 - tax^2 + bx + a = (x - \beta_1)(x - \beta_2)(x - \beta_3) \ .$$

Let α be an element in $\mathbb{F}_{p^2}$ such that $\alpha^2 = t$. Then $\{1, \alpha\}$ is a basis of $\mathbb{F}_{p^2}$ over $\mathbb{F}_p$. Since $\beta_1 + \beta_2 + \beta_3 = ta$ and $\beta_1\beta_2\beta_3 = -a$, we have $(1 + \beta_1\alpha)(1 + \beta_2\alpha)(1 + \beta_3\alpha) \quad \mathbb{F}_p$.

Put

$$F(x) = (x - (1 + \beta_1\alpha)^{p^3-1})(x - (1 + \beta_2\alpha)^{p^3-1})(x - (1 + \beta_3\alpha)^{p^3-1}) \ .$$

Since $(1 + \beta_1\alpha)(1 + \beta_2\alpha)(1 + \beta_3\alpha)$ is in $\mathbb{F}_p$, the constant term of $F(x)$ is $-(1 + \beta_1\alpha)^{p^3-1}(1 + \beta_2\alpha)^{p^3-1}(1 + \beta_3\alpha)^{p^3-1} = -1$. Put

$$c = (1 + \beta_1\alpha)^{p^3-1} + (1 + \beta_2\alpha)^{p^3-1} + (1 + \beta_3\alpha)^{p^3-1} \ .$$

We claim that the coefficient of x of $F(x)$ equals c^p. Note that $\alpha^p = \alpha^{p-1}\alpha = (\alpha^2)^{\frac{p-1}{2}}\alpha = t^{\frac{p-1}{2}}\alpha = -\alpha$. Thus

$$\begin{aligned} c^p &= (1-\beta_1^p\alpha)^{p^3-1} + (1-\beta_2^p\alpha)^{p^3-1} + (1-\beta_3^p\alpha)^{p^3-1} \\ &= \frac{1+\beta_1^{p^4}\alpha}{1-\beta_1^p\alpha} + \frac{1+\beta_2^{p^4}\alpha}{1-\beta_2^p\alpha} + \frac{1+\beta_3^{p^4}\alpha}{1-\beta_3^p\alpha} . \end{aligned}$$

On the other hand, the coefficient of x of $F(x)$ is

$$\begin{aligned} &(1+\beta_1\alpha)^{p^3-1}(1+\beta_2\alpha)^{p^3-1} + (1+\beta_2\alpha)^{p^3-1}(1+\beta_3\alpha)^{p^3-1} \\ &\qquad +(1+\beta_1\alpha)^{p^3-1}(1+\beta_3\alpha)^{p^3-1} \\ &= \frac{1}{(1+\beta_1\alpha)^{p^3-1}} + \frac{1}{(1+\beta_2\alpha)^{p^3-1}} + \frac{1}{(1+\beta_3\alpha)^{p^3-1}} \\ &= \frac{(1+\beta_1\alpha)}{(1-\beta_1^{p^3}\alpha)} + \frac{(1+\beta_2\alpha)}{(1-\beta_2^{p^3}\alpha)} + \frac{(1+\beta_3\alpha)}{(1-\beta_3^{p^3}\alpha)} . \end{aligned}$$

There are three cases to consider:

(i) all β_i's are in $\mathbb{F}_p$, in which case $\beta_i^p = \beta_i$,
(ii) one of β_i, say β_1, is in $\mathbb{F}_p$, and the other two are in $\mathbb{F}_{p^2}$, in which case $\beta_1^p = \beta_1$ and for $i = 2, 3$, $\beta_i^{p^4} = \beta_i^{p^2} = \beta_i$ and $\beta_i^{p^3} = \beta_i^p$,
(iii) all β_i's are in $\mathbb{F}_{p^3}$, in which case $\beta_i^{p^3} = \beta_i$, $\beta_i^{p^4} = \beta_i^p$ and they are conjugate over $\mathbb{F}_p$, so that we may assume $\beta_1^p = \beta_2$, $\beta_2^p = \beta_3$ and $\beta_3^p = \beta_1$.

In any case, one can check that the coefficient of x of $F(x)$ equals c^p as desired. Therefore $F(x)$ is a polynomial of the form $F(x) = x^3 - cx^2 + c^px - 1$.

Next we claim that

(a) $c \quad \mathbb{F}_{p^2}$,
(b) $G(x)$ is irreducible over $\mathbb{F}_p$ if and only if $F(x)$ is irreducible over $\mathbb{F}_{p^2}$,
(c) If $G(x)$ is irreducible over $\mathbb{F}_p$, then $tb = -1$ and $c = \frac{-tb+3}{tb+1} + \frac{4ta}{tb+1}\alpha.$

(a) follows easily from the observation that $c^{p^2} = c$. To prove (b), note that if $G(x)$ is irreducible, then we are in situation (iii). In this case the roots of $F(x)$ are conjugates over $\mathbb{F}_{p^2}$, thus $F(x)$ is irreducible. On the other hand, if $G(x)$ is reducible, we are in the case (i) or (ii). Then the roots of $F(x)$ lie in F_{p^2}, so $F(x)$ is reducible. Hence we obtain (b). Finally we examine (c). If $G(x)$ is irreducible, then $tb = -1$, for otherwise, ta would be a root of $G(x)$. To get the formula for c, we expand the expression for c:

$$\begin{aligned} c &= (1+\beta_1\alpha)^{p^3-1} + (1+\beta_2\alpha)^{p^3-1} + (1+\beta_3\alpha)^{p^3-1} \\ &= \frac{1-\beta_1^{p^3}\alpha}{1+\beta_1\alpha} + \frac{1-\beta_2^{p^3}\alpha}{1+\beta_2\alpha} + \frac{1-\beta_3^{p^3}\alpha}{1+\beta_3\alpha} \\ &= \frac{1-\beta_1\alpha}{1+\beta_1\alpha} + \frac{1-\beta_2\alpha}{1+\beta_2\alpha} + \frac{1-\beta_3\alpha}{1+\beta_3\alpha} \end{aligned}$$

$$= \frac{3 - (\beta_1\beta_2 + \beta_2\beta_3 + \beta_1\beta_3)t + \{(\beta_1 + \beta_2 + \beta_3) - 3\beta_1\beta_2\beta_3 t\}\alpha}{1 + (\beta_1\beta_2 + \beta_2\beta_3 + \beta_1\beta_3)t}$$
$$= \frac{3 - bt + 4at\alpha}{1 + bt} \ .$$

To summarize, we have the following theorem:

Theorem 1. *Let t be a quadratic nonresidue mod p and $G(x)$ be an irreducible polynomial in $\mathbb{F}_p[x]$ of the form $G(x) = x^3 - atx^2 + bx + a$. Then $F(c, x) = x^3 - cx^2 + c^p x - 1$ with $c = \frac{-tb+3}{tb+1} + \frac{4ta}{tb+1}\alpha$ is an irreducible polynomial in $\mathbb{F}_{p^2}[x]$, where $\alpha^2 = t$.*

Remark 1. Let $H(x) = x^3 + lx^2 + mx + n$ be an irreducible polynomial in $\mathbb{F}_p[x]$ with $lmn = 0$. Then $H(x)$ is of the form $G(x) = x^3 - tax^2 + bx + a$ for some quadratic nonresidue t if and only if $-\frac{l}{n}$ is a quadratic nonresidue, which can be tested by computing the Jacobi symbol $\left(\frac{-\frac{l}{n}}{p}\right)$. So the probability for $H(x)$ to be of the form $x^3 - tax^2 + bx + a$ is $\frac{1}{2}$. Hence, once we have an irreducible polynomial $H(x)$, by considering $H(x + k)$ for $k = 0, \pm 1, \pm 2, \cdots$, we obtain an irreducible polynomial of the form $x^3 - tax^2 + bx + a$ before long, since the probability that $H(x + k)$ is not of the desired form is $\frac{1}{2}$ for each k.

3 Examples

Example 1. $p \equiv 2$ or 5 *mod* 9

Let $\zeta = \zeta_9$ be a primitive 9th root of 1. Then $Irr(\zeta, \mathbb{F}_p) = x^6 + x^3 + 1$, $\mathbb{F}_{p^6} = \mathbb{F}_p(\zeta)$ and $\mathbb{F}_{p^3} = \mathbb{F}_p(\zeta + \zeta^{-1})$. Now we compute the irreducible polynomial for $\zeta + \zeta^{-1}$ over $\mathbb{F}_p$. Note that the conjugates of $\zeta + \zeta^{-1}$ over $\mathbb{F}_p$ are $\zeta + \zeta^{-1}$, $\zeta^p + \zeta^{-p}$ and $\zeta^{p^2} + \zeta^{-p^2}$. Since $p \equiv 2$ or 5 *mod* 9, they are $\zeta + \zeta^{-1}$, $\zeta^2 + \zeta^{-2}$ and $\zeta^4 + \zeta^{-4}$. Since

$$\begin{aligned} & (\zeta + \zeta^{-1}) + (\zeta^2 + \zeta^{-2}) + (\zeta^4 + \zeta^{-4}) = 0, \\ & (\zeta + \zeta^{-1})(\zeta^2 + \zeta^{-2}) + (\zeta^2 + \zeta^{-2})(\zeta^4 + \zeta^{-4}) + (\zeta + \zeta^{-1})(\zeta^4 + \zeta^{-4}) = -3, \\ \textit{and} \quad & (\zeta + \zeta^{-1})(\zeta^2 + \zeta^{-2})(\zeta^4 + \zeta^{-4}) = -1, \end{aligned}$$

we get $Irr(\zeta + \zeta^{-1}, \mathbb{F}_p) = x^3 - 3x + 1$. Let

$$H(x) = Irr(3(\zeta + \zeta^{-1}) - 1, \mathbb{F}_p) = x^3 + 3x^2 - 24x + 1 \ .$$

Since $p \equiv 2$ mod 3, $\left(\frac{-3}{p}\right) = -1$. Thus $H(x)$ is of form $x^3 - atx^2 + bx + a$ with $a = 1$, $b = -24$ and $t = -3$ which is a quadratic nonresidue. Therefore the corresponding polynomial $F(c, x) = x^3 - cx^2 + c^p x - 1$ with $c = -\frac{69}{73} - \frac{12}{73}\alpha$ is irreducible over $\mathbb{F}_{p^2}$, where $\alpha^2 = -3$.

Example 2. $p \equiv \pm 1 \ mod \ 7$

Let $\zeta = \zeta_7$ be a primitive 7th root of 1. If $p \equiv 3$ or $5 \ mod \ 7$, then $\mathbb{F}_p(\zeta) = \mathbb{F}_{p^6}$. So $\mathbb{F}_p(\zeta + \zeta^{-1}) = \mathbb{F}_{p^3}$. If $p \equiv 2$ or $4 \ mod \ 7$, then $\mathbb{F}_p(\zeta) = \mathbb{F}_{p^3}$. Note that $\mathbb{F}_p(\zeta + \zeta^{-1}) = \mathbb{F}_{p^3}$ in this case, too. Thus conjugates of $\zeta + \zeta^{-1}$ over $\mathbb{F}_p$ are $\zeta + \zeta^{-1}$, $\zeta^2 + \zeta^{-2}$ and $\zeta^4 + \zeta^{-4}$. Note that

$$\begin{aligned} & (\zeta + \zeta^{-1}) + (\zeta^2 + \zeta^{-2}) + (\zeta^4 + \zeta^{-4}) = -1, \\ & (\zeta + \zeta^{-1})(\zeta^2 + \zeta^{-2}) + (\zeta^2 + \zeta^{-2})(\zeta^4 + \zeta^{-4}) + (\zeta + \zeta^{-1})(\zeta^4 + \zeta^{-4}) = -2, \\ \text{and} \quad & (\zeta + \zeta^{-1})(\zeta^2 + \zeta^{-2})(\zeta^4 + \zeta^{-4}) = 1 \ . \end{aligned}$$

Therefore $Irr(\zeta + \zeta^{-1}, \mathbb{F}_p) = x^3 + x^2 - 2x - 1$. Let

$$H(x) = Irr(\zeta + \zeta^{-1}, \mathbb{F}_p) = x^3 + x^2 - 2x - 1 \ .$$

Suppose that $p \equiv 3 \ mod \ 4$, and consider

$$H(x + 2) = Irr(\zeta + \zeta^{-1} - 2, \mathbb{F}_p) = x^3 + 7x^2 + 14x + 7 \ .$$

Since $p \equiv 3 \ mod \ 4$, -1 is a quadratic nonresidue $mod \ p$. Thus $H(x + 2)$ is an irreducible polynomial of the form $G(x) = x^3 - atx^2 + bx + a$ with $a = 7$, $b = 14$ and $t = -1$. Hence the polynomial $F(c, x) = x^3 - cx^2 + c^p x - 1$ with $c = -\frac{17}{13} + \frac{28}{13}\alpha$ is irreducible over $\mathbb{F}_{p^2}$, where $\alpha^2 = -1$.

Example 3. $p \equiv \pm 2$ or ± 3 or ± 4 or $\pm 6 \ mod \ 13$

Let $\zeta = \zeta_{13}$ be a primitive 13th root of 1. As in example 2, one can check that $\mathbb{F}_{p^3} = \mathbb{F}(\zeta + \zeta^{-1} + \zeta^5 + \zeta^{-5})$. The conjugates of $y_1 = \zeta + \zeta^{-1} + \zeta^5 + \zeta^{-5}$ over $\mathbb{F}_p$ are $y_1 = \zeta + \zeta^{-1} + \zeta^5 + \zeta^{-5}$, $y_2 = \zeta^2 + \zeta^{-2} + \zeta^3 + \zeta^{-3}$ and $y_3 = \zeta^4 + \zeta^{-4} + \zeta^6 + \zeta^{-6}$. Note that

$$\begin{aligned} & y_1 + y_2 + y_3 = -1, \\ & y_1 y_2 + y_2 y_3 + y_1 y_3 = -4, \\ \text{and} \quad & y_1 y_2 y_3 = -1 \ . \end{aligned}$$

Therefore, $Irr(y_1, \mathbb{F}_p) = x^3 + x^2 - 4x + 1$. Let

$$H(x) = Irr(y_1, \mathbb{F}_p) = x^3 + x^2 - 4x + 1 \ .$$

For a $\mathbb{F}_p$, we have

$$H(x + a) = Irr(y_1 - a) = x^3 + (3a + 1)x^2 + (3a^2 + 2a - 4)x + a^3 + a^2 - 4a + 1 \ .$$

In particular,

$$H(x - 1) = x^3 - 2x^2 - 3x + 5 \quad \text{and} \quad H(x + 3) = x^3 + 10x^2 + 29x + 25 \ .$$

Thus, if $\left(\frac{10}{p}\right) = -1$, we make use of the first polynomial with $a = 5$, $b = -3$ and $t = \frac{2}{5}$ to get an irreducible polynomial $F(c, x)$. In this case, $c = -21 - 40\alpha$, where $\alpha^2 = \frac{2}{5}$. While if $\left(\frac{10}{p}\right) = 1$, we use $H(x + 3) = x^3 + 10x^2 + 29x + 25$. In this case, $a = 25$, $b = 29$ and $t = -\frac{2}{5}$. Then we get an irreducible polynomial $F(c, x)$ with $c = -\frac{73}{53} + \frac{200}{53}\alpha$, where $\alpha^2 = -\frac{2}{5}$.

References

1. A. K. Lenstra, E. R. Verheul : The XTR public key system. Proceedings of Crypto 2000, LNCS 1880, Springer-Verlag (2000) 1-19
2. A. K. Lenstra, E. R. Verheul : Key improvements to XTR, Proceedings of Asiacrypt 2000, LNCS 1976, Springer-Verlag (2000) 220-233
3. A. K. Lenstra, E. R. Verheul : Fast irreducibility and subgroup membership testing in XTR, Proceedings of PKC 2001, LNCS 1992,l Springer-Verlag (2001) 73-86
4. A. K. Lenstra, E. R. Verheul : An overview of the XTR public key system, Proceedings of the Eurocrypto 2001, LNCS, Springer-Verlag (to appear)

Cheating Prevention in Secret Sharing over $GF(p^t)$

Josef Pieprzyk[1] and Xian-Mo Zhang[2]

[1] Department of Computing, Macquarie University,
Sydney, NSW 2109, Australia
josef@ics.mq.edu.au
[2] School of IT and CS, University of Wollongong,
Wollongong NSW 2522, Australia xianmo@cs.uow.edu.au

Abstract. The work investigates cheating prevention in secret sharing. It is argued that cheating is immune against cheating if the cheaters gain no advantage over honest participants by submitting invalid shares to the combiner. This work addresses the case when shares and the secret are taken from $GF(p^t)$. Two models are considered. The first one examines the case when cheaters consistently submit always invalid shares. The second model deals with cheaters who submit a mixture of valid and invalid shares. For these two models, cheating immunity is defined, properties of cheating immune secret sharing are investigated and their constructions are given.

Keywords: Secret Sharing, Nonlinear Secret Sharing, Cheating Immunity

1 Introduction

Secret sharing is widely used to produce group-oriented cryptographic algorithms, systems and protocols. Tompa and Woll [11] showed that Shamir secret sharing can be subject to cheating by dishonest participants. It is easy to see that, in fact, dishonest participants can cheat in any linear secret sharing. Cheating prevention has been addressed in literature for conditionally and unconditionally secure secret sharing. For conditionally secure secret sharing, the combiner checks validity of submitted shares before attempting to compute the secret. Any invalid share (and the cheater) is likely to be detected before the secret reconstruction (see [2,1,6]). Publicly verifiable secret sharing (see [3,5,9,7]) provide a solution to this problem in the conditionally secure setting. We argue that instead of setting an expensive verification infrastructure to detect cheaters, it is possible to discourage them from cheating. It is likely that cheaters will be discouraged if they are not able to reconstruct the valid secret from the invalid one returned by the combiner. Ideally, submission of invalid shares should not give any advantage to the cheaters over the honest participants in recovery of the valid secret. In this work shares and the secret are from $GF(p^t)$. The structure of the paper is as follows. First we introduce a basic model of cheating in

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 79–90, 2001.

which, cheaters always submit invalid shares. Cheating immunity is defined and constructions of cheating immune secret sharing are given. Further we generalise our model for the case where the collaborating cheaters may submit an arbitrary mixture of their valid and invalid shares. Again, the notion of strict immunity is introduced, its properties are investigated and constructions are shown.

2 Basic Model of Cheating

Let $GF(p^t)$ denote a finite field with p^t elements where p is a prime number and t in a positive integer. We write $GF(p^t)^n$ to denote the vector space of n tuples of elements from $GF(p^t)$. Then each vector α $GF(p^t)^n$ can be expressed as $\alpha = (a_1, \dots, a_n)$ where $a_1, \dots, a_n$ $GF(p^t)$. We consider a mapping f from $GF(p^t)^n$ to $GF(p^t)$. Usually we write f as $f(x)$ or $f(x_1, \dots, x_n)$ where $x = (x_1, \dots, x_n)$ and each x_j $GF(p^t)$. f is also called a *function* on $GF(p^t)^n$. f is said to be *balanced* if $f(x)$ takes each element of $GF(p^t)$ precisely $p^{t(n-1)}$ times while x goes through each vector in $GF(p^t)^n$ once. The *Hamming weight* of a vector α $GF(p^t)^n$, denoted by $HW(\alpha)$, is the number of nonzero coordinates of α. An *affine* function f on $GF(p^t)^n$ is a function that takes the form of $f(x_1, \dots, x_n) = a_1x_1 + \cdots + a_nx_n + c$, where $+$ denotes the addition in $GF(p^t)$, a_j, c $GF(p^t)$, $j = 1, 2, \dots, n$. Furthermore f is called a *linear* function if $c = 0$. It is easy to verify that any non-constant affine function is balanced.

We see secret sharing as a set of distribution rules combined into a single table T (see [10]) with entries from $GF(p^t)$. We also assume that we are dealing with (n, n) threshold scheme where any n participants are able to determine a single entry from T which indicates the secret. Our considerations are restricted to the case of (n, n) secret sharing. The general case of (n, N) secret sharing can be seen as a concatenation of (n, n) secret sharing with a system of N “consistent” linear equations. Shares are generated for N participants using the linear equations. Any n participants can get a system of linear equations with a unique solution which points out the unique row of the table T. Let $x = (x_1, \dots, x_n)$ and $\delta = (\delta_1, \dots, \delta_n)$ be two vectors in $GF(p^t)^n$. Define a vector x_δ^+ $GF(p^t)^n$, whose j-th coordinate is x_j if $\delta_j = 0$, or 0 if $\delta_j = 0$. In addition, we define a vector x_δ^- $GF(p^t)^n$, whose j-th coordinate is 0 if $\delta_j = 0$, or x_j if $\delta_j = 0$. Let $\tau = (\tau_1, \dots, \tau_n)$ and $\delta = (\delta_1, \dots, \delta_n)$ be two vectors in $GF(p^t)^n$. We write τ δ to denote the property that if $\tau_j = 0$ then $\delta_j = 0$. In addition, we write τ δ to denote the property that τ δ and $HW(\tau) < HW(\delta)$. In particular, if δ δ and $HW(\delta) = HW(\delta)$ we write δ □ δ. It is easy to verify that δ □ δ δ δ and δ δ both $x_\delta^+ = x_\delta^+$ and $x_\delta^- = x_\delta^-$ hold for any x $GF(p^t)^n$, where denotes “if and only if”. We define the following notation that will be frequently used in this paper. Let δ be a nonzero vector in $GF(p^t)^n$, τ δ and u $GF(p^t)$. Set

$$R_f(\delta, \tau, u) = \{x_\delta^- | f(x_\delta^- + \tau) = u\} \tag{1}$$

We also simply write $R_f(\delta, \tau, u)$ as $R(\delta, \tau, u)$ if no confusions occur.

Lemma 1. *Let δ be a nonzero vector in $GF(p^t)^n$, $\tau \quad \delta$, and $u \quad GF(p^t)$. Then for any given function f on $GF(p^t)^n$, (i) $R(\delta, \tau, u) = R(\delta\ , \tau, u)$ if δ □ δ, (ii) $R(\delta, \alpha^+_\delta, u) = R(\delta, \gamma^+_\delta, u)$ for any $\alpha, \gamma \quad GF(p^t)^n$ with $\alpha^+_\delta = \gamma^+_\delta$, (iii) there exists some $b \quad GF(p^t)$ such that $R(\delta, \tau, b) = \ $, where   denotes the empty set.*

Proof. As (i) and (ii) hold obviously, we only prove (iii). Let γ be any vector in $GF(p^t)^n$. Set $f(\gamma^-_\delta + \tau) = b$. By definition, $\gamma^-_\delta \quad R_f(\delta, \tau, b)$ and thus $R(\delta, \tau, b) = \ $.

Given a function f on $GF(p^t)^n$, we introduce the following notations:

- Let $\alpha \quad GF(p^t)^n$ be the sequence of shares held by the group $P = \{P_1, \dots, P_n\}$ of n participants and the secret $K = f(\alpha)$.
- The collection of cheaters is determined by the sequence $\delta = (\delta_1, \delta_2, \dots, \delta_n)$ where P_i is a cheater δ_i is nonzero.
- At the pooling time, the cheaters submit their shares. It is assumed that cheaters always submit invalid shares. The honest participants always submit their valid shares. We consider the vector $\alpha + \delta$. From the properties of x^+_δ and x^-_δ, $\alpha + \delta = \alpha^-_\delta + \alpha^+_\delta + \delta$. Thus the combiner obtains $\alpha + \delta$ that splits into two parts: α^-_δ – the part submitted by honest participants, and $\alpha^+_\delta + \delta$ – the part submitted by cheaters. The combiner returns an invalid secret $K\ \ = f(\alpha + \delta)$. Note that the cheaters always change their shares. We assume that there exists at least one cheater, in other words, δ is nonzero or $HW(\delta) > 0$.
- α^+_δ determines valid shares held by the cheaters. The set $R(\delta, \alpha^+_\delta, K)$, or $\{x^-_\delta | f(x^-_\delta + \alpha^+_\delta) = K\}$, determines a collection of rows of T with the correct secret K and valid shares held by the cheaters.
- The set $R(\delta, \alpha^+_\delta + \delta, K\ \ )$, or $\{x^-_\delta | f(x^-_\delta + \alpha^+_\delta + \delta) = K\ \ \}$, represents the view of the cheaters after getting back K from the combiner.

The function f is called the *defining function* as it determines the secret sharing. The nonzero vector $\delta = (\delta_1, \dots, \delta_n)$ is called a *cheating vector*, α is called a *original vector*. The value of $\rho_{\delta,\alpha} = \#(R(\delta, \alpha^+_\delta + \delta, K\ \ ) \cap R(\delta, \alpha^+_\delta, K)) / \#R(\delta, \alpha^+_\delta + \delta, K\ \ )$, expresses the probability of cheater success with respect to δ and α, where $\#X$ denotes the number of elements in the set X. As an original vector α is always in $R(\delta, \alpha^+_\delta + \delta, K\ \ ) \cap R(\delta, \alpha^+_\delta, K)$, the probability of successful cheating always satisfies $\rho_{\delta,\alpha} > 0$. Clearly the number of cheaters is equal to $HW(\delta)$.

Theorem 1. *Given a secret sharing scheme with its defining function f on $GF(p^t)^n$. Let $\delta \quad GF(p^t)^n$ with $0 < HW(\delta) < n$ be a cheating vector and α be an original vector in $GF(p^t)^n$. If $\rho_{\delta,\alpha} < p^{-t}$ then there exists a vector $\gamma \quad GF(p^t)^n$ such that $\rho_{\delta,\gamma} > p^{-t}$.*

Proof. Let $f(\alpha) = K$ and $f(\alpha + \delta) = K\ \ $. By definition, $R(\delta, \alpha^+_\delta, K) = \{x^-_\delta | f(x^-_\delta + \alpha^+_\delta) = K\}$ and $R(\delta, \alpha^+_\delta + \delta, K\ \ ) = \{x^-_\delta | f(x^-_\delta + \alpha^+_\delta + \delta) = K\ \ \}$. We partition $R(\delta, \alpha^+_\delta + \delta, K\ \ )$ into p^t parts: $R(\delta, \alpha^+_\delta + \delta, K\ \ ) = \ {}_{u \quad GF(p^t)} Q_u$ where $Q_u = R(\delta, \alpha^+_\delta + \delta, K\ \ ) \cap R(\delta, \alpha^+_\delta, u + K)$. Clearly $\#R(\delta, \alpha^+_\delta + \delta, K\ \ ) = \ {}_{u \quad GF(p^t)} \#Q_u$. Note that $R(\delta, \alpha^+_\delta + \delta, K\ \ ) \cap R(\delta, \alpha^+_\delta, K) = Q_0$. Therefore

$\rho_{\delta,\alpha} = \#(R(\delta, \alpha_\delta^+ + \delta, K\ ) \cap R(\delta, \alpha_\delta^+, K))/\#R(\delta, \alpha_\delta^+ + \delta, K\ ) = \#Q_0/\#R(\delta, \alpha_\delta^+ + \delta, K\ )$. Since $\rho_{\delta,\alpha} < p^{-t}$, we have $\#Q_0/\#R(\delta, \alpha_\delta^+ + \delta, K\ ) < p^{-t}$. It follows that $\#Q_0 < p^{-t}\#R(\delta, \alpha_\delta^+ + \delta, K\ )$. Thus we know that $_{u \quad GF(p^t), u=0} \#Q_u > (1 - p^{-t})\#R(\delta, \alpha_\delta^+ + \delta, K\ )$. Thus there exists some $b \quad GF(p^t)$ with $b = 0$ such that $\#Q_b > p^{-t}\#R(\delta, \alpha_\delta^+ + \delta, K\ )$. By definition, $Q_b = \{x_\delta^- | f(x_\delta^- + \alpha_\delta^+ + \delta) = K\ ,\ \ f(x_\delta^- + \alpha_\delta^+) = b + K\}$. Then there exists a vector $\beta_\delta^- \quad Q_b$ and then $f(\beta_\delta^- + \alpha_\delta^+ + \delta) = K\ $, $f(\beta_\delta^- + \alpha_\delta^+) = b + K$. Set $\gamma = \beta_\delta^- + \alpha_\delta^+$. Thus $f(\gamma + \delta) = K\ $ and $f(\gamma) = b + K$. Clearly $\gamma_\delta^+ = \alpha_\delta^+$ and $\gamma_\delta^- = \beta_\delta^-$. Next we choose γ as an original vector. Due to $R(\delta, \gamma_\delta^+ + \delta, K\ ) = \{x_\delta^- | f(x_\delta^- + \gamma_\delta^+ + \delta) = K\ \}$, $R(\delta, \gamma_\delta^+, b + K) = \{x_\delta^- | f(x_\delta^- + \gamma_\delta^+) = b + K\}$ and $\gamma_\delta^+ = \alpha_\delta^+$, we know that $R(\delta, \gamma_\delta^+ + \delta, K\ ) \cap R(\delta, \gamma_\delta^+, b + K) = Q_b$ and $\rho_{\delta,\gamma} = \#(R(\delta, \gamma_\delta^+ + \delta, K\ ) \cap R(\delta, \gamma_\delta^+, b + K))/\#R(\delta, \gamma_\delta^+ + \delta, K\ ) = \#Q_b/\#R(\delta, \gamma_\delta^+ + \delta, K\ ) = \#Q_b/\#R(\delta, \alpha_\delta^+ + \delta, K\ ) > p^{-t}$.

2.1 k-Cheating Immune Secret Sharing Scheme

Given a secret sharing with its defining function f on $GF(p^t)^n$. For a fixed nonzero $\delta \quad GF(p^t)^n$, due to Theorem 1, $\min\{\rho_{\delta,\alpha} | \alpha \quad GF(p^t)^n\} < p^{-t}$ implies that $\max\{\rho_{\delta,\alpha} | \alpha \quad GF(p^t)^n\} > p^{-t}$. Therefore it is desirable that $\rho_{\delta,\alpha} = p^{-t}$ holds for every $\alpha \quad GF(p^t)^n$. A secret sharing is said to be *k-cheating* if $\rho_{\delta,\alpha} = p^{-t}$ holds for every $\delta \quad GF(p^t)^n$ with $1 \le HW(\delta) \le k$ and every $\alpha \quad GF(p^t)^n$.

Theorem 2. *Given a secret sharing with its defining function f on $GF(p^t)^n$. Then this secret sharing is k-cheating immune for any integer l with $1 \le l \le k$, any $\delta \quad GF(p^t)^n$ with $HW(\delta) = l$, any $\tau \quad \delta$ and any $u, v \quad GF(p^t)$, the following conditions hold simultaneously: (i) $\#R(\delta, \tau, v) = p^{t(n-l-1)}$, (ii) $\#(R(\delta, \tau, v) \cap R(\delta, \tau + \delta, u)) = p^{t(n-l-2)}$.*

Proof. Assume that the secret sharing is k-cheating immune. Choose δ as a cheating vector and any vector $\alpha \quad GF(p^t)^n$ as an original vector. Due to Lemma 1, there exist $a, b \quad GF(p^t)$ such that $R(\delta, \alpha_\delta^+ + \delta, a) = \ $ and $R(\delta, \alpha_\delta^+, b) = \ $. Note that $R(\delta, \alpha_\delta^+ + \delta, a)$ can be partitioned into p^t parts: $R(\delta, \alpha_\delta^+ + \delta, a) = {}_{v \ GF(p^t)} R(\delta, \alpha_\delta^+ + \delta, a) \cap R(\delta, \alpha_\delta^+, v)$. Assume that $R(\delta, \alpha_\delta^+ + \delta, a) \cap R(\delta, \alpha_\delta^+, v) = \ $ for some $v \quad GF(p^t)$. Then there exists a vector $\beta_\delta^- \quad R(\delta, \alpha_\delta^+ + \delta, a) \cap R(\delta, \alpha_\delta^+, v)$. Set $\gamma = \beta_\delta^- + \alpha_\delta^+$. Since the secret sharing is k-cheating immune, $\#(R(\delta, \gamma_\delta^+ + \delta, a) \cap R(\delta, \gamma_\delta^+, v))/\#R(\delta, \gamma_\delta^+ + \delta, a) = \rho_{\delta,\gamma} = p^{-t}$, where $\gamma_\delta^+ = \alpha_\delta^+$. Thus $\#R(\delta, \alpha_\delta^+ + \delta, a) = p^t\#(R(\delta, \alpha_\delta^+ + \delta, a) \cap R(\delta, \alpha_\delta^+, v))$ whenever $R(\delta, \alpha_\delta^+ + \delta, a) \cap R(\delta, \alpha_\delta^+, v) = \ $. It follows that $\#R(\delta, \alpha_\delta^+ + \delta, a) = {}_{v \ GF(p^t)} \#(R(\delta, \alpha_\delta^+ + \delta, a) \cap R(\delta, \alpha_\delta^+, v))$. Combing the above two equalities, we know that $R(\delta, \alpha_\delta^+ + \delta, a) \cap R(\delta, \alpha_\delta^+, v) = \ $ for every $v \quad GF(p^t)$ and thus $\#(R(\delta, \alpha_\delta^+ + \delta, a) \cap R(\delta, \alpha_\delta^+, v)) = p^{-t}\#R(\delta, \alpha_\delta^+ + \delta, a)$ for every $v \quad GF(p^t)$. Replacing α, δ, by $\alpha + \delta$, $(p-1)\delta$ respectively, due to the same arguments, we have $\#(R((p-1)\delta, \alpha_\delta^+ + p\delta, b) \cap R((p-1)\delta, \alpha_\delta^+ + \delta, u)) = p^{-t}\#R((p-1)\delta, \alpha_\delta^+ + p\delta, b)$ for every $u \quad GF(p^t)$. Since the characteristic of the finite field $GF(p^t)$ is p, $pe = 0$ for every $e \quad GF(p^t)$. It follows that $\#(R((p-1)\delta, \alpha_\delta^+, b) \cap R((p-1)\delta, \alpha_\delta^+ +$

$\delta, u)) = p^{-t}\#R((p-1)\delta, \alpha_\delta^+, b)$ for every u $GF(p^t)$. Using Lemma 1, we obtain $\#(R(\delta, \alpha_\delta^+, b) \cap R(\delta, \alpha_\delta^+ + \delta, u)) = p^{-t}\#R(\delta, \alpha_\delta^+, b)$ for every u $GF(p^t)$. Recall that $R(\delta, \alpha_\delta^+ + \delta, a) =$ and $R(\delta, \alpha_\delta^+, b) =$. Therefore we have proved that $R(\delta, \alpha_\delta^+, v) =$ and $R(\delta, \alpha_\delta^+ + \delta, u) =$ for every u, v $GF(p^t)$. Due to the same reasoning, we have $\#(R(\delta, \alpha_\delta^+ + \delta, u) \cap R(\delta, \alpha_\delta^+, v)) = p^{-t}\#R(\delta, \alpha_\delta^+ + \delta, u)$ and $\#(R(\delta, \alpha_\delta^+, v) \cap R(\delta, \alpha_\delta^+ + \delta, u)) = p^{-t}\#R(\delta, \alpha_\delta^+, v)$ for every u, v $GF(p^t)$. Comparing the above two equalities, we conclude that $\#R(\delta, \alpha_\delta^+ + \delta, u) = \#R(\delta, \alpha_\delta^+, v)$ for every u, v $GF(p^t)$. Therefore both $\#R(\delta, \alpha_\delta^+ + \delta, u)$ and $\#R(\delta, \alpha_\delta^+, v)$ are constant. Note that $_{v \quad GF(p^t)} \#R(\delta, \alpha_\delta^+, v) = p^{t(n-l)}$. We have proved that $\#R(\delta, \alpha_\delta^+, v) = p^{t(n-l-1)}$ for any v $GF(p^t)$. Thus we have proved that $\#(R(\delta, \alpha_\delta^+ + \delta, u) \cap R(\delta, \alpha_\delta^+, v)) = p^{t(n-l-2)}$ for every u, v $GF(p^t)$. For any τ δ, choose α $GF(p^t)^n$ such that $\alpha_\delta^+ = \tau$. Clearly both conditions (i) and (ii) hold. Conversely assume the defining function f satisfies conditions (i) and (ii). Choose any δ $GF(p^t)^n$ with $HW(\delta) = l$, where $1 \le l \le k$, as a cheating vector and any α as an original vector. Set $f(\alpha) = K$ and $f(\alpha + \delta) = K$. By definition, $\rho_{\delta,\alpha} = \#(R(\delta, \alpha_\delta^+ + \delta, K\ ) \cap R(\delta, \alpha_\delta^+, K))/\#R(\delta, \alpha_\delta^+ + \delta, K\ )$. Due to conditions (i) and (ii), $\rho_{\delta,\alpha} = p^{-t}$. Thus we have proved that the secret sharing is k-cheating immune.

Theorem 3. *Given a secret sharing with its defining function f on $GF(p^t)^n$. Then the following statements are equivalent: (i) this secret sharing is k-cheating immune, (ii) for any integer l with $1 \le l \le k$, any δ $GF(p^t)^n$ with $HW(\delta) = l$, any τ δ and any u, v $GF(p^t)$, we have $\#(R(\delta, \tau, v) \cap R(\delta, \tau + \delta, u)) = p^{t(n-l-2)}$, (iii) for such l, δ, τ, u and v mentioned in (ii), the system of equations:*

$$\begin{array}{l} f(x_\delta^- + \tau + \delta) = u \\ f(x_\delta^- + \tau) = v \end{array} \quad \text{has precisely } p^{t(n-l-2)} \text{ solutions on } x_\delta^-.$$

Proof. Clearly (ii) (iii). Due to Theorem 2, (i) = (ii). To complete the proof, we only need prove that (ii) = (i). Assume that (ii) holds. Thus $\#(R(\delta, \tau, v) \cap R(\delta, \tau + \delta, u)) = p^{t(n-l-2)}$ for every u, v $GF(p^t)$. Note that $R(\delta,\tau,v) = {}_{u \quad GF(p^t)} R(\delta, \tau, v) \cap R(\delta, \tau + \delta, u)$ and then $\#R(\delta, \tau, v) = {}_{u \quad GF(p^t)} \#(R(\delta, \tau, v) \cap R(\delta, \tau + \delta, u))$. This proves that $\#R(\delta, \tau, v) = p^{t(n-l-1)}$. Using Theorem 2, we have proved that (i) holds.

2.2 Constructions of k-Cheating Immune Secret Sharing

Let h is a function of degree two on $GF(p^t)^n$ and $\delta = \{\delta_1, \dots, \delta_n)$ be a nonzero vector in $GF(p^t)^n$. Set $J = \{j \mid \delta_j = 0,\ 1 \le j \le n\}$. Let τ be any vector in $GF(p^t)^n$ with τ δ. It is easy to verify that $x_j x_i$ is a term in $h(x_\delta^+ + \tau)$ $x_j x_i$ is a term in h also j, i J $x_j x_i$ is a term in $h(x_\delta^+ + \tau + \delta)$. Thus $h(x_\delta^+ + \tau)$ and $h(x_\delta^+ + \tau + \delta)$ have the same quadratic terms, and thus $h(x_\delta^+ + \tau + \delta) - h(x_\delta^+ + \tau)$ must be an affine function. The function h of degree two is said to have the *property B(k)* if for any δ $GF(p^t)^n$ with $1 \le HW(\delta) \le k$ and any τ δ, $h(x_\delta^+ + \tau + \delta) - h(x_\delta^+ + \tau)$ is a non-constant affine function.

Lemma 2. *Let f_1 and f_2 be two functions on $GF(p^t)^{n_1}$ and $GF(p^t)^{n_2}$ respectively. Set $f(x) = f_1(y) + f_2(z)$ where $x = (y, z)$ where y $GF(p^t)^{n_1}$ and z $GF(p^t)^{n_2}$. Then f is balanced if f_1 or f_2 is balanced,*

The above lemma can be verified directly. The special case of $p = 2$ and $t = 1$ was given in Lemma 12 of [8]. Using Lemma 2, we can prove

Lemma 3. *Let f_1 and f_2 be two functions of degree two on $GF(p^t)^{n_1}$ and $GF(p^t)^{n_2}$ respectively. Set $f(x) = f_1(y) + f_2(z)$ where $x = (y, z)$ where y $GF(p^t)^{n_1}$ and z $GF(p^t)^{n_2}$. Then f has the property B(k) if both f_1 and f_2 have the property B(k).*

Theorem 4. *Let k and s be two positive integers with $s \geq k+1$, h_j be a balanced function of degree two on $GF(p^t)^{n_j}$ satisfying the property B(k), $j = 1, \ldots, s$. Set $n = n_1 + \cdots + n_s$. Define a function f on $GF(p^t)^n$ such as $f(x) = h_1(y) + \cdots + h_s(z)$ where $x = (y, \ldots, z)$, h_i and h_j have disjoint variables if $i = j$. Then the secret sharing with the defining function f is k-cheating immune.*

Proof. Let $\delta = (\delta_1, \ldots, \delta_n)$ $GF(p^t)^n$ with $HW(\delta) = l$, where $1 \leq l \leq k$. Let τ be any vector in $GF(p^t)^n$ with τ δ. Consider the system of equations: $\begin{array}{l} f(x_{\overline{\delta}} + \tau + \delta) = u \\ f(x_{\overline{\delta}} + \tau) = v \end{array}$. Set $J = \{j \mid \delta_j = 0, \ 1 \leq j \leq n\}$. Note that $\#J = HW(\delta) = l$. We write $J = \{j_1, \ldots, j_l\}$. Since $l \leq k \leq s-1$, there exists some j_0 with $1 \leq j_0 \leq s$ such that each variable of h_{j_0} is not in $\{x_{j_1}, \ldots, x_{j_l}\}$. For the sake of convenience, we assume that $j_0 = s$ and thus h_s remains in both equations above. Thus if j J then $j \leq n - n_s$. Write $x = (\mu, z)$, where μ $GF(p^t)^{n-n_s}$ and z $GF(p^t)^{n_s}$. Define a vector σ $GF(p^t)^{n-n_s}$ such that $\sigma = (\sigma_1, \ldots, \sigma_{n-n_s})$ satisfying $\sigma_j = \delta_j$, $j = 1, \ldots, n - n_s$. Thus $HW(\sigma) = HW(\delta) = \#J = l$ and $x_{\overline{\delta}} = (\mu_{\overline{\sigma}}, z)$. We rewrite the above system of equations as $\begin{array}{l} g_1(\mu_{\overline{\delta}}) + h_s(z) = u \\ g_2(\mu_{\overline{\delta}}) + h_s(z) = v \end{array}$ where both g_1 and g_2 are functions on $GF(p^t)^{n-n_s}$. Note that $x_j x_i$ is a term in $g_1 + h_s$ $x_j x_i$ is a term in f and j, i J $x_j x_i$ is a term in $g_2 + h_s$. Thus $g_1 + h_s$ and $g_2 + h_s$ have the same quadratic terms. Therefore $g_1 - g_2$ is an affine function. Set $g_2 - g_1 = \psi$. Note that the above system of equations is equivalent to $\begin{array}{l} g_1(\mu_{\overline{\sigma}}) + h_s(z) = u \\ \psi(\mu_{\overline{\sigma}}) = u - v \end{array}$. Since each h_j has the property B(k), ψ is a non-constant affine function and thus the equation $\psi(\mu_{\overline{\sigma}}) = u - v$ has $p^{t(n-n_s-l-1)}$ solutions on $\mu_{\overline{\sigma}}$. For each fixed solution $\mu_{\overline{\sigma}}$ of the equation $\psi(\mu_{\overline{\sigma}}) = u - v$, since h_s is balanced, $g_1(\mu_{\overline{\sigma}}) + h_s(z)$ takes u precisely $p^{t(n_s-1)}$ times while z runs through $GF(p^t)^s$ once. Therefore the above system of equations has precisely $p^{t(n-n_s-l-1)} \cdot p^{t(n_s-1)} = p^{t(n-l-2)}$ solutions on $(\mu_{\overline{\sigma}}, z) = x_{\overline{\delta}}$. Due to Theorem 3, we have proved that the secret sharing with the defining function f, defined in the theorem, is k-cheating immune.

Lemma 4. *Define a function χ_{2k+1} on $GF(p^t)^{2k+1}$ by $\chi_{2k+1}(x_1, \ldots, x_{2k+1}) = x_1x_2 + x_2x_3 + \cdots + x_{2k}x_{2k+1} + x_{2k+1}x_1$. Then (i) the function χ_{2k+1} is balanced, (ii) χ_{2k+1} satisfies the property B(k).*

Proof. By a nonsingular linear transform on the variables, the function χ_{2k+1} can be transformed to the form of $y_1y_2+y_2y_3+\cdots+y_{2k-1}y_{2k}\pm y_{2k+1}^2$. It is easy to verify that the function $h(y_1,\ldots,y_{2k+1}) = y_{2k+1}^2$ is balanced. Due to Lemma 2, χ_{2k+1} is balanced. Next we prove the part (ii) of the lemma. Let $\delta \quad GF(p^t)^{2k+1}$ with $HW(\delta) = l$, where $1 \le l \le k$, and $\tau \quad \delta$. Write $\delta = (\delta_1,\ldots,\delta_{2k+1})$ and $J = \{j \mid \delta_j = 0,\ 1 \le j \le 2k+1\}$. Clearly, $\#J = HW(\delta) = l$. The index $i \quad J$ is said to be *associated* with $j \quad J$ if x_jx_i is a term in χ_{2k+1}. Due to the structure of χ_{2k+1}, each $i \quad J$ is associated at most two elements of J. Since $l \le k$, it is easy to verify that there exists some j_0 such that $j_0 \quad J$, $j_0+1 \quad J$ and j_0+1 is associated with j_0 only – Case 1, otherwise there exists some j_0 such that $i_0 \quad J$, $i_0-1 \quad J$ and i_0-1 is associated with i_0 only – Case 2. Assume Case 1 occurs. Write $\tau = (\tau_1,\ldots,\tau_{2k+1})$. Since $j_0 \quad J$, we know that $\delta_{j_0} = 0$. Therefore $\delta_{j_0}x_{j_0+1}$ must appear in $\chi_{2k+1}(x_\delta^+ + \tau + \delta) - \chi_{2k+1}(x_\delta^+ + \tau)$. This proves that χ_{2k+1} has the property B(k) in Case 1. Similarly we can prove that χ_{2k+1} has the property B(k) in Case 2.

Using Lemmas 2, 3 and 4, we obtain the following:

Lemma 5. *Define a function χ_{4k+2} on $GF(p^t)^{4k+2}$ by $\chi_{4k+2}(x_1,\ldots,x_{4k+2}) = \chi_{2k+1}(x_1,\ldots,x_{2k+1}) + \chi_{2k+1}(x_{2k+2},\ldots,x_{4k+2})$. Then (i) the function χ_{4k+2} is balanced, (ii) χ_{4k+2} satisfies the property B(k).*

χ_n in Lemma 4 or 5 has been defined for odd n and even n with $n \equiv 2 \bmod 4$. Due to Lemma 4, Lemma 5 and Theorem 4, we have the following construction.

Theorem 5. *Let k and s be positive integers with $s \ge k+1$. Let $n_1,\ldots,n_s = 4k+1$ or $4k+2$, and $n = n_1+\cdots+n_s$. Define a function on $GF(p^t)^n$ such as $f(x) = \chi_{n_1}(y)+\cdots+\chi_{n_s}(z)$ where $x = (y,\ldots,z)$, $y \quad GF(p^t)^{n_1},\ldots,z \quad GF(p^t)^{n_s}$, each χ_{n_j} has been defined in (4) or (5), and $\chi_{n_1},\ldots,\chi_{n_s}$ have disjoint variables mutually. Then the secret sharing with the defining function f is k-cheating immune.*

Note that $n = n_1+\cdots+n_s$, defined in Theorem 5, can be expressed as $n = (4k+1)r + (4k+2)q$ where $r \ge 0$ and $q \ge 0$ are integers. Since $4k+1$ and $4k+2$ are relatively prime, any integer can also be written as $(4k+1)r+(4k+2)q$ where r and q are integers. Furthermore it is easy to verify that any integer n with $n \ge (4k+1)^2$ can be expressed as $n = (4k+1)r + (4k+2)q$ where $r, q \ge 0$. Since $n \ge (4k+1)^2$, $s = r+q > k+1$ where s was mentioned in Theorem 5. Using Theorem 5, we can construct k-cheating immune secret sharing with n participants where $n \ge (4k+1)^2$.

3 Generalised Model of Cheating

Given a function f on $GF(p^t)^n$, we introduce the following notations:

- Let $\alpha \quad GF(p^t)^n$ be the sequence of shares held by the group $P = \{P_1,\ldots,P_n\}$ of n participants and the secret $K = f(\alpha)$.

- The collection of cheaters is determined by the sequence $\delta = (\delta_1, \delta_2, \ldots, \delta_n)$ where P_i is a cheater   if $\delta_i = 0$.
- At the pooling time, the cheaters submit their shares. This time it is assumed that cheaters may submit a mixture of valid and invalid shares. The honest participants always submit their valid shares. The collection of cheaters who submit invalid shares is determined by the sequence $\tau = (\tau_1, \ldots, \tau_n)$ where $\tau_j = 0$ P_j is honest or P_j is a cheater who submits a valid share, in other words, $\tau_j = 0$ P_j is a cheater who submits an invalid share. Clearly τ δ. We assume that there exists at least one cheater who submits invalid share, in other words, we only consider the case that τ is nonzero or $HW(\tau) > 0$. We consider the vector $\alpha + \tau$. Due to the properties of operations x_δ^+ and x_δ^-, $\alpha + \tau = \alpha_\delta^- + \alpha_\delta^+ + \tau$. The combiner obtains $\alpha + \tau$ that splits into two parts: α_δ^- – the part submitted by honest participants and $\alpha_\delta^+ + \tau$ the part submitted by cheaters. The combiner returns an invalid secret $K \ = f(\alpha + \tau)$.
- $R(\delta, \alpha_\delta^+ + \tau, K \ )$, or $\{x_\delta^- | f(x_\delta^- + \alpha_\delta^+ + \tau) = K \ \}$, where α_δ^+ determines valid shares held by the cheaters, represents the view of the cheater after getting back K from the combiner.
- The set $R(\delta, \alpha_\delta^+, K)$, or $\{x_\delta^- | f(x_\delta^- + \alpha_\delta^+) = K\}$, determines a collection of rows of T with the correct secret K and valid shares held by the cheaters.

In generalised model of cheating, τ is used to determine how to cheat while δ is only used to determine which participants are dishonest, therefore we can define δ as a $(0,1)$-vector in $GF(p^t)^n$. However, in basic model of cheating, δ is not only used to determine which participants are dishonest but also used to determine how to cheat, thus δ has a more general form.

The function f is called the *defining function*. The nonzero vector $\delta = (\delta_1, \ldots, \delta_n)$ is called a *cheating vector*, the nonzero vector τ δ is called an *active cheating vector*, α is called a *original vector*. The value of $\rho_{\delta,\tau,\alpha} = \#(R(\delta, \alpha_\delta^+ + \tau, K \ ) \cap R(\delta, \alpha_\delta^+, K))/\#R(\delta, \alpha_\delta^+ + \tau, K \ )$ expresses the probability of cheater success with respect to δ, τ and α. As an original vector α is always in $R(\delta, \alpha_\delta^+ + \tau, K \ ) \cap R(\delta, \alpha_\delta^+, K)$, the probability of successful cheating always satisfies $\rho_{\delta,\tau,\alpha} > 0$. Clearly the number of cheaters is equal to $HW(\delta)$ and the number of active cheaters is equal to $HW(\tau)$. In particular, if $\tau = \delta$, we regain basic model of cheating. ¿From now, we consider secret sharing against cheating by generalised model of cheating.

3.1 Strictly k-Cheating Immune Secret Sharing Scheme

By using the same arguments as in the proof of Theorem 1, we can state.

Theorem 6. *Given a secret sharing with its defining function f on $GF(p^t)^n$. Let δ $GF(p^t)^n$ with $0 < HW(\delta) < n$ be a cheating vector, τ δ with $\tau = 0$ be an active cheating vector, and α $GF(p^t)^n$ be an original vector. If $\rho_{\delta,\tau,\alpha} < p^{-t}$ then there exists a vector γ $GF(p^t)^n$ such that $\rho_{\delta,\tau,\gamma} > p^{-t}$.*

For the same reason mentioned in Section 2.1, we introduce the concept of k-cheating immune secret sharing scheme. Given a secret sharing with its defining function f on $GF(p^t)^n$. Let k be an integer with $1 \le k \le n-1$. The secret sharing is said to be *strictly k-cheating immune* if the probability of successful cheating satisfies $\rho_{\delta,\tau,\alpha} = p^{-t}$ for every $\delta \in GF(p^t)^n$ and any $\tau \preceq \delta$ with $1 \le HW(\tau) \le HW(\delta) \le k$ and every $\alpha \in GF(p^t)^n$. The following is a relationship between the two models of cheating immune secret sharing.

Theorem 7. *Given a secret sharing with its defining function f on $GF(p^t)^n$. Then the secret sharing is strictly k-cheating immune $\iff$ for any integer r with $0 \le r \le k-1$, any subset $\{j_1,\dots,j_r\}$ of $\{1,\dots,n\}$ and any $a_1,\dots,a_r \in GF(p^t)$, $f(x_1,\dots,x_n)|_{x_{j_1}=a_1,\dots,x_{j_r}=a_r}$, as a function on $GF(p^t)^{n-r}$ with the variables $x_{i_1},\dots,x_{i_{n-r}}$, where $\{i_1,\dots,i_{n-r}\} \cup \{j_1,\dots,j_r\} = \{1,\dots,n\}$, is the defining function on $GF(p^t)^{n-r}$ of a $(k-r)$-cheating immune secret sharing.*

Proof. Assume that the secret sharing is strictly k-cheating immune. Let g be a function on $GF(p^t)^{n-r}$ given by $g = f(x_1,\dots,x_n)|_{x_{j_r}=a_1,\dots,x_{j_r}=a_r}$. Comparing basic model of cheating with generalised model of cheating, since f is the defining function on $GF(p^t)^n$ of a strictly k-cheating immune secret sharing in generalised model of cheating, we know that g is the defining function on $GF(p^t)^{n-r}$ of a $(k-r)$-cheating immune secret sharing against basic model of cheating. We have proved the necessity. By definition, we can invert the above reasoning and prove the sufficiency.

3.2 Construction of Strictly k-Cheating Immune Secret Sharing

Lemma 6. *Let a function f of degree two on $GF(p^t)^n$ do not have a nonzero constant term, in other words, $f(0,\dots,0) = 0$, where 0 denotes the zero element in $GF(p^t)$. Then f is balanced $\iff$ there exists a nonzero vector $\alpha \in GF(p^t)^n$ such that $f(x+\alpha) - f(x)$ is constant and $f(\alpha) = 0$.*

Lemma 6 with $p = 2$ and $t = 1$ is a special case of the lemma in [4]. Lemma 6 can be proved using the same arguments as those used for the proof of the lemma in [4].

Lemma 7. *Let $\lambda_{n,p}$ be a function on $GF(p^t)^n$ $(n \ge 2p^2 + p)$ defined by $\lambda_{n,p}(x_1,\dots,x_n) = x_1 + \sum_{j=1}^{n}(x_j x_{[j+1]_{(n)}} + x_j x_{[j+2]_{(n)}} + \cdots + x_j x_{[j+p]_{(n)}})$ where $[i]_{(n)}$ denotes the integer j such that $1 \le j \le n$ and $j \equiv i \bmod n$ (we replace i by $[i]_{(n)}$ as i is possibly greater than n). Then (i) $\lambda_{n,p}$ is balanced, (ii) for any r with $0 \le r \le p-1$, any subset $\{j_1,\dots,j_r\}$ of $\{1,\dots,n\}$ and any $a_1,\dots,a_r \in GF(p^t)$, $\lambda_{n,p}(x_1,\dots,x_n)|_{x_{j_1}=a_1,\dots,x_{j_r}=a_r}$, as a function on $GF(p^t)^{n-r}$ with the variables $x_{i_1},\dots,x_{i_{n-r}}$, where $\{i_1,\dots,i_{n-r}\} \cup \{j_1,\dots,j_r\} = \{1,\dots,n\}$, satisfies the property B(p).*

Proof. From the construction of $\lambda_{n,p}$, for any j with $1 \le j \le n$, there precisely exist $2p$ quadratic terms of $\lambda_{n,p}$: $x_j x_{[j+i]_{(n)}}$ and $x_j x_{[j-i]_{(n)}}$ containing x_j where $i = 1,\dots,p$. It is easy to verify that $\lambda_{n,p}$ has precisely np quadratic terms, in addition, a linear term x_1. Set $g = \lambda_{n,p} - x_1$ or $g(x_1,\dots,x_n) = \sum_{j=1}^{n}(x_j x_{[j+1]_{(n)}} + x_j x_{[j+2]_{(n)}} + \cdots + x_j x_{[j+p]_{(n)}})$, and $\alpha = (1,\dots,1)$ where 1 denotes the identity in

$GF(p^t)$. Recall that the characteristic of the finite field $GF(p^t)$ is p. Then $pe = 0$ holds for any element $e \quad GF(p^t)$. Thus it is easy to verify that $g(x+\alpha)-g(x) = 0$ and $g(\alpha) = 0$. Therefore $\lambda_{n,p}(x+\alpha) - \lambda_{n,p}(x) = 1$ and $\lambda_{n,p}(\alpha) = 1$. Due to Lemma 6, we know that $\lambda_{n,p}$ is balanced. Next we prove the part (ii) of the lemma. Write $h(x_{i_1},\dots,x_{i_{n-r}}) = \lambda_{n,p}(x_1,\dots,x_n)|_{x_{j_1}=a_1,\dots,x_{j_r}=a_r}$. Set $x_{i_1} = y_1,\dots,x_{i_{n-r}} = y_{n-r}$ and $y = (y_1,\dots,y_{n-r})$. Then we consider the function $h(y_1,\dots,y_{n-r})$. Recall that for each j, $1 \le j \le n$, x_j appears precisely in $2p$ quadratic terms of $\lambda_{n,p}$: $x_j x_{[j+i]_{(n)}}$ and $x_j x_{[j-i]_{(n)}}$ where $i = 1,\dots,p$. Since $r \le p-1$, it is easy to see that for each j, $1 \le j \le n-r$, there at least two quadratic terms of h. Let $\delta \quad GF(p^t)^{n-r}$ be a cheating vector with $HW(\delta) = l$, where $1 \le l \le p$, and $\tau \quad \delta$ be an active cheating vector. Write $\delta = (\delta_1,\dots,\delta_{n-r})$ and $J = \{j \mid \delta_j = 0,\ 1 \le j \le n-r\}$. Clearly $\#J = HW(\delta) = l$. We do not need to consider any term $y_j y_i$ in h with $j, i \quad J$ as it does not appear in $h(y_\delta^+ + \tau + \delta) - h(y_\delta^+ + \tau)$. Since $n-r \ge 2p^2+1$, there exist some integers j_0 and m such that $m \ge 2p+1$, $[j_0+m]_{(n-r)} \quad J$ and $\{[j_0+1]_{(n-r)}, [j_0+2]_{(n-r)},\dots,[j_0+m-1]_{(n-r)}\} \cap J = \quad$. Due to the structures of $\lambda_{n,p}$ and h, there exists some $[i_0]_{(n-r)} \quad \{[j_0+1]_{(n-r)}, [j_0+2]_{(n-r)},\dots,[j_0+m-1]_{(n-r)}\}$ such that $y_{j_0} y_{[i_0]_{(n-r)}}$ is a term in h but $y_{[j_0+m]_{(n-r)}} y_{[i_0]_{(n-r)}}$ is not a term in h. Furthermore, due to the structures of $\lambda_{n,p}$ and h, $y_j y_{[i_0]_{(n-r)}}$ cannot be a term in h for any $j \quad J$ with $j = j_0$. Since $[i_0]_{(n-r)} \quad J$, as the discussion before, any term $y_j y_{[i_0]_{(n-r)}}$ with $j \quad J$ does not appear in $h(y_\delta^+ + \tau + \delta) - h(y_\delta^+ + \tau)$. Since $j_0 \quad J$, we know that $\delta_{j_0} = 0$. Therefore $\delta_{j_0} y_{[i_0]_{(n-r)}}$ appears in $h(y_\delta^+ + \tau + \delta) - h(y_\delta^+ + \tau)$. This proves that h has the property B(p).

Based on Theorem 7 and Lemma 7, we have the following construction.

Theorem 8. *Let $GF(p^t)$ be a finite field, s be an integer with $s \ge 2p$. Let $n_1,\dots,n_s = 2p^2+p$ or $2p^2+p+1$, and $n = n_1+\cdots+n_s$. Define a function on $GF(p^t)^n$ such as $f(x) = \lambda_{n_1,p}(y)+\cdots+\lambda_{n_s,p}(z)$ where $x = (y,\dots,z)$, $y \quad GF(p^t)^{n_1},\dots,z \quad GF(p^t)^{n_s}$, each $\lambda_{n_j,p}$ has been defined in Lemma 7 and $\lambda_{n_i,p},\dots,\lambda_{n_j,p}$ have disjoint variables if $i = j$. Then the secret sharing with the defining function f is strictly p-cheating immune.*

Proof. Let r be an integer with $0 \le r \le p-1$ and $\{j_1,\dots,j_r\}$ be a subset of $\{1,\dots,n\}$. Since $r \le p-1$, there exist at least $s-r \ge p+1$ functions among $\lambda_{n_1,p},\dots,\lambda_{n_s,p}$, each of whose variables is not included in $\{x_{j_1},\dots,x_{j_r}\}$. Without loss of generality, we assume that each variable of $\lambda_{n_{r+1},p},\dots,\lambda_{n_s,p}$ is not included in $\{x_{j_1},\dots,x_{j_r}\}$. Therefore for any $a_1,\dots,a_r \quad GF(p^t)$, f can be expressed as $f|_{x_{j_1}=a_1,\dots,x_{j_r}=a_r} = g + \lambda_{n_{r+1},p} + \lambda_{n_{r+2},p} + \cdots + \lambda_{n_s,p}$ where $g = (\lambda_{n_1,p}+\cdots+\lambda_{n_r,p})|_{x_{j_1}=a_1,\dots,x_{j_r}=a_r}$. Due to Lemmas 7, $\lambda_{n_j,p}|_{x_{j_1}=a_1,\dots,x_{j_r}=a_r}$ has the property B(p), $j = 1,\dots,r$ and thus from Lemma 3, g has the property B(p) and thus $f|_{x_{j_1}=a_1,\dots,x_{j_r}=a_r}$ has the property B(p). Since each $\lambda_{n_j,p}$ is balanced, due to Lemma 2, $f|_{x_{j_1}=a_1,\dots,x_{j_r}=a_r}$ is balanced. Applying Theorem 4 to $f|_{x_{j_1}=a_1,\dots,x_{j_r}=a_r} = g + \lambda_{n_{r+1},p} + \lambda_{n_{r+2},p} + \cdots + \lambda_{n_s,p}$, we conclude the secret sharing with the defining function $f|_{x_{j_1}=a_1,\dots,x_{j_r}=a_r}$ is p-cheating immune. Finally, using Theorem 7, we know that the secret sharing with the defining function f is strictly p-cheating immune.

By using the same arguments as in the last paragraph of Section 2.2, it is easy to verify that any integer $n \geq (2p^2 + p)^2$ can be expressed as $n = r(2p^2 + p) + q(2p^2 + p + 1)$ where $r, q \geq 0$. Since $n \geq (2p^2 + p)^2$, $s = r + q \geq 2p$ where s was mentioned in Theorem 7. Using Theorem 7, we can construct p-cheating immune secret sharing with n participants where $n \geq (2p^2 + p)^2$.

4 Conclusions and Remarks

We have considered secret sharing over finite field and its resistance against cheating by a group of k dishonest participants. We have proved that the probability of successful cheating is always higher than p^{-t}. The secret scheme is said to be k-cheating immune if the probability of successful cheating is p^{-t} for any group of k or less participants. We have characterised k-cheating immune secret sharing scheme by examining its defining function. This characterisation enables us to construct k-cheating immune secret sharing scheme. Being more precise, we have studied two cases. In the first case, the group of cheaters always submit invalid shares. While in the second case, the group is more flexible as they collectively decide which of their shares should be modified and which should be submitted in their original form.

Acknowledgement

The first author was supported by the Large ARC Grant A00103078. The second author was supported by a Queen Elizabeth II Fellowship (227 23 1002).

References

1. M. Carpentieri. A perfect threshold secret sharing scheme to identify cheaters. *Designs, Codes and Cryptography*, 5(3):183–187, 1995.
2. M. Carpentieri, A. De Santis, and U. Vaccaro. Size of shares and probability of cheating in threshold schemes. *Advances in Cryptology - EUROCRYPT'93*, LNCS No. 765, pages 118–125. Springer-Verlag, 1993.
3. P. Feldman. A practical scheme for non-interactive verifiable secret sharing. In *Proceedings of the* 28*th IEEE Symposium on Foundations of Computer Science*, pages 427–437. IEEE, 1987.
4. K. Nyberg and L. R. Knudsen. Provable security against differential cryptanalysis. In *Advances in Cryptology - CRYPTO'92*, LNCS No. 740, pages 566–574. Springer-Verlag, 1993.
5. T.P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In J. Feigenbaum, editor, *Advances in Cryptology - CRYPTO'91*, LNCS No. 576, pages 129–140. Springer-Verlag , 1992.
6. T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of 21st ACM Symposium on Theory of Computing*, pages 73–85, 1989.
7. B. Schoenmakers. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In M. Wiener, editor, *Advances in Cryptology - CRYPTO'99*, LNCS No. 1666, pages 148–164. Springer - Verlag, 1999.

8. J. Seberry, X. M. Zhang, and Y. Zheng. Nonlinearity and propagation characteristics of balanced boolean functions. *Information and Computation*, 119(1):1–13, 1995.
9. M. Stadler. Publicly verifiable secret sharing. In U. Maurer, editor, *Advances in Cryptology - EUROCRYPT'96*, LNCS No. 1070, pages 190–199. Springer-Verlag, 1996.
10. D.R. Stinson. *Cryptography: Theory and Practice*. CRC Press, 1995.
11. Martin Tompa and Heather Woll. How to share a secret with cheaters. In A.M. Odlyzko, editor, *Advances in Cryptology - CRYPTO'86*, LNCS No. 263, pages 261–265. Springer-Verlag, 1987.

An Application of Sieve Methods to Elliptic Curves

S. Ali Miri[1] and V. Kumar Murty[2]

[1] School of Information Technology and Engineering (SITE), Univesity of Ottawa,
Ottawa, ON K1N 6N5, Canada,
samiri@site.uottawa.ca

[2] Department of Mathematics, University of Toronto,
100 St. George Street, Toronto, ON M5S 3G3, Canada,
murty@math.toronto.edu

Abstract. Let E be an elliptic curve defined over the rationals. Koblitz conjectured that the number of primes $p \leq x$ such that the number of points $|E(\mathbb{F}_p)|$ on the curve over the finite field of p elements has prime order is asymptotic to

$$C_E \frac{x}{(\log x)^2}$$

for some constant C_E. We consider curves without complex multiplication. Assuming the GRH (that is, the Riemann Hypothesis for Dedekind zeta functions) we prove that for

$$\frac{x}{(\log x)^2}$$

primes $p \leq x$, the group order $|E(\mathbb{F}_p)|$ has at most 16 prime divisors. We also show (again, assuming the GRH) that for a random prime p, the group order $|E(\mathbb{F}_p)|$ has $\log\log p$ prime divisors.

1 Introduction

In cryptographic applications, one works with elliptic curves E over a finite field $\mathbb{F}_q$ with the property that the group order $|E(\mathbb{F}_q)|$ is prime or nearly prime.

Let E be an elliptic curve over $\mathbb{Q}$. Koblitz [3] considers the problem of estimating the number of primes $p \leq x$ so that $|E(\mathbb{F}_p)|$ is of prime order. He conjectured that this number is

$$C_E \frac{x}{(\log x)^2}$$

where C_E is an explicit constant depending only on E. Let us set

$$N_p = |E(\mathbb{F}_p)|.$$

In this paper, we shall prove the following results.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 91–98, 2001.

Theorem 1 *Assume the GRH (i.e. the Riemann Hypothesis for all Dedekind zeta functions of number fields). Let $E/\mathbb{Q}$ be an elliptic curve without complex multiplication. Then there are*

$$\frac{x}{(\log x)^2}$$

primes $p \leq x$ so that $N_p = |E(\mathbb{F}_p)|$ has at most 16 prime factors (counting multiplicity).

We also show that for a random prime p, the group order should be divisible by about $\log\log p$ primes. We make this precise in the following theorem.

Theorem 2 *Assume the GRH. Let $\quad > 0$. Except possibly for*

$$O(\pi(x)/(\log\log x)^{2 \quad})$$

of the primes $p \leq x$, the number $\nu(N_p)$ of prime divisors of N_p satisfies

$$\log\log p - (\log\log p)^{\frac{1}{2}+ \quad} < \nu(N_p) < \log\log p + (\log\log p)^{\frac{1}{2}+ \quad} .$$

We remark that the question of the primality of the number of points on an elliptic curve has been studied by other authors. In particular, Galbraith and Mckee [2] fix a prime p and consider elliptic curves over the field $\mathbb{F}_p$. They make a precise conjecture about the probability that such an elliptic curve will have a prime number of points.

2 An Application of the Chebotarev Density Theorem

Given an elliptic curve $E/\mathbb{Q}$ and a prime l, we can consider the Galois representation

$$\rho_l : Gal(\bar{\mathbb{Q}}/\mathbb{Q}) \to GL_2(\mathbb{Z}_l)$$

where $\mathbb{Z}_l$ denotes the l-adic integers and $\bar{\mathbb{Q}}$ is an algebraic closure of $\mathbb{Q}$. This representation comes from the action of the Galois group on the Tate module

$$T_l(E) = \varprojlim E[l^n] = \mathbb{Z}_l \quad \mathbb{Z}_l.$$

If E does not have complex multiplication, then a result of Serre [6] states that for $l \quad 1$, ρ_l is surjective. Multiplying together these representations for different $l|d$, we get a representation

$$\bar{\rho_d} : Gal(\bar{\mathbb{Q}}/\mathbb{Q}) \to GL_2(\mathbb{Z}/d).$$

If N is the conductor of E, this representation is unramified outside dN. Moreover, if $p \nmid dN$, the characteristic polynomial of $\bar{\rho_d}$ (mod d) is

$$T^2 - (a_p \pmod{d})T + (p \pmod{d})T.$$

Using this information, and assuming the GRH, one can use the Chebotarev density theorem (in the effective form proved by Lagarias and Odlyzko [4]) to deduce that for some $\delta(d)$,

$$\pi_E(x,d) = \left(\frac{1}{\delta(d)}\right) \operatorname{Li} x + O(d^3 x^{\frac{1}{2}} \log dNx).$$

For $d = l$ prime and for l large, $1/\delta(l)$ is the density of elements g in $GL_2(\mathbb{Z}/l)$ with the property that $tr\, g \equiv \det g + 1 \mod l$. Hence

$$\delta(l) = l + O(1).$$

Moreover, δ is a multiplicative function.

3 Selberg's Sieve Method

We follow the notation of Bombieri [1]. The general set up is as follows: Let f be a non zero multiplicative function $f : \mathbb{N} \to \mathbb{Z}$ from the natural numbers to the ring of integers. Let

$$P = \{p : \quad p \le x\}$$

be the set of primes up to x. For an integer d, let us set

$$P_d = \{p \in P : \quad f(p) \equiv 0 \mod d\}.$$

Suppose that

$$|P_d| = \frac{1}{\delta(d)}|P| + R_d,$$

where δ is a multiplicative function and R_d is the remainder. Set

$$\delta_1 = \delta * \mu$$

where $*$ denotes Dirichlet convolution. Let λ_ν be a sequence of real numbers with $\lambda_\nu = 0$ for ν sufficiently large or if ν is not squarefree. Set

$$\zeta_r = \mu(r)\delta_1(r) \sum_\nu \frac{\lambda_{\nu r}}{\delta(\nu r)}.$$

Consider the quantity

$$S = \sum_{p \in P} \left(\sum_{d|f(p)} \alpha_d \right) \left(\sum_{\nu|f(p)} \lambda_\nu \right)^2 . \tag{1}$$

Then, by [1], Théorème 18, we have

$$S = |P|\mathfrak{G} + \mathrm{O}\left(\sum_m \Big(\sum_{d|m} |\alpha_d| \Big) \Big(\sum_{\nu|m} |\lambda_\nu| \Big)^2 |R_m| \right) \tag{2}$$

where

$$\mathfrak{S} = \sum_{m}\sum_{\substack{d \\ (m,d)=1}} \frac{\mu^2(m)}{\delta_1(m)}\frac{\alpha_d}{\delta(d)}\Big(\sum_{r|d}\mu(r)\zeta_{rm}\Big)^2.$$

We take for f the function defined by $f(p) = N_p$. Thus,

$$R_d \ll d^3x^{1/2}\log dNx. \tag{3}$$

Let $y, z > 1$ be parameters to be chosen later. We will choose the α_d to be bounded and $\alpha_d = 0$ if $d > y$. We will choose the λ_ν so that $|\lambda_\nu| < 1$, and $\lambda_\nu = 0$ if ν is not squarefree or if $\nu > z$.

By using (3), we see that the error term in (2) is

$$\sum_{m\le yz^2} \mathrm{d}(m)^3m^3x^{1/2}(\log mNx),$$

where $\mathrm{d}(m)$ denotes the number of positive divisors of m. This sum is easily seen to be

$$\ll (yz^2)^4x^{1/2+\varepsilon}(\log xN)$$
$$\ll x^{1-\varepsilon},$$

provided that $(yz^2) \ll x^{1/8-\varepsilon}$.

Thus (2) becomes

$$S = |P|\mathfrak{S} + O(x^{1-\varepsilon}). \tag{4}$$

Now we apply this with two choices of the $\{\alpha_d\}$ and $\{\zeta_r\}$. Suppose that

$$\alpha_d = \begin{cases} 1 & \text{if } d = 1, \\ 0 & \text{if } d > 1, \end{cases}$$

and

$$\zeta_r = \begin{cases} \zeta_1 & \text{if } r < z \text{ is squarefree,} \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\mathfrak{S} = \sum_{m<z}\frac{\mu^2(m)}{\delta_1(m)}\zeta_1^2.$$

Hence,

$$\sum_{p\le x}\Bigg(\sum_{\substack{\nu|N_p \\ \nu<z}}\lambda_\nu\Bigg)^2 = \zeta_1^2\Bigg(\sum_{m<z}\frac{\mu^2(m)}{\delta_1(m)}\Bigg)\mathrm{Li}\,x + \mathrm{O}(x^{1-\varepsilon}).$$

On the other hand, if we choose

$$\alpha_d = \begin{cases} 1 & \text{if } d \text{ is a prime } < y, \\ 0 & \text{otherwise,} \end{cases}$$

and the same choice of $\{\zeta_r\}$ as above, then

$$\mathfrak{S} = \sum_{m<z} \sum_{\substack{l<y \\ l\nmid m}} \frac{\mu^2(m)}{\delta_1(m)} \frac{1}{\delta(l)} \left(\sum_{\substack{r|l \\ r\le z/m}} \mu(r)\zeta_{rm} \right)^2$$

$$= \sum_{m<z} \frac{\mu^2(m)}{\delta_1(m)} \left(\sum_{\substack{\frac{z}{m}<l<y \\ l\nmid m}} \frac{1}{\delta(l)} \right) \zeta_1^2.$$

By easy estimates, we deduce that assuming the GRH,

$$\sum_{p\le x} \left(\sum_{\substack{d|N_p \\ d<y}} \alpha_d \right) \left(\sum_{\substack{v|N_p \\ v<z}} \lambda_v \right)^2 \quad (\mathrm{Li}\ x) \left(1+\log(\frac{\log y}{\log z}) \right) \left(\sum_{m<z} \frac{\mu^2(m)}{\delta_1(m)} \right) \zeta_1^2$$

provided $yz^2 < z^{1/8-}$.

Now we choose

$$y = z^{1/16+}$$
$$z = z^{1/32-}$$

Then

$$\sum_{p\le x} \left(2-\sum_{\substack{d|N_p \\ d<y}} \alpha_d \right) \left(\sum_{\substack{v|N_p \\ v<z}} \lambda_v \right)^2 \quad \zeta_1^2 \left(\sum_{m<z} \frac{\mu^2(m)}{\delta_1(m)} \right) \left(2-1-\log(\frac{\log y}{\log z}) \right) \mathrm{Li}\ x. \tag{5}$$

Now as $x \longrightarrow \infty$,

$$1-\log(\frac{\log y}{\log z}) \longrightarrow 1-\log(2+\ \ ) > 0.$$

Hence, for many primes p, we have

$$2-\sum_{\substack{d|N_p \\ d<y}} \alpha_d > 0. \tag{6}$$

Now, $\sum_{\substack{d|N_p \\ d<y}} \alpha_d$ represents the number of prime divisors of N_p which are less than y. This means that for many primes, N_p has at most one prime $< y$. But as $N_p \quad p$, it has at most 15 prime divisors $> y$. Hence, in all, N_p has at most 16 prime divisors.

How many such primes p have we identified? By Möbius inversion, we have the formula

$$\lambda_v = \mu(v)\delta(v) \sum_r \frac{\mu^2(rv)}{\delta_1(rv)} \zeta_{rv}.$$

With our choice of the ζ_ν, we see that

$$\sum_{\substack{\nu | N_p \\ \nu < z}} \lambda_\nu \quad \sum_{\nu < z} \delta(\nu) \quad \sum_{r < \frac{z}{\nu}} \frac{1}{\delta_1(r\nu)} |\zeta_1|$$
$$\log z \quad \log x.$$

Hence the number of primes $p \le x$ with (6) holding

$$\frac{x}{(\log x)^2}.$$

This proves Theorem 1.

Remark. We can refine this slightly. For any $a < \frac{1}{16} +\ $, choose

$$\alpha_d = \begin{cases} 2 & \text{if } d < x^a \text{ is prime,} \\ 1 & \text{if } x^a < d < y \text{ is prime,} \\ 0 & \text{otherwise.} \end{cases}$$

Then the above argument allows us to conclude that for $\quad \frac{x}{(\log x)^2}$ primes $p \le x$, we have N_p composed of ≤ 16 primes all of which are $> p^a$.

4 The Number of Prime Divisors of N_p

Let $F : \mathbb{R} \to \mathbb{R}$ be a monotone increasing function with $F(x) = O(\log x)$ and $F(x) \ge z$. Let

$$y = y(x) = x^{\frac{1}{F(x)}}.$$

Define as before

$$\pi_E(x, d) = \#\{p \le x : \quad N_p \equiv 0 \mod d\}.$$

Set $\nu_u(n)$ to be the number of distinct prime divisors of n which are less than u. Suppose that

$$\sum_{d \le y} |\pi_E(x, d) - \frac{1}{\delta(d)} Lix| = \mathrm{O}(\pi(x)).$$

Then by a result of [5],

$$\sum_{p \le x} (\nu_u(N_p) - \log\log p)^2 \quad \pi(x)\{\log\log u + F(x)^2\}.$$

In particular, if we assume the GRH, we may choose F bounded. Then, choosing $u = x$, we get

$$\sum_{p \le x} (\nu(N_p) - \log\log p)^2 \quad \pi(x) \log\log x.$$

This means that for any > 0,

$$|\nu(N_p) - \log\log p| < (\log\log p)^{\frac{1}{2}+}$$

holds except possibly for

$$\frac{\pi(x)}{(\log\log x)^2}.$$

primes $p \leq x$.

This proves Theorem 2.

5 Numerical Results

The curves used here are $y^2 + y = x^3 - x$ (Koblitz's curve A [3]) and $y^2 + y = x^3 + x^2$ (Koblitz's curve B [3]) over $\mathbb{Q}$.

In the following tables, $\mathbb{P}$ denotes the number of primes $p \leq x$ for which $|E(\mathbb{F}_p)|$ is prime and $\mathbb{P}_{16}$ denotes the number for which □$|E(\mathbb{F}_p)|$ has at most 16 prime divisors. We compare both numbers with sum = $_{p \leq x} \frac{1}{\log p} \quad \frac{x}{(\log x)^2}$.

Notice that in both cases, the ratio of $\mathbb{P}$/sum is growing. The reason for this is that as shown in Theorem 2, for a general prime p, $|E(\mathbb{F}_p)|$ has $\log\log p$ prime

Curve A

x	$\mathbb{P}$	$\mathbb{P}_{16}$	$\mathbb{P}$/sum	$\mathbb{P}_{16}$/sum
2000	27	295	0.5251573439157927	5.737830239079958
10000	72	1217	0.4515210283966590	7.631959604982417
20000	119	2246	0.4453847927662430	8.508378040511230
30000	177	3225	0.4857878136075451	8.916431258897687
500000	1763	41475	0.5040228786621391	11.85725972348963
1000000	3147	78413	0.5047240217277622	12.57608030369845
50000000	91564	3000724	0.5057166880846511	16.57328429444024
100000000	168513	5760864	0.5053138649370887	17.27489542775297

Curve B

x	$\mathbb{P}$	$\mathbb{P}_{16}$	$\mathbb{P}$/sum	$\mathbb{P}_{16}$/sum
50000	282	5108	0.5195410982580403	9.410694786886773
100000	504	9557	0.5356395779869580	10.15695921988364
1000000	3144	78414	0.5042421014445694	12.57622141942572
5000000	12391	348337	0.5062077219244572	14.23056082898875
70000000	123646	4117486	0.5079079824044065	16.91364060979239
90000000	154100	5216302	0.5071762319513201	17.16797140220723
100000000	168867	5760781	0.5063753774849848	17.27464604382933

divisors. For $p < e^{e^{16}}$, $\log\log p \leq 16$. Note that $e^{e^{16}} > 10^{3,000,000}$. Hence $\mathbb{P}_{16}$ includes almost all primes in the range of our computations!

In the next table, the ratio is calculated with respect to a new sum function for the second curve newsum $= \sum_{p \leq x} \dfrac{\log\log p}{\log p}$.

Curve B

x	$\mathbb{P}_{16}$	$\mathbb{P}_{16}$ / newsum
1500000	114055	5.077703705381576486359745075
4500000	315780	5.342066297428407961234328273
7500000	508055	5.463691349810055780139294703
10000000	664338	5.531694315435302779590602582
15000000	970418	5.627286090928976577998808347
20000000	1270273	5.694755271679060610517382234
50000000	3000632	5.908162470684291146461777003
60000000	3561564	5.950331982556687066367401342
70000000	4117486	5.985957408602426528340496046
80000000	4668771	6.016740549442994361295284568
90000000	5216302	6.043845873790258861760026661
100000000	5760781	6.068092257839014532708832806
105000000	6032009	6.079312197063872087216809876

References

1. E.Bombieri, Les grand crible dans la théorie analytique des nombres, *Asterique*,**18**, Société Math. de France, 2nd ed., 1987
2. S. Galbraith, and J. McKee, the probability that the number of points on an elliptic curve over a finite field is prime, *J. London Math. Soc.*, **62**(2000), 671-684.
3. N. Koblitz, Primality of the number of points on an elliptic curve over a finite field, *Pacific J. Math.*, **131** (1) (1988) 157–165.
4. J. Lagarias and A. Odlyzko, Effective versions of the Chebotarev density theorem, in: Algebraic Number Fields, pp. 409-464, ed. A. Fröhlich, Academic Press, New York, 1977.
5. M. Ram Murty and V. Kumar Murty, Prime divisors of Fourier coefficients of modular forms, *Duke Math. J.*, **51** (1984), 57–76.
6. J. -P. Serre, Propriétés galoisiennes des points d'ordre fini des courbes elliptiques, Invent. Math. **15**(1972), 259-331.

Elliptic Curves of Prime Order over Optimal Extension Fields for Use in Cryptography

Harald Baier

Darmstadt University of Technology, Computer Science Department,
Alexanderstr. 10, 64283 Darmstadt, Germany,
hbaier@cdc.informatik.tu-darmstadt.de

Abstract. We present an algorithm for generating elliptic curves of prime order over Optimal Extension Fields suitable for use in cryptography. The algorithm is based on the theory of Complex Multiplication. Furthermore, we demonstrate the efficiency of the algorithm in practice by giving practical running times. In addition, we present statistics on the number of cryptographically strong elliptic curves of prime order for Optimal Extension Fields of cardinality $(2^{32} + c)^5$ with $c < 0$. We conclude that there are sufficiently many curves in this case.

Keywords: complex multiplication, cryptography, elliptic curve, Optimal Extension Field

1 Introduction

Since their proposal for use in cryptography about 15 years ago ([Kob87], [Mil86]), elliptic curve cryptography has gained a lot of attention in the cryptographic community due to their short key lengths. However, as of today, only two families of finite fields have found consideration in practice: Finite fields of characteristic 2 and finite prime fields of large characteristic. Algorithms to find elliptic curves for use in cryptography are well known for both families of fields.

Recently, a new type of finite fields was proposed for use in practice: Optimal Extension Fields ([BP98], [BP01]). Optimal Extension Fields consider the hardware in use (i.e. the word size of the processor) and thus yield an efficient way of implementing finite field arithmetic, especially the inversion. As the inversion is the most time-consuming step for adding points on elliptic curves over finite fields, Optimal Extension Fields have the potential to be considered as a third family of finite fields for elliptic curve cryptography.

In order to decide whether an elliptic curve is suitable for use in cryptography, we have to know its group order. However, when choosing random curves and using the efficient point counting algorithms, we have to choose a couple of curves before finding a suitable one. This turns out to be rather slow. Hence, we make

supported by FairPay, a project funded by the German Department of Trade and Industry

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 99–107, 2001.

use of the Complex Multiplication Theory to find suitable elliptic curves over Optimal Extension Fields. Due to security reasons we restrict to elliptic curves of prime order. We will develop a closed algorithm solving this task.

Processors of word size 32-bit play a crucial role in practice. Hence, we will show that our algorithm is very fast in this case. Let p be a 32-bit prime with $2^{32} - p < 2^{16}$. Our algorithm finds a cryptographically strong elliptic curve of prime order over an Optimal Extension Field $\mathbb{F}_{p^5}$ in about 22 seconds using an ordinary PC. In addition, we present data on the number of suitable elliptic curves over Optimal Extension Fields of the form $\mathbb{F}_{p^5}$. We conclude that, for fields of this form, their quantity is sufficiently large.

The paper is organized as follows: In the next section we review the basic definitions of Optimal Extension Fields and elliptic curves suitable for use in cryptography. We present our generating algorithm in Sect. 3. Finally, in Sect. 4 we present sample running times of our implementation and discuss statistics on the number of elliptic curves of prime order over fields of the form $\mathbb{F}_{p^5}$ with a 32-bit prime p.

2 Elliptic Curves over Optimal Extension Fields

We review the definition and some properties of Optimal Extension Fields and elliptic curves. Furthermore, we list the conditions on elliptic curves suitable for use in cryptography. Let us first turn to Optimal Extension Fields.

Definition 1. *Let c be a rational integer, and let $p = 2^n + c$ be prime with $n \in \mathbb{N}$. Furthermore, assume $|c| \le \sqrt{2^n}$, and let $m \in \mathbb{N}$. If there is a $\omega \in \mathbb{F}_p$ such that the binomial $X^m - \omega$ is irreducible in $\mathbb{F}_p[X]$, then $\mathbb{F}_{p^m}$ is called an* Optimal Extension Field.

The basic idea of introducing Optimal Extension Fields is to adapt the arithmetic over finite extension fields to the hardware in use (see [BP98], [BP01]). For instance, when implementing an elliptic curve cryptosystem on a 32-bit processor, one may choose $n = 32$ and $c < 0$ such that $2^{32} + c$ is prime. Hence, the arithmetic in $\mathbb{F}_p$ fits in a word size. Furthermore, let ω be as in definition 1. We represent $\mathbb{F}_{p^m}$ as the factor ring $\mathbb{F}_p[X]/(X^m - \omega)$ with respect to the polynomial basis $\{1, X, X^2, \dots, X^{m-1}\}$. Hence, in $\mathbb{F}_{p^m}$ the identity $X^m = \omega$ holds, yielding an easy reduction of X^k for $k \ge m$.

Bailey and Paar [BP01] distinguish two special types of Optimal Extension Fields: First, if $|c| = 1$, the according Optimal Extension Field is called a *Type I* OEF. Second, if $X^m - 2$ is irreducible in $\mathbb{F}_p[X]$, they name the according field *Type II* OEF. In this paper we do not make use of Type I OEFs.

In order to decide whether an irreducible binomial of degree m exists in $\mathbb{F}_p[X]$ we make use of the following theorem, which we prove in [Bai01b].

Theorem 1. *Let p and m be rational primes. For $\omega \in \mathbb{F}_p^\times$ the following properties are equivalent:*

1. *The binomial $X^m - \omega$ is irreducible in $\mathbb{F}_p[X]$.*

2. *m divides the order e of ω in $\mathbb{F}_p^\times$, but not $\frac{p-1}{e}$.*
3. *We have $m \mid p-1$ and $\omega^{\frac{p-1}{m}} \equiv 1 \bmod p$.*

Using the property that $\mathbb{F}_p^\times$ is a cyclic group, the following corollary is an easy consequence of property 3 in theorem 1.

Corollary 1. *Let p and m be primes. There exists an irreducible binomial of degree m in $\mathbb{F}_p[X]$ if and only if $m \mid p-1$.*

Next, we review a few basic facts concerning elliptic curves over finite fields and define cryptographically strong ones. Let p be a prime number, $p > 3$, and let $q = p^m$ with m $\mathbb{N}$. An *elliptic curve* over the field $\mathbb{F}_q$ is a pair $E = (a, b)$ $\mathbb{F}_q^2$ with $4a^3 + 27b^2 = 0$. A *point* on E is a solution (x, y) $\mathbb{F}_q^2$ of $y^2 = x^3 + ax + b$ or the point at infinity O obtained by considering the projective closure of this equation. The set of points on E over $\mathbb{F}_q$ is denoted by $E(\mathbb{F}_q)$. It turns out that $E(\mathbb{F}_q)$ carries a group structure with the point at infinity acting as the identity element.

We call the elliptic curve E *cryptographically strong* if it satisfies the following conditions: We have $|E(\mathbb{F}_q)| = k \cdot r$ with a prime $r > 2^{159}$ and a positive integer $k \leq 4$. The first requirement avoids generic attacks as the ρ-algorithm of Pollard, while the second one is due to efficiency reasons. If $m \geq 2$ and $p \geq 11$ this condition implies that E is not defined over $\mathbb{F}_p$. In addition, in order to avoid anomalous curves, the primes r and p are different. Finally, the order of q in the multiplicative group $\mathbb{F}_r^\times$ is at least $\frac{2000}{\log_2(q)}$; hence, we exclude curves which are amenable to the attack of Menezes, Okamoto, and Vanstone. An explanation of either attack may be found in [BSS99].

In addition, the German Information Security Agency (GISA) requires the class number of the maximal order containing the endomorphism ring of E to be at least 200. Although there is no consensus on this requirement in the community, we take it into account for the following two reasons: First, in order to provide curves for digital signatures being in conformance with the German Digital Signature Act, we have to respect the requirements of the GISA. Second, we want to show that our algorithm is *not* restricted to discriminants of small class numbers. However, our algorithm is applicable to the case of small class numbers either.

In this paper we focus on Optimal Extension Fields of the form p^5 with a 32-bit prime p. The reason for the choice $m = 5$ is twofold. Due to a theorem of Hasse we have $|E(\mathbb{F}_q)| \approx q$. Hence, in order to generate an elliptic curve of prime order r with $r \approx 2^{160}$ we have to ensure $m \geq 5$. Second, we restrict to extension fields of prime degree as some of our sub-algorithms of section 3 are very efficient in this case. However, the security implications of the Weil-descent ([GHS01]) on these curves are not yet clear. Nevertheless, the generalization to composite m is easy.

We are not aware of any further efficient algorithm to find an elliptic curve over an Optimal Extension Field of characteristic ≥ 5 respecting all these requirements. Although the Schoof-Elkies-Atkin (SEA) algorithm is polynomial time for arbitrary finite fields and efficiently implemented for Optimal Extension Fields, it turns out to be much slower in practice. The main reason is that we have to choose a number of curves and determine their cardinalities before finding a suitable one. Furthermore, the very efficient Satoh-algorithm for fields of characteristic 2 ([FGH01]) does not apply to Optimal Extension Fields.

3 The Generating Algorithm

Our generating algorithm `oefCurve`, presented at the end of this section, makes use of the theory of Complex Multiplication. A good reference of this theory in the scope of elliptic curve cryptography may be found in [AM93], [LZ94], and [BB00]. We sketch the most important theory used in our algorithm. A central term is that of an *imaginary quadratic discriminant*, which is a negative integer Δ congruent 0 or 1 modulo 4. Our aim is to find a prime power p^m and an elliptic curve defined over a field $\mathbb{F}_{p^m}$, but not over $\mathbb{F}_p$. In order to do this we first have to find a prime power p^m and a discriminant Δ, such that the norm equation

$$t^2 - \Delta y^2 = 4p^m \tag{1}$$

has a solution $(t, y) \in \mathbb{Z}^2$, while the equation $t^2 - \Delta y^2 = 4p$ does not have a solution $(t, y) \in \mathbb{Z}$. If this is true, using Complex Multiplication, we find elliptic curves $E_{1,q}$ and $E_{2,q}$ over $\mathbb{F}_{p^m}$, both not defined over $\mathbb{F}_p$, with

$$|E_{1,q}(\mathbb{F}_{p^m})| = p^m + 1 - t, \quad |E_{2,q}(\mathbb{F}_{p^m})| = p^m + 1 + t \tag{2}$$

analogously as explained in [BB00].

Let $H \in \mathbb{Z}[X]$ be the the minimal polynomial of $j(\frac{\Delta + \sqrt{\Delta}}{2})$ where j denotes the well-known modular function j. Modulo p the polynomial H splits into irreducible factors of degree m, while it splits in $\mathbb{F}_{p^m}[X]$ into pairwise distinct linear factors. Let $j_q \in \mathbb{F}_{p^m}$ be a zero of H mod p. If $\Delta < -4$, we have $j_q \notin \{0; 1728\}$, and for any non-square $s_q \in \mathbb{F}_{p^m}$ we set

$$\kappa_q = \frac{j_q}{1728 - j_q}, \quad (a_q, b_q) = (3\kappa_q, 2\kappa_q) . \tag{3}$$

Then we have

$$\{E_{1,q}, E_{2,q}\} = \{(a_q, b_q), (a_q s_q^2, b_q s_q^3)\}. \tag{4}$$

After this construction it is not known which of the curves is $E_{1,q}$ and which is $E_{2,q}$. However, by choosing points on each curve and testing whether their order is a divisor of $p^m + 1 + t$ or $p^m + 1 - t$, the curves $E_{1,q}$ and $E_{2,q}$ can be identified.

Thus we can decide whether one of the curves $E_{1,q}$ or $E_{2,q}$ is cryptographically strong before we actually construct those curves. We only need to know the primes p and m and the norm representation of p^m as in (1). From (2) we deduce the orders of $E_{1,q}$ and $E_{2,q}$, and we can check whether one of the curves respects *all* conditions from the previous section.

Input of our algorithm `oefCurve`(n, m, h_0) is a positive integer n (e.g. the word size of the processor in use), the degree m of the Optimal Extension Field over its prime field, and an integer $h_0 \geq 200$. The algorithm returns a prime $p < 2^n$ such that $\mathbb{F}_{p^m}$ is an Optimal Extension Field, an irreducible binomial $X^m - \omega$ in $\mathbb{F}_p[X]$, and an elliptic curve E of prime order r defined over $\mathbb{F}_{p^m}$ respecting all requirements of Sect. 2. Furthermore, the endomorphism ring of E is a maximal order of class number at least h_0. In addition, `oefCurve` returns a generating point of $E(\mathbb{F}_{p^m})$. In order to get reasonable results we have to ensure $n \cdot m \geq 160$ and that m is prime.

We next explain our main algorithm `oefCurve`. It splits into several sub-algorithms, which we discuss in what follows. The first sub-algorithm `findField`(n, m, h_0) determines an Optimal Extension Field of cardinality p^m and a prime r being the group order of a cryptographically strong elliptic curve defined over $\mathbb{F}_{p^m} \setminus \mathbb{F}_p$. To be more precise, `findField` computes among other things a prime p of the form $2^n + c$ with $c < 0$ and $|c| < \sqrt{2^n}$ such that $m \mid p - 1$. Although it is not clear if such a prime p exists for a random tuple (n, m), the asymptotic density of such primes for growing n is $\frac{1}{(m-1)\cdot\log(2^n)}$ due to the Prime Number Theorem and a theorem of Dirichlet on the number of primes in arithmetic progressions. Hence, for example, if $n = 32$ and $m = 5$ (i.e. the case we are most interested in), there should be about $\frac{2^{16}}{4\log(2^{32})} = 739$ primes congruent 1 modulo 5 in the interval $[2^{32} - 2^{16}, 2^{32}]$. However, the exact number is 733. Thus we may assume, that an appropriate prime p exists.

In order to be successful, `findField` has to solve the norm equation (1) for some Δ and p. We explain how to find appropriate Δ and p. A necessary condition on Δ for E to be of prime order is $\Delta \equiv 5 \bmod 8$. We assume that a sufficiently large database of fundamental imaginary quadratic discriminants $\Delta \equiv 5 \bmod 8$ of class number at least 200 is to our disposal. In our tests we make use of a database containing all such fundamental discriminants $\Delta > -6000000$. Our function `nextDiscriminant`(h, Δ) returns the maximal fundamental discriminant $\Delta \equiv 5 \bmod 8$ of class number h with $\Delta < \Delta$.

The algorithm is exponential in $\log(h)$ In addition, it depends on the bitlength of Δ. Thus we want h and $|\Delta|$ to be as small as possible. A necessary condition, due to class field theory, we have to take care of is $m \mid h(\Delta)$. Hence we set $h = \min\{h \quad \mathbb{N} : h \geq h_0, m \mid h\}$. Let $\Delta \equiv 5 \bmod 8$ be maximal of class number h. We set $p = \max\{p \quad \mathbb{Z} : p < 2^n, p \equiv 1 \bmod m, p \text{ prime}\}$. We determine whether the norm equation $t^2 - \Delta y^2 = 4p$ has a solution $(t, y) \quad \mathbb{Z}^2$ by using an algorithm due to Cornacchia ([Coh95], p.34-36): `cornacchia`(Δ, p) gets an imaginary quadratic discriminant Δ and a prime p as input and returns $t = 0$ if the according norm equation has an integer solution, and 0 otherwise. If $t^2 - \Delta y^2 = 4p$ has no integer solution, we turn to the norm equation $t^2 - \Delta y^2 = 4p^m$. In order to decide whether this equation has an integer solution or not, we extended the algorithm of Cornacchia to prime powers: `cornacchiaPrimePower`(Δ, p^m) gets an imaginary quadratic discriminant Δ and a prime power p^m as input. It returns $t = 0$ if the norm equation (1) has an integer solution, and 0 otherwise.

If we have found a prime p with an integer solution of the norm equation for p^m, but not for p, we make use of (2) to check for the conditions of section 2. Analogously to [BB00] this task is performed by the function `isStrong`(p^m, N); it returns the prime r if N turns out to be the order of a cryptographically strong elliptic curve over $\mathbb{F}_{p^m}$, and 0 otherwise. This yields our algorithm `findField`(n, m, h_0).

```
findField(n, m, h_0)
Input: A positive integer $n$, a prime $m$, such that $nm \geq 160$, and an integer $h_0 \geq 200$.
Output: A prime $p$ of bit-length $n$, such that $\mathbb{F}_{p^m}$ is an Optimal Extension Field, if
  such a $p$ exists.
  A prime $r$ and a discriminant $\Delta$, such that $r$ is the cardinality of a cryptographically
  strong elliptic curve defined over $\mathbb{F}_{p^m} \setminus \mathbb{F}_p$ having a maximal order of discriminant
  $\Delta$ as endomorphism ring with $h(\Delta) \geq h_0$.

  $p \leftarrow \max\{p' \ \mathbb{Z} : p < 2^n, p' \equiv 1 \bmod m, p' \text{ prime}\}$;
  if $2^n - p > \sqrt{2^n}$ then
    output("No OEF found. Terminating."); terminate;
  $h \leftarrow \min\{h' \ \mathbb{N} : h' \geq h_0, m \mid h'\}$;
  while true do
    $\Delta \leftarrow$ nextDiscriminant$(h, 0)$;
    while $\Delta > -6000000$ do
      $p \leftarrow \max\{p' \ \mathbb{N} : p' < 2^n, p' \equiv 1 \bmod m, p' \text{ prime}\}$;
      while $2^n - p < \sqrt{2^n}$ do
        $t \leftarrow$ cornacchia$(\Delta, p)$;
        if $t = 0$ then
          $t \leftarrow$ cornacchiaPrimePower$(\Delta, p^m)$;
          if $t = 0$ then
            if $(r \leftarrow$ isStrong$(p^m, p^m + 1 - t)) = 0$ AND $r = p^m + 1 - t$ then
              return$(p, r, \Delta)$;
            else if $(r \leftarrow$ isStrong$(p^m, p^m + 1 + t)) = 0$ AND $r = p^m + 1 + t$ then
              return$(p, r, \Delta)$;
        $p \leftarrow \max\{p' \ \mathbb{Z} : p' < p, p' \equiv 1 \bmod m, p' \text{ prime}\}$;
      $\Delta \leftarrow$ nextDiscriminant$(h, \Delta)$;
    $h \leftarrow h + m$;
```

Once knowing the cardinality p^m of an Optimal Extension Field, we turn to the computation of an irreducible binomial $X^m - \omega$ in $\mathbb{F}_p[X]$. Our algorithm `findBinomial`(p, m) is a straightforward consequence of theorem 1 and corollary 1.

We remark that if $X^m - \omega$ is reducible in $\mathbb{F}_p[X]$, $X^m - \omega^d$ is reducible either for all $d \ \mathbb{N}$. However, due to the simplicity of algorithm `findBinomial`(p, m) we do not take this fact into account.

Finally, we turn to algorithm `findOEFCurve`(Δ, p, r). This algorithm bases on `findCurve`(Δ, p, l) in [BB00]. The main differences come from the sub-algorithm `findRoot`. As explained above, given a root j_q of H mod p in $\mathbb{F}_{p^m}$, `findOEFCurve`

```
findBinomial($p, m$)
Input: Rational primes $p$ and $m$ with $p \equiv 1 \bmod m$.
Output: An irreducible binomial $X^m - \omega$ in $\mathbb{F}_p[X]$ with minimal $\omega$ $\mathbb{N}$.

  $\omega \leftarrow 2$;
  while true do
    $d \leftarrow \omega^{\frac{p-1}{m}} \bmod p$;
    if $d = 1$ then
      return($X^m - \omega$);
    $\omega \leftarrow \omega + 1$;
```

```
findOEFCurve($\Delta, p, r$)
Input: A fundamental imaginary quadratic discriminant $\Delta \equiv 5 \bmod 8$.
  A prime power $p^m$ such that there exists an elliptic curve of prime order $r$ over $\mathbb{F}_{p^m}$.
Output: An elliptic curve $E$ over $\mathbb{F}_{p^m}$ with $|E(\mathbb{F}_{p^m})| = r$ and endomorphism ring of
  discriminant $\Delta$.
  A generating point $G$ of $E(\mathbb{F}_{p^m})$.

  $j_q \leftarrow$ findRoot($\Delta, p^m$);
  Select a non-square $s_q$ $\mathbb{F}_{p^m}$;
  $E_1 \leftarrow (a_q, b_q)$; $E_2 \leftarrow (a_q s_q^2, b_q s_q^3)$; //assign curve parameters
  $G_1$ $_R$ $(E_1(\mathbb{F}_{p^m})) \setminus \{O\}$; $G_2$ $_R$ $(E_2(\mathbb{F}_{p^m})) \setminus \{O\}$; //choose random points
  if $rG_1 = O$ AND $rG_2 = O$ then
    return $(E_1, G_1)$;
  else
    return $(E_2, G_2)$;
```

computes the coefficients of elliptic curves over $\mathbb{F}_{p^m}$ of order $p^m + 1 \pm t$, and it decides by trial and error, which of these curves is of order r.

We next discuss findRoot(Δ, p^m), i.e. the proceeding to determine a root of $H \bmod p$ in $\mathbb{F}_{p^m}$. The first step of findRoot(Δ, p^m) consists in determining a generating polynomial of the Hilbert class field of $\mathbb{Q}(\sqrt{\Delta})$. In the literature one finds some proposals of polynomials with rather small coefficients. If $3 \nmid \Delta$ we compute a polynomial due to Atkin and Morain (see [AM93]). To be more precise, in this case we determine the minimal polynomial of $e^{2\pi i/3} \cdot \gamma_2(\frac{\Delta + \sqrt{\Delta}}{2})$ over $\mathbb{Q}(\sqrt{\Delta})$, where γ_2 is the unique cube root of j which is real-valued on the imaginary axis. We denote this polynomial by P_γ. Let $\gamma_{2,q}$ $\mathbb{F}_{p^m}$ be a root of $P_\gamma \bmod p$. Then $\gamma_{2,q}^3$ is a root of $H \bmod p$. If we have $3 \mid \Delta$, we compute the Hilbert polynomial H. For an efficient computation of P_γ or H we refer to [Bai01a].

It remains to explain how to get a root of a polynomial $P \bmod p$ that splits completely to linear factors in $\mathbb{F}_{p^m}[X]$. As in [BB00] we make use of the LiDIA-function find_root(p^m, P). As input this function requires a prime power p^m and a polynomial P $\mathbb{Z}[X]$, such that $P \bmod p$ splits into linear factors in $\mathbb{F}_{p^m}[X]$.

It returns a zero of $P \bmod p$ in $\mathbb{F}_{p^m}$. find_root uses the Cantor-Zassenhaus split (see [Coh95]) and a polynomial arithmetic due to Shoup [Sho95].

We finally present the main algorithm oefCurve. Given n, m, and h_0, oefCurve first invokes findField(n, m, h_0). Once p, r, and Δ are determined, it calls the functions findBinomial and findOEFCurve. Finally, oefCurve returns (p, r, f, E, G).

4 Running Times and Statistics

We implemented our algorithms in C++ using the library LiDIA 2.0 and the GNU compiler 2.95.2 setting the optimization flag -O2 and using gmp 2.0.2 as underlying multiprecision package. The timings were measured on a Pentium III running Linux 2.2.14 at 850 MHz and having 128 MB of main memory. Hence the timings may be measured on any modern personal computer either. We present some sample running times of oefCurve$(32, 5, h_0)$ and CPU-timings for $200 \leq h_0 \leq 250$, $10 \mid h_0$, in table 1. More timings and statistical data may be found in [Bai01b].

Table 1. Data delivered by oefCurve$(32, 5, h_0)$.

h_0	h	Δ	p	ω	CPU-time in seconds
200	200	-125579	4294920991	2	21.8
210	210	-265235	4294903891	7	52.8
220	220	-268931	4294931761	2	65.3
230	230	-405803	4294931071	2	64.0
240	240	-170651	4294946191	2	38.5
250	250	-254579	4294940641	3	54.6

Finally, we give some statistical data on the number of non-isomorphic elliptic curves of prime order over Optimal Extension Fields $\mathbb{F}_{p^5}$ where p is a 32-bit prime. First, we determine for each class number h with $200 \leq h \leq 400$, h divisible by 5, the number of pairs (Δ, p), where $\Delta > -6000000$ is a fundamental discriminant congruent $5 \bmod 8$ and p a 32-bit prime, such that there exists a cryptographically strong elliptic curve of prime order r over $\mathbb{F}_{p^5}$ having an endomorphism ring of discriminant Δ. In all, there are 5579 such tuples. Furthermore, in 4563 of the cases, the according field $\mathbb{F}_{p^5}$ is a Type II OEF. Next, we determine the number of non-isomorphic elliptic curves for the tuples (Δ, p) as above. For each such tuple (Δ, p) there are $h(\Delta)$ non-isomorphic elliptic curves having the properties cited above. In all, there are 1546830 non-isomorphic curves, and 1263850 of them are defined over a Type II OEF. We deduce that, even in our special case, the set of non-isomorphic curves for use in cryptography is sufficiently large to choose from.

References

[AM93] A.O.L. Atkin and F. Morain. Elliptic curves and primality proving. *Mathematics of Computation*, 61:29–67, 1993.

[Bai01a] H. Baier. Efficient Computation of Singular Moduli with Application in Cryptography. In *Fundamentals of Computing Theory, Proceedings of FCT 2001*, LNCS 2138, pages 71–82, Berlin, 2001. Springer-Verlag.

[Bai01b] H. Baier. Elliptic Curves of Prime Order over Optimal Extension Fields for Use in Cryptography. Technical Report, Darmstadt University of Technology, 2001. Technical Report No. TI-11/01.

[BB00] H. Baier and J. Buchmann. Efficient Construction of Cryptographically Strong Elliptic Curves. In *Progress in Cryptology - INDOCRYPT2000*, LNCS 1977, pages 191–202, Berlin, 2000. Springer-Verlag.

[BP98] D. Bailey and C. Paar. Optimal Extension Fields for fast Arithmetic in Public-Key Algorithms. In *Advances in Cryptology - CRYPTO'98*, LNCS 1462, pages 472–485, Berlin, 1998. Springer-Verlag.

[BP01] D. Bailey and C. Paar. Efficient Arithmetic in Finite Field Extensions with Application in Elliptic Curve Cryptography. *Journal of Cryptology*, 2001. to appear.

[BSS99] I. Blake, G. Seroussi, and N. Smart. Elliptic Curves in Cryptography. Cambridge University Press, 1999.

[Coh95] H. Cohen. A Course in Computational Algebraic Number Theory. Springer-Verlag, 1995.

[FGH01] M. Fouquet, P. Gaudry, and R. Harley. Finding Secure Curves with the Satoh-FGH Algorithm and an Eary-Abort Strategy. In *Proceedings of Eurocrypt 2001*, LNCS 2045, pages 14–29, Berlin, 2001. Springer-Verlag.

[GHS01] P. Gaudry, F. Hess, and N.P. Smart. Constructive and descructive facets of Weil descent on elliptic curves. *Journal of Cryptology*, 2001. to appear.

[Kob87] N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.

[LZ94] G.-J. Lay and H.G. Zimmer. Constructing elliptic curves with given group order over large finite fields. In *Proceedings of ANTS I*, LNCS 877, pages 250–263, 1994.

[Mil86] V. Miller. Use of Elliptic Curves in Cryptography. In *Proceedings of CRYPTO '85*, LNCS 218, pages 417–426, Berlin, 1986. Springer-Verlag.

[Sho95] V. Shoup. A new polynomial factorization algorithm and its implementation. *Journal of Symbolic Computation*, 20:363–397, 1995.

A Secure Family of Composite Finite Fields Suitable for Fast Implementation of Elliptic Curve Cryptography

Extended Abstract

Mathieu Ciet , Jean-Jacques Quisquater, and Francesco Sica

UCL Crypto Group,
Place du Levant 3, B-1348 Louvain-la-Neuve, Belgium,
{ciet,quisquater,sica}@dice.ucl.ac.be

Abstract. In 1999 Silverman [21] introduced a family of binary finite fields which are composite extensions of $\mathbb{F}_2$ and on which arithmetic operations can be performed more quickly than on prime extensions of $\mathbb{F}_2$ of the same size.

We present here a fast approach to elliptic curve cryptography using a distinguished subset of the set of Silverman fields $\mathbb{F}_{2^N} = \mathbb{F}_{h^n}$. This approach leads to a theoretical computation speedup over fields of the same size, using a standard point of view (cf. [7]). We also analyse their security against prime extension fields $\mathbb{F}_{2^p}$, where p is prime, following the method of Menezes and Qu [12]. We conclude that our fields do not present any significant weakness towards the solution of the elliptic curve discrete logarithm problem and that often the Weil descent of Galbraith-Gaudry-Hess-Smart (GGHS) does not offer a better attack on elliptic curves defined over $\mathbb{F}_{2^N}$ than on those defined over $\mathbb{F}_{2^p}$, with a prime p of the same size as N.

A noteworthy example is provided by $\mathbb{F}_{2^{226}}$: a generic elliptic curve $Y^2 + XY = X^3 + \alpha X^2 + \beta$ defined over $\mathbb{F}_{2^{226}}$ is as prone to the GGHS Weil descent attack as a generic curve defined on the NIST field $\mathbb{F}_{2^{233}}$.

Keywords. Finite fields, Weil descent, elliptic curve cryptography, fast performance.

1 Introduction

Elliptic curve cryptography was introduced in 1986 independently by Koblitz [10] and Miller [14] as a rich context where one can apply cryptographic protocols based on the discrete logarithm problem in a multiplicative group G: given a, b G such that $b = a^d$, find d.

However, the rich structure of elliptic curves made possible a wide variety of attacks that must be avoided in the design of elliptic curve cryptosystems such

The work described in this paper has been supported by the European Commission through the IST Programme under Contract IST-1999-12324.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 108–116, 2001.

as ECIES or ECDSA. Some of these attacks rely on a peculiarity of a curve or a family of curves, such as supersingular elliptic curves [13], or elliptic curves of trace one [22,18,19].

Nevertheless, in general, such elliptic curves can be easily avoided, except in the case when the field $\mathbb{F}_{h^n}$ is intrinsically weak, and this may happen when $h = 2^l$ with $l \geq 4$ [23]. Indeed Galbraith-Gaudry-Hess-Smart devised a practical implementation of the Weil descent to compute discrete logarithms in $\mathbb{F}_{2^N}$ for composite N's. This result seemed to preclude the use of composite binary fields for elliptic curve cryptography.

On the other hand, Silverman [21] (and independently in 1989 Ito-Tsujii [9]) proved that basic field operations can be implemented very quickly on certain composite binary extensions, namely extensions $\mathbb{F}_{2^{p-1}}$, with prime p such that 2 is a primitive root modulo p, which we will call Silverman fields.

The goal of the present article is to resuscitate elliptic curve cryptography over the Silverman fields $\mathbb{F}_{2^{p-1}}$. The idea is to choose *Sophie Germain* primes (*SG*-primes) q so that $\mathbb{F}_{2^{2q}} = \mathbb{F}_{2^{p-1}}$. In this way we will be able to keep a good performance record since we are working with Silverman fields while at the same time ensuring an excellent security against the Galbraith-Gaudry-Hess-Smart attack, since $\mathbb{F}_{2^{2q}}$ is a “quasi-prime extension” ($l = 2$).

2 Definitions, Setup and Performance

In this section we describe the working representations of the binary fields $\mathbb{F}_{2^n}$ as well as of the ring R_p which is used to speed up computations in $\mathbb{F}_{2^{p-1}}$.

Let n be a positive integer. The field $\mathbb{F}_{2^n}$ is generally regarded as a quotient $\mathbb{F}_2[X]/(P(X))$ where $P(X)$ is an irreducible polynomial of degree n over $\mathbb{F}_2$. Each element α of $\mathbb{F}_{2^n}$ is viewed as a polynomial $\sum_{i=0}^{n-1} \alpha_i X^i$ modulo $P(X)$ and denoted $(\alpha_0, \ldots, \alpha_{n-1})$. In the case of NIST fields, one chooses n to be prime and $P(X)$ to be a trinomial or a pentanomial in order to minimise field operation cost on machines.

Let p be a prime. We denote $\Phi_p(X) := X^{p-1} + X^{p-2} + \ldots + X^2 + X + 1 \pmod 2$. It is well known that $\Phi_p(X)$ is an irreducible polynomial over $\mathbb{F}_2[X]$ if and only if 2 is a primitive root modulo p. This condition is equivalent to $2^{(p-1)/l} = 1 \pmod p$ for every prime l dividing $p - 1$. A prime p such that 2 is a primitive root modulo p is called a *primitive prime* (to the base 2).

Examples of primitive primes include 101, 107, 131, 139 etc. There is a famous conjecture by E. Artin that there are infinitely many such primes, and that they have a natural density. However, neither of those two assertions has been proved yet, although Hooley [8] deduced the Artin conjecture from the generalised Riemann hypothesis.

Following [21] we introduce $R_p = \mathbb{F}_2[X]/(X^p + 1)$. In the sequel we will suppose that the prime p is primitive. If this is the case then

$$R_p = \mathbb{F}_{2^{p-1}} \times \mathbb{F}_2.$$

We can pass from R_p to $\mathbb{F}_{2^{p-1}}$ in both directions very easily and this canonical projection is very fast.

Here is the list of primitive primes $100 < p < 1200$ such that $q = (p-1)/2$ is also prime: 107, 179, 227, 1019, 1187. Their number is quite sparse but from probabilistic methods one conjectures that the number of such primes less than x is $\approx x/\log^2 x$,[1] hence we may find them as large as needed.

Let us consider the performance of the relevant arithmetic operations, used on elliptic curve cryptosystem, over the field $\mathbb{F}_{2^N}$, where, following Silverman, we have denoted $N = p - 1$. Algorithms for each basic operation are available in [21].

Considering $\mathbb{F}_{2^p}$ as a $\mathbb{F}_2$-space vector of dimension p, we define the *trinomial basis* (resp. *pentanomial basis*) to be the canonical basis of $\mathbb{F}_{2^p}$ under the isomorphism $\mathbb{F}_{2^p} = \mathbb{F}_2[X]/P(X)$, with $P(X)$ irreducible trinomial (resp. pentanomial) in $\mathbb{F}_2$ of degree p.

Addition in R_p is a very straightforward operation taking up as much time as in $\mathbb{F}_{2^p}$, since we have to XOR words with similar size.

Squaring of R_p elements proceeds by defining two other elements which are XORed to produce the output. The squaring operation is related to reordering the α_i's and is as efficient as if using optimal normal bases [3].

Multiplication of two elements of R_p is twice as efficient as optimal normal basis multiplication or Montgomery multiplication. In the particular case of trinomial or pentanomial basis, this achievement is less significant.

Modular inversions are somewhat simple using a modified Almost Inverse Algorithm (AIA). For more details and implementations see [7,21].

Finally the speedup comes only from the underlying field arithmetic and not from a specific curve, like Koblitz curves [11], or specific computation techniques [24]. All generic exponentiation methods [6], like for example the NAF method [15], can be used in this particular type of extension.

A notable feature of our analysis is the conjunct use of primitive primes p and of SG-primes $q = (p-1)/2$. The former property is necessary to insure a good performance, while the latter leads to the claimed security of the field $\mathbb{F}_{2^{2q}} = \mathbb{F}_{2^N} = \mathbb{F}_{2^{p-1}}$.

We therefore analyse the security of the Silverman fields $\mathbb{F}_{2^N}$ for elliptic curve cryptography.

3 On the Security of the Elliptic Curve Discrete Logarithm over $\mathbb{F}_{2^{2q}}$

In this section we will use the following notation: we let n, l be two positive integers, $h = 2^l$, $K = \mathbb{F}_{h^n}$ and $k = \mathbb{F}_h$.

[1] If f and g are non-negative functions, we write $f(x) \approx g(x)$ if there exist c_1, c_2 positive numbers such that $c_1 g(x) \le f(x) \le c_2 g(x)$. We are not interested in formulating a precise asymptotic formula here, only a lower bound (the upper bound is classical).

3.1 Generic Attacks on Elliptic and Hyperelliptic Curves

When using elliptic curve cryptography, one must prevail against known attacks on the elliptic curve discrete logarithm problem. For an elliptic curve E defined over $\mathbb{F}_{2^N}$ these are:

1. Pollard's ρ algorithm [17], which has a running time of $O(2^{N/2}t_o)$, where t_o is the time to perform an addition of two points on the curve,
2. the baby step-giant step algorithm, due to Shanks [20], which runs in $O(N2^{N/2}t_o)$.

To solve the discrete logarithm problem on the Jacobian of a hyperelliptic curve H of genus g defined over k, one resorts to five methods:

1. Pollard's ρ [17], which has a running time of $O(g^2h^{g/2}\log^2 h)$ bit operations, since $\operatorname{card}\operatorname{Jac}(H) \approx h^g$ and a group operation on $\operatorname{Jac}(H)$ takes $O(g^2\log^2 q)$ bit operations using Cantor's algorithm [1],
2. the baby step-giant step algorithm, due to Shanks [20], which runs in $O(g^3h^{g/2}\log^3 h)$,
3. the Pohlig-Hellman algorithm [16], which is not better than Pollard's ρ if $\operatorname{card}\operatorname{Jac}(H)$ has a large prime factor, which is the case by the Gaudry-Hess-Smart construction of $\operatorname{Jac}(H)$,
4. the Enge-Gaudry subexponential [2] algorithm with estimated running time $O\left(\exp((\sqrt{2}+o(1))\sqrt{g\log h(\log g+\log\log h)})\right)$ as $g/\log h$ goes to infinity; this method is not applicable when h^g is too large, say around 2^{1024}, hence when $g \geq 2^{10}$.
5. Gaudry's algorithm [4], a variation of the classical index-calculus algorithm, with running time $O(g^3h^2\log^2 h + g^2g!\,h\log^2 h)$. Actually when g is fixed, Gaudry's algorithm runs in $O(h^2)$ which is better than Pollard's ρ when $g > 4$. However this method is impractical when $g \geq 31$ (using a modified version due to Enge-Gaudry).

The recent work of Gaudry, Hess and Smart [5] (GHS) shows how, for a large proportion of elliptic curves E defined over a binary field $\mathbb{F}$, the discrete logarithm problem on a subgroup of $E(\mathbb{F})$ can be transposed to the same problem on a subgroup of the Jacobian of an hyperelliptic curve. Gaudry's algorithm [4] then manages to solve this equivalent discrete logarithm in a substantially quicker time than the standard methods of Pollard or Shanks.

3.2 Description of the GHS Implementation of the Weil Descent

We give here an account of the Weil descent method of Gaudry, Hess and Smart. Let E/K be an elliptic curve. A theorem of Weil says that one can define an abelian variety A/k (defined over the smaller field) such that canonically $A(k) = E(K)$. In our case $A = E \times E^\sigma \times \cdots \times E^{\sigma^{n-1}}$, where σ is the Frobenius automorphism of k.

In practice, one starts from a Weierstraß equation over K, say

$$Y^2 + XY = X^3 + \alpha X^2 + \beta, \quad \beta = 0. \qquad (1)$$

Given a k-basis $\{\psi_0, \psi_1, \ldots, \psi_{n-1}\}$ of K, we express

$$\begin{cases} \alpha &= a_0\psi_0 + a_1\psi_1 + \cdots + a_{n-1}\psi_{n-1}, \\ \beta &= b_0\psi_0 + b_1\psi_1 + \cdots + b_{n-1}\psi_{n-1}, \\ X &= x_0\psi_0 + x_1\psi_1 + \cdots + x_{n-1}\psi_{n-1}, \\ Y &= y_0\psi_0 + y_1\psi_1 + \cdots + y_{n-1}\psi_{n-1}. \end{cases}$$

Substituting the latter equations into the former defining the elliptic curve and equating coefficients of the ψ_i's, we have defined an abelian variety A over k, obtained by Weil descent from E. Note that $|\mathrm{card}\, A(k) - h^n - 1| \leq 2h^{n/2}$ by the Hasse bound.

Let G be a point of E of large prime order (say about h^n) and G the cyclic group generated by G. Let $P = dG$ for some unknown d $[1, \; - 1]$. The problem of the discrete logarithm in G E consists in finding d, knowing P, G and of course E.

Although the statement of the discrete logarithm problem involves only the cyclic structure of G , the solution to this problem often depends on a suitable embedding of the group into a richer algebraic structure. Also, since $A(k) = E(K)$, we deduce that G can be embedded into an irreducible subvariety B of A.

It happens that under some hypothesis, it is possible to explicitly find an hyperelliptic curve H A of genus g such that its Jacobian has an irreducible component isogenous to B. One can also give a formula for g, namely $g = 2^{m-1}$ or $2^{m-1} - 1$, where $1 \leq m \leq n$ is the $\mathbb{F}_2$-dimension of some vector space (see below).

To put it otherwise, there exists an explicitly computable homomorphism $E(K) \to \mathrm{Jac}(H)$ such that its kernel does not contain G . Hence the problem of solving the discrete logarithm in G E is translated into finding the same d as above with respect to a subgroup isomorphic to G sitting inside $\mathrm{Jac}(H)$. Since there exists a fast (in $O(h^{2+})$) algorithm, due to Gaudry, to find discrete logarithms there, the problem is noticeably simplified.

A consequence is that such elliptic curves as those we started with should be avoided for cryptographical purposes. In general, this reasoning has brought the conclusion that elliptic curves defined over composite extension fields of $\mathbb{F}_2$ should be eschewed by cryptographers. However, specific curves, such as Koblitz curves (defined over $\mathbb{F}_2$), currently thwart this kind of attack.

On the other hand it should be noticed that the current approach to the Weil descent breaks down if $n < 4$, since in this case the Pollard ρ method solves the discrete logarithm problem on $E(K)$ in $O(h^{n/2}) = O(h^{3/2})$, that is faster than through the aforementioned approach.

Similarly Menezes and Qu [12] proved that the fields $\mathbb{F}_{2^p}$ are immune to the GHS version of the Weil descent attack. Our goal is next to extend their approach to establish the security of the fields $\mathbb{F}_{2^{2q}}$, when q is prime.

3.3 The Menezes and Qu Analysis

Suppose that the elliptic curve is given in Weierstraß form as in (1). Let $\sqrt{\ }$ denote the inverse of Frobenius in $\mathbb{F}_2$. The definition of the number m in the genus formula above is given by

$$m(\beta) = \dim_{\mathbb{F}_2}\left(\mathrm{Span}_{\mathbb{F}_2}\left\{(1, \sqrt{\beta_0}), \ldots, (1, \sqrt{\beta_{n-1}})\right\}\right),$$

where $\beta_i = \beta^{h^i}$ is the i-th power of the Frobenius automorphism σ (over k).

Menezes and Qu define another value, $\bar{m}(\beta)$, closely related to $m(\beta)$, by the formula

$$\bar{m}(\beta) = \dim_{\mathbb{F}_2}\left(\mathrm{Span}_{\mathbb{F}_2}\left\{\sqrt{\beta_0}, \ldots, \sqrt{\beta_{n-1}}\right\}\right).$$

To see how the two values are related, let $n = 2^e n_1$, where n_1 is odd, and let $t = 2^e$. The polynomial $x^n + 1$ factors in $\mathbb{F}_2[x]$ as $(f_0 f_1 \cdots f_s)^t$, where $f_0 = x + 1$ and the f_i's are distinct irreducible polynomials in $\mathbb{F}_2[x]$ with $\deg f_i = d_i$.

We view K as a $\mathbb{F}_2$-vector space and σ as a $\mathbb{F}_2$-endomorphism of K. The unique polynomial f of least degree in $\mathbb{F}_2[x]$ such that $f(\sigma) = 0$ in $\mathrm{End}(K)$ is $x^n + 1$. In particular K is the null space of $\sigma^n + 1$.

The idea of Menezes and Qu is to decompose the field K into a direct sum of subspaces corresponding to the null spaces of the factors $f_i^t(\sigma)$. One has

$$K = \bigoplus_{i=0}^{s} W_i,$$

where $W_i = \ker f_i^t(\sigma)$.

Let $\gamma \in K$. By what precedes, we can write uniquely $\gamma = \sum_{i=0}^{s} \gamma_i$, where $\gamma_i \in W_i$. For $0 \le i \le s$, define

$$j_i = j_i(\gamma) = \min\left\{ j \ge 0 \colon \gamma_i \in \ker f_i^j(\sigma) \right\}.$$

We define the *type* of γ to be $(j_0, \ldots, j_s)$.

The relation between $\bar{m}(\beta)$ and $m(\beta)$ appears as Theorem 6 in [12]. It states

Theorem 1 (Menezes and Qu). *Let $\beta \in K = \mathbb{F}_{h^n}$. Then*

$$m(\beta) = \begin{cases} \bar{m}(\beta), & \text{if } j_0(\sqrt{\beta}) = 0, \\ \bar{m}(\beta) + 1, & \text{if } j_0(\sqrt{\beta}) = 0. \end{cases}$$

Furthermore, Menezes and Qu give a complete description of the values taken on by $\bar{m}(\beta)$ when $\beta \in K$. They also give the number of elements of K with given value $\bar{m}$. Their result appears as Theorem 5, which we recall here.

Theorem 2 (Menezes and Qu). *Let $\gamma \in K = \mathbb{F}_{h^n}$. Then the admissible values for $\bar{m}(\gamma)$ are $\sum_{i=0}^{s} j_i d_i$ where each $j_i \in [0, t]$. Moreover, there are pre-*

cisely $\prod_{i=0, j_i=0}^{s} \left(h^{j_i d_i} - h^{(j_i-1)d_i} \right)$ *elements* $\sqrt{\bar{\gamma}}$ K *of type* $(j_0, \ldots, j_s)$ *with* $\bar{m}(\gamma) = \sum_{i=0}^{s} j_i d_i$.

3.4 GHS Weil Descent from the Fields $\mathbb{F}_{2^{2q}}$

We are interested in this paper in fathoming in the fashion of Menezes and Qu the security of the field $K = \mathbb{F}_{h^n}$, where $h = 2^l$ and $n = q$ is prime, as we descend to the subfield $k = \mathbb{F}_h$. In particular we will consider the case where $l = 2$, thus producing highly secure fields $\mathbb{F}_{2^{2q}}$, since they are “quasi-prime extensions” of $\mathbb{F}_2$. We follow the reasoning of [12].

Since card Jac$(H) = h^g + O(h^{g/2})$ and the success of Gaudry’s algorithm depends clearly on the magnitude of Jac(H), Menezes and Qu observe that currently the GHS approach to the Weil descent is ineffective whenever $h^g \geq 2^{1024}$. When $h = 2$, this imposes a lower security bound such as $m \geq 11$. Also in the case when $m = 1$, the GHS method is ineffective, since the curve obtained by Weil descent is elliptic (this is the case of Koblitz curves).

From the Menezes and Qu analysis, more specifically from Theorem 2 one immediately deduces that the admissible values of $\bar{m}$ (and hence m) in the Weil descent do not depend on l, that is on the degree of $\mathbb{F}_h$ over $\mathbb{F}_2$, but only on n, the degree of $\mathbb{F}_{h^n}$ over $\mathbb{F}_h$. Notice that $m(\beta) = 1$ if and only if β $\mathbb{F}_h$.

More specifically, if $h = 4$, then the field $\mathbb{F}_{2^{2q}}$ contains $\mathbb{F}_{2^q}, \mathbb{F}_4$ and $\mathbb{F}_2$ as proper subfields.

Going down from $\mathbb{F}_{2^{2q}}$ to $\mathbb{F}_4$ by the previous observation and the experimental results of [12] we deduce that the admissible values (greater than 1) of m are greater than 16 when q $[100, 600]$ and $q = 127$. Hence the same holds a fortiori when we descend to $\mathbb{F}_2$, since the relative m does not decrease (cf. the definition of m).

As for the descent from $\mathbb{F}_{2^{2q}}$ to $\mathbb{F}_{2^q}$, the degree of the descent is 2 and the hyperelliptic curve found by the method of GHS has genus at most two. For such curves, it is well known that Pollard rho is still more efficient than Gaudry’s algorithm to compute discrete logarithms, hence the GHS attack fails for all elliptic curves over $\mathbb{F}_{2^{2q}}$ with q $[100, 600]$ and $q = 127$ and in general we can affirm that, when considering Weil descent via GHS, the security of the field $\mathbb{F}_{2^{2q}}$ for elliptic curve cryptography is at least as strong as the security of the field $\mathbb{F}_{2^q}$.

4 Conclusion

We have produced a sequence of fields $\mathbb{F}_{2^N}$, such as

$$\mathbb{F}_{2^{178}},\ \mathbb{F}_{2^{226}},\ \mathbb{F}_{2^{1018}},\ \mathbb{F}_{2^{1186}},$$

which are secure for elliptic curve cryptography. Indeed the GHS Weil descent attack on elliptic curves defined over these fields produces hyperelliptic curves of genus at least $2^{m-1} - 1$, where $m \geq 12, 29, 509, 149$ respectively. Therefore the elliptic discrete logarithm problem on these curves is currently out of reach of known attacks. As an example the field $\mathbb{F}_{2^{226}}$ offers the same order of security against the GHS attack as the NIST field $\mathbb{F}_{2^{233}}$ where the corresponding lower bound on m is 30.

Moreover the performance of basic field operations in the fields $\mathbb{F}_{2^N}$ is faster than in the fields $\mathbb{F}_{2^{N+1}} = \mathbb{F}_{2^p}$.

Acknowledgments

We would like to thank anonymous referees for their helpful and detailed remarks.

References

1. D.G. Cantor. Computing in the Jacobian of a Hyperelliptic Curve. *Mathematics of Computation*, 48(177):95–101, 1987.
2. A. Enge and P. Gaudry. A General Framework for Subexponential Discrete Logarithm Algorithms. In *LIX/RR/00/04-Laboratoire d'Informatique-Ecole Polytechnique-Palaiseau, to appear in Acta Arithmetica*, Available at http://www.math.uni-augsburg.de/~enge/Publikationen.html, June 2000.
3. S. Gao and H.W. Lenstra JR. Optimal Normal Bases. *Designs, Codes and Cryptography*, 2:315–323, 1992.
4. P. Gaudry. An Algorithm for Solving the Discrete Logarithm Problem on Hyperelliptic Curves. In Springer-Verlag, editor, *Advances in Cryptography - EUROCRYPT '2000*, LNCS, 2000.
5. P. Gaudry, F. Hess, and N.P. Smart. Constructive and Destructive Facets of Weil Descent on Elliptic Curves. *Journal of Cryptology*, to appear.
6. D. M. Gordon. A Survey of Fast Exponentiation Methods. *Journal of Algorithms*, 27(1):129–146, 1998.
7. D. Hankerson, J. L. Hernandez, and A. Menezes. Software Implementation of Elliptic Curve Cryptography over Binary Fields. *Proceedings of CHES2000*, pages 1–24, 2000.
8. C. Hooley. On Artin's Conjecture. *J. Reine Angew. Math.*, 225:209–220, 1967.
9. B. Ito and S. Tsujii. Structure of a Parallel Multiplier for a Class of Fields GF(2^n). *Information and Compuers*, 83:21–40, 1989.
10. K. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.
11. N. Koblitz. CM-curves with good cryptographic properties. In Joan Feigenbaum, editor, *Advances in Cryptology - Crypto '91*, pages 279–287, Berlin, 1991. Springer-Verlag. Lecture Notes in Computer Science Volume 576.
12. A. Menezes and M. Qu. Analysis of the Weil Descent Attack of Gaudry, Hess and Smart. In *Proceedings RSA 2001*, 2001.
13. A.J. Menezes, T. Okamoto, and S. Vanstone. Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. *IEEE Transactions on Information Theory*, 39:1639–1646, 1993.
14. V. Miller. Use of Elliptic Curves in Cryptography. In Springer-Verlag, editor, *Advances in Cryptology, CRYPTO86*, volume 263 of *LNCS*, pages 417–426, 1986.

15. F. Morain and J. Olivos. Speeding up the Computations on an Elliptic Curve using Addition-Subtraction Chains. *Inform. Theor. Appl.*, 24:531–543, 1990.
16. S. Pohlig and M. Hellman. An Improved Algorithm for Computing Logarithms over GF(p) and its Cryptographic Significants. *IEEE Transactions on Infomation Theory*, 24:106–110, 1978.
17. J. Pollard. Monte Carlo Methods for Index Computation $\pmod p$. *Mathematics of Computation*, 32:918–924, 1978.
18. T. Satoh and K. Araki. Fermat Quotients and the Polynomial Time Discrete Log Algorithm for Anomalous Elliptic Curves. *Commentarii Math. Univ. St. Pauli*, 47:81–92, 1998.
19. I.A. Semaev. Evaluation of Discrete Logarithms in a Group of p-torsion Points of an Elliptic Curve in Characteristic p. *Mathematics of Computation*, 67:353–356, 1998.
20. D. Shanks. A Theory of Factorization and Genera. *In Proc. Symp. Pure Math.*, 20:415–440, 1971.
21. J. H. Silverman. Fast Multiplication in Finite Fields GF(2^n). *Proceedings CHES '99*, pages 122–134, 1999.
22. N. P. Smart. The Discrete Logarithm Problem on Elliptic Curves of Trace One. *Journal of Cryptology*, 12(3):193–196, 1999.
23. N. P. Smart. How Secure are Elliptic Curves over Composite Extension Fields? *Proceedings EUROCRYPT 2001*, 2045:30–39, 2001.
24. J. A. Solinas. An Improved Algorithm for Arithmetic on a Family of Elliptic Curves. In Burton S. Kaliski Jr., editor, *Advances in Cryptology, CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science, Springer-Verlag*, pages 357–371, 1997.

Frameproof and IPP Codes

Palash Sarkar and Douglas R. Stinson

Centre for Applied Cryptographic Research,
Department of Combinatorics and Optimization, University of Waterloo,
200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1,
{psarkar,dstinson}@cacr.math.uwaterloo.ca

Abstract. Frameproof and identifying parent property codes and their relationship to hash families and error correcting codes are studied. A upper bound on the size of such codes is presented. A generalization of identifying parent property is introduced and studied in terms of a new class of hash families – the strong separating hash families. Asymptotic consequences of some recursive construction techniques are described.

1 Introduction

Codes providing certain forms of traceability have been studied for their applications to protection of intellectual property rights. The weakest form of such codes provides protection against the possibility of a user being framed by a coalition of users. Such codes are called frameproof codes and were introduced by Boneh and Shaw [5]. A stronger notion introduced in [12] and called secure frameproof code is protection against two disjoint coalition being able to produce a common descendant. The strongest version studied in this paper is called the identifying parent property (IPP) and requires the condition that if a set of coalitions is able to produce a common descendant then the coalitions must have a codeword in common (the identifying parent). This concept was introduced in [8].

The class of codes with traceability property have been studied by several authors. A systematic combinatorial study have been carried out in [10]. In a recent work, Blackburn [2] have provided an upper bound on the cardinality of frameproof codes. Also [2] provides the counterexample of length 5, 3-frameproof code to show that the coefficient of the leading term in the upper bound is not tight. We build on this counterexample to show that in most cases the coefficient of the upper bound is not tight.

A sufficient condition for the existence of IPP codes has been obtained in terms of perfect hash families in [10]. This raised the question of existence of IPP codes for certain values of the parameters. This question was resolved in [4] by the probabilistic method and using a new family of hash functions called the partially hashing family. We introduce a natural generalization of perfect and separating hash family called the strong separating hash family which turns out to be equivalent to the class of partially hashing property. We introduce a generalization of IPP codes and obtain a hierarchy of codes between the frameproof

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 117–126, 2001.

and IPP codes. The existence of these codes are studied in terms of the strong separating hash family and error correcting codes.

Frameproof and IPP codes can be constructed from suitable hash families which in turn can be construted from error correcting codes. We explore this theme in conjunction with a recursive composition type construction used in [8,13]. It turns out that for each size of the alphabet and strength of the code it is possible to construct these classes of codes having a fixed relation between the length and cardinality.

Due to lack of space, some of the proofs cannot be presented in the paper. These proofs can be found in the technical report [9].

2 Hash Functions

Let H be a family of N functions from $\{1, \dots, n\}$ to the alphabet Q of cardinality q. The family H is called an (N, n, q) hash family. Given an (N, n, q) hash family we can obtain an $N \times n$ matrix by enumerating the values of all the functions. Sometimes it is easier to consider hash families as such matrices.

Definition 1. *We consider the following kinds of hash families.*

1. **Perfect hash family** : *An (N, n, q) hash family H is called an (N, n, q, w) perfect hash family ((N, n, q, w)-PHF) if for any subset $S \subseteq \{1, \dots, N\}$ of cardinality w, there is a function $f \in H$ such that f is injective on S.*
2. **Separating Hash Family** : *An (N, n, q) hash family H is called an (N, n, q, w_1, w_2) separating hash family ((N, n, q, w_1, w_2)-SHF) if for any two disjoint subsets A, B of $\{1, \dots, n\}$ with $|A| = w_1$ and $|B| = w_2$, there is a function f in H such that $f(A)$ and $f(B)$ are also disjoint.*
3. **Strong Separating Hash Family** : *An (N, n, q) hash family H is called an (N, n, q, w_1, w_2) strong separating hash family ((N, n, q, w_1, w_2)-SSHF) if for any two disjoint subsets $A, B \subseteq \{1, \dots, n\}$ with $|A| = w_1$ and $|B| = w_2$, there is a function $f \in H$ such that f is injective on A and $f(A) \cap f(B) = \emptyset$.*
4. **Partially Hashing Family** : *[4] An (N, n, q) hash family H is called an (N, n, q, t, u) partially hashing family ((N, n, q, t, u)-PAHF) if for any two subsets $T, U \subseteq \{1, \dots, n\}$ such that $T \subseteq U$, $|T| = t$, $|U| = u$, there is a function f in H such that for any $x \in T$ and any $y \in U$, if $y \neq x$, we have $f(x) \neq f(y)$.*

See [6] for a survey on perfect hash families. The concept of separating hash families was introduced in [11] and that of partially hashing families was introduced in [4].

We now relate strong separating hash and partially hashing properties to the other hashing properties. The following is immediate from the definition of strong separating hash property.

Theorem 1. *Let H be an (N, n, q) hash family.*

1. *If H is an (N,n,q,w_1,w_2)-SSHF, then it is simultaneously an (N,n,q,w_1,w_2)-SHF and an (N, n, q, w_1)-PHF.*

2. *If H is an $(N, n, q, w_1 + w_2)$-PHF, then it is an (N, n, q, w_1, w_2)-SSHF.*
3. *If H is an (N, n, q, w_1, w_2)-SSHF, then it is an (N, n, q, t_1, t_2)-SSHF for any $t_1 \leq w_1$ and $t_2 \leq w_2$.*

It is not difficult to see that the converse of Theorem 1(1) does not in general hold. We now show the equivalence of strong separating hash and partially hashing property.

Theorem 2. *A hash family H is an (N, n, q, w_1, w_2)-SSHF iff it is an $(N, n, q, w_1, w_1 + w_2)$-PAHF.*

Proof : Suppose H is an (N, n, q, w_1, w_2)-SSHF. We will show that H is an $(N, n, q, w_1, w_1 + w_2)$-PAHF. Let $A \subseteq B \subseteq \{1, \ldots, n\}$, with $|A| = w_1$, $|B| = w_1 + w_2$. Let $C = B - A$. Then $|C| = w_2$ and $C \cap A = \emptyset$. By the strong separating hash property there is an f such that f is injective on A and $f(A) \cap f(C) = \emptyset$. Let $x \in A$ and $y \in B$ with $x \neq y$. There are two cases to consider.

1. $y \in A$. Since f is injective on A, $f(x) \neq f(y)$.
2. $y \in B - A$. Since $C = B - A$ and $f(A) \cap f(C) = \emptyset$, $f(x) \neq f(y)$.

Thus in both cases $f(x) \neq f(y)$ and hence H is an $(N, n, q, w_1, w_1 + w_2)$-PAHF.

Conversely, suppose H is an $(N, n, q, w_1, w_1 + w_2)$-PAHF. We show H is an (N, n, q, w_1, w_2)-SSHF. Let $A, B \subseteq \{1, \ldots, n\}$, with $A \cap B = \emptyset$, $|A| = w_1$, $|B| = w_2$. Let $C = A \cup B$. Then $|C| = w_1 + w_2$ and $A \subseteq C$. Thus by the partially hashing property there is a function f such that if $x \in A$ and $y \in C$ and $x \neq y$, then $f(x) \neq f(y)$. It is not difficult to argue that this f must be injective on A and $f(A) \cap f(B) = \emptyset$. Hence H is an (N, n, q, w_1, w_2)-SSHF.

We present a sufficient condition for the existence of separating hash families based on error correcting codes. An (n, N, q, D) error correcting code is a set $C \subseteq Q^N$ of n codewords where $|Q| = q$ and the distance between any pair of codewords is at least D. We denote by $hd(x, y)$ the Hamming distance between two codewords x, y.

The ideas in the following theorem are present in [7,1,13] and a proof can be found in [9].

Theorem 3. *Let C be an (n, N, q, D) error correcting code with*

$$D > N \left(1 - \frac{1}{\binom{w_1}{2} + w_1 w_2} \right). \tag{1}$$

Then $H(C)$ is an (N, n, q, w_1, w_2)-SSHF.

It is now easy to see that all the construction techniques (direct and recursive) used in [13] for the construction of perfect and separating hash functions can be applied to construct strong separating hash functions. In particular, we get the following result using the recursive construction technique of [13].

Theorem 4. *For positive integers q, w_1 and w_2, there exists an infinite class of (N, n, q, w_1, w_2)-SSHF for which N is $O((w_1(w_1 + w_2))^{\log^* n} \log n)$.*

3 Traceability Codes

Let Q be an alphabet of size q and C be a subset of Q^N of cardinality n. We will call C an (n, N, q) code and the elements of C to be codewords or simply words. It is sometimes convenient to consider $C = (c_{ij})$ to be an $n \times N$ matrix with elements from the set Q. Let $\mathrm{elem}_C(j)$ be the set of all elements that occur in the jth column of C, i.e., $\mathrm{elem}_C(j) = \{a \in Q : a = c_{ij} \text{ for some } 1 \leq i \leq n\}$. When the code is understood we will write $\mathrm{elem}(i)$ instead of $\mathrm{elem}_C(i)$. Define

$$\mathrm{desc}(C) = \mathrm{elem}_C(1) \times \ldots \times \mathrm{elem}_C(N).$$

The set $\mathrm{desc}(C)$ is called the descendant code of C. Clearly $C \subseteq \mathrm{desc}(C)$. We make a few remarks on the connection between C and $\mathrm{desc}(C)$.

The connection of codes and hash families follows from the fact that if C is an (n, N, q) code then C^T can be considered to be the matrix obtained from an (N, n, q) hash family and conversely. Here C^T denotes the transpose of the matrix C. Thus given an (n, N, q) code C, we think of the columns of the matrix C as the functions of the associated hash family. This makes it easier to state many of the results below. Following [10], we will denote by $H(C)$ the hash family obtained from the code C. The proof of the following result can be found in [9].

Lemma 1. *Let $x \in Q^N$ and C be a minimal (n, N, q) code over Q such that $x \in \mathrm{desc}(C) - C$. Then $H(C)$ is an $(N, n, q, n-1, 1)$-SHF. Conversely, if C is an (n, N, q) code such that $H(C)$ is a minimal $(N, n, q, n-1, 1)$-SHF, then there is an $x \in Q^N$ such that $x \in \mathrm{desc}(C) - C$.*

Suppose C is an (n, N, q) code such that $|\mathrm{elem}(j)| = q$ for all $1 \leq j \leq N$. Then $C = \mathrm{desc}(C)$ iff $C = Q^N$. Here we note that if we consider the codewords to be one way infinite strings over Q (i.e., elements of $Q^{\mathbb{N}}$) and C is an infinite subset of $Q^{\mathbb{N}}$, then C is a proper subset of $\mathrm{desc}(C)$, since by diagonalization it is possible to construct a string in $\mathrm{desc}(C)$ which is not in C.

Let C be an (n, N, q) code. We will call a subset D of C to be a coalition and if a codeword x is in $\mathrm{desc}(D)$, then we say that D produces x. We define $\mathrm{desc}_w(C)$, the w-descendant code, in the following manner.

$$\mathrm{desc}_w(C) = \bigcup_{C_0 \subseteq C, |C_0| \leq w} \mathrm{desc}(C_0).$$

The set $\mathrm{desc}_w(C)$ consists of all codewords which could have been produced by subsets of C of size atmost w. We next define the important classes of codes.

Definition 2. *Let C be an (n, N, q) code. We now define certain traceability properties such a code might possess. These are the proerties we will consider in this paper. A stronger notion of traceability exists (see for example [10]).*

1. **Frameproof** : *We say that C is an (n, N, q, w)-FP code if the following condition holds. For any subset C_0 of C of cardinality atmost w, if $x \in \mathrm{desc}(C_0) \cap C$ then $x \in C_0$.*

2. **Secure Frameproof** : *We say that C is (n, N, q, w)-SFP code if the following condition holds. For any two subsets C_0, C_1 of C of cardinality atmost w, we have that* $\mathrm{desc}(C_0) \cap \mathrm{desc} C_1 = \emptyset$ *implies* $C_0 \cap C_1 = \emptyset$.
3. **Identifying Parent Property** : *We say that C is an (n, N, q, w)-IPP code if the following condition holds. Let $\{C_1, \ldots, C_\alpha\}$ be a family of subsets of C where each C_i is of cardinality atmost w. Then we must have that*

$$\bigcap_{1 \leq i \leq \alpha} \mathrm{desc}(C_i) = \emptyset \quad \textit{implies} \quad \bigcap_{1 \leq i \leq \alpha} C_i = \emptyset .$$

The relationships among these classes of codes and their relationships to other combinatorial structures such as PHF, SHF and cover free families have been studied in [10]. Here we briefly mention some of these relations. We use the notation $P_1 \leq P_2$ to denote the fact that property P_2 implies property P_1.

Proposition 1. *[10] The following relationships hold for any (n, N, q) code.*

$$(n, N, q, w)\text{-}FP \leq (n, N, q, w)\text{-}SFP \leq (n, N, q, w)\text{-}IPP.$$

Theorem 5. *[11]*

1. *A code C is an (n, N, q, w)-FP code iff $H(C)$ is an $(N, n, q, w, 1)$-SHF.*
2. *A code C is an (n, N, q, w)-SFP code iff $H(C)$ is an (N, n, q, w, w)-SFP.*

Theorem 6. *[8, Lemma 1] A code C is an $(n, N, q, 2)$-IPP iff $H(C)$ is both an $(N, n, q, 3)$-PHF and an $(N, n, q, 2, 2)$-SHF.*

A characterization of $(n, N, q, 3)$-IPP codes appear in [4].

4 Frameproof Codes

In [10], it was shown that for an (n, N, q, w)-FP code $n \leq w(q^{\frac{N}{w}} - 1)$. This was improved in [2] where it has been shown that

$$n \leq \max\{q^{\frac{N}{w}}, r(q^{\frac{N}{w}} - 1) + (w - r)(q^{\frac{N}{w}} - 1)\}, \tag{2}$$

where r is the unique integer in $\{1, \ldots, w\}$ such that $r \equiv N \bmod w$. Also in [2], an example of an $(n, 5, q, 3)$-FP code was presented where $n \leq \frac{5}{3}q^2 + 3q$. This example shows that the coefficient of the leading term of (2) is not always tight. Here we build on the counterexample of [2] to provide an upper bound which shows that the coefficient of the leading term of (2) is in most cases not tight.

Following [2], we define

$$U_S = \{x \in C : \text{ there exists no } y \in C - \{x\} \text{ such that } x_i = y_i \text{ for all } i \in S\}.$$

The following crucial result has been proved in [2,10].

Lemma 2. *Let C be an (n, N, q, w)-FP code and $S_1, \ldots, S_w$ is a partition of $\{1, \ldots, N\}$. Then*

$$C = U_{S_1} \cup \ldots \cup U_{S_w}.$$

We require some preliminary results to prove the upper bound. These results are generalizations of the ideas present in [2].

In the following discussion we will fix an (n, N, q, w)-FP code C, where we write $N = wt + r$ with r to be the unique integer in the range $\{1, \ldots, w\}$. Further we will only consider codes with r in the range $1 < r < w$. Let $L = \{1, \ldots, N\}$. Let T_1, T_2 be subsets of L satisfying

1. $|T_1| = |T_2| = t + 1$,
2. $T_1 \cap T_2 = \emptyset$,
3. $|U_{T_1} \cap U_{T_2}|$ is the maximum among all pairs of subsets T_1, T_2 satisfying 1 and 2 above.

We put

$$|U_{T_1} \cap U_{T_2}| = kq^{t+1}. \tag{3}$$

Clearly $0 \leq k \leq 1$. The next lemma and its corollary are important in obtaining our upper bound. The proofs can be found in [9].

Lemma 3. *Let S_1, S_2 be subsets of $\{1, \ldots, N\}$ satisfying*

1. *$|S_1| = |S_2| = t+1$,*
2. *$|S_1 \cap S_2| = 1$.*

Then $|U_{S_2} - U_{S_1}| \leq (r-1)kq^{t+1} + (w-r)q^t$.

Corollary 1. *Let T_1, T_2 be subsets of L such that $|T_1| = |T_2| = t+1$ and $T_1 \cap T_2 = \emptyset$. Then $|U_{T_2} - U_{T_1}| \leq 2(r-1)kq^{t+1} + 2(w-r)q^t$.*

Now we are in a position to prove the upper bound.

Theorem 7. *Let C be an (n, N, q, w)-FP code where $N = wt + r$ and r is the unique integer in the set $\{1, \ldots, w\}$ such that $N \equiv r \bmod w$. If $1 < r < w$, then*

$$n \leq \left(r - \frac{\lfloor \frac{r}{2} \rfloor}{1 + 2\lfloor \frac{r}{2} \rfloor (r-1)} \right) q^{t+1} + (w-r)\left(2 + \left\lfloor \frac{r}{2} \right\rfloor \right) q^t.$$

Proof : Let $T_1, \ldots, T_w$ be a partition of $L = \{1, \ldots, N\}$, such that $T_1, \ldots, T_r$ are of cardinalities $(t+1)$ each and $T_{r+1}, \ldots, T_w$ are of cardinalities t each. Further we choose T_1, T_2 such that $|U_{T_1} \cap U_{T_2}| = kq^{t+1}$. Let $r_1 = \lfloor \frac{r}{2} \rfloor$ and assume $U_{T_0} = \emptyset$. We compute $|C|$ as follows.

$$\begin{aligned} |C| &= |U_{T_1} \cup U_{T_2} \cup \ldots \cup U_{T_w}| \\ &\leq (|U_{T_1}| + |U_{T_2}| - |U_{T_1} \cap U_{T_2}|) + |U_{T_3}| + \ldots + |U_{T_w}| \end{aligned}$$

Using (3), we get the following bound on $|C|$.

$$|C| \leq (r - k)q^{t+1} + (w - r)q^t. \tag{4}$$

Computing $|C|$ in a different way we get the following.

$$\begin{aligned}
|C| &= |U_{T_1} \cup U_{T_2} \cup \ldots \cup U_{T_w}| \\
&\leq (|U_{T_1} \cup U_{T_2}|) + \ldots + (|U_{T_{2r_1-1}} \cup U_{T_{2r_1}}|) + |U_{T_{r-2r_1}}| + |U_{T_{r+1}}| + \ldots + |U_{T_w}| \\
&= (|U_{T_1}| + |U_{T_2} - U_{T_1}|) + \ldots + (|U_{T_{2r_1-1}}| + |U_{T_{2r_1}} - U_{T_{2r_1-1}}|) \\
&\quad + |U_{T_{r-2r_1}}| + |U_{T_{r+1}}| + \ldots + |U_{T_w}|
\end{aligned}$$

For $1 \leq i \leq r_1$, we have using Corollary 1,

$$|U_{T_{2i}} - U_{T_{2i-1}}| \leq 2(r-1)kq^{t+1} + (w-r)q^t.$$

We use this to bound $|C|$ in another way as follows.

$$|C| \leq (r - r_1 + 2kr_1(r-1))q^{t+1} + (w-r)(2+r_1)q^t. \tag{5}$$

Combining (4) and (5) and eliminating k we get,

$$|C| \leq \left(r - \frac{r_1}{1 + 2r_1(r-1)}\right) q^{t+1} + (w-r)(2+r_1)q^t.$$

Note : Subsequent to our work, Blackburn [3] has improved upon the upper bound in Theorem 7 by using techniques from extremal set theory based upon the Erdos-Ko-Rado theorem.

We briefly consider the construction problem for frameproof codes. The first construction is simple and doubles both the alphabet size and cardinality of the code while maintaining the parameters w and N constant.

Union Construction : Let C be an (n, N, q, w)-FP code on alphabet $Q = \{a_1, \ldots, a_q\}$. Define $Q' = \{a'_1, \ldots, a'_q\}$. This ensures that $Q \cap Q' = \emptyset$. Let C' be an (n, N, q, w)-FP code on alphabet Q'. Clearly C' can be obtained from C by changing each element $c_{i,j}$ of C to $c'_{i,j}$. Let $D = C \cup C'$. Then D is an $(2n, N, 2q, w)$-FP code on alphabet $Q \cup Q'$.

The correctness of the construction can be found in [9]. This construction immediately gives us the following result.

Theorem 8. *Let C be an (n, N, q, w)-FP code. Then it is possible to construct $(2^i n, N, 2^i q, w)$-FP codes for each integer $i \geq 1$.*

Recursive Construction : From Theorem 5 we know that C is an (n, N, q, w)-FP code iff $H(C)$ is an $(N, n, q, 1, w)$-SHF. A recursive construction for (N, n, q, w_1, w_2)-SHF was presented in [13]. Specializing this construction to the case of $w_1 = 1$ and $w_2 = w$ we get the following result.

Theorem 9. *For any positive integers q and w, there exists an infinite class of (n, N, q, w)-FP codes for which N is $O(w^{\log^* n} \log n)$.*

5 IPP Codes

In this section we study IPP codes. We present a necessary condition for IPP codes and introduce a generalization of IPP codes. Proofs for the next two results can be found in [9].

Theorem 10. *Let C be an (n, N, q, w)-IPP code and $x \in \mathrm{desc}_w(C)$. Then there is a nonempty set of indices $S \subseteq \{1, \ldots, N\}$ and a unique $y \in C$, such that $x_i = y_i$ for all $i \in S$.*

Theorem 11. *Let C be an (n, N, q) code. Suppose for every $x \in \mathrm{desc}_w(C)$, there is a nonempty set $S \subseteq \{1, \ldots, N\}$ of columns and a unique $y \in C$, such that for any $i \in S$ and any codeword $z \in C$, $x_i = z_i$ implies $z = y$. Then C is an (n, N, q, w)-IPP code.*

Corollary 2. *Let C be an (n, N, q) code. Suppose for every $x \in \mathrm{desc}_w(C)$, there is an index i and a $y \in C$, such that for any $z \in C$, $z_i = x_i$ implies $z = y$. Then C is an (n, N, q, w)-IPP code.*

We now introduce a hierarchy of codes between the SFP and the IPP codes.

Definition 3. *Let C be an (n, N, q) code. Suppose for every k coalitions $(1 \leq k \leq t)$ $C_1, \ldots, C_k$ of C, each of size atmost w, $x \in \cap_{i=1}^{i=k} \mathrm{desc}(C_i)$ implies $\cap_{i=1}^{i=k} C_i \neq \emptyset$. Then we call C to be an (n, N, q, w, t)-IPP code.*

Clearly the case $t = 2$ correspond to the SFP codes. The hierarchy relation is made clear in the following result which is immediate from the definition of (n, N, q, w, t)-IPP codes.

Proposition 2. *For an (n, N, q) code the following holds.*

$$\begin{aligned}(n, N, q, w)\text{-}SFP = (n, N, q, w, 2)\text{-}IPP &\leq (n, N, q, w, 3)\text{-}IPP \\ &\leq \ldots \leq (n, N, q, w, \ )\text{-}IPP.\end{aligned}$$

Thus an (n, N, q, w)-IPP code is always an (n, N, q, w, t)-IPP code for any fixed t. Thus the new codes are between the SFP and the IPP codes.

An $(n, N, q, w, \binom{n}{w})$-IPP code is certainly an (n, N, q, w)-IPP code. So a relevant question is whether the hierarchy of codes in Definition 3 is strict from $t = 1$ to $t = \binom{n}{w}$. The next result shows that for $w = 2$, this is not the case. See [9] for a proof.

Proposition 3. *A code C is an $(n, N, q, 2)$-IPP code iff it is an $(n, N, q, 2, 3)$-IPP code.*

A sufficient condition for the existence of (n, N, q, w)-IPP codes can be obtained in terms of perfect hash families [10] and partially hashing families [4]. We provide a sufficient condition for the existence of (n, N, q, w, t)-IPP codes. The idea of the proof is from Theorem 2.8 of [10].

Theorem 12. *Let C be an (n, N, q) code. If $H(C)$ is an $(N, n, q, w, k(w + 2 - k) - w)$-SSHF, where k is such that $k(w + 2 - k) = \max_{2 \le i \le t}\{i(w + 2 - i)\}$. Then C is an (n, N, q, w, t)-IPP code.*

Proof : Let $D = \{C_1, \ldots, C_r\}$ be a minimal set of coalitions such that $x \in \cap_{i=1}^{i=r} \mathrm{desc}(C_i)$ and $\cap_{i=1}^{i=r} C_i = \emptyset$. Thus if we drop any C_i from D, then there is at least one codeword $y_i \in \cap_{j \neq i} C_j$. Let $E = \cup_{i=1}^{i=r} C_i$ and $|E| = \beta$. We now obtain an upper bound on β. Each C_i contains the codewords $y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_r$ and it can contain atmost $w - (r - 1)$ other codewords. Hence

$$\beta \le r + r(w - r + 1) = r(w + 2 - r). \tag{6}$$

We have $H(C)$ to be an $(N, n, q, w, \beta - w)$-SSHF. Let A be a subset of E such that $C_1 \subseteq A$ and $|A| = w$. Clearly such an A exists. Let $B = E - A$. Then $|B| = \beta - w$. By the SSHF property, there is an $f \in H(C)$ (an index $j \in \{1, \ldots, n\}$) such that $|\mathrm{elem}_A(j)| = w$ and $\mathrm{elem}_A(j) \cap \mathrm{elem}_B(j) = \emptyset$. But this implies $|\mathrm{elem}_{C_1}(j)| = |C_1|$ and $\mathrm{elem}_{C_1}(j) \cap \mathrm{elem}_{E-C_1}(j) = \emptyset$. Since $x \in \mathrm{desc}(C_1)$, there is a codeword $z \in C_1$, such that $z_j = x_j$. Since no codeword in E is present in all coalitions in D, let C' be a coalition of D not containing z. But then $x \notin \mathrm{desc}(C')$, since no codeword in C' can equal x on index j. This is a contradiction.

Corollary 3. *Let C be an (n, N, q) code. If $H(C)$ is an $(N, n, q, w, \lfloor(\frac{w+2}{2})^2\rfloor - w)$-SSHF, then C is an (n, N, q, w)-IPP code.*

Proof : If t is not fixed, then the maximum value of β in (6) is $\lfloor(\frac{w+2}{2})^2\rfloor$. The rest of the argument is similar.

Corollary 3 was obtained in [4] in terms of partially hashing property. In [4], the probablistic method based on Corollary 3 was used to show the existence of (n, N, q, w)-IPP codes for all $q \ge w + 1$. This also implies the existence of (n, N, q, w, t)-IPP codes for each t and $q \ge w + 1$.

Theorem 3 can be used to provide a sufficient condition for the existence of IPP codes in terms of error correcting codes. Let C be an (n, N, q, D) error correcting code with

$$D > N\left(1 - \frac{1}{\binom{w}{2} + w(\beta - w)}\right). \tag{7}$$

Using Theorem 3 we have $H(C)$ is an $(N, n, q, w, \beta - w)$-SSHF. ¿From Theorem 12 and Corollary 3 we have the following.

1. If $\beta = k(w + 2 - k)$, where k is such that $k(w + 2 - k) = \max_{2 \le i \le t}\{i(w + 2 - i)\}$, then C is an (n, N, q, w, t)-IPP code.
2. If $\beta = \lfloor\frac{(w+2)^2}{4}\rfloor$, then C is an (n, N, q, w)-IPP code.

Using this and Theorem 4, we get the following result. See [9] for a proof.

Theorem 13. *For any poitive integers q and w, there exists an infinite class of (n, N, q, w)-IPP codes for which N is $O((w^3)^{\log^* n} \log n)$.*

References

1. N. Alon. Explicit construction of exponential sized families of k-independent sets. *Discrete Mathematics*, Volume 58(1986), pp 191-193.
2. S.R. Blackburn. Frameproof codes. *Preprint*, 2000.
3. S.R. Blackburn. Frameproof codes. *Preprint*, 2001.
4. A. Barg, G. Cohen, S. Encheva, G. Kabatiansky and G. Zémor. A hypergraph approach to identifying parent property: the case of multiple parents. DIMACS Technical Report 2000-20.
5. D. Boneh and J. Shaw. Collision-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, Volume 44(1998), pp 1897-1905.
6. Z.J. Czech, G. Havas and B.S. Majewski. Perfect hashing. *Theoretical Computer Science*, Volume 182 (1997), pp 1-143.
7. J. Friedman. Constructing $O(n \log n)$ size monotone formulae for the kth elementary symmetric polynomial of n Boolean variables. *Proceedings of the* 25*th Annual Symposium on Foundations of Computer Science*, (1984), 506-515.
8. H.D.L. Hollman, J.H. van Lint, J-P. Linnartz and L.M.G.M. Tolhuizen. On codes with the identifiable parent property. *Journal of Combinatorial Theory A*, Volume 82 (1998), pp 121-133.
9. P. Sarkar and D. R. Stinson. Frameproof and IPP codes. CACR Technical Report CORR 2001-12, University of Waterloo, http://www.cacr.math.uwaterloo.ca, 2001.
10. J.N. Staddon, D.R. Stinson and R. Wei. Combinatorial properties of frameproof and traceability codes. *CACR Technical Report CORR 2000-16*, University of Waterloo, http://cacr.math.uwaterloo.ca, 2000.
11. D.R. Stinson, T. van Trung and R. Wei. Secure frameproof codes, key distribution patterns, group testing algorithms and related structures. *Journal of Statistical Planning and Inference*, Volume 86 (2000), pp 596-617.
12. D.R. Stinson and R. Wei. Combinatorial properties and constructions of traceability schemes and frameproof codes. *SIAM Journal of Discrete Mathematics*, Volume 11 (1998), pp 41-53.
13. D.R. Stinson, R. Wei and L. Zhu. New constructions for perfect hash families and related structures using combinatorial designs and codes. *Journal of Combinatorial Designs*, Volume 8 (2000), pp 189-200.

Linear Authentication Codes: Bounds and Constructions

Rei Safavi-Naini[1], Huaxiong Wang[1], and Chaoping Xing[2]

[1] School of IT and CS, University of Wollongong, Australia,
rei, huaxiong@uow.edu.au
[2] Department of Mathematics, National University of Singapore, Singapore,
matxcp@nus.edu.sg

Abstract. In this paper, we consider a new class of unconditionally secure authentication codes, called *linear authentication code* (or *linear A-code*). We show that a linear A-code can be characterised by a family of subspaces of a vector space over a finite field. We then derive an upper bound on the size of source space when other parameters of the systems, that is the size of the key space and the authenticator space, and the deception probability, are fixed. We give constructions that are asymptotically close to the bound and show application of these codes in constructing distributed authentication systems.

Key Words: Authentication Codes, Linear Authentication Codes, Distributed Authentication Codes.

1 Introduction

Unconditionally secure authentication codes (A-codes) allow two trusting parties to communicate in the presence of an opponent who may construct a fraudulent message, and/or substitute a transmitted message with a fraudulent one.

The construction of unconditionally secure authentication codes relies on a number of theoretical areas including design theory, finite geometry, coding theory, and information theory. Previous research on authentication theory has been mainly focussed on deriving bounds on parameters of A-codes and construction of codes with desirable properties such as having the minimum possible deception probabilities and the minimum number of keys. In general, to describe the model of A-codes and characterise optimal codes, a combinatorial approaches is used. For example, numerous results are in the form “an A-code with certain properties exists if and only if a certain combinatorial structure exists”.

In this paper, we introduce a new class of authentication codes, called *linear A-codes*. *Linearity* requires some additional algebraic properties for the A-codes; that is, we require both the key space and the authenticator space of the codes be vector spaces, and a source state to induce a linear mapping between them. The main motivation of linear A-codes stems from the study of distributed authentication systems in which the functionality of authentication is to be distributed

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 127–135, 2001.

among a number of participants. The extra algebraic property allows more efficient construction of such distributed systems.

We characterise linear A-codes in terms of a family vector spaces over finite fields such that the dimension of the intersection of a pair of such subspaces does not exceed certain desired value (security parameter). We derive an upper bound on the number of possible source states of an A-code for given deception probabilities and number of keys, and give constructions that meet, or asymptotically meet, the bound.

The paper is organised as follows. In section 2 we give definitions and review known results on A-codes that will be required for the rest of the paper. In section 3, we introduce linear A-codes and give characterisation of A-codes in terms of the families of subspaces of a vector space. Bounds on the number of source states and constructions that asymptotically meet the bounds, are given in sections 4 and 5. We show how linear A-codes can be used to construct distributed authentication schemes in section 6. Finally, we conclude the paper in section 7 and propose new research problems.

2 Authentication Codes

Authentication codes were first considered by Gilbert, MacWilliams and Sloane [9]. The general theory of unconditionally secure authentication systems has been initiated by Simmons ([17,18]) and extended by a number of authors (see, for example, [1,2,3,6,7,10,14,19,20,21,22,23]).

In the conventional model for unconditionally secure authentication system, there are three participants: a *transmitter*, a *receiver* and an *opponent*. The transmitter wants to communicate a message to a receiver using a public channel which is subject to active attacks. That is, the opponent may impersonate the transmitter and insert a message into the channel, or replace a transmitted message with a fraudulent one. To protect against these attacks, the transmitter and the receiver share a secret key which is used to choose an authentication rule from an authentication code (A-code for short).

A *systematic* A-code (or A-code *without secrecy*) is a code in which a *message* sent over the channel, consists of a *source state* (i.e. plaintext) concatenated with an *authenticator* (or a *tag*). Such a code is a triple (S, E, A) of finite sets together with a (authentication) function $f : S \times E \rightarrow A$. We sometimes denote the A-code by (S, E, A, f). Here S is the set of source states, E is the set of keys and A is the set of authenticators. When the transmitter wants to send the source state $s \quad S$ using a key $e \quad E$, which is secretly shared with the receiver, he transmits the message (s, a), where $a = f(s, e) \quad A$. When the receiver receives (s, a), she checks the authenticity of the message by verifying if $a = f(s, e)$ holds, using the secret key $e \quad E$. If the equality holds, she accepts s as authentic.

Suppose the opponent has the ability to insert messages into the channel and/or to modify the existing messages. An *impersonation attack* is when the opponent inserts a new message (s , a) into the channel. A *substitution attack* is when the opponent sees a message (s, a) and changes it to (s , a) where $s = s$.

A message (s, a) is called *valid* if there exists a key e such that $a = f(s, e)$. We assume that there is a probability distribution on the source states, which is known to all the participants. Given the probability distribution on the source states, the receiver and the transmitter will choose a probability distribution for E. We will denote the probability of success of the opponent in impersonation and substitution attack, by P_I and P_S, respectively. In the remainder of the paper, we will always assume that the keys and the source states are uniformly distributed. In this case we can represent P_I and P_S as follows.

$$P_I \ = \max_{s,a} \frac{|\{e \ \ E \mid a = f(s,e)\}|}{|E|},$$

$$P_S = \max_{s,a} \max_{s \ \ = s,a} \frac{|\{e \ \ E \mid a = f(s,e), a \ \ = f(s \ ,e)\}|}{|\{e \ \ E : a = f(s,e)\}|}.$$

One of the goal of authentication theory is to derive bounds on various parameters of A-codes and to construct A-codes with desired properties. For a review of different bounds and constructions for A-codes, refer to [10,21,12].

3 Linear Authentication Codes

Consider an A-code (S, E, A, f). For each key $e \ \ E$, the authentication function $f : S \times E \to A$ induces a mapping ψ_e from S to A defined by $\psi_e(s) = f(s, e), \ \ s \ \ S$. Thus, the A-code (S, E, A, f) can be characterised completely by the family of mappings $\{\psi_e \mid e \ \ E\}$, and vice versa. An attractive family of mappings is obtained from *almost strongly universal hash families*, which was introduced by Wegman and Carter [22] and has been the basis of the most combinatorial constructions. More details on the connection between almost strongly universal hash families and A-codes can be found in [2,22,21].

A source state $s \ \ S$ in an A-code (S, E, A, f) can also be uniquely associated with a mapping φ_s from E to A defined by $\varphi_s(e) = f(s, e), \ \ e \ \ E$. Then, again, the A-code (S, E, A, f) can be characterised by a family of mapping $\Phi = \{\varphi_s \mid s \ \ S\}$. In a conventional authentication system the key space E and the authenticator space A do not have any algebraic structures. We Consider A-codes in which E and A have algebraic structures. In particular, E and A are vector spaces over a finite filed $\mathbb{F}_q$, and Φ is a family of $\mathbb{F}_p$-linear mappings from E to A. These codes are called *linear* A-codes. As will be shown in section 6, linear A-codes are useful in constructing distributed authentication schemes.

Definition 1. *An A-code (S, E, A, f) is linear over $\mathbb{F}_q$ if*

(i) E and A are finite dimensional vector spaces over $\mathbb{F}_q$;
(ii) For all $s \ \ S$, φ_s defined by $\varphi_s(e) = f(s, e)$ is an $\mathbb{F}_q$-linear mapping from E to A.

We identify S with $\Phi = \{\varphi_s \mid s \ \ S\}$, and write the A-code as (Φ, E, A, f) to emphasis that the source states are represented as linear mappings. We may assume that $E = \mathbb{F}_q^n$ and $A = \mathbb{F}_q^m$. Given a basis $e_1, e_2, \ldots, e_n$ of E and a basis

$a_1, a_2, \ldots, a_m$ of A, a linear mapping $\varphi \quad \Phi$, can be represented by a *unique* $n \times m$ matrix A over $\mathbb{F}_q$ such that $\varphi(e) = eA$, $e \quad E$. If V and W are two vector spaces over $\mathbb{F}_q$, and φ is a linear mapping from V to W, we will denote $\mathrm{Ker}(\varphi) = \{v \quad V \mid \varphi(v) = 0\}$. Obviously, $\mathrm{Ker}(\varphi)$ is a subspace of V and its dimension is denoted by $\dim(\mathrm{Ker}(\varphi))$.

Next, we compute the success probabilities of impersonation and substitution attacks for a linear A-code. For the impersonation attack, we have

$$\begin{aligned} P_I &= \max_{\varphi \quad \Phi} \max_{a \quad A} \frac{|\{e \mid \varphi(e) = a\}|}{|E|} \\ &= q^{n-\gamma}, \end{aligned}$$

where $\gamma = \max_{\varphi \quad \Phi}\{\dim(\mathrm{Ker}(\varphi)) \mid \varphi \quad \Phi\}$. Clearly, $\gamma \leq n - m$, and if equality holds then P_I achieves the maximal value. In this case, each φ is onto, i.e., $\varphi_s(E) = A$, $s \quad S$.

For the substitution attach, we have

$$P_S = \max_{\substack{\varphi, \varphi \quad \Phi \\ \varphi = \varphi}} \max_{a, a \quad A} \frac{|\{e \mid \varphi(e) = a\} \cap \{e \mid \varphi\ (e) = a\ \}|}{|\{e \mid \varphi(e) = 0\}|}.$$

It follows that both P_I and P_S must be the reciprocals of a power q. That is $P_I = q^{-t}$ and $P_S = q^{-d}$ for some integers t and d, $t \geq d$, and so performance of a linear A-code over $\mathbb{F}_q$ can be determined by the parameters $|\Phi|, n, m, t$ and d. For a given t and d (which correspond to the security level of the A-code), and n and m (which correspond to the key size and the length of tag), we would like to have $|\Phi|$ as large as possible. Equivalently, for given t, d and $|S|$ (the number of sources), we would like to construct linear A-code with $|\Phi| = |S|$ such that n and m are as small as possible.

Let $V(n, q)$ denote the n dimensional linear space over $\mathbb{F}_q$.

Definition 2. *A linear A-code* (S, E, A) *is called an* $[n, M, t, d]$ *linear A-code if* $|S| = M, |E| = q^n, P_I = 1/q^t$ *and* $P_S = 1/q^d$.

Theorem 1. *There exists an* $[n, M, t, d]$ *linear A-code if and only if there exists a family of subspaces of* $V(n, q)$,

$$L = \{L \mid L \text{ is a subspace of } V(\alpha, q)\}$$

such that

(i) $|L| = M$;
(ii) $\dim(L) = n - t, \quad L \quad L$;
(iii) $\dim(L \cap L\ ) \leq n - (t + d), \quad L, L \quad L, L = L\ $.

4 Bounds on Linear A-codes

In an $[n, M, t, d]$ linear A-code over $\mathbb{F}_q$, given n, t and d we would like to have M as large as possible. In this section, we will derive some upper bounds on M. We denote $M(n, t, d, q)$ the maximal M for which an $[n, M, t, d]$ linear A-code over $\mathbb{F}_q$ exists.

Let $\begin{bmatrix} n \\ k \end{bmatrix}_q$ denotes the *Gaussian coefficient.*[1] Then the number of k-dimensional subspaces of $V(n, q)$ is $\begin{bmatrix} n \\ k \end{bmatrix}_q$, which gives an upper bound for $M(n, t, d, q)$. The following theorem improves the result.

Theorem 2. *In an $[n, M, t, d]$ linear A-code over $\mathbb{F}_q$, we have*

$$M[n, t, d, q] \leq \begin{bmatrix} n \\ n-(t+d)+1 \end{bmatrix}_q \Big/ \begin{bmatrix} n-t \\ n-(t+d)+1 \end{bmatrix}_q .$$

For any fixed n and k, as $q \to \infty$ we have

$$\begin{aligned} \begin{bmatrix} n \\ k \end{bmatrix}_q &= \frac{(q^n-1)(q^{n-1}-1)\dots(q^{n-k+1}-1)}{(q^k-1)(q^{k-1}-1)\dots(q-1)} \\ &\approx q^{(n-k)k}. \end{aligned}$$

It follows that

$$\begin{aligned} M &\leq \begin{bmatrix} n \\ n-(t+d)+1 \end{bmatrix}_q \Big/ \begin{bmatrix} n-t \\ n-(t+d)+1 \end{bmatrix}_q \\ &\approx \frac{q^{(n-(t+d)+1)(t+d-1)}}{q^{(n-(t+d)+1)(d-1)}} \\ &= q^{(n-(t+d)+1)t}. \end{aligned} \tag{1}$$

In the next section we give a construction that meets the asymptotic bound in (1).

It is also worth pointing out that while in the general theory of A-codes, it is possible that the size of source states grow exponentially with the size of key, for example the construction based on universal hash family (see, for example, [22,2,21,23]), because of the structure restriction, this will not be true for linear A-codes. In fact, from Theorem 1 it is easy to see that $\log_q |S| \leq n^2 = (\log_q |E|)^2$, and this bound can be asymptotically achieved. For example, if $(t+d)-1 \approx t \approx t \approx n/2$, then, as we will show in the next section, we will have a linear A-code with $\log_q M(n, t, d, q) \approx n^2/4$.

[1] The Gaussian coefficient is defined as

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \frac{(q^n-1)(q^{n-1}-1)\dots(q^{n-k+1}-1)}{(q^k-1)(q^{k-1}-1)\dots(q-1)}.$$

5 Constructions

Rank distance codes [8] have been used to construct distributed authentication schemes such as A^2-codes by Johansson [11] and group authentication by van Dijk *et al* [7]. Inspired by their work, we show that linear A-codes can be constructed from rank distance codes. It turns out that such constructions result in linear A-codes that asymptotically meet the bound in the previous session.

We first review rank distance codes studied by Gabidulin in [8]. Let $\Lambda = \{A_i\}$ be a set of m by r matrices over $\mathbb{F}_q$. The distance $d(A,B)$ between two matrices A and B in Λ is defined by $d(A,B) = \mathrm{rank}(A-B)$ and the minimum distance of Λ , denoted by $d(\Lambda)$, is defined as $d(\Lambda) = \min_{A,B \quad \Lambda,\ A = B} d(A,B)$. Let $d = d(\Lambda)$ and $M = |\Lambda|$. We call Λ an $(m \times t, M, d)$ *rank distance code*. The following theorem establishes the relation between linear A-codes and rank distance codes.

Theorem 3. *If there exists a* $(m \times t, M, d)$ *rank distance code over* $\mathbb{F}_q$*, then there exists a* $[m+t, M, t, d]$ *linear A-code over* $\mathbb{F}_q$*.*

As shown in [11], in an $(m \times t, M, d)$ rank distance code, we always have $d \leq m - k + 1$, where $k = \log_{q^t} M$. Codes for which the equality holds are called *Maximum-Rank-Distance codes* (or MRD-codes for short). Johansson [11] showed that MRD-codes can be constructed from lineralized polynomials, briefly recalled in the following.

Recall that a polynomial of the form $F(z) = \square_{i=0}^{m} f_i z^{q^i}$, where $f_i \quad \mathbb{F}_{q^t}$ is called a *linearised polynomial* over $\mathbb{F}_{q^t}$. Let k, m, t be integers satisfying $0 < k \leq m \leq t$. By $P_{k,m,t}$, we denote the set of all linearised polynomials of degree at most q^{k-1}. Assume that $g_1, g_2, \ldots, g_m$ are specified elements of the field $\mathbb{F}_{q^r}$ which are linearly independent over $\mathbb{F}_q$. For each $F(z) \quad P_{k,m,t}$, set $c_{F(z)} = (F(g_1), F(g_2), \ldots, F(g_m))^T$.

We associate $c_{F(z)}$ with an $m \times t$ matrix $A(c_{F(z)}) = (a_{ij})$, which is obtained by writing $F(g_i)$ (expressed in a fixed base) as a row vector with entries $a_{ij} \quad \mathbb{F}_q$.

Lemma 1 ([11]). *$\{A(c_{F(z)}) \mid F(z) \quad P_{k,m,t}\}$ is an MRD-code. That is, $\{A(c_{F(z)}) \mid F(z) \quad P_{k,m,t}\}$ is an* $(m \times r, q^{tk}, m-k+1)$ *rank distance code.*

Applying Theorem 3 and Lemma 1, we obtain the following result.

Corollary 1. *Let k, m, t be integers satisfying* $0 < k \leq m \leq t$ *and let q be a prime. The above construction from linearised polynomials results in a* $[t+m, q^{tk}, t, m-k+1]$ *linear A-code.*

Corollary 2. *The parameters given in Corollary 1 asymptotically meet the bounds in Theorem 2.*

6 Applications

Linear A-codes have been implicitly used in constructing distributed authentication schemes, for example, A^2-codes [11], group authentication schemes[7] and one-time fail-stop signatures [16]. With appropriate modification, these constructions can be generalised to *any* linear A-codes. In this section we show how linear A-codes can be used as a building block for constructing group authentication schemes.

Group authentication schemes, also known as *threshold authentication* schemes, were introduced by Desmedt *et al* [6] to generalise conventional authentication codes. In group authentication schemes there are multiple senders and the generation of authenticator requires collaboration of an autheorized subset of senders. In a $(k, \ )$ threshold authentication schemes, there are senders and generation of authenticator for a message requires collaboration of at least k senders. A general method of constructing a threshold authentication system is by combining a $(k, \ )$ secret sharing scheme [15] and an authentication code, by sharing the authentication key among the n senders. It is known that a direct combination will fail to fulfil the security requirement of such systems; and caution must be paid to the authentication operation for the generation of authenticator such that one can not recover the underlying authentication key even if he/she has seen the authenticated message from the autheorized group. To our best knowledge, all the previous constructions use Shamir's secret sharing and some particular examples of linear A-codes ([6,7]). We show that this construction method is generic in the sense that one can always construct group authentication schemes by combining *any* linear A-codes and a (linear) secret sharing scheme.

The construction of a $(k, \ )$ group authentication scheme proceeds as follows. Let (S, E, A, f) be an $[n, M, t, d]$ linear A-code over F_q. Assume that there are n senders $P_1, \ldots, P$ and a receiver R. Assume $q >$ and $x_1, x_2, \ldots, x$ are distinct elements of F_q (x_i is associated to P_i). Let $e_0, e_1, \ldots, e_{k-1}$ be k random values in E. The key of R is e_0 and the key of P_i is

$$\alpha_i = \sum_{j=0}^{t-1} x_i^j e_j. \tag{2}$$

since E is an n-dimensional vector space over F_q, the right-hand side of equation (2) is well defined. Assume that k senders $P_{i_1}, \ldots, P_{i_t}$ want to authenticate a message s S. Each P_{i_j} computes $b_{i_j} = \prod_{u \ B, u = i_j} \frac{-x_u}{(x_{i_j} - x_u)} \cdot \alpha_{i_j}$, and sends $a_{i_j} = f(s, b_{i_j})$ to the receiver R, where $B = \{i_1, \ldots i_k\}$. The receiver computes $a = \sum_{j=1}^{k} a_{i_j}$ and accepts s as authentic if $a = f(s, e_o)$.

The security proof of the above schemes is similar to [7]. Thus, various group authentication codes can be obtained through different choices of the underlying linear A-codes. In general, we can combine a linear A-code and a linear secret sharing scheme to construct a group authentication code. Details will be given in the full version of the paper.

7 Conclusions

Linear A-codes are an interesting class of authentication codes. We have showed that such A-codes can be characterised in terms of families of subspaces of a vector spaces over finite fields. We derived an upper bound on the number of source states of these codes and gave constructions that asymptotically meet the bound. However, the construction that is closed to the asymptotic bound is only when q, the size field, is sufficiently large. An interesting research problem is whether the bound in Theorem 2 can be met for general q, and in particular, when q is small.

A linear A-code $C = (S, E, A, f)$ is defined using vector spaces over finite fields. It is an interesting question that if we relax the algebraic structure to Abelian groups (or modules over rings), can we improve the bound of Theorem 2 or give other non-trivial constructions?

We believe linear A-codes can be used in other distributed systems in which A-codes play a role and so exploring such applications needs further work.

Acknowledgements

Thanks to H. Niederreiter for some useful discussions which led to the proof of Theorem 2, thanks also go to C. Ding for his interests in this work.

References

1. J. Bierbrauer, “Universal hashing and geometric codes”, *Designs, Codes and Cryptography,* Vol.11, pp. 207-221,1997.
2. J. Bierbrauer, T. Johansson, G. Kabatianskii and B. Smeets, “On families of hash functions via geometric codes and concatenation”, *Advances in Cryptology–CRYPTO’93, Lecture Notes in Computer Science*, 773, pp. 331-342, 1994.
3. E. F. Brickell, A few results in message authentication, *Congressus Numerantium,* Vol.43 (1984), 141-154.
4. Y. Desmedt, Society and group oriented cryptology: a new concept, *Advances in Cryptography–CRYPTO ’87*, Lecture Notes in Compute. Sci. 293, 1988, 120-127.
5. Y. Desmedt, Some recent research aspects of threshold cryptography, 1997 *Information Security Workshop, Japan* (JSW ’97), LNCS, 1396 (1998), 99-114.
6. Y. Desmedt, Y. Frankel and M. Yung, Multi-receiver/Multi-sender network security: efficient authenticated multicast/feedback, *IEEE Infocom’92*, 1992, 2045-2054.
7. M. van Dijk, C. Gehrmann and B. Smeets, Unconditionally Secure Group Authentication, *Designs, Codes and Cryptography*, 14 (1998), 281-296.
8. E. M. Gabidulin, Theory of codes with maximum rank distance, *Problems of Information Transmission*, 21(1) (19850, 1-12.
9. E. N. Gilbert, F. J. MacWilliams and N. J. A. Sloane, Codes which detect deception, *The Bell System Technical Journal*, 33 (1974), 405-424.
10. T. Johansson, *Contributions to unconditionally secure authentication*, Ph.D. thesis, Lund University, 1994.
11. T. Johansson, Authentication codes for non-trusting parties obtained from rank metric codes, *Designs, Codes and Cryptography*, 6:205-218, 1995.
12. G. Kabatianskii, B. Smeets, and T. Johansson, “On the cardinality of systematic authentication codes via error correcting”, *IEEE Trans. Inform. Theory*, Vol. 42, pp. 566-578, 1996.

13. F. J. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*, New-York; NorthHolland, 1977.
14. K. Martin and R. Safavi-Naini, Multisender Authentication Schemes with Unconditional Security, *Information and Communications Security*, LNCS, **1334** (1997), 130-143.
15. A. Shamir, How to Share a Secret, *Communications of the ACM*, **22**, 1979, 612-613.
16. R. Safavi-Naini, W. Susilo and H. Wang, Fail-Stop Signature for long messages, *Indocrypt'00,* LNCS, **1977**(2000), 165-177.
17. G. J. Simmons, Authentication theory/coding theory, In *Advances in Cryptology-Crypto '84*, LNCS, **196** (1984), 411-431.
18. G. J. Simmons, A survey of information authentication, in *Contemporary Cryptology, The Science of Information Integrity*, G.J. Simmons, ed., IEEE Press, 1992, 379-419.
19. B. Smeets, P. Vanroose and Zhe-Xian Wan, On the construction of authentication codes with secrecy and codes withstanding spoofing attacks of order $L \geq 2$, In *Advances in Cryptology-Eurocrypt '90*, LNCS, **473** (1990), 306-312.
20. D. R. Stinson, The combinatorics of authentication and secrecy codes, *J. Cryptology*, **2** (1990), 23-49.
21. D. R. Stinson, Universal Hashing and authentication codes, *Designs, Codes and Cryptography* **4** (1994), 369-280.
22. M. N. Wegman and J. L. Carter, "New hash functions and their use in authentication and set equality", *Journal of Computer and System Sciences*, Vol. 22, pp. 265-279, 1981.
23. C. Xing, H. Wang and K. Y. Lam, Constructions of authentication codes from algebraic curves over finite fields, *IEEE Trans.on Info.Theory,* **46**(2000), 886-892.

Selective Receipt in Certified E-mail

Steve Kremer and Olivier Markowitch

Université Libre de Bruxelles,
Bd du Triomphe C.P. 212, 1050 Brussels, Belgium,
{skremer,omarkow}@ulb.ac.be

Abstract. Traditional pen and paper transactions are becoming more and more replaced by equivalent electronic services. Therefore electronic e-mail should also provide enhanced services as those provided by traditional mail. In this paper we present new optimistic protocols for certified e-mail. The major contribution of our paper is the definition of a new property, specific to certified *no author-based selective receipt*. This property requires that once the identity of the author of the mail is known, the receipt can not be refused any more. We present two certified e-mail protocols respecting this property.

1 Introduction

Due to the tremendous growth of open network in general and the Internet in particular, traditional 'pen and paper' transactions are becoming more and more replaced by equivalent electronic services. Probably the best example to illustrate this fact is electronic mail. Millions of e-mails are sent every day over the Internet. However the traditional post offers more sophisticated services, such as express mail, compensation for lost mail or certified mail. Hence, the electronic equivalent should offer similar services. Express mail deals with a fast delivery of mails before a fixed deadline. On networks, fast delivery is part of the nature of electronic mails and is naturally provided, unless at least one link on each route between the sender and the recipient is broken. However, the Internet is designed to resist a high number of failures. In a similar way, compensation of lost mails is almost never necessary when using an electronic item does not physically exist and cannot be lost in the same way as a traditional letter. Although e-mails can be mis-routed or lost due to network failures, the loss is insignificant as multiple copies of an item can easily be created. If the mail has to be delivered before a given deadline, the loss of a message, or a permanent network failure can be a problem. In that case, compensation could be an expected property, but this paper does not focus on this issue. The third service we mentioned is certified mail. This service provides a proof of delivery to the sender. In opposition to express mail and compensation for lost items, the electronic version of certified delivery is more complicated to provide than the traditional one. When we send a traditional certified mail, a person assures the exchange of the mail against a receipt. The mail is only handed to the receiver, after he has signed the receipt. On the other hand the receiver is sure that once he signed the receipt he will

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 136–148, 2001.

receive the mail, as the postman, holding the mail, stands in front of him. In an electronic environment, guaranteeing a certified mail is more complicated. Imagine the following scenario where Alice wants to send a mail to Bob: she sends an e-mail to Bob who is expected to return a receipt. In this scenario, Bob can first read the mail, and then decide whether he wants to admit having received it or not. He can choose not to send the receipt although he read the mail. If we change the scenario such that Bob sends the receipt before having received the message, we face a different problem: Alice could get the receipt and claim that Bob received a message he never saw. In order to overcome these problems we have to use specialized protocols.

Before looking at existing certified e-mail protocols, we first discuss some related topics: fair exchange [1,2,11] and non-repudiation [8,9,14,16] protocols. All of these topics have the same underlying problem: an exchange of secrets in which none of the entities gains a significant advantage over the other one. In a fair exchange protocol an entity wants to exchange one or several items against one or several other items in a fair way, i.e. in a way that at the end of the protocol either both got their expected items or none of them got any valuable information. A certified e-mail protocol can be seen as an instance of a fair exchange protocol. However, in fair exchange protocols the expected item, or a description of it, is generally known a priori, before the exchange takes place. In a certified e-mail protocol, the message that is sent is not known to the receiver. In a non-repudiation protocol, an entity, Alice, sends a message to another entity Bob. At the end of the protocol, Alice is expected to have a non-repudiation of receipt evidence, i.e. an evidence that Bob received the message, and Bob is expected to have a non-repudiation of origin evidence, i.e. an evidence that Alice is the author of the message. Those evidences can, in case of dispute be presented to a judge. The difference between certified e-mail and non-repudiation protocols is not very clear in literature and will be discussed later in more detail.

First solutions to these exchange problem were based on gradual exchange protocols [13]. The disadvantages of this approach are the requirement of equivalent computational power and the network overhead. The second approach is a probabilistic one [4,9]. Generally, the probability to cheat the other entity can be decreased by increasing the number of messages necessary in the protocol. To avoid the communication overhead, a different approach using a trusted third party (TTP) has been introduced. Both entities can send their items to the TTP that forwards them to the respective entities. However, this may create a communication and computation bottleneck at the TTP. To overcome this bottleneck, independently, Micali and Asokan et al. [1] introduced the optimistic approach in the context of fair exchange protocols. The rationale is that the TTP only intervenes in case of a problem, i.e. an entity is trying to cheat or a communication fails at a crucial moment. In an optimistic protocol, the TTP is said to be offline, while it is online in non-optimistic protocols. The optimistic approach has received most attention in recent literature.

In literature, there have been several papers addressing explicitly certified e-mail with either online TTP [7,10,15] or offline TTP [2,3,11]. However, all except Zhou et al. in [15] do not make a difference between a certified e-mail protocol and a non-repudiation protocol. In [15], Zhou et al. argue that non-repudiation of origin is not a required service—in fact, traditional certified mail does not provide it—and define a certified e-mail protocol to be a service that must be offered by an external delivery agent, in the same way it is realized by the post in traditional mail. We do not follow this last point, as the definition of a service provided by a protocol should not depend on *how* it is provided. Moreover, Zhou's approach does not permit to use an optimistic approach, which is, in most cases, more efficient. In this paper, we propose an optimistic certified e-mail protocol and we introduce an additional property, specific to certified e-mail, that has not been discussed previously: *author based selective receipt*. Generally, it is required that once the message is known to the recipient, he cannot prevent the protocol from delivering the receipt. It seems to us that this property is not sufficient: we believe that once the identity of the author is revealed, the receipt has to be delivered to the sender, and of course the message to the recipient. In traditional certified mail, one does not get any information about the mail, neither the content nor the origin, before having signed the receipt. To better understand the crucial importance of that property, consider the following real life example. If a person does not pay the rent of his flat, he can refuse a certified mail coming from his landlord. He does not need to read the letter to guess that the landlord is claiming the rent and wants a proof for this claim. Knowing the identity of the sender would reveal enough information to guess the content of the letter. If we suppose that a network address, e.g. an IP address is sufficient to identify a person we need to introduce an additional mechanism. We propose to use a third party (or several third parties) as an anonymity provider. Hence, our protocol is not entirely optimistic. However a third party providing anonymity is different from the TTP classically used in this kind of protocols. The third parties used for anonymizing communications do not have to generate evidences that need to be verified by an adjudicator during a dispute. An anonymity provider does not need to have a jurisdiction on evidence generation, as a TTP. Also, they do not need to be entirely trusted as they are observable by any exterior party, that could detect "strange" behaviors.

The rest of the paper will be structured as follows. In section 2, we will define the properties required by a certified e-mail protocol. In section 3, we present two variants of a new certified e-mail protocol: the first protocol does not provide data confidentiality, while the second one does. We go on discussing several solutions to the anonymity requirements of our protocols and finally conclude.

2 Properties

In this section we define all the properties a certified e-mail protocol is required to provide. We define them with respect to a sender, that we call Alice and a recipient we call Bob. Throughout he rest of this paper we also assume that no party acts against its own interests. This assumption is rather natural and discards scenarios, where a dishonest party, i.e. a party not following the protocol, could break some of the underneath defined properties by harming itself.

The first property is *non-repudiability*. Two kinds of non-repudiation services are required: non-repudiation of receipt and non-repudiation of origin.

Definition 1 (Non-repudiation of receipt). *A certified e-mail protocol provides non-repudiation of receipt, if and only if it generates a non-repudiation of receipt evidence, destined to Alice, that can be presented to an adjudicator, who can unambiguously decide whether Bob received a given message or not.*

Definition 2 (Non-repudiation of origin). *A certified e-mail protocol provides non-repudiation of origin, if and only if it generates a non-repudiation of origin evidence, destined to Bob, that can be presented to an adjudicator, who can unambiguously decide whether Alice is the author of a given message or not.*

Non-repudiation of origin is not necessary in a certified e-mail protocol, as it is not provided by a classic certified mail service. However, in most papers on certified e-mail it is treated as a mandatory property [3,7,10]. Moreover, in optimistic protocols a non-repudiation of origin evidence is provided quite naturally, as Bob must prove he received a message from Alice, when contacting the TTP. Both protocols that we present in the following section will provide non-repudiation of origin.

A second property the protocol must respect is fairness.

Definition 3 (Fairness). *A certified e-mail protocol is* fair *if and only if at the end of a protocol execution either Alice got the non-repudiation of receipt evidence, and Bob got the corresponding mail (as well as the non-repudiation of origin evidence if required), or none of them got any valuable information.*

Fairness ensures that none of the entities can cheat the other, i.e. arrives in a situation where either Alice or Bob has got his expected item, and the other has no mean of receiving his item anymore.

Another important property is timeliness.

Definition 4 (Timeliness). *A certified e-mail protocol provides* timeliness *if and only if all honest parties always have the ability to reach, in a finite amount of time, a point in the protocol where they can stop the protocol while preserving fairness.*

Timeliness assures that an entity does not need to keep open protocol runs for an infinite amount of time. Such a situation could occur if an entity is not sure whether the other entity stopped the protocol or not. It must always be possible for an entity to quit a protocol, without giving the possibility to the other entity to gain any advantage.

An optional property that can sometimes be required is confidentiality.

Definition 5 (Data confidentiality). *A certified e-mail protocol is said to provide* data confidentiality, *if and only if Alice and Bob are the only entities that can extract the content of the sent mail out of the protocol messages.*

This property ensures that no one can read the content of a sent mail, by for instance listening to the communication channels. In pen and paper transactions this property should be ensured by the envelope. Moreover, confidentiality is not always required. As adding confidentiality harms the efficiency of the protocol and is not always requested, we will present two protocols, where the first one does not provide confidentiality. It could also be interesting to anonymize Alice's identity from all entities exterior to the protocol and to provide *entity confidentiality*. In the remaining of this paper, we only consider data confidentiality.

The last property we require is *no selective receipt*. Selective receipt comes in two flavors.

Definition 6 (No message based selective receipt). *A certified e-mail protocol does not allow* message based selective receipt *if and only if once the message is known to Bob, he cannot prevent delivery of a receipt to Alice.*

No message based selective receipt is directly implied by the fairness requirement. We mention this property, as it has been discussed before in literature.

Definition 7 (No author based selective receipt). *A certified e-mail protocol does not allow* author based selective receipt *if and only if once the identity of the author is known to Bob, he cannot prevent delivery of a receipt to Alice.*

No author based selective receipt is a new property introduced in this paper. We believe that hiding the content of a mail, while stopping the protocol is still possible, is not sufficient. One should also hide the author's identity. In many cases the origin of a message leaks enough information to guess the mail's content[1].

3 The Protocols

In this section, we present two variants of a new protocol. The first protocol does not provide confidentiality, while the second one does. We believe that for efficiency reasons it is important that a user can decide to require confidentiality or not, as any additional property requires additional signatures or ciphers to be computed. We will discuss the security of each of the protocols with respect to the different properties. In both protocols, we assume that the communication channels between the TTP and respectively Alice and Bob are resilient, i.e. all data sent on such a channel arrive correctly after a finite, but unknown amount of time, although there may be delays. Channels between Alice and Bob may be unreliable, that means that data may accidentally be lost. Each of the protocols consists of three sub-protocols: a main protocol, an abort protocol and a recovery protocol. The main protocol performs the exchange. Both the abort protocol and the recovery protocol are used in case of problems to contact the TTP and either cancel the exchange or force a correct termination of the exchange.

[1] Of course, *every* sent certified e-mail has to provide this property. Otherwise, a recipient always rejects anonymous messages.

3.1 A Certified E-mail Protocol

Notations. We use the following notations to describe the protocol.

- $X \longrightarrow Y$: transmission from entity X to entity Y
- $X \xrightarrow{@} Y$: anonymous transmission from entity X to entity Y
- $X \xleftarrow{@} Y$: reply from entity Y to entity X on an anonymous channel
- $h()$: a collision resistant one-way hash function
- $E_k()$: a symmetric encryption function under key k
- $D_k()$: a symmetric decryption function under key k
- $E_X()$: a public-key encryption function under X's public key
- $S_X()$: the signature function of entity X
- m: the mail sent from A to B
- k: a fresh session key A uses to cipher m
- $c = E_k(m)$: the result of a symmetric ciphering of m under the key k
- $\quad = h(m, A, B, k)$: a label that, in conjunction with the identities (A,B), uniquely identifies a protocol run[2]
- f: a flag indicating the purpose of a message

The protocol generates the following evidences.

- $\text{EOO} = S_A(f_{\text{EOO}}, A, B, TTP, \ , h(c), k)$
- $\text{EOR}_c = S_B(f_{\text{EOR}_c}, B, TTP, \ , h(c), E_{TTP}(f_{\text{EOO}}, A, B, \ , k, \text{EOO}))$
- $\text{EOR}_k = S_B(f_{\text{EOR}_k}, A, B, \ , k)$
- $\text{Con}_k = S_{TTP}(f_{\text{Con}_k}, A, B, \ , k)$
- $\text{Abort} = S_A(f_{\text{Abort}}, \ )$
- $\text{Con}_{abort} = S_{TTP}(f_{\text{Con}_{abort}}, \ )$

Main Protocol

```
1. $A \xrightarrow{@} B$ :   $f_{\text{Com}}, B, TTP, \ , c, E_{TTP}(f_{\text{EOO}}, A, B, \ , k, \text{EOO})$
2. $A \xleftarrow{@} B$ :   $f_{\text{EOR}_c}, \ , \text{EOR}_c$
if $A$ times out then abort
else
3. $A \longrightarrow B$ :   $f_{\text{EOO}}, A, B, \ , k, \text{EOO}$
if $B$ times out then recovery[$X := B, Y := A$]
else
4. $B \longrightarrow A$ :   $f_{\text{EOR}_k}, A, B, \ , \text{EOR}_k$
if $A$ times out then recovery[$X := A, Y := B$]
```

The main protocol is composed of four messages. Alice starts by sending a commitment to the mail to Bob. This commitment consists of a cipher of the mail

[2] Although the label □ contains the identities of both Alice and Bob, we have to add them to the identification, as the TTP is unable to verify the content of the label (the TTP only verifies that the label is coherent with previous messages).

m, as well as a cipher *c* of the key *k* and the evidence of origin for *m*, ciphered with the TTP's public key. We also include the identity of the TTP that can be contacted in case of problem, as well as a label that in conjunction with the identities (A,B) uniquely identifies a protocol run. This label is added to each protocol message and avoids interference of different protocol runs. Moreover the purpose of each message is indicated by a flag. Also note that all the signatures in the protocol are on the hash of the cipher *c*, and not on the cipher itself. This fact allows us not to send the entire ciphertext, that can be very large, to the TTP in case of a recovery protocol, but only its hash. The first message is sent via an anonymous channel, hiding the sender's identity. Realization of such a channel will be discussed in the next section. Bob replies, via the open anonymous channel, sending an evidence of receipt for the cipher *c*. If Alice does not receive a valid response before a reasonable amount of time, she launches an abort protocol to stop the protocol. Here, a valid response means a response coherent with the previous message, i.e. the label must match the label in message 1 and the signature has to be correct. Otherwise Alice sends the key *k* and the evidence of origin EOO to Bob. If Bob does not receive a valid third protocol message, he executes the recovery protocol. Otherwise he sends an evidence of receipt for the key *k* to Alice. If Alice does not receive a valid evidence of receipt for the key, she has to contact the TTP in order to recover the protocol.

Recovery Protocol

```
1. $X \longrightarrow TTP: \quad f_{\mathrm{Rec}}, B, \ , h(c), \mathrm{EOR}_c, E_{TTP}(f_{\mathrm{EOO}}, A, B, \ , k, \mathrm{EOO})$
*if* (aborted or recovered) *then* stop
*else*
recovered = true
2. $TTP \longrightarrow A: \quad f_{\mathrm{Rec}_A}, A, B, \ , k, \mathrm{Con}_k, \mathrm{EOR}_c$
3. $TTP \longrightarrow B: \quad f_{\mathrm{Rec}_B}, A, B, \ , k, \mathrm{EOO}$
```

The aim of the recovery protocol is to enable either Alice or Bob to force a successful end of the protocol. The recovery protocol can be executed, once the protocol has reached a certain state. Bob can recover the protocol once he has got the first message from Alice, and Alice can launch it after having received the second message of the main protocol, i.e. the first message from Bob. When an invalid recovery request arrives, i.e. the signatures do not match the content of the cipher for the TTP, the TTP sends a signed message to alert *X* that the request is invalid. Receiving an *invalid request* token assures to *X* that *Y* will not be able to perform a valid recovery request and *X* can stop the protocol. Note that an invalid request does not disable the possibility to abort or recover the protocol later. When a valid recovery request arrives, the TTP must first ensure that this protocol run has not been aborted before. The protocol run is identified by the label and the identities (A,B). The abort protocol and the recovery protocol are mutually exclusive. In a first message either Alice or Bob sends all information the TTP needs to complete the protocol and to be sure the protocol has been started between Alice and Bob. The first message contains the hash of the cipher c, needed for the verification of EOO and EOR_c, the evidence of receipt for the cipher c, as well as the key and the evidence of origin ciphered with the TTP's public key. After having verified the validity of the signatures the TTP sends a confirmation of the key k, replacing the evidence of receipt for k, and the evidence of receipt for c to Alice. The TTP also sends the key k and the evidence of origin to Bob.

Abort Protocol

```
1. $A \xrightarrow{@} TTP : E_{TTP}(f_{\mathrm{Abort}}, A, B, \ , \mathrm{Abort})$
if (recovered or aborted) then stop
else
aborted=true
2. $A \xleftarrow{@} TTP : f_{\mathrm{Con}_{abort}}, \ , \mathrm{Con}_{abort}$
3. $TTP \longrightarrow B : f_{\mathrm{Con}_{abort}}, \ , \mathrm{Con}_{abort}$
```

The abort protocol can be launched by Alice, if Alice does not receive the second message of the main protocol. When Alice aborts the protocol, the TTP first has to verify that the protocol has not yet been recovered. Once the abort protocol has been engaged, recovery is not possible anymore. The communication between Alice and the TTP is ciphered and anonymous, in order to avoid Bob tracing the abort request to its originator. To abort the protocol Alice sends a signed abort request including the label that identifies the protocol. The request is ciphered using the TTP's public key to hide Alice's identity from Bob. The TTP sends a signed abort confirmation to both Alice and Bob. Once Alice started the abort protocol, she must not continue the main protocol anymore, even if the second message arrives. If the second message arrives and Alice continues the protocol, she does not have the ability to recover the protocol anymore. If Bob does not send the last message of the main protocol, Alice will not receive a receipt for her mail. Hence, Alice must stop the protocol after having received an abort token.

Properties

Non-repudiability. The protocol generates a non-repudiation of origin evidence (NRO) and a non-repudiation of receipt evidence (NRR). We have that

- NRO=EOO,
- NRR=$(\mathrm{EOO}_c, \mathrm{EOR}_k)$ or NRR=$(\mathrm{EOO}_c, \mathrm{Con}_k)$.

In case of *repudiation of receipt*, i.e. Bob denies having received a given mail, Alice can prove the receipt by presenting EOR_c, EOR_k or Con_k, EOO, , m, c, k, as well as the identities of A, B and the TTP to an adjudicator. The adjudicator verifies that:

- $\text{EOO} = S_A(f_{\text{EOO}}, A, B, TTP, \;, h(c), k)$
- $\text{EOR}_c = S_B(f_{\text{EOR}_c}, B, TTP, \;, h(c), E_{TTP}(f_{\text{EOO}}, A, B, \;, k, \text{EOO}))$
- $\text{EOR}_k = S_B(f_{\text{EOR}_k}, A, B, \;, k)$ or $Con_k = S_{TTP}(f_{\text{Con}_k}, A, B, \;, k)$
- $\; = h(m, A, B, k)$
- $c = E_k(m)$

If all the tests hold, the adjudicator concludes that Bob received the message.

In a similar way, in case of *repudiation of origin*, i.e. Alice denies being the author of the message, Bob can present EOO, , m, c, k, as well as the identities of A, B and the TTP to an adjudicator to prove the mail's origin. The adjudicator verifies that

- $\text{EOO} = S_A(f_{\text{EOO}}, A, B, TTP, \;, h(c), k)$
- $\; = h(m, A, B, k)$
- $c = E_k(m)$

If all the tests hold, the adjudicator concludes that Alice is the author of the mail.

Fairness. We will now show that the proposed protocol is fair: neither Alice nor Bob can receive a valuable item without the other one having the possibility to also do so. Therefore we will look at the different possible executions of the protocol. If the main protocol is entirely executed, it is trivial to see that the protocol provides fairness.

We will look at the possible implications of the abort protocol. The abort protocol can only be executed by Alice, as the abort request is signed. Note that a protocol is identified by and $(A, B)(= (B, A))$. Hence, this protocol run can only be aborted by Alice. Alice has the possibility to execute the abort protocol at any moment. However, executing it after having sent the third message of the main protocol could harm Alice. We therefore suppose that a honest Alice only executes the abort protocol before the third message of the main protocol. That means that neither the mail m nor any of the non-repudiation evidences has been exchanged. Any recovery request from either Alice or Bob will be refused by the TTP. In order not to harm herself, Alice will not continue the main protocol, once she executed the abort protocol. There is no possibility, neither for Alice nor Bob, to receive any valid item in this protocol run.

We will continue examining the consequences of a recovery protocol. Mutual exclusion of the abort protocol and the recovery protocol is ensured by the TTP. We can not have a situation where Alice has stopped the protocol, after having aborted it and Bob can recover the protocol by receiving the mail and the non-repudiation of origin message. The recovery protocol sends the expected items to both Alice and Bob. The only way of breaking fairness would be to send an invalid item to one of the entities and not to the other. However sending any invalid item will result in invalid evidences. Consider for instance, that Alice sends an invalid key in the first message of the main protocol. This key is always included in the non-repudiation of receipt evidence and so the evidence will be invalid.

Now consider the case where an invalid recovery request is sent to the TTP. Bob cannot verify the validity of his recovery request and could possibly not be able to generate a valid recovery request, as a part of it has been ciphered by Alice for the TTP. If the cipher contains incorrect information, the recovery cannot be performed and Bob is informed of this by the mean of an incorrect request token. However in that case neither Alice nor Bob can perform a recovery, as the cipher containing incorrect data has been signed by Bob in EOR_c and can not be replaced by Alice. We conclude that our protocol provides fairness.

Timeliness. Timeliness is provided by the fact that the communication channels between the TTP and both Alice and Bob are resilient. This means that all sent messages are received correctly after a finite amount of time. Looking at the protocol, we see that Alice at each moment of the protocol can contact the TTP to end the protocol: before having sent the third message Alice can abort the protocol and thereafter Alice can recover it. Bob can always execute a recovery protocol or at least receive an incorrect request token signed by the TTP. Hence at any moment in the protocol, both Alice and Bob have the ability to finish the protocol in a finite amount of time. In the previous paragraph we showed that executing the abort or the recovery protocol results in a fair termination of the protocol. Hence our protocol provides timeliness.

No selective receipt. No message based selective receipt is implied by fairness. To show that the protocol respects *no author based selective receipt*, we look at all possible executions of the protocol. After arrival of the first message, as the transmission is anonymized, Bob does not know Alice's identity. At this moment, Bob has the possibility to either execute the recovery protocol, stop the protocol or continue it by sending the second message. If Bob launches the recovery protocol, the protocol ends succesfully, so Alice's identity may be revealed. If Bob stops the protocol after having received the first message, Alice will execute the abort protocol. The abort protocol needs to be anonymized to avoid the following attack: Bob stops the protocol after having received the first message and waits for the TTP to receive an abort request for the protocol. Even if this request is ciphered, one could try to trace all incoming requests and recover Alice's identity. Therefore we use anonymous transmissions that can not be traced. All data sent to an anonymity provider must also be ciphered for this provider. For instance the abort request, is first ciphered for the TTP, and then, additionally, for the anonymity provider. Otherwise, Bob can permanently observe several "potential Alices", and compare outgoing messages to the certified e-mails he is receiving and to the requests arriving at the TTP. If Bob sends the second message, Alice will send message 3 of the main protocol. Once the third message is sent the fairness property ensures succesful termination of the protocol.

3.2 A Confidential Certified E-mail Protocol

Notations. The notation will be the same as in the previously described protocol. The changed evidences generated in the protocol are the following.

- $\text{EOO} = S_A(f_{\text{EOO}}, A, B, TTP, \ , h(c), E_B(k))$
- $\text{EOR}_c = S_B(f_{\text{EOR}_c}, B, TTP, \ , h(c), E_{TTP}(f_{\text{EOO}}, A, B, \ , k, \text{EOO}))$
- $\text{EOR}_k = S_B(f_{\text{EOR}_k}, A, B, \ , E_B(k))$
- $\text{Con}_k = S_{TTP}(f_{\text{Con}_k}, A, B, \ , E_B(k))$

The main protocol is very similar to the previous protocol. The difference with the first protocol is that the key k is systematically ciphered using Bob's public key. There are two reasons to do so. Firstly, we avoid an additional ciphering operation at the TTP during a recovery request. Secondly, we avoid that the TTP could gain knowledge or help some external attacker to gain knowledge of the mail. The only change we made to the recovery protocol, is that the first message contains the key ciphered with Bob's public key, and in messages 2 and 3 the ciphered key is sent instead of the plain key. Neither the TTP, nor an external observer, ever gains knowledge of the mail content. The abort protocol is identical to the abort protocol in the previous section.

Properties. Most of the reasonings for the previously discussed properties also hold for this protocol, as only minimal changes have been done. When we contact the adjudicator, we have to send all data on a confidential channel, in order for the mail content to stay confidential. When the adjudicator verifies the evidences, he additionally needs to check that ciphering k with Bob's public key results to $E_k(B)$ signed in the respective evidences. Therefore we must not use a probabilistic ciphering algorithm. We need to use a ciphering algorithm, that associates exactly one cipher to one plain text. Confidentiality is easily proved. The plain mail is never sent over the network. To recover the mail, one has to know the cipher c and the key k. However, the key k only intervenes ciphered under Bob's public key. If the ciphering algorithms producing both c and $E_B(k)$ are secure, the only party that can gain knowledge of m is Bob.

4 Anonymous Transmissions

In this section we will briefly discuss how anonymous transmissions may be realized. Consider first a situation where the network address does not reveal someone's identity. We can be the case of public places offering an Internet access, e.g. a cybercafe. In that case no additional mechanisms to provide anonymity are required.

In many cases, the network address can be thought to be equivalent to a person's identity. Then we need more elaborate mechanisms. A simple and commonly used way to provide anonymity on a network consists in special hosts, such as *anonymizers* or *re-mailers*. These hosts are used as an intermediate to directly hide the link between two hosts. Although this solution is very easily implemented, it suffers from several drawbacks. It does not resist more elaborate attacks, based on traffic analysis. Examples of such attacks are described in [5].

If we have to care about a powerful receiver, that can attack simple anonymizers, one has to use solutions, such as *mixnets*, proposed by Chaum in [6]. Mixnets are chains of anonymizers, called *mixes* that only know the address of the preceding and the following host. All sent message blocks have same length and the hosts always wait for a lower bound of messages before forwarding them. The order of the different incoming messages is mixed, making tracing of messages impossible. Moreover, $k - 1$ out of k mixes may collude, without compromizing anonymity. Although mixnets are rather inefficient, they offer secure anonymity services.

However, in our context, mixnets are not suitable, as they do not offer the possibility of *anonymous replies*: in our protocols, Bob has to reply to Alice on an anonymous channel without knowing her identity. On simple re-mailers, solutions based on pseudonyms are available. If we need secure anonymity services, resisting traffic analysis we may use the *Onion Routing* system [12], proposed and maintained by the US Naval Research Center. The onion routing system is based on an anonymous connection, set up once for the whole transmission, and defines a solution to the anonymous reply problem.

In practice a trade-off between security and efficiency needs to be made. The choice of the chosen solution will depend on the importance of the mail.

5 Conclusion

In this paper we presented a new protocol for certified e-mail. We discussed the properties of certified e-mail and their links to related problems, namely fair exchange, contract signing and non-repudiation. We introduced a new property, no author based selective receipt, that is specific to certified e-mail. The property claims that once the identity of the author of a mail is known, the receipt of the mail can not be refused anymore. Then we presented two variants of a new protocol: the first one does not provide confidentiality, while the second one does. As confidentiality is not always required, and harms the efficiency of the protocol, we suggest to leave the choice to the user, whether confidentiality is provided or not. Both protocols provide no author-based selective receipt. Finally, we discuss some mechanisms to provide anonymous transmissions that are needed in our protocols.

References

1. N. Asokan, M. Schunter, and M. Waidner. Optimistic protocols for fair exchange. In T. Matsumoto, editor, *4th ACM Conference on Computer and Communications Security*, pages 8–17, Zurich, Switzerland, Apr. 1997. ACM Press.
2. N. Asokan, V. Shoup, and M. Waidner. Asynchronous protocols for optimistic fair exchange. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 86–99, Oakland, CA, May 1998. IEEE Computer Society, Technical Committee on Security and Privacy, IEEE Computer Society Press.
3. G. Ateniese, B. de Medeiros, and M. T. Goodrich. TRICERT: A distributed certified E-mail scheme. In *Symposium on Network and Distributed Systems Security (NDSS 2001)*, San Diego, CA, Feb. 2001. Internet Society.

4. M. Ben-Or, O. Goldreich, S. Micali, and R. L. Rivest. A fair protocol for signing contracts. *IEEE Transactions on Information Theory*, 36(1):40–46, 1990.
5. O. Berthold, H. Federrath, and M. Köhntopp. Anonymity and unobservability in the internet. In *Workshop on Freedom and Privacy by Design*, Toronto, Canada, Apr. 2000.
6. D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, Feb. 1981.
7. R. H. Deng, L. Gong, A. A. Lazar, and W. Wang. Practical protocols for certified electronic mail. *Journal of Network and System Management*, 4(3), 1996.
8. S. Kremer and O. Markowitch. Optimistic non-repudiable information exchange. In J. Biemond, editor, *21st Symp. on Information Theory in the Benelux*, pages 139–146, Wassenaar (NL), May25-26 2000. Werkgemeenschap Informatie- en Communicatietheorie, Enschede (NL).
9. O. Markowitch and Y. Roggeman. Probabilistic non-repudiation without trusted third party. In *Second Conference on Security in Communication Networks'99*, Amalfi, Italy, Sept. 1999.
10. J. Riordan and B. Schneier. A certified E-mail protocol. In *14th Annual Computer Security Applications Conference*. ACM, 1998.
11. M. Schunter. *Optimistic Fair Exchange*. PhD thesis, Technische Fakultät der Universität des Saarlandes, Saarbrücken, 2000.
12. P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *IEEE Symposium on Security and Privacy, May 4–7, 1997, Oakland, California*, pages 44–54. IEEE Computer Society Press, 1997.
13. T. Tedrick. Fair exchange of secrets. In G. R. Blakley and D. C. Chaum, editors, *Advances in Cryptology: Proceedings of Crypto'84*, volume 196 of *Lecture Notes in Computer Science*, pages 434–438. Springer-Verlag, 1985.
14. J. Zhou, R. Deng, and F. Bao. Evolution of fair non-repudiation with TTP. In *ACISP: Information Security and Privacy: Australasian Conference*, volume 1587 of *Lecture Notes in Computer Science*, pages 258–269, 1999.
15. J. Zhou and D. Gollmann. Certified electronic mail. In *Proc. ESORICS '96*, volume 1146 of *Lecture Notes in Computer Science*, pages 160–171, 1996.
16. J. Zhou and D. Gollmann. An efficient non-repudiation protocol. In *PCSFW: Proceedings of The 10th Computer Security Foundations Workshop*. IEEE Computer Society Press, 1997.

Spatial Domain Digital Watermarking with Buyer Authentication

Subhamoy Maitra[1] and Dipti Prasad Mukherjee[2]

[1] Computer and Statistical Service Center, Indian Statistical Institute,
203, B T Road, Calcutta 700 035, India,
subho@isical.ac.in

[2] Electronics and Communication Science Unit, Indian Statistical Institute,
203, B T Road, Calcutta 700 035, India,
dipti@isical.ac.in

Abstract. This paper presents watermark insertion algorithm applied to spatial domain of an image. It is assumed that a possible forger knows the proposed watermarking technique. The retrieval of watermark is based on the principles of error correcting codes using secure image key and the original image. The scheme could identify a buyer specific signature from the image in the form of a unique buyer key assigned to every copy of the image being sold. The survival of the watermark is demonstrated for a wide range of image transformations and forging attempts both in spatial and frequency domains.

Keywords: Buyer Key, Digital Watermarking, Error Correcting Code, Image Key.

1 Introduction

With the increased use of digital media and internet technologies, the distribution and dissemination of digital multimedia objects become wide spread. One of the important issues in this area is the introduction of secure copyright technique for multimedia data. In this paper, we present an invisible digital watermarking technique for images and show that effective recovery of watermark is possible even under a variety of attacks.

In general the watermark insertion strategy revolves around inserting watermarks in the perceptually significant regions of the image [7]. This motivation is based on the fact that any attempt to modify the watermark results in visible distortion of the image. A number of watermarking techniques exist that introduce watermarks in the spatial domain [14,4,2], most of which are not robust enough in case of intentional attacks in frequency domain. In one of the important frequency domain watermarking, Cox et al have proposed spread spectrum based insertion of watermark refining the major DCT components [3]. The inserted watermark is recovered using a statistical similarity measure with the original watermark. Similar approach by statistically modeling the DCT coefficients is reported in [6]. Ruanaidh et al [13] have introduced watermark by

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 149–161, 2001.

modifying the phase of the DFT as it is perceptually more important than the magnitudes of the Fourier coefficients. The differential-energy watermarking algorithm embeds labeled bits by selectively discarding high frequency discrete cosine transform (DCT) coefficients in certain selective image regions [8]. An information theoretic model for steganography is given in [1] where uncertainty about the embedded watermark is resolved using principles of hypothesis testing.

Our proposed technique of watermarking does not depend on the perceptually significant regions of the image rather it is based on the concept of an image key and a buyer key. The buyer key ensures a footprint, specific to the buyer of a particular multimedia object, while the image key is dependent on the spatial organizations of pixels. Therefore, the recovery of watermark in this case not only protects the copyright but also authenticates the possible owner in case multiple copies of the same image or some modifications of it are sold. This is achieved without any additional computational cost to the watermarking process. The desirable objectives of the proposed watermarking scheme are as follows.

1. We consider that the algorithm for watermarking is known to the possible attacker. However, the key(s) associated with the process are as usual secure.
2. We use error correcting codes to generate secure buyer key(s). A random partitioning of the image is used as the image key.
3. In no time the perceptual quality of the image could be compromised. The process of watermark insertion is controlled to the extent that the image pixels are well within just noticeable distortions.
4. Every multimedia object has a single image key and multiple buyer keys depending on the number of copies sold. So, parameters for key generation could be different for an expensive multimedia object compared to a low cost one. It is presumed that a high value item is sold less in number compared to a low value one. Accordingly, a high value item is secured in greater detail.

Craver et al [4] have introduced watermark after dividing the image into two blocks and then modifying the intensity of the blocks by the same amount but in reverse ways. The safeguard to watermark is expected because of the inability of the attacker to guess the exact partitioning of the original image. The process however fails in case of frequency domain attack and also there is not enough randomness to fight a variety of spatial attacks. The image key that we have used divides the image into a large number of blocks such that it is impossible for the attacker to guess the image partitions.

The question of copyright protection should ideally take care of both intentional attack and common transformations on the watermark [12,9]. Ruanaidh et al [13] have assessed an exhaustive list of number of possible threats and exploitation in case of digital watermarking in images. We demonstrate that our present approach is robust enough to survive common image transformations like rotation, isotropic scaling, cropping etc., besides intentional attacks in both spatial and frequency domains.

The paper is organized as follows. In Section 2, we present the generation of cryptographic parameters viz., the image and the buyer key. In Section 3, the proposed watermarking scheme is outlined including the watermark recovery process. The simulation for watermarking and key retrieval results are presented in Section 4. This is followed by conclusion.

2 Generation of Image and Buyer Key

2.1 The Image Key

Consider an image I, which can be seen as a matrix of size $2^a \times 2^b$. Let us consider that the image is divided into $m = 2^n$ subgroups, each containing 2^{a+b-n} pixel locations. Let us denote the subgroups by $G_0, G_1, \ldots, G_{m-2}, G_{m-1}$. Each subgroup G_k is thus a set of 2^{a+b-n} tuples of the form row index, column index . Note that row index i varies from 0 to $2^a - 1$ and the column index j varies from 0 to $2^b - 1$. It is also clear that $G_1 \cap G_2 =$ for $0 \le k_1 = k_2 \le m - 1$.

Example1.

	0	1	2	3
0	5	7	11	233
1	34	14	79	61
2	123	211	47	11
3	0	254	1	191

(a) Matrix for I^e

	0	1	2	3
0	3	0	3	3
1	2	0	2	1
2	2	1	2	0
3	1	2	0	3

(b) Label matrix L^{I^e}

	0	1	2	3
0	5	7+1	11	233
1	34-1	14+1	79	61
2	123-1	211	47-1	11+1
3	0	254-1	1+1	191

(c) Matrix for I^{ew}

Let us explain the scenario with the Example 1. Consider the $2^2 \times 2^2$ matrix in Example 1(a) where $a = b = 2$. This matrix corresponds to an example image I^e. Each location of the matrix can be referred as i, j pair for $0 \le i \le 2^a - 1, 0 \le j \le 2^b - 1$. Each location i, j contains some value, typically between 0 to 255 for an 8-bit intensity image. For example, the location 2, 3 contains the value 11. We take $m = 2^n = 2^2 = 4$ for example and assign $G_0 = \{ 0,1 , 3,2 , 1,1 , 2,3 \}$, $G_1 = \{ 1,2 , 3,0 , 2,1 , 1,3 \}$, $G_2 = \{ 1,0 , 2,0 , 2,2 , 3,1 \}$, $G_3 = \{ 0,0 , 0,2 , 0,3 , 3,3 \}$. Storing this kind of groups of the image pixels is easier if we maintain a label matrix L^I corresponding to the image I, which is of the size $2^a \times 2^b$. Each location of this matrix contains the value k, $0 \le k \le m - 1$, if the corresponding pixel location in the image I belongs to the group G_k. This is shown in Example 1(b). Now we provide the following proposition.

Proposition 1. *Consider an image I of size $2^a \times 2^b$. Consider the partition of the image I in $m = 2^n$ groups $G_0, \ldots, G_{m-1}$, each containing equal number of pixel locations 2^{a+b-n}. Then the total number of options to select such groups is greater than $2^{2^{n-1}(a+b-1)}$.*

Proof. Given the partition of 2^n groups, the number of pixels at each group is 2^{a+b-n}. Thus the total number of choices to select such groups is equal to

$$\psi = \binom{2^{a+b}}{2^{a+b-n}} \times \binom{2^{a+b}-2^{a+b-n}}{2^{a+b-n}} \times \binom{2^{a+b}-2\times 2^{a+b-n}}{2^{a+b-n}} \times \ldots \times \binom{2^{a+b}-(2^n-1)\times 2^{a+b-n}}{2^{a+b-n}}$$

$$= \prod_{k=0}^{2^n-1} \binom{2^{a+b}-k\times 2^{a+b-n}}{2^{a+b-n}} > \prod_{k=0}^{2^{n-1}-1} \binom{2^{a+b}-k\times 2^{a+b-n}}{2^{a+b-n}} >$$

$$\binom{2^{a+b}-2^{n-1}\times 2^{a+b-n}}{2^{a+b-n}}^{2^{n-1}} = \binom{2^{a+b-1}}{2^{a+b-n}}^{2^{n-1}} > (2^{a+b-1})^{2^{n-1}} = 2^{2^{n-1}(a+b-1)}$$

Corresponding to an image I, it is now clear that the total number of options in choosing the label matrix L^I is prohibitively large. We select a random label matrix from this set. We use this random label matrix as the image key K^I. Thus the size of the image key is 2^{a+b} locations, each containing an integer value in between 0 to $2^n - 1$. This integer value can be represented using n bits. Thus the total size of the image key K^I is $n2^{a+b}$ bits. Given p number of image pixels, generation of image key needs $O(p)$ operation. For an image size 256 × 256, if we divide the image in $m = 2^n = 2^8$ groups, we need 2^{16} bytes or 64 kbytes. The image key will be stored with the owner of the image and there is no need to communicate this key. Thus, the amount of space required for storing this key is moderate. Each location of the image key can be accessed as $K^I_{i,j}$ for $0 \le i \le 2^a - 1, 0 \le j \le 2^b - 1$.

2.2 The Buyer Key

Depending on the number of groups $m = 2^n$ in the image, we take a binary vector of length 2^n. This vector is considered as the buyer key B. Each location of the bit vector B can be accessed as B_k for $0 \le k \le 2^n - 1$. Vector B is selected from a set of binary error correcting codes C. The set contains M distinct code words such that Hamming distance [10] between any two code words is at least d. For experimentation, we use the set of 2^n length code words containing $M = 2^{n+1}$ distinct codes with minimum distance 2^{n-1} [10]. The motivation of selecting buyer keys from a set of error correcting codes will be clear in the following section where watermark insertion and retrieval issues are discussed.

The actual buyer key used is not directly derived from the error correcting codes that we will use. Rather it is a random permutation $\pi(B)$ of the code word B. Note that this permutation $\pi(.)$ is selected randomly but it is specific for an image. So this can be considered as a part of the image key. This provides additional robustness given that a possible forger may know the error correcting codes but not the image specific permutation. Given a moderate value of the code length m, such possible permutations are $m!$, which is prohibitively large. In subsequent discussions, this transformation is not incorporated due to brevity of space. Also it has no additional influence on the watermarking and retrieval algorithms described next.

3 Watermarking Scheme

The overall approach of the proposed scheme is presented in Figure 1. In the watermarking module, the original image is spatially divided into a number of blocks based on image key. Image intensity of each block is then modulated depending on the bit values of the buyer key. This process generates the watermarked image. In the retrieval process, the original and the watermarked (may be forged) images are compared block by block. The block information is obtained from the image key. Depending on the extent of intensity modification in each block a probable buyer key is generated. This key is then mapped to exact buyer key by correcting the errors using the theory of error correcting codes [10].

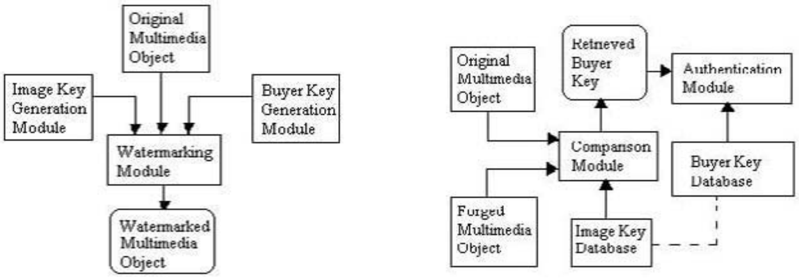


Fig. 1. The watermarking and the watermark retrieval process.

3.1 Insertion of Watermark

The process of generating watermarked image I^w from the original image I is described next.

Algorithm 1

```
1. For $0 \leq i \leq 2^a - 1, 0 \leq j \leq 2^b - 1$
    (a) Let $I_{i,j}$ is the pixel value of the image $I$ at the pixel location $i, j$ .
    (b) Let $i, j$ belongs to the group $G_k$, i.e., $L^I_{i,j} = k$.
    (c) If $B_k = 0$, $I^w_{i,j} = I_{i,j}$, else if $B_k = 1$, $I^w_{i,j} = I_{i,j} + \beta_{i,j}$.
```

Now we denote the sum $\sum_{i,j \quad G_k}(I^w_{i,j} - I_{i,j}) = \sum_{i,j \quad G_k}\beta_{i,j} = \delta_k$. We will show that this δ_k value plays an important role in this digital watermarking technique. We consider that $\beta_{i,j}$'s are either all positive or all negative corresponding to a group G_k. Thus the values of δ_k may be either positive or negative. Also it is important to decide the values of $\beta_{i,j}$ such that the quality of the image is not degraded. We follow the principles of Weber ratio (WR) [5, Pages 16–18] in selecting the values of $\beta_{i,j}$. To maintain perceptual clarity, $WR = \sum_{i,j}|I^w_{i,j} - I_{i,j}| / \sum_{i,j} I_{i,j}$ is taken as less or equal to 2%. Let us discuss the situation in terms of the example image I^e. From I^e we can construct the watermarked image I^{ew} with $\beta_{i,j} = 1$, or $\beta_{i,j} = -1$, for $0 \leq i \leq 3, 0 \leq j \leq 3$ and the buyer key $B^e = 1010$. This is shown in Example 1(c). For G_0 we take $\beta_{i,j} = 1$ and for G_2, we take $\beta_{i,j} = -1$. In this case $\delta_0 = 4, \delta_2 = -4$ and $\delta_1 = \delta_3 = 0$. Hence, given the image I, image key K^I and the unaltered watermarked image I^w, one can find the buyer key B as follows.

Algorithm 2

```
1. For $0 \le k \le 2^n - 1$, initialize values $\sigma_k = 0$, bit values $q_k = 0$.
2. For $0 \le i \le 2^a - 1, 0 \le j \le 2^b - 1$
       If $i, j$ belongs to the group $G_k$, $\sigma_k = \sigma_k + I^w_{i,j} - I_{i,j}$.
3. For $0 \le k \le 2^n - 1$
   (a) if $\sigma_k$ and $\delta_k$ are equal with the value 0, $q_k = 0$.
   (b) if $\sigma_k$ and $\delta_k$ are equal with nonzero value, $q_k = 1$.
4. Report $q$ as the buyer key $B$.
```

Thus if Bob buys an watermarked image I^w from the owner Alice and then resells I^w to Oscar, then from I^w (no matter whether Alice gets it from Bob or Oscar), Alice can easily find out the buyer key and identifies Bob. However, the real situation is little more complicated. Bob can very well make some intensional processing over the image so that the scheme get disturbed and in such a situation Alice can not identify the buyer key as given in the above algorithm. This is discussed next. Again, for p pixels, the retrieval of buyer key is executed in $O(p)$ time.

3.2 Identifying Buyer Key from Attacked Watermarked Image

In this subsection we give the idea how we can find out the buyer key successfully from the attacked watermarked image $I^{w\#}$. Let us first describe the algorithm. Algorithm 3

```
1. For $0 \le k \le 2^n - 1$, initialize values $\sigma_k = 0$, bit values $q_k = 0$.
2. For $0 \le i \le 2^a - 1, 0 \le j \le 2^b - 1$
       If $|I^{w\#}_{i,j} - I_{i,j}| > |\beta_{i,j}|$ then $I^{w\#}_{i,j} = I_{i,j} + \beta_{i,j}$.
3. For $0 \le i \le 2^a - 1, 0 \le j \le 2^b - 1$
       If $i, j$ belongs to the group $G_k$, $\sigma_k = \sigma_k + I^{w\#}_{i,j} - I_{i,j}$.
4. For $0 \le k \le 2^n - 1$
   (a) if $|\sigma_k| \le c_k|\delta_k|$, $q_k = 0$.
   (b) else if $|\sigma_k| > c_k|\delta_k|$, $q_k = 1$.
5. Find out the codeword $q$ closest to $q$ and report $q$ as $B$.
```

3.3 Analysis of the Watermark Retrieval Process

Given a wide range of image transformations and intentional attacks on watermark, exact determination of c_k is not possible. Instead, we would be using a range of values for c_k. We call this as tolerance factor. In simulation, we present our retrieval results as the number of bit wise matches between B and q against a range of values for c_k. Our experimentation shows that it is always possible to find out the buyer keys properly with some tuning of this method. We like to elaborate a few issues in this regard.

The method works over some disjoint subsets of the image. Given the enormous number of options to select one such image key (see Proposition 1), it is clear that guessing the image key is almost impossible. Also, while watermarking, the intensity values may be increased or decreased for an individual group. Thus if the attacker tries to change some pixel values in the same way and at the same time tries to regroup the pixels, the effect will be nullified over the whole region corresponding to a subgroup G_k.

Given the watermarking scheme and the $\beta_{i,j}$ values, the difference between $I_{i,j}$ and $I^w_{i,j}$ is known. Thus, if we receive an attacked watermarked image $I^{w\#}$, then we can prune the image as follows. If $|I^{w\#}_{i,j} - I_{i,j}|$ is greater than $|\beta_{i,j}|$, then it is clear that this can not be a true value in the watermarked object. Thus, in this situation the value of $I^{w\#}_{i,j}$ should be replaced by $I_{i,j} + \beta_{i,j}$ before further processing.

Once again recapitulate that $\sum_{i,j \in G_k}(I^w_{i,j} - I_{i,j}) = \sum_{i,j \in G_k}\beta_{i,j} = \delta_k$. After the pruning, $\sum_{i,j \in G_k}|I^{w\#}_{i,j} - I_{i,j}| \leq \sum_{i,j \in G_k}|I^w_{i,j} - I_{i,j}|$ and thus $0 \leq |\sigma_k| = \sum_{i,j \in G_k}|I^{w\#}_{i,j} - I_{i,j}| \leq |\delta_k|$. At this point the decision problem is should we interpret the value of σ_k as 0 or as δ_k. If we interpret σ_k as 0, then B_k is 0, else we interpret B_k as 1. The value of the tolerance factor plays an important role in this respect.

It is most natural that we should consider $q_k = 0$, if $|\sigma_k| \leq 0.5|\delta_k|$ and $q_k = 1$, if $|\sigma_k| > 0.5|\delta_k|$. However, depending on different kinds of attack it is not always possible to fix $c_k = 0.5$. It is important to tune the value of c_k in between 0 to 1 to decide the proper decision boundary for choosing the value of q_k. Also the values of c_k may differ for different k. Thus we should write the relation as $q_k = 0$, if $|\sigma_k| \leq c_k|\delta_k|$ and $q_k = 1$, if $|\sigma_k| > c_k|\delta_k|$.

It may very well happen that the attacked image $I^{w\#}$ is such that there are some errors in deciding the bits of q. We already know that the buyer key B is chosen from a set of error correcting codes. If q itself is a codeword, then we choose $q' = q$. Else we will try to find out a codeword q' closest to q. We then decide the buyer key B as q'. Since the minimum distance between the codewords is d, even if there are $\frac{d}{2} - 1$ errors in selecting q (i.e., the Hamming distance between q and B is at most $\frac{d}{2} - 1$), at the time of finding the closest codeword q' these errors can be corrected [10]. Thus we will get the proper buyer key $B = q'$. On the other hand, if the number of errors is more than $\frac{d}{2} - 1$, then a wrong buyer key will be estimated and the scheme will fail. Given the scheme, it is intuitively clear that the probability of errors in properly estimating the regional sum δ_k from σ_k is very low, as without knowing the image key (the division of groups) it is almost impossible to change the bias in the region. Moreover, for an erroneous detection of buyer key, such judgements need to be wrong for more than $\frac{d}{2} - 1$ regions, which further reduces the error probability. It seems extremely complicated to provide a closed form mathematical expression in calculating the probability of error. To justify our claim, with experimental results we will show that the possibility of wrong estimation is negligible. In fact in all the experiments we could detect the buyer key properly.

For expensive items, it is natural that we need to provide additional security. Also, it is expected that less number of copies will be sold. This means we need comparatively less number of code words. This helps in selecting a larger value for d and in turn ensures better performance while retrieving watermarks.

In the next section, we simulate the insertion and retrieval of watermark in two dimensional images and show the robustness of the methodology under a variety of image transformations and forging attacks.

4 Simulation

We have used Reed Muller codes [10] to generate 2^n length buyer keys with minimum distance between any two keys being 2^{n-1}. There are 2^{n+1} such codes. For experimentation, we have taken $n = 7$. This makes the length of the key 128 bits and can provide maximum 256 number of distinct code words. Bit wise matching value of 128 gives the exact match for the buyer key. The scheme can correct a maximum of 31 bit errors and bit wise matching in at least 97 (= 128-31) positions ensure complete decoding of the buyer key.

In the experiments, watermark is added in the spatial domain and its robustness is tested against a set of possible image transformations and simulated attacks. In this paper, we present the results from an 128 × 128 digitized image. The image is divided into 128 different image blocks. The results are presented in the form of graphs where bit wise matching values are plotted against the tolerance factor described in Section 3. Throughout the experiments, we have used all the $\beta_{i,j}$ values equal to $\beta = 1$, as it is the minimal level of intensity modification of the pixels in spatial domain. Naturally, this is the most favorable situation for the attacker and consequently the retrieval process is most challenging. In the subsequent discussions, we show that our method survives satisfactorily in retrieving the buyer key.

To evaluate the performance of our scheme, a wide range of simulations has been carried out and retrieval process is successful in all the cases. We present here a representative set of results that highlight the contribution of our method. We have implemented the algorithms in MATLAB on an Intel PIII 800 MHz computer. For 128 bit buyer keys applied on 128 × 128 digitized image, the watermarking process is almost instantaneous while the buyer key retrieval takes approximately 0.3 to 0.7 seconds depending on the range of tolerance factor used.

4.1 Performance with Respect to Common Image Transformations

The different image transformations tested are as follows. (a) Scaling : In this case both expansion and reduction of image size are considered. (b) Rotation : We rotate the image at certain angle. (c) Cropping : In this case we have set the constraint that cropping area can be no higher than 40% of the original size. (d) Combined transformation : A combination of (a), (b) and (c).

In case of scaling and rotation, the watermarked image is reverted back to the original size and orientation for testing of retrieval of watermark. The cropped image is reverted back to original dimension by adding the missing part of the cropped image from the original image. The original *fruit* image of size 128 × 128 and its watermarked version are shown in Figure 2(a) and 2(b) respectively. Note that there is no perceptual difference between the original and the watermarked version of the image. For simulation, the watermarked image is separately subjected to 127% expansion, 79% reduction and a rotation of 13^o. The cropped image for testing watermark retrieval is shown in Figure 2(c). In case of combined transformation, the sequence of transformations includes 83% reduction followed by 13^o rotation and cropping maintaining 80% of the watermarked image. The resultant image is shown in Figure 2(d). The buyer key retrieval result is shown in Figure 3(a). Table 1 presents image transformation type (column 1) and parameters (column 2), the Weber ratio value (column 3), and tolerance factor (c_k) range (column 4) for bit wise matching value ≥ 97. Note that for the image in Figure 2(d), we could successfully recover the buyer key even if the quality degrades beyond perceptually acceptable limit after the transformations. Note that as an estimate of degradation of the images after the attack we provide the Weber Ratio $WR = \sum_{i,j} |I^{w\#}_{i,j} - I_{i,j}| / \sum_{i,j} I_{i,j}$ in the tables. See Figure 3(a) for the graphical representation. In graphical representations we plot the range of c_k in the horizontal axis. For the watermarking scheme we select a buyer key B and insert that in the image. Using the recovery scheme we get back a key q. In the graph we plot at vertical axis in how many places B and q are matching corresponding to a specific value of c_k. Note that in the experiments we vary c_k for the complete range 0 to 1 and also for all the values of k we choose the same c_k value. If we get the bit matching value between B and q in at least 97 places, we can recover the key exactly [10].

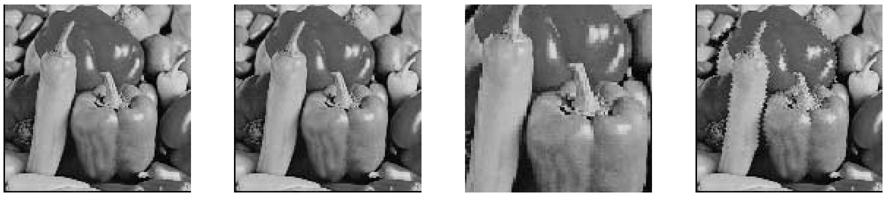


Fig. 2. Original, watermarked and attacked images, (a), (b), (c), (d) from left to right.

Table 1. Experimental results for common image transformations.

Transformation type	Transformation parameters	Weber ratio	Range of c_k
Expansion	1.27×	0.008	0 to 1
Reduction	0.79×	0.087	0 to 0.87
Rotation	13^o	0.173	0.35 to 0.63
Cropping	73% of original area	0.191	0 to 0.5
Combined	0.83×, 13^o rotation, 80% of original area	0.442	0.17 to 0.28

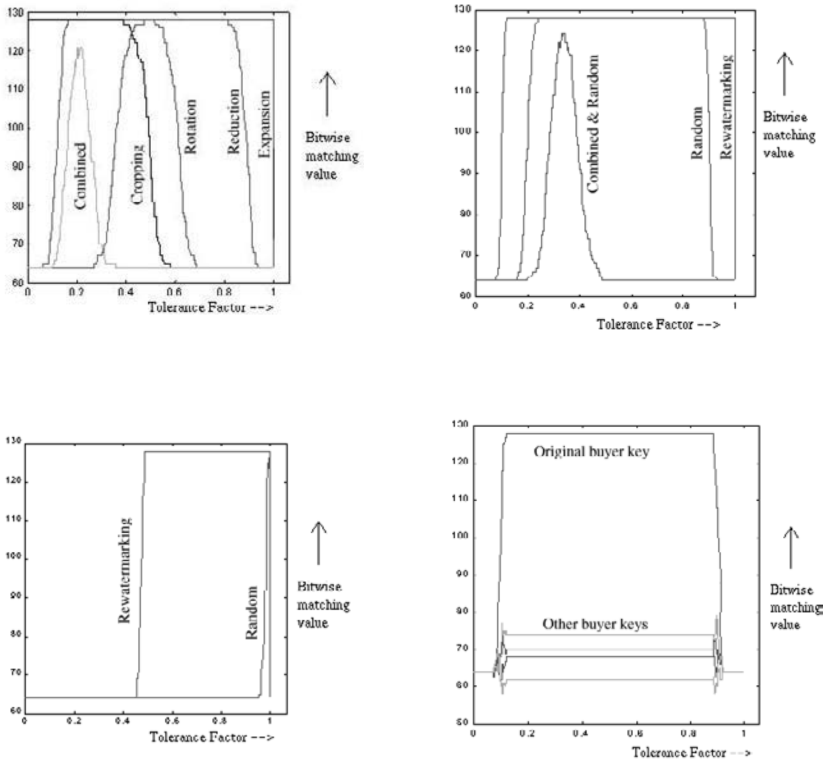


Fig. 3. Retrieval Graphs, (a), (b), (c), (d) from top left to bottom right.

4.2 Performance against Spatial Domain Attack

Attempt in destruction of the watermark in the spatial domain is the most common type of attack in digital watermarking. We have simulated three such conditions and the performance of our proposed scheme against such attacks is shown in Figure 3(b). In the first case, rewatermarking is done on the watermarked image. The parameters for the process are exactly identical as the original watermarking except that different image and buyer keys are used for rewatermarking. This is in line with our assumption that the attacker is aware of the watermarking algorithm. The next test is to corrupt the intensity values of the watermarked image by either increasing or decreasing it by at most 2 based on random decision. Finally, random attack is implemented on the watermarked image already subjected to combined image transformations of scaling, rotation and cropping. In all these three cases, proposed watermarking scheme is successful. Numerical results are as follows. See Figure 3(b) for the graphical representation.

Table 2. Experimental results for spatial domain attacks.

Transformation type	Transformation parameters	Weber ratio	Range of c_k
Rewatermarking		0.02	0.2 to 1
Random		0.079	0.1 to 0.9
Combined and random	0.83×, 13^o rotation, 80% of original area	0.541	0.3 to 0.4

Table 3. Experimental results for frequency domain attacks.

Transformation type	Weber ratio	Range of c_k
Rewatermarking	0.039	0.5 to 1
Random	0.102	0.97 to 1

4.3 Performance against Frequency Domain Attack

The simulation of forging attempt is further extended to frequency domain. The watermarked image is transformed to frequency domain using FFT. The FFT domain image including the imaginary part is subjected to following attacks. First we insert a separate watermark in the frequency space following watermarking principles explained in Algorithm 1. With the knowledge of the watermarking process, this could be a valid attack with the intention that the spatial watermark is going to be distorted. The $\beta_{i,j}$ values selected for this purpose is taken as 5% of the original value. The second case is similar to random attack in spatial domain except that in this case amplitude values in frequency space are randomly manipulated. The amplitude is changed to a maximum of ±10% of the original value.

After these attacks, inverse FFT is applied and the image is passed through retrieval process. In both the cases successful retrieval of buyer key is possible. The corresponding watermarking parameters are shown in Table 3. See Figure 3(c) for the graphical representation.

4.4 Authentication of Buyer Key

We once again refer to the step 5 of Algorithm 3. In this step, we find the correct code word q closest to q. Our hypothesis is that the selection of a wrong buyer key is improbable for the complete range of tolerance factor. We substantiate this with the following experiment. We use the buyer key B to watermark the image and then perform the random attack in spatial domain as in Table 2. We get back the key q using our retrieval algorithm. We also select four other buyer keys B_1, B_2, B_3, B_4 (from the same error correcting code) which are different from B. Figure 3(d) shows that varying the tolerance factor c_k, the original code word B provides the highest bit wise matching with the retrieved bit pattern q. For the rest of the code words, bit wise matching between B_i ($i = 1, 2, 3, 4$) and q do not ever reach the threshold 97 necessary for buyer key authentication.

4.5 Collusion Attack

Note that if we use same value of $\beta_{i,j}$ for all buyer keys, then using more than one images, it is possible to guess the true value of the image pixels by comparing watermarked images. Our motivation in this direction is that the collusion attack should not succeed in getting the original image. For this we propose different sets of $\beta_{i,j}$ values for different users. Corresponding to the image I, for each pixel value $I_{i,j}$, we define a range of values $\mu^-_{i,j} \leq \beta_{i,j} \leq \mu^+_{i,j}$ ($\mu^-_{i,j} < 0, \mu^+_{i,j} > 0$) such that $I_{i,j} + \beta_{i,j}$ does not make any perceptual change for the complete range. Now for a specific user u, we choose $\beta^u_{i,j}$ from this range of $\beta_{i,j}$ values. We modify the step 1(c) of Algorithm 1 as follows. Corresponding to the buyer key B, if the bit value $B_k = 0$, then for the group G_k we use $I^w_{i,j} = I_{i,j} - \beta^u_{i,j}$, otherwise for $B_k = 1$, we take $I^w_{i,j} = I_{i,j} + \beta^u_{i,j}$. Thus it is not possible for the collusion attackers to decide on the exact value of each pixel. In such a case the pruning step 2 of Algorithm 3 needs to be modified also. The choice of $\beta_{i,j}$ values in this case will identify different buyers and it is of interest to explore whether this helps in identifying one or more of the attackers who participate in collusion attack.

5 Conclusion

We have proposed a novel watermarking technique that survived attacks both in frequency and spatial domains. The watermark retrieval is based on the secure image key and the original image. Since the retrieval is basically the identification of the buyer key, the trail of forging could be identified through the buyer key. The proposed technique is computationally attractive and has the potential for improvement which we are working on. We are extending this for multimedia objects incorporating watermarking in spatial, frequency and wavelet domains.

References

1. C. Cachin. An Information-Theoretic Model for Steganography. In *2nd Workshop on Information Hiding*, in volume 1525 of Lecture Notes in Computer Science. Springer Verlag, 1998.
2. I. J. Cox and M. L. Miller. A review of watermaking and the importance of perceptual modeling. In *Proceedings of Electronics Imaging*, February 1997.
3. I. J. Cox, J. Kilian, T. Leighton and T. Shamoon. Secure Spread Spectrum Watermaking for Multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997.
4. S. Craver, B. Yeo and M. Yeung. Technical Trials and Legal Tribulations. *Communication of the ACM* July 1998, 41(7), July 1998.
5. R. C. Gonzalez and P. Wintz. *Digital Image Processing*. Addison-Wesley Publishing (MA, USA), 1988.
6. J. R. Hernansez, M. Amadeo and F. P. Gonzalez. DCT-Domain Watermaking Techniques for Still Images: Detector Performance Analysis and a New Structure. *IEEE Transactions on Image Processing*, 9(1):55–68, 2000.

7. N. F. Johnson, Z. Duric and S. Jajodia. Information Hiding: Steganography and Watermarking – Attacks and Countermeasures. Kluwer Academic Publishers Boston, USA, 2000.
8. G. C. Langelaar and R. L. Lagendijk. Optimal Differential Energy Watermarking of DCT Encoded Images and Video. *IEEE Transactions on Image Processing*, 10(1):148–158, 2001.
9. C. S. Lu, S. K. Huang, C. J. Sze and H. Y. Liao. *A New Watermarking Technique for Multimedia Protection*. Multimedia Image and Video Processing. L. Guan, S. Y. Kung and J. Larsen (ed.), pages 507–530, CRC Press (Boca Raton, USA), 2001.
10. F. J. MacWillams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. North Holland, 1977.
11. F. Mintzer, G. W. Braudaway and A. E. Bell. Opportunities for Watermaking Stndards. *Communication of the ACM* July 1998, 41(7), July 1998.
12. F. A. P. Petitcolas, R. J. Anderson, M. G. Kuhn and D. Aucsmith. Attacks on Copyright Marking Systems. In *2nd Workshop on Information Hiding*, pages 218–238 in volume 1525 of Lecture Notes in Computer Science. Springer Verlag, 1998.
13. J. O. Ruanaidh, H. Petersen, A. Herrigel, S. Pereira and T. Pun. Cryptographic copyright protection for digital images based on watermarking techniques. *Theoretical Computer Science* 226:117–142, 1999.
14. M. M. Yeung. Digital Watermaking. *Communication of the ACM* July 1998, 41(7), July 1998.

Efficient Public Auction with One-Time Registration and Public Verifiability

Byoungcheon Lee, Kwangjo Kim, and Joongsoo Ma

Information and Communications University,
58-4, Hwaam-dong, Yusong-gu, Taejon, 305-732, Korea
{sultan,kkj,jsma}@icu.ac.kr

Abstract. In public auction, all bid values are published, but each bidder participates in auction protocol in anonymous way. Recently, Omote and Miyaji [OM01] proposed a new model of public auction in which any bidder can participate in plural rounds of auction with one-time registration. They have introduced two managers, registration manager (RM) and auction manager (AM), and have used efficient tools such as bulletin board and signature of knowledge [CS97].In this scheme, even if a bidder is identified as a winner in a round, he can participate in next rounds of auction maintaining anonymity for RM, AM, and any bidder. But a problem of this protocol is that the identity of winner cannot be published. In the winner announcement stage, RM informs the vendor of winner's identity secretly. Therefore RM's final role cannot be verified, and AM and any participating bidder can not be sure of the validity of auction.

In this paper, we propose a new public auction scheme which can solve this problem. In the proposed scheme, both RM and AM execute randomization operation in round setup process which makes the publication of winner's identity be possible while keeping anonymity of winner in next rounds of auction. Moreover, AM provides ticket identifier based on Diffie-Hellman key agreement which is recognized only by the bidder. Our scheme provides real anonymity in plural rounds of auction with one-time registration in a verifiable way.

Keywords: public auction, English auction, anonymity, one-time registration, public verifiability, hash chain, signature of knowledge, anonymous signature scheme

1 Introduction

Electronic auction is an attractive form of electronic commerce and recently many kind of auction services are provided over the Internet. Electronic auction can be classified into sealed-bid auction and public auction according to the way it runs.

In sealed-bid auction [FR96,SKM00,Sako00,OM00,SM00], each bidder secretly submits a bid only once in bidding stage. In opening stage, a bidder

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 162–174, 2001.

who has offered the highest price is announced as a winner. In this type of auction, bid secrecy is of prime concern. Possible problems of sealed-bid auction are that the competition principle does not work well and a winning bid may be much higher than market price. In public auction [OM01,NT00,SS99], also called English auction, all the bid values are published, but each bidder participates in auction protocol in anonymous way. Each bidder offers higher price one by one and can bid multiple times in a round of auction. Finally, a bidder who has offered the highest price becomes a winner. In this case anonymity of bidder is of prime concern. Traditionally, sealed-bid auction and public auction are two different ways of running auction, and one is preferred than the other according to applications. Recently, many online auction services are provided on the Internet and most of them are public auction. In this paper we consider how to improve public auction.

Requirements of public auction can be listed as follows [OM01].

1. Anonymity: Nobody can identify a bidder from a bid.
2. Traceability: A winner who has submitted the winning bid can be traced.
3. No framing: Nobody can impersonate a certain bidder.
4. Unforgeability: Nobody can forge a bid with a valid signature.
5. Non-repudiation: The winner cannot repudiate the fact that he has bidded the winning bid.
6. Fairness: All bids should be dealt with in a fair way.
7. Public verifiability: Anybody can verify the validity of a bidder, the validity of a bid, and the correctness of winner announcement.
8. Unlinkability (among different rounds of auction): Nobody can link the same bidder's bids among different rounds of auction.
9. Linkability (in a round of auction): Anybody can link which bids are placed by the same bidder and knows how many times a bidder places a bid in a round of auction.
10. Efficiency of bidding: The computation and communication amount in both bidding and verifying should be practical.
11. One-time registration: Bidder can participate in plural rounds of auction anonymously with one-time registration.
12. Easy revocation: RM can revoke certain bidder easily.

Note that we have added the public verifiability and non-repudiation compared with [OM01].

[NT00] proposed a public auction protocol which keeps bidder privacy using group signature scheme. They used the useful property of group signature that a member of a group can sign anonymously on behalf of the group, and the group manager can identify the signer later. But the public auction based on group signature requires complicated signature generation and verification procedure. Moreover the group signature does not satisfy the anonymity for group manager (GM) at all since GM has special power to identify bidders. Revocation of a bidder is also difficult in group signature.

Recently, [OM01] proposed an efficient model of public auction. In their scheme, two managers, registration manager (RM) and auction manager (AM), are introduced to provide the anonymity of bidder. As an anonymous signature scheme, they used the signature of knowledge [CS97] with an anonymous challenge. They made the overall protocol very simple and efficient by using bulletin board as a public communication channel. But a problem of this protocol is that the identity of winner cannot be published. In the winner announcement stage, RM secretly informs the vendor of the winner's identity. Therefore AM and all participating bidders cannot be sure whether RM has executed his role correctly and winner was decided, i.e., the winner announcement is not publicly verifiable. If winner's identity is published (exposed to AM), the anonymity of winner for AM is not satisfied in future rounds of auction because AM uses the same public key in future rounds of auction.

To solve this problem, we propose a new public auction protocol. In our protocol, both RM and AM execute randomization operation in round setup process to prepare auction ticket, so RM or AM alone cannot identify bidders. Moreover winner's identity can be published in the winner announcement stage while keeping the anonymity of winner in future rounds of auction. Therefore, plural rounds of auction with one-time registration is possible in a verifiable way. Moreover, AM provides ticket identifier using Diffie-Hellman key agreement which is recognized only by the bidder.

This paper is organized as follows. First, [OM01] scheme is describe briefly and its problem is discussed in Section 2. Next, cryptographic primitives such as signature of knowledge, hash chain, and Diffie-Hellman key agreement are described in Section 3. Then, the proposed public auction protocol is described in detail in Section 4 and various features of the proposed protocol are discussed in Section 5. Finally, we conclude in Section 6.

2 Omote and Miyaji's Scheme

The public auction scheme proposed by Omote and Miyaji [OM01] is an efficient model of public auction in which bidders can participate in plural rounds of auction with one-time registration. In this scheme, two kind of managers are introduced. Registration manager (RM) secretly knows the correspondence of bidder's identity and bidder's registration key, and works as an identity escrow agency. Auction manager (AM) hosts the auction and prepares auction tickets in each round.

Consider a discrete logarithm based cryptosystem. Let p and q be two large primes satisfying $q|p-1$ and g be a generator of multiplicative group Z_p with order q. AM has private key x_A and public key $y_A = g^{x_A}$. The i-th bidder B_i has private key x_i and public key $y_i = g^{x_i}$.

2.1 Procedure

Bidder registration: A bidder B_i registers his public key y_i to RM as follows. He chooses a random number t_i and sends (y_i, t_i) with a proof that he knows the private key x_i (discrete logarithm of y_i to the base g). When RM accepts the proof, he publishes (y_i, t_i) on his bulletin board and keeps bidder's identity B_i secretly in his secure database.

AM's round setup: Assume that AM holds the k-th round of auction. She gets (y_i, t_i) of every participating bidders B_i from RM's bulletin board. She computes shared secret keys $y_i^{x_A}$ for every bidders B_i by using Diffie-Hellman key agreement technique. She generates random numbers r_i for every bidders and keeps them secretly. She computes the following auction keys T_i for B_i

$$T_i = (Enc^k(y_i^{x_A}, t_i), y_i^{r_i}, g^{r_i})$$

where $Enc^k(y_i^{x_A}, t_i) = Enc(y_i^{x_A}, Enc^{k-1}(y_i^{x_A}, t_i))$ is the k-time encryption of t_i using a shared key $y_i^{x_A}$. She publishes the auction keys T_i of all bidders on her bulletin board in a shuffled way.

Bidding: Bidder B_i who wants to participate in the k-th round of auction can easily find his auction key T_i from AM's bulletin board because he can compute $Enc^k(y_i^{x_A}, t_i)$ in advance by using $y_A^{x_i} = y_i^{x_A}$. When he places a bid, he sends the following bid information $(m_i, y_i^{r_i}, g^{r_i}, V_2)$ to AM.

- a bid m_i (m_i = auction ID || bid value)
- $y_i^{r_i}$ and g^{r_i} published by AM
- $V_2 = SK[\alpha : y_i^{r_i} = (g^{r_i})^{\alpha}](m_i)$

Here V_2 is a signature of knowledge [CS97] on message m_i and implies that B_i knows the value $\alpha = x_i$.

Winner decision and announcement: Assume that m_j be a winning bid. AM proves to RM that the public information $y_j^{r_j}$ added to a winning bid m_j corresponds to the public key y_j by sending r_j^{-1}. Then, RM informs a vendor of winner's identity secretly after the winner decision procedure.

2.2 Problem of this Scheme

This scheme is a very efficient public auction in the sense that the bidding and verifying procedures are very simple and each bidder can participate in plural rounds of auction with one-time registration. But a problem of this scheme is that the winner announcement stage is not publicly verifiable. AM's proof to RM (sending r_j^{-1}) and RM's secret identification of the winner to a vendor are not published at all. This kind of secret proof is not a good way in public auction over a distributed network like the Internet. In the winner announcement stage, every bidders can just recognize what the highest bid value is, but they cannot verify whether two managers have executed their job correctly and who the winner is. They just have to trust the honesty of two managers. In AM's point of view, she sends r_j^{-1} to RM, but cannot verify whether RM gives proper identification of winner to the vendor. Therefore this kind of auction protocol that cannot be verified publicly cannot be used in real application and it does not have the one-time registration property.

The reason that winner's identity cannot be published is that anonymity of winner for AM is not provided in future rounds of auction. Since AM uses the same key material y_i for every rounds of auction, she can identify the winner easily in future rounds of auction. So fairness and unlinkability are not provided.

In this paper we propose a new public auction protocol which can solve this problem. The basic idea is that RM executes an additional randomization operation in round setup procedure such that the winner's identity can be published in the winner announcement stage and the winner anonymity for AM is kept in future rounds of auction.

3 Cryptographic Primitives

3.1 Signature of Knowledge

We use the signature of knowledge (SK) of discrete logarithm introduced by Camenisch and Stadler [CS97] as an anonymous signature scheme. Let x be a private key of a signer and $y = g^x$ be the corresponding public key. A pair $(c, s) \quad \{0, 1\}^l \times Z_q$ satisfying $c = h(m||y||g||g^s y^c)$ where l is a security parameter of hash function, is a signature of knowledge of the discrete logarithm of the element $y \quad Z_p$ to the base g on the message m. Such a signature of knowledge can be computed if the private key $x = \log_g y$ is known, by choosing a random number $k \quad Z_q$ and computing

$$c = h(m||y||g||g^k) \quad \text{and} \quad s = k - cx \bmod q.$$

It is verified by checking $c \overset{?}{=} h(m||y||g||g^s y^c)$. We denote this signature of knowledge as

$$V = SK[x : y = g^x](m).$$

SK represents both the proof of knowledge of the private key x and a signature on message m.

This scheme can be used as an anonymous signature scheme if (y^r, g^r) are challenged for a secret random number $r \quad Z_q$ instead of (y, g). The signer computes (c, s) satisfying $c = h(m||y^r||g^r||(g^r)^s(y^r)^c)$ for challenged (y^r, g^r). We denote this signature as

$$V = SK[x : y^r = (g^r)^x](m).$$

3.2 Hash Chain

Assume that a bidder B_i and RM are sharing secret bidder information t_i. In each round k, they compute a special hash chain

$$h^k(t_i) \equiv h(t_i, h^{k-1}(t_i))$$

which can be computed only by the bidder B_i and RM who know t_i. If $h()$ is a collision-resistant cryptographic hash function, computing $h^k(t_i)$ without knowing t_i is infeasible even though all $h^j(t_i)$ for $j < k$ are known.

This is a kind of secure channel between B_i and RM. Using this primitive, a bidder can easily identify his round key generated by RM while keeping the anonymity of the round key against any other party including AM.

3.3 Diffie-Hellman Key Agreement

Assume that a bidder B_i has a key pair (x_i, y_i) and AM has a key pair (x_A, y_A). B_i and AM can share a secret key $K_i = y_i^{x_A} = y_A^{x_i}$ using Diffie-Hellman key agreement technique. Using the shared secret key K_i, bidder B_i can easily identify his auction ticket generated by AM, while AM does not know which is B_i's auction ticket.

4 Proposed Public Auction Scheme

In this Section, we describe the proposed public auction scheme which is a modification of [OM01] such that RM executes an additional randomization operation in round setup procedure and winner's identity is published on bulletin board.

4.1 System Set-Up

The entities of our scheme consists of the registration manager (RM), the auction manager (AM), and n bidders B_i $(i = 1, ..., n)$. The role of each entity is as follows:

RM
- He is in charge of the one-time registration process and has secret database to keep secret user information.
- He participates in round key setup process to publish round keys in shuffled way on his bulletin board.
- He publishes winner specific information on his bulletin board in the winner announcement stage.

AM
- She prepares auction tickets in each round of auction using a random number and round keys. She publishes them on her bulletin board in a shuffled way. She has secret database to keep random numbers.
- She publishes winner specific information on her bulletin board in the winner announcement stage.
- She has private key x_A and public key $y_A = g^{x_A}$.

Bidder (B_i where $i = 1, ..., n$)
- Bidder has to register to RM to participate in auction.
- He participates in a round of auction using his auction ticket.
- He has private key x_i and public key $y_i = g^{x_i}$.

In [OM01], winner's identity is secretly informed to the vendor by RM, therefore vendor is an important entity. But in our scheme the vendor of auction does not have any role because winner's identity is published on bulletin board. In this setting we assume that RM and AM do not collude each other to open the anonymity of bidder. If they collude, they can identify any bidder.

In our scheme, five bulletin boards are used, i.e., bulletin boards for registration, round key, auction ticket, bidding, and winner announcement. Bulletin board is a kind of public communication channel which can be read by anybody, but can be written only by legitimate party in an authentic way. All communications are executed publicly via bulletin boards except the one-time registration message of bidder to RM. The registration and round key boards are written only by RM and the auction ticket board is written only by AM. The information posted on each bulletin board is as follows.

Registration board (written by RM)

- RM publishes the identities and public keys of registered bidders.

Round key board (written by RM)

- RM computes round keys for every registered bidders and publishes them in a shuffled way.

Auction ticket board (written by AM)

- AM computes auction tickets for every valid bidders listed in round key board of RM and publishes them in a shuffled way.

Bidding board (written by bidder)

- Each bidder posts his bidding information on this board. Only higher bid than the previous highest one can be posted. Posting of a bid cannot be prevented by anybody.

Winner announcement board (written by AM and RM)

- In the winner announcement stage, AM publishes the winner dependent secret random number.
- In the winner announcement stage, RM publishes the winner dependent secret information.

To identify a winner in the winner announcement stage, RM and AM should keep bidder dependent secret information. Therefore, the following two secret databases are used.

User information DB (managed by RM)

- RM maintains secret user information for registered bidders.

Random number DB (managed by AM)

- AM maintains secret random numbers used to generate auction tickets in each round of auction.

4.2 Public Auction Protocol

The proposed public auction protocol consists of the following 5 stages. Registration of bidder is only one-time in the auction protocol, but other 4 stages are executed in each round of auction. We depict the overall auction protocol in Figure 1.

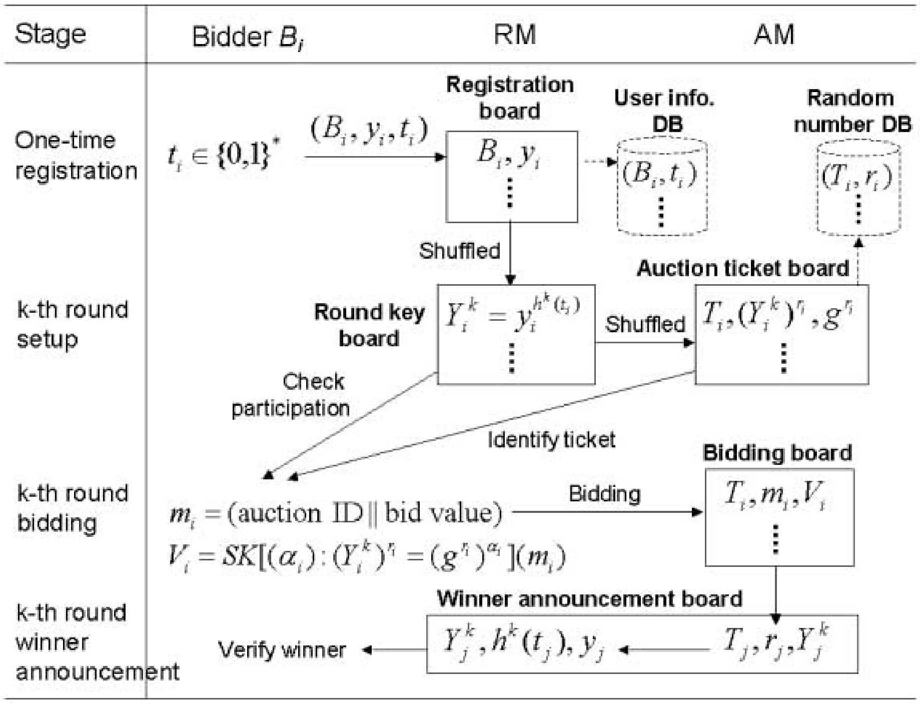


Fig. 1. Public auction protocol

Stage 1. One-time registration:

A bidder B_i registers to RM as follows:

1. B_i chooses his private key $x_i \in_R Z_q$ and computes his public key $y_i = g^{x_i}$ (Or a certified key with certificate can be used).
2. B_i chooses a random string $t_i \in \{0,1\}^*$ and keeps it secretly.
3. B_i sends (B_i, y_i, t_i) to RM secretly and proves his knowledge of the private key x_i in zero-knowledge.
4. If RM accepts B_i's registration, he publishes (B_i, y_i) on his registration board and keeps (B_i, t_i) secretly in his secure user info DB.

Stage 2. RM's round key setup (k-th round auction):

Now assume that RM, AM and all n bidders are involved in the k-th round of auction. RM computes n round keys $Y_i^k = y_i^{h^k(t_i)}$ for all n bidders using y_i and t_i. Then he shuffles and publishes them on his round key board. Note that a bidder B_i can check easily whether his round key is listed on the round key board because he can also compute round key Y_i^k. But anybody except RM and B_i does not know the correspondence between y_i and Y_i^k. If RM wants to revoke a bidder, then he just removes the bidder from the registration board and removes the round key from the round key board.

Stage 3. AM's auction ticket preparation (k-th round auction):
AM gets the list of all the round keys Y_i^k of n valid bidders from RM's round key board. Then she executes the following steps.

1. She chooses n random numbers $\{r_1, ..., r_n\} \quad _R \; Z_q$.
2. She computes the auction keys $(Y_i^k)^{r_i}, g^{r_i}$.
3. She computes the ticket identifiers $T_i = h((Y_i^k)^{x_A})$.
4. She shuffles and publishes the auction tickets $(T_i, (Y_i^k)^{r_i}, g^{r_i})$ on the auction ticket board.
5. She keeps (T_i, r_i) secretly in her secure random number DB.

Note that a bidder B_i can find the ticket identifier T_i easily as he can compute $T_i = h(y_A^{h^k(t_i)x_i}) = h(K_i^{h^k(t_i)})$ in advance, while AM and RM cannot identify B_i from T_i.

Stage 4. Bidding (k-th round auction):
A bidder B_i who wants to participate in the k-th round of auction executes the following steps.

1. He computes his round key as $Y_i^k = y_i^{h^k(t_i)}$ and checks whether his round key is listed in RM's round key board. If his round key is not listed, he complains to RM.
2. He computes his ticket identifier as $T_i = h(Y_A^{h^k(t_i)x_i})$ and gets his auction ticket $(T_i, (Y_i^k)^{r_i}, g^{r_i})$ from the auction ticket board. If his auction ticket is not listed in the auction ticket board, he complains to AM.
3. He checks the validity of his auction ticket by $(g^{r_i})^{h^k(t_i)x_i} \overset{?}{=} (Y_i^k)^{r_i}$. If it does not hold, he complains to AM.
4. He prepares his bid information (T_i, m_i, V_i) as follows and posts them on the bidding board.
   - m_i=(auction ID || bid value), or any relevant information can be included.
   - $V_i = SK[\alpha_i : (Y_i^k)^{r_i} = (g^{r_i})^{\alpha_i}](m_i)$ where $\alpha_i = h^k(t_i)x_i$.

The bid value should be higher than the previous highest one. Note that only the bidder B_i who knows $\alpha_i = h^k(t_i)x_i$ (knows both t_i and x_i) can compute the signature of knowledge V_i.

Stage 5. Winner announcement (k-th round auction):
Assume that a bid m_j of bidder B_j is the highest bid at the end of the bidding stage. AM and RM jointly publish the winner on the winner announcement board as follows.

1. AM announces that (T_j, m_j, V_j) is a winning bid.
2. AM posts (T_j, r_j, Y_j^k) on the winner announcement board which reveals the correspondence between Y_j^k and $(Y_j^k)^{r_j}$.
3. RM posts $(Y_j^k, h^k(t_j), y_j)$ on the winner announcement board which reveals the correspondence between $Y_j^k = y_j^{h^k(t_j)}$ and y_j. It shows that B_j is the winner.

4. Anyone verifies that B_j is the winner using the published values r_j and $h^k(t_j)$.

Although $h^k(t_j)$ is published, t_j is not revealed because of the one-wayness of hash function. $h^{k+1}(t_j)$ cannot be computed from $h^l(t_j)$ for $l \leq k$ without knowing t_j.

The ticket identifier T_i can be recognized only by B_i who knows both t_i and x_i. B_i recognizes the correspondence between y_i and Y_i^k using t_i and recognizes the correspondence between Y_i^k and T_i using x_i. Anybody else including RM and AM cannot identify the two correspondence together. Therefore anonymity of bidder is provided while giving an efficient ticket identifier.

Public verifiability of winner is provided by publishing r_j and $h^k(t_j)$ together. r_j can be published safely after the bidding is finished because it is a random number chosen by AM in a round of auction. $h^k(t_j)$ can also be published safely after the bidding is finished, because $h^{k+1}(t_j)$ is not exposed if t_j is kept secretly.

5 Features

We discuss various features of the proposed public auction protocol according to the list of requirements.

1. Anonymity: We assume that RM and AM do not collude to break the anonymity of bidders. If they collude, they can identify any bidder. They corporate only for winning bid in a public way.
   - Anonymity for RM: RM cannot identify B_i from the auction tickets $(T_i, (Y_i^k)^{r_i}, g^{r_i})$ published by AM on the auction ticket board or bidding information (T_i, m_i, V_i) posted by B_i on bidding board. Identifying Y_i^k from $(Y_i^k)^{r_i}$ is a discrete logarithm problem. Without knowing the secret shared key K_i between B_i and AM, RM cannot identify B_i from T_i. RM also cannot identify B_i from V_i because of the zero-knowledge property of SK.
   - Anonymity for AM: AM cannot identify B_i from the round key Y_i^k published by RM without knowing t_i. Identifying y_i from $Y_i^k = y_i^{h^k(t_i)}$ is a discrete logarithm problem. Although AM knows the previous values of $h^l(t_i)$ for $l < k$, she cannot compute $h^k(t_i)$ because of the collision-resistance of the cryptographic hash function $h()$. AM also cannot identify B_i from V_i because of the zero-knowledge property of SK.
2. Traceability: A winner's identity B_j can be identified with the corporation of AM (publishing r_j) and RM (publishing $h^k(t_j)$) together as shown in the winner announcement stage.
3. No framing: Nobody can impersonate a bidder B_i because the signature of knowledge V_i can be computed only with $\alpha_i = h^k(t_i)x_i$ and the bidder B_i is the only person who knows α_i. Even though RM and AM collude, they cannot impersonate B_i.

4. Unforgeability: Anybody including RM and AM cannot forge a valid bid of a bidder B_i with a signature V_i.
5. Non-repudiation: The winner B_j cannot repudiate his bidding because it contains his valid signature V_j.
6. Fairness: Because all bids are anonymous and are posted on the bidding board by the bidder, all bids are dealt with in a fair way.
7. Public verifiability: Because all the relevant information is published on bulletin board, anybody can verify the validity of a bid (by signature of knowledge V_i), the validity of bidder (by round key and auction ticket), and the correctness of winner announcement (by r_j and $h^k(t_j)$).
8. Unlinkability (among different rounds of auction): Because the auction ticket is generated by two randomization operations by RM (round key generation) and AM (auction ticket generation), the auction ticket cannot be linked to a bidder. Therefore, nobody can link the same bidder's bids among plural rounds of auction.
9. Linkability (in a round of auction): Because the same auction ticket is used in a round of auction, anybody can link which bids are placed by the same bidder and knows how many times a bidder places a bid in a round of auction.
10. Efficiency of bidding: In our protocol, most of communication is executed in very simple way, posting on public bulletin boards. Only one exception is that a bidder transmits (B_i, y_i, t_i) to RM through a secure channel in the one-time registration stage. Any complex protocol such as non-repudiation protocol as introduced in [OM01] is not required because a bidder posts his bid on the bidding board. Any secure channel between RM and vendor is not required. The overall computation for one-time registration to RM (1GSK+1VSK), round key generation by RM (1E), auction ticket generation by AM (3E), computing bidding information by bidder (2E+1GSK), and verifying the winner announcement (2E+1VSK) are very efficient, where E, GSK, and VSK represent modular exponentiation, generation of signature of knowledge, and verification of signature of knowledge, respectively.
11. One-time registration: Although the winner's identity in a round of auction is published, the anonymity of auction ticket is maintained in next rounds of auction. Therefore, bidders can participate in plural rounds of auction anonymously with one-time registration.
12. Easy revocation: When a bidder wants to withdraw from an auction or RM wants to revoke a bidder, RM can simply delete the bidder from his registration board and the round key from the round key board.

We compare the features of proposed protocol with [OM01] in Table 1. In [OM01] AM can distinguish winner's public key although she does not know winner's identity because the same public keys are used by AM repeatedly. Therefore, anonymity for AM, fairness, and unlinkability are not satisfied. As discussed in Section 2, public verifiability and one-time registration are not satisfied in [OM01]. But the proposed scheme satisfies all these requirements. In terms of computational load, the proposed scheme requires a little more exponentiation than [OM01], but both systems are very practical for real application. In com-

Table 1. Comparison of proposed public auction scheme with [OM01]

Features	[OM01]	Proposed
Anonymity for RM	O	O
Anonymity for AM	X	O
Traceability	O	O
No framing	O	O
Unforgeability	O	O
Non-repudiation	O	O
Fairness	X	O
Public verifiability	X	O
Unlinkability	X	O
Linkability	O	O
One-time registration	X	O
Easy revocation	O	O
Registration	1GSK+1VSK	1GSK+1VSK
Round setup by RM	–	1E
Round setup by AM	2E	3E
Bidding	1E+1GSK	2E+1GSK
Winner announcement	1E+1VSK	2E+1VSK
Non-repudiation protocol	required	not required

munication model, our scheme does not require any non-repudiation protocol because bidding information is posted on bidding board by the bidder.

6 Conclusion

We have pointed out the problem of [OM01], lack of public verifiability in the winner announcement stage, and proposed a new public auction scheme which solves this problem. In our scheme both RM and AM execute randomization operations in each round setup process such that RM or AM alone cannot identify bidders, which makes the publication of winner's identity be possible. An efficient ticket identifier is provided such that only a legitimate bidder can identify his auction ticket easily while any other party cannot identify it.

Compared with [OM01], our scheme has following advantages.

1. All the stages of public auction including the winner announcement stage are publicly verifiable because all the relevant information is published on bulletin boards.
2. The overall communication is more efficient. In our scheme winner's identity is published on bulletin boards while it is secretly informed to vendor by RM in [OM01]. Therefore, secure channel is not required in winner announcement stage and non-repudiation protocol for fairness is not required.

3. Plural rounds of auction with one-time registration is possible in a verifiable way.

One drawback of our scheme compared with [OM01] is that the round setup process is executed by two entities, RM and AM, but it is an essential cost to provide public verifiability together with anonymity in one-time registration.

References

[CS97] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups", In *Crypto'97*, pages 410–424, 1997.

[FR96] M. Franklin and M. Reiter, "The design and implementation of a secure auction service", In *IEEE Transactions on Software Engineering*, pages 302–312, 1996.

[NT00] K. Nguyen and J. Traore, "An online public auction protocol protecting bidder privacy", In *ACISP'2000*, pages 427–442, 2000.

[OM00] K. Omote and A. Miyaji, "An anonymous auction protocol with a single non-trusted center using binary trees", In *ISW'2000*, pages 108–120, 2000.

[OM01] K. Omote and A. Miyaji, "A practical English auction with one-time registration", In *ACISP'2001*, pages 221–234, 2001.

[Sako00] K. Sako, "An auction protocol which hides bids of losers", In *PKC'2000*, pages 422–432, 2000.

[SKM00] K. Suzuki, K. Kobayashi, and H. Morita, "Efficient sealed-bid auction using hash chain", In *ICISC'2000*, pages 189–197, 2000.

[SM00] K. Sakurai and S. Miyazaki, "An anonymous electronic bidding protocol based on a new convertible group signature scheme", In *ACISP'2000*, pages 385–399, 2000.

[SS99] S. G. Stubblebine and P. F. Syverson, "Fair online auctions without special trusted parties", In *Financial Cryptography'1999*, pages 230–240, 1999.

An Analysis of Integrity Services in Protocols[*]

Kapali Viswanathan, Colin Boyd, and Ed Dawson

Information Security Research Centre, Queensland University of Technology,
GPO Box 2434, Brisbane, Q 4001, Australia,
kapali@isrc.qut.edu.au, {boyd,dawson}@fit.qut.edu.au

Abstract. An analysis of integrity services in cryptologic protocols is presented. The informal syntax, to be presented, attempts to model the integrity service as a property that is transferred from a key to a message. The message can, in turn, be a key. The modeling presupposes confidentiality and integrity to be the atomic properties or services offered by cryptologic algorithms. More complex algorithms and protocols, such as those for digital signature, identification protocols and non-malleable encryption, are considered to be ensembles of these services. This paper concentrates only on the analysis of the integrity service in signature techniques based on the proof of knowledge of discrete logarithm. The paper will demonstrate the usefulness of this modeling by identifying flaws in the recent proposals for an efficient electronic cash system and a key-recovery system.

Keywords : Confidentiality, integrity, representation of cryptologic goals.

1 Introduction

Confidentiality and integrity services are the atomic properties that are required for the construction of cryptologic protocols. The work on network security architectures by Rueppel [14] is an example for a research with a similar view. These properties can be viewed as follows: keys provide service (confidentiality or integrity) to messages. The importance of entities (like Alice or Bob) is deliberately avoided in subsequent definitions and analyses in order to facilitate a *key-centric view* of cryptosystems[1], which may be more appropriate for the representation, analysis and design of cryptosystems. Such an approach does not require any form of protocol idealisation [5], which may create more difficulties in the analysis of protocols. Moreover, since the idealisation functions do not have an inverse mapping (de-idealisation functions), the analysis techniques employing such functions may not be useful directly in the design of protocols.

A cryptosystem can be viewed to be a composition of integrity and confidentiality services, which can be considered to be independent of each other. Although integrity and confidentiality services are not totally independent, the

[*] Research Supported by the Australian Research Council grant A49804059

[1] This is as opposed to an *entity-centric view*, such as that of the BAN logic [5].

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 175–187, 2001.

results of this paper will be logically consistent. This is because if the relationship between the two services were to be represented syntactically, the syntax will only add more functionalities to the model and will not remove any.

Due to this view, cryptosystems may be decomposed into an integrity component and a confidentiality component. This decomposition when represented in a suitable fashion will result in a simple characterisation of the goals of the cryptosystem – that is the integrity goal and the confidentiality goal. Many proposals in recent times, knowingly or unknowingly, have neglected the integrity goal of the cryptosystem. The negligence often results in deficient cryptosystems, which may be highly undesirable for many applications.

The concern of this paper is an informal technique for the representation of the integrity goal. There exists many papers that have attempted to represent the confidentiality goal, such as the paper by Abadi and Rogaway [1]. So, this paper will not deal with the representation of the confidentiality service. Section 2 presents an analysis of the integrity goal. The subsequent sections will employ the proposed technique to analyse the the electronic cash system proposed by Radu, Govaerts and Vandewalle [12], and the fraud detectable key recovery scheme by Verheul and van Tilborg [17].

2 An Integrity Verification Technique

The informal working definitions for the integrity and confidentiality services are as follows:

Definition 1 Confidentiality *is the service that grants* access *to the message corresponding to the cipher-text when the* access *to the key is available.*

Definition 2 Integrity *is the service that determines the* immutability *of a message corresponding to a cipher-text when the* immutability *of the key has been determined.*

These definitions express succinctly the importance of the confidentiality and integrity properties of the keys in cryptosystems. The aim of any cryptosystem is to *maintain* the confidentiality and integrity properties of the messages with respect to the corresponding properties of the keys.

The transfer of a cryptologic property from a key to a message will be represented as follows:

$$K \xrightarrow{SERVICE,C} M$$

where, $SERVICE \quad \{C, I\}$ is the type of service, C is the keyword for the confidentiality service and I is the keyword for the integrity service. Confidentiality is the private view of the participants and integrity is the public view. The terms private and public are relative depending upon the assumptions about the ownership of various keys. Since, this paper is interested only in the characterisation of the integrity service, the subsequent representations will present only the graphs for the transfer of the integrity service.

While characterising the integrity goal of any system (such as the Schnorr signature scheme, Brands' e-cash scheme), the model will account only for the verification equations. This abstraction is essential to model the *unpredictable behaviour* of the signer. The signer's behaviour is unpredictable because the verifier does not necessarily trust the signer. The behaviour of the verifier is not modeled because it is assumed that the verifiers perform the verifications to safe-guard their interests. Moreover, it is not the concern of cryptology to force the verifier to act properly during and after the verification process.

This section presents a protocol developer's view, as opposed to a cryptologic algorithm developer's view, of the general purpose signature schemes and an informal syntax for the representation of the transfer of service from keys to messages. The results are then extended to represent the Schnorr signature scheme [15] in Section 2.1. Section 2.2 contains a discussion on Schnorr-type blind signature schemes [7,4] and outlines the subtleties that protocol designers must be aware of.

2.1 Characterising Signature Schemes

The following representation for the signature schemes will be employed in this paper:

$$PublicKey \xrightarrow{Ciphertexts} Message$$

The term *Ciphertexts* includes the result of any cryptographic operation, such as encryption and signature operations. For example, if $y = g^x \bmod p$ for a suitable value of p and g, then y is a cipher-text. There may be one or more individual cipher-texts in the system. Usually, the signature process is computationally expensive and the messages are arbitrarily long. Additionally, the use of secure hash functions improve the security of the verification equations. Therefore, suitable message digest (symmetric key) techniques are employed. This gives raise to two techniques.

The first technique is to sign the message digest. Suppose that an RSA public-key pair [13], $[e, n]$, is employed to sign a message, m, employing a secure hash function, H, to generate the following verification equations:

$$c \overset{?}{=} H(m, A)$$
$$r \overset{?}{=} c^e \bmod n$$

then $[c, r]$ are the signature tuples. This technique is represented as follows:

$$^{\square}(A \xrightarrow{c} m) \quad ([e, n] \xrightarrow{r} c)^{\square}$$

where:

1. c is the message digest;

2. A is the symmetric key. When an unkeyed hash function is employed, $A =$, which is the *null key*.
3. m is the message to be signed.
4. $[e, n]$ is the public key of the signer.
5. r is the signature cipher-text.

Henceforth, the logical “and” operation will be represented by the symbol . This operator suggests that individual verification equations must output `true` for the verification system to output `true`. Note that the key represents the no key scenario and is known globally to all participants. Also note the myriad of protocol design possibilities when *SymmetricKey* is not equal to the key.

The second technique is to sign a symmetric key that would provide integrity service to the message. The technique proposed by Fiat and Shamir [9], and adopted by Schnorr [15] is a good example. Such a signature technique is represented as follows:

$$\left(PublicKey \xrightarrow{SignatureCiphertext} SymmetricKey \xrightarrow{MessageDigest} Message \right)$$

The symmetric key, in this case, cannot be (null key). Note that the representation, by itself, does not suggest that the signature cipher-text provides non-repudiation service to the message, rather it suggests the integrity service for the symmetric key, which in turn provides integrity service to the message. This is because the representation deals with a lower level view to trace the flow of integrity service, which is more basic than the non-repudiation service. In order to achieve the non-repudiation service for the message, a one-to-one relationship between the symmetric key and the message, which in the Schnorr signature scheme is achieved by a one-to-one relationship between the signature cipher-text and the message digest, is essential. The rest of this section will explain this form of representation in detail.

A tuple $[r, c]$ is a valid Schnorr signature on a set of messages m by the public key $[g, y, p]$ (henceforth the symbol p, representing the prime number, will be omitted whenever it can be implicitly understood), if the following equation holds:

$$c \overset{?}{=} H(m, A)$$

where, H is a secure hash function, c is the message digest and $A = y^c g^r$ is the symmetric key. The integrity goal of the Schnorr signature scheme can be expressed as follows:

$$\square [g, y] \xrightarrow{[c,r]} A \xrightarrow{c} m \square \qquad (1)$$

That is a *trusted* public key, $[g, y]$, provides integrity service to a symmetric key, A, by employing the cipher-texts, $[c, r]$. The symmetric key, A, in turn provides integrity service to the message, m, by employing the cipher-text c. The same value of the cipher-text, c is employed by the public key and the symmetric key.

It is important to note that in Schnorr-type signature schemes, the structure of A with respect to g, is similar to the structure of y with respect to g. That is by knowing the discrete logarithm, $\log_g y$ and the signature tuple, it is possible to know the value of $\log_g A$, and vice versa. This is an important requirement to prevent the generation of multiple signature transcripts from a single Schnorr signature. Henceforth, the □□ delimiter will separate verification equations from each other.

The proof of equality of discrete logarithms employed by Chaum and van Antwerpen [6] resembles the Schnorr signature. It proves that $\log_g y = \log_v u$ for some u and v. Note that $[g, y]$ or $[u, v]$ must be *trusted* or *certified*. The verification equation for such a scheme is as follows:

$$c \overset{?}{=} H(m, A, B)$$

where,

1. c is the message digest;
2. H is a secure hash;
3. m is the set of messages;
4. $[c, r]$ is the signature cipher-text; and
5. $A = y^c g^r$ and $B = u^c v^r$ are the symmetric keys.

The integrity goal of this scheme can be expressed as follows:

$$^{\square}(([g, y] \overset{[c,r]}{\to} A) \quad ([v, u] \overset{[c,r]}{\to} B)) \overset{c}{\to} m^{\square} \tag{2}$$

The symmetric keys A and B provide integrity service to m. It is crucially important to note that $[g, y]$ or $[v, u]$ *must be certified* (using some private or public certification scheme) before any integrity deductions can be made. The protocol *associates* the integrity of $[g, y]$ (or $[v, u]$) with the integrity of $[v, u]$ (or $[g, y]$). Once this association is made and the *absolute* integrity of at least one of the key tuples is deduced, then the integrity of the symmetric keys $[A, B]$, and thereby the message m, can be deduced. Without certification of any of the keys, no meaningful deductions on the integrity service can be made. Note that this requirement is *inherited* from the Schnorr signature scheme represented in Equation 1.

2.2 Characterising Schnorr-Type Blind Signature Schemes

The blind signature technique [8] allows an entity to obtain a signature tuple on a message from a signer without revealing either the signature tuple or the message. This allows the entity to prove to any other entity that it was authorised by the signer without revealing its identity – the entity is anonymous.

A well known method to obtain blind signature requires the signer to engage in a honest-verifier zero-knowledge identification protocol with the receiver (of the signature), who would play the role of a *skewed honest-verifier* to obtain the blind signature. Chaum and Pedersen [7] demonstrated the technique to obtain a blind Schnorr signature, which was later modified by Brands [4] to obtain a specialised version called restrictive blind signature. The purpose of this section is to characterise both these schemes in order to highlight their subtle and important properties, which are usually ignored by some protocol designers. This oversight introduces many deficiencies in the integrity goal of the resulting cryptosystem.

A Schnorr-type blind signature was first proposed by Chaum and Pedersen [7]. The signature tuple is the same as that of Schnorr signature scheme (see Section 2.1) and has the same signature verification equation. The only difference is that the signer *cannot know* the message that is being signed, which in the case of Schnorr signature is the symmetric key and *not the message itself*. This is a subtle point that should actually mean that the signer is authorising the symmetric key only and *does not necessarily* authorise the message that the symmetric key may provide integrity to – as was the case in the original Schnorr signature scheme. Interestingly, this problem has a counterpart in the key recovery research (and cryptologic research as a whole), where it is a difficult problem to *restrict the use of certified keys* [10].

Since the verification equation for a blind Schnorr signature is the same as the Schnorr signature scheme, this subtlety is introduced in the representation of the integrity goal by employing a *modifier*. This is because the blinding process provides *confidentiality service* and the syntax presented in this paper deals only with the *integrity service*. Since the blinding process does not alter the integrity goal of the protocol, any alteration of the representation of the integrity goal for the Schnorr signature, to introduce the subtlety, must be purely a convention. The best way to accomplish this requirement would be to introduce a modifier. In Equation 1, the message that is signed, m, is represented employing a modifier as $\overline{m}$. Syntactically, Equation 1 is otherwise unchanged. The integrity goal is represented as follows:

$$\square\, [g, y] \xrightarrow{\quad [c, r] \quad} A \overset{c}{\to} \overline{m}\, \square \tag{3}$$

Note that the signature generation procedure may or may not be blinded[2], so the modifier is intended only for the interpretation of a potential weakness in argument. In other words, the modifier is *a statement of intent* and *not of a fact*. In the previous equation, the modifier suggests that the signer *may have no control over* the message, m.

The restrictive blind signature by Brands [4] is similar to the blind Schnorr signature scheme [7], with an additional property that the signer guarantees the *structure* of the symmetric key, A. In the original proposal [4], the signer employs the Schnorr variant (by Chaum and van Antwerpen, see Section 2.1) represented by Equation 2 and guaranteed the representation (structure) of one of the symmetric keys with respect to the bases $[g_1, g_2]$. The verification equations

[2] In the case of an e-cash system the customer could engage in a normal Schnorr signature protocol with the bank, and the merchant cannot discern this fact.

employed by the merchant (during the spending phase) and the bank (during the deposit phase) in Brands' scheme are as follows:

$$
\begin{aligned}
c &= H(A, B, z, a, b)\\
a &\overset{?}{=} g^{r} y^{-c}\\
b &\overset{?}{=} A^{r} z^{-c}\\
d &= H_0(A, B, \cdots)\\
B &\overset{?}{=} g_1^{r_1} g_2^{r_2} A^{-d}
\end{aligned}
$$

where:

1. c and d are message digests;
2. H and H_0 are secure hash functions;
3. $[A, z]$ is a temporary key pair;
4. B is a message;
5. $[a, b]$ is the symmetric key tuple blindly *authorised* by the bank; and,
6. $[g, g_1, g_2, y, y_1, y_2]$ is the public key of the bank such that $y = g^{x_B}$, $y_1 = g_1^{x_B}$ and $y_2 = g_2^{x_B}$, where x_B is the banks private key;
7. $[r, c]$ is the signature tuple by the bank; and,
8. $[r_1, r_2]$ is the signature tuple on B employing the key$[g_1, g_2, A]$.

The integrity goal of this scheme is represented as follows:

$$
\begin{aligned}
&{}^{\square}(([g, y] \xrightarrow{\quad [c, r] \quad} a) \quad ([A, z] \xrightarrow{\quad [c, r] \quad} b)) \overset{c}{\to} \overline{B}^{\square}\\
&{}^{\square}[g_1, g_2, A] \xrightarrow{\quad [r_1, r_2, d] \quad} B \overset{d}{\to} [A, \cdots]^{\square} \qquad (4)
\end{aligned}
$$

It can be read as: the bank authorises the symmetric keys $[a, b]$ using its public key $[g, y]$ and, $[A, z]$ by its *association* with $[g, y]$. The symmetric keys provide integrity service to B (note the use of the modifier as $\overline{B}$ to represent the blind operation). This is the joint statement of the first verification equation. The second verification equation provides integrity service to B by employing the public key $[g_1, g_2, A]$ and the cipher-texts $[r_1, r_2, d]$. B, in turn, provides integrity service to a predetermined set of messages and A. This is not a blinded operation. The implicit assumption for the goal of this proposal is the *association* of the bases $[g_1, g_2]$ with the key A, which was a part of the key $[A, z]$ which was *associated* with $[g, y]$ by the blind signature process. Thereby, whoever possessed the signature (the first verification equation) must also possess the knowledge of the representation of A with respect to the base $[g_1, g_2]$ (just as the Schnorr signature scheme required the signer to possess the representation of the public key y with respect to the base g), and therefore the representation of B. This additional check allowed the bank (which took part in the signature generation process) to gain another implicit confidence: the blind signature transcript contains a valid, *hidden* identity that is a representation of the bases $[g_1, g_2]$. In the case of electronic cash systems employing blind signature, the merchant, without trusting the bank, *cannot* gain this knowledge as it can make no logical deductions about the withdrawal protocol (signature generation process).

3 Analysis of an Efficient E-Cash Proposal

Electronic cash systems, like physical cash systems and unlike electronic payments systems like credit cards, allow the users to anonymously spend legitimate amounts of currency. The anonymity property is mutually exclusive of the properties for tracing transactions.

This section will analyse the e-cash proposal of Radu, Govaerts and Vandewalle [12]. The proposal is a three-phased withdrawal mechanism presented briefly as follows:

1. *get_pseudonym* protocol between the user and the bank to obtain a restrictive blind signature on a pseudonym, π, by employing the Brands withdrawal protocol (see Equation 4). This allows the bank to guarantee that the pseudonym π is derived from a registered identity π_0.
2. *withdraw_big_coin* protocol between the user and the bank allows the user to obtain a blind Schnorr signature (see Equation 3) on a *big coin* that associates a pseudonym, β with a valid long-term pseudonym π; and,
3. *exchange_big_coin* protocol between the user and the bank that allows the user to *anonymously* withdraw many *small coins* after providing the bank with a valid *big coin* and the corresponding long term pseudonym π.

The user can spend the *small coins* with any merchant. Radu *et al.* proposed the use of a smart-card during the spending protocol that will act as an observer to prevent double spending of small coins (refer to the paper by Chaum and Pedersen [7] for a detailed discussion on this topic). The certified public keys of the bank is represented by the tuple, $[g, P, P_1]$ such that the bank possesses the representation of P and P_1 to the base g.

As stated previously in Section 2.2, a blind signature must be considered as an authorisation for a symmetric key and not for the message that could be serviced by the symmetric key. Radu *et al.* did not observe this caution in their proposal for an efficient e-cash. As will be shown, this oversight results in a weakness in their proposal that allows unaccounted transfer of funds between accounts, that is the property of non-transferability is not achieved.

The verification equations that the bank employs to verify the long-term pseudonym during the *exchange_big_coin* phase are as follows:

$$\begin{aligned} c &= H(\pi, z, A, B) \\ A &\stackrel{?}{=} g^r P^c \\ B &\stackrel{?}{=} \pi^r z^c \\ d &= H(\beta, \alpha) \\ \alpha &\stackrel{?}{=} g_1^{r_1} g_2^{r_2} \pi^d \end{aligned}$$

These are the verification equations of Brands' restrictive blind signature scheme discussed in Section 2.2. The representation for the verification of long-term pseudonym component of the big-coin is as follows:

$$^{\square}(([g,P] \xrightarrow{[c,r]} A) \quad ([\pi,z] \xrightarrow{[c,r]} B)) \overset{c}{\to} \overline{[\pi,z]}^{\square}$$

$$^{\square}[g_1,g_2,\pi] \xrightarrow{[r_1,r_2,d]} \alpha \overset{d}{\to} [\beta]^{\square} \qquad (5)$$

In the Brands' scheme, the symmetric key α (B in Equation 4) was serviced by c, which restricted the use of α to only one servicing – otherwise the private key of the user would be revealed (a deficiency of Schnorr-type signature schemes). Whereas, in the scheme proposed by Radu *et al.*, the symmetric key α was not serviced by c. Thereby the value for α can be changed (mutable) to allow for multiple servicing of multiple values of β by π.

The verification equation that the bank employs to verify the big coin during the *exchange_big_coin* phase are as follows:

$$e = H(\beta,\pi,D)$$
$$D \overset{?}{=} g^{r_3}P_1^e$$

This is a blind Schnorr signature explained in Section 2.2 by Equation 3. The representation for the verification of the short-term pseudonym (β) component of the big-coin is as follows:

$$^{\square}[g,P_1] \xrightarrow{[e,r_3]} D \overset{e}{\to} \overline{[\beta,\pi]}^{\square} \qquad (6)$$

Note that the claimed association between a long term pseudonym, π, and the short term pseudonym, β, happens during this protocol. Also, note the modified term, $\overline{[\beta,\pi]}$, which suggests that the signer (the bank) with the public key $[g,P_1]$ *can have no control over* the values $[\beta,\pi]$.

Radu, Govaerts and Vandewalle analysed $[e,r_3]$ as a *signature* on $[\beta,\pi]$ by the key pair $[g,P_1]$, the certified public key of the bank. Therefore, they argued that association was authorised by the bank. The flaw in this argument is: $[e,r_3]$ is a *blind signature* on $[\beta,\pi]$. Referring to equation 6, clearly the integrity check relies on the *use of the key*, D, which was authorised by the bank, to associate the tuple $[\beta,\pi]$ and this problem is similar to the generic situation explained in Section 2.2. That is, the bank is *trusting* the user to correctly associate one of his/her long-term pseudonyms, π, with a short-term pseudonym, β. This allows the user to associate the π value of another user with the β value that resulted from his/her withdrawal. In effect, this would allow *unaccounted* money transfer between users, which may result in perfect black-mailing and/or money-laundering [18]. Although Radu *et al.* did not comment about the property of *non-transferability*[3] in their paper, many practical monetary systems require this property for their proper functioning. Therefore, their scheme lacks the *non-transferability* property, primarily due to the lack of consistent integrity checks.

In order to visualise this problem let the long-term pseudonym of a blackmailer be π, which was derived from his/her long term identity π_0 using the

[3] The property which is essential to prevent unaccounted transfer of funds.

get_pseudonym protocol. The black-mailer can perform the following actions to achieve a perfect-blackmail;

1. allow the victim user to participate in the mutual-authentication protocol that takes place before the *withdraw_big_coin* transaction;
2. logically or physically hijack the withdrawal terminal from the victim user to prevent him/her from registering the value of π; and,
3. perform the *withdraw_big_coin* transaction with the bank as prescribed by the protocol, employing π as the pseudonym.

4 Analysis of the Binding ElGamal Proposal

Key recovery infrastructures aim to provide *restricted* confidentiality channel for users communications. The confidentiality property of the channel is restricted because, unlike the traditional key establishment systems, the messages communicated by the users can be *accessed* or *wire-tapped* by *authorised* entities called escrow agents. Such systems were primarily motivated by the needs of law enforcement agencies.

Verheul and van Tilborg [17] proposed a fraud detectable key recovery scheme. The proposal was aimed to allow any third party to verify if a sender has encrypted the session key value to the receiver and the escrow agents. The verification equations, which were proposed to detect activities that could by-pass the key-recovery infrastructure, were:

$$\begin{aligned} c &= H(E, C, R_A, R_B, R_M, D, F, I, \cdots) \\ D &\overset{?}{=} g^c C^r \\ F &\overset{?}{=} (y_A/y_M)^c (R_A/R_M)^r \\ I &\overset{?}{=} (y_B/y_M)^c (R_A/R_M)^r \end{aligned} \tag{7}$$

where: H is a secure hash function, $[c, r]$ is a Schnorr signature tuple. This check was aimed to show that the message encrypted in $R_A = Sy_A^k$ and $R_M = Sy_M^k$ $(C = g^k)$ is the same, *without revealing the message*.

Using the notation presented in Equation 2, Section 2.1, the following representation for the verification equations of the key recovery scheme can be determined:

$$\begin{aligned} &{}^{\square}(([g, C] \overset{[c,r]}{\rightarrow} D) \\ &([y_A/y_M, R_A/R_M] \overset{[c,r]}{\rightarrow} F) \\ &([y_B/y_M, R_B/R_M] \overset{[c,r]}{\rightarrow} I)) \overset{c}{\rightarrow} [E, C, R_A, R_B, R_M, \cdots]^{\square} \end{aligned} \tag{8}$$

Note that none of the key pairs ($[g, C]$, $[y_A/y_M, R_A, R_M]$ $[y_B/y_M, R_B/R_M]$) providing integrity service are certified. It is evident that this representation is similar to the representation provided in Equations 1 and 2. By comparing the above representation with Equations 1 and 2, the following observations can be made:

1. none of the key pairs ($[g, C]$, $[y_A/y_M, R_A/R_M]$ and $[y_B/y_M, R_B/R_M]$) can be trusted because they are uniformly chosen by the sender (who is not trusted for certification procedures);
2. ratios of keys provide the integrity service to the symmetric keys F and I, which is not a *standard* assumption of Schnorr-type signatures.

These observations suggest a deficiency in the system that allows the sender to manipulate the keys, which were meant to be the *starting point* of the integrity service – that is if the starting point is corrupted then the integrity service that it transfers is also corrupted. This weakness in the integrity service could potentially result in attacks on the protocol, like the attack to be presented in this section.

Prior to discussing an attack on the key recovery system, the meaning of a non-trivial attack must be understood. A key recovery protocol is deficient if successful adversaries abide with the message formats suggested by the protocol and *procure legitimate services from the key recovery infrastructure* to ensure secure communication. For example, if a public-key based key recovery system provides robust certification mechanism, such as robust public key infrastructures, and requires key recovery enablement before the certification can be employed, then an adversary is successful when certified public keys are employed and key recovery is avoided. The attack on the proposal, by Verheul and van Tilborg [17], by Pfitzmann and Waidner [11] need *not necessarily* be an attack on the protocol proposed by Verheul and van Tilborg, rather it is an attack on all session-key recovery systems without any form of private-key recovery. It outlines the *generic* concealed-encryption attack[4] on key recovery protocols and *fails* to explain the manner in which the *concealed key* may be established. Although the attack proposed in this section exploits the property of concealed-encryption attack, it is not a generic attack on all session-key recovery protocols, rather it is a specialised attack on the proposal [17], *which resulted from an oversight in the protocol design*. Moreover, this section will detail the manner in which an illegal session key can be established using *the key recovery infrastructure*. This distinction is important for protocol designers, who may employ the proposed fraud detection mechanism [17] for a different application that may not have properties similar to that of key-recovery applications. For example, refer to the paper by Abe [2], which successfully employed a similar integrity verification mechanism for a mix network proposal.

Suppose that the sender and a hidden receiver ($\tilde{M}$) would like to communicate using the actual receiver (M) as the decoy. The sender can accomplish this by employing the following steps:

1. Choose a random session key, $\tilde{S}$.
2. Encrypt the message with $\tilde{S}$ to obtain the cipher-text, E.
3. Obtain the public keys of the hidden receiver, y_H, the decoy, y_M and the authorities (y_A, y_B).

[4] There is no technique available to check if a claimed key was used during the encryption process — verifiable encryption for symmetric key systems is not currently available

4. Choose a random value for k.
5. Compute a decoy session key, $S = \tilde{S} y_H^k / y_M^k$.
6. Encrypt the decoy session key for the decoy and the authorities, $R_M = S y_M^k = \tilde{S} y_H^k$, $R_A = S y_A^k$, $R_B = S y_B^k$ and $C = g^k$.
7. Form the verification equation as suggested by the representation in Equation 8.
8. Send the cipher-texts and verification parameters to decoy.

The hidden receiver performs the following steps:

1. Wiretap the communication to decoy to obtain E, R_M and C.
2. Obtain session key, $\tilde{S} = R_M / C^{x_H}$, where x_H is the private key of the hidden receiver.
3. Decrypt E using $\tilde{S}$ to obtain the message.

The monitor will verify the equations properly, the decoy receiver and the authorities will retrieve the decoy session key, S, from the respective cipher-texts employing the respective private keys and, the decoy session key, S, will not decrypt E correctly. Also note that it will be difficult to find the hidden receiver, y_H, or the actual session key, $\tilde{S}$ (finding the hidden receiver would imply breaking of the multi-ElGamal cryptosystem proposed in the paper [17]).

5 Conclusion

The paper presented a novel technique to represent the integrity goal of a system by accounting for all the verification equations and ignoring the unnecessary protocol complexities that produced the equations. An abstraction to encompass the *unpredictability* of the protocol participants was also proposed. The use of the technique was demonstrated by the identification of *similar* protocol deficiencies in *seemingly* different scenarios.

Many proposals for *compliant* systems tend to ignore the importance of the integrity service, while in pursuit of the confidentiality service. Blaze [3] formulated an attack on the integrity service in the Clipper proposal [16], which was predominantly focused on the confidentiality service. Unfortunately, many protocols in various fields of cryptologic application still succumb to attacks similar to those detailed in Sections 3 and 4, namely attacks exploiting weaknesses in integrity services. In order to design robust and secure protocols the integrity and the confidentiality services must be carefully designed and integrated.

Prospective formal syntax that can represent precisely both the confidentiality and the integrity goals will greatly improve protocol logic development. Research for such a syntax will be very useful, both theoretically and practically.

References

1. Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *IFIP International Conference on Theoretical Computer Science (IFIP TCS2000), Sendai, Japan*, 2000. To appear.

2. Masayuki Abe. Mix-networks on permutations networks. In K. Lam, E. Okamoto, and C. Xing, editors, *Advances in Cryptology – ASIACRYPT'99*, volume 1716 of *LNCS*, pages 258–273. Springer-Verlag, 1999.
3. Matt Blaze. Protocol failure in the escrowed encryption standard. In *The 2nd ACM Conference on Computer and Communications Security*, November 1994.
4. Stefan Brands. Untraceable Off-line Cash in Wallet with Observers. In Tor Helleseth, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *LNCS*, pages 344–359. Springer-Verlag, 1993.
5. M. Burrows, M. Abadi, and R. M. Needham. A logic of authentication. In *Proceedings of the Royal Society of London*, volume 426, pages 233–271, 1989.
6. D. Chaum and H. van Antwerpen. Undeniable signatures. In G. Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *LNCS*, pages 212–216. Springer-Verlag, 1989.
7. David Chaum and T. Pedersen. Wallet Databases with Observers. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *LNCS*, pages 89–105. Springer-Verlag, 1992.
8. David Chaum. Blind Signatures for Untraceable Payments. In Sherman A.T. Chaum D., Rivest R.L., editor, *Advances in Cryptology – CRYPTO'82*, pages 199–203. Plenum Press, 1983.
9. A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer-Verlag, 1986.
10. Lars R. Knudsen and Torben P. Pedersen. On the difficulty of software key escrow. In U. M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *LNCS*, pages 237–244. Springer-Verlag, 1996.
11. Birgit Pfitzmann and Michael Waidner. How to break fraud-detectable key recovery. *Operating Systems Review, ACM press*, 32(1):23–28, January 1998.
12. Cristian Radu, René Govaerts, and Joos Vandewalle. Efficient electronic cash with restricted privacy. In Rafael Hirschfeld, editor, *Financial Cryptography, FC'97*, volume 1318 of *LNCS*, pages 24–28. Springer-Verlag, 1997.
13. Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
14. Rainer A. Rueppel. A formal approach to security architectures. In Donald W. Davies, editor, *Advances in Cryptology – EUROCRYPT'91*, volume 547 of *LNCS*, pages 387–398. Springer-Verlag, 1991.
15. C.P. Schnorr. Efficient signature generation for smart cards. *Journal of Cryptology*, 4:161–174, 1991.
16. U.S. DEPARTMENT OF COMMERCE / National Institute of Standards and Technology. *Federal Information Processing Standard 185—Escrowed Encryption Standard*, February 1994.
17. Eric R. Verheul and Henk C.A. van Tilborg. Binding ElGamal: A fraud-detectable alternative to key-escrow proposals. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *LNCS*, pages 119–133. Springer-Verlag, 1997.
18. B. von Solms and D. Naccache. On Blind Signatures and perfect crimes. *Computers and Security*, pages 581–583, October 1992.

Cryptanalysis of the Nonlinear FeedForward Generator

S.S. Bedi and N. Rajesh Pillai

Scientific Analysis Group, DRDO, Delhi 110054

Abstract. The nonlinear feedforward generator is one of the commonly used building blocks of stream ciphers. This paper describes a novel known-plaintext attack for cryptanalyzing nonlinear feedforward generator. The plaintext requirement of the attack is only twice the length of the shift register. The implementation of this attack could identify the initial settings of the system for a 128 stage register and randomly chosen nonlinear feedforward function of 10 variables in few minutes on a P-II 300 MHz machine.

1 Introduction

The nonlinear feedforward generator is one of the commonly used building blocks of stream ciphers. This paper describes a new technique for cryptanalyzing feedforward generator. Given just 2n bits of the plain-text, where n is the length of shift register, the attack determines the initial setting of the shift register. The basic idea is to form a system of Boolean equations describing relationship between the keysequence (obtained by xoring crypt and the known plain text), and the initial settings. We then use the techniques developed by Zakrevskij & Vasilkova [8] for solving a system of nonlinear equations.

The proposed attack is very efficient and in most of the cases we got results within few minutes. We believe that this attack can be extended to attack other building blocks of stream ciphers also.

The Nonlinear feedforward generator is made up of a linear feedback shift register (LFSR) and a nonlinear Boolean function. The shift register is allowed to run (with its feedback polynomial deciding the bit to be shifted in). The nonlinear feedforward function (NLFF) takes some of the bits of the LFSR as input and calculates the output bit. The location in LFSR from where the input bit for NLFF is picked is called a tappoint or tap for short. Figure.1 shows a block diagram of a Nonlinear feedforward generator.

It is known (corollary 5.6 of [4]) that the lower bound for Linear complexity (LC) of NLFF Generator can be expressed in terms of L, the length of shift register, k, the number of taps, under suitable conditions as $LC \geq {}^{n}C_{k}$.

For L = 50, k = 8, we have $LC \geq {}^{50}C_{8} = 536878650 \approx 5.4\ 10^{8}$

In this paper we will be dealing with *feedforward functions with equidistant taps*.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 188–194, 2001.

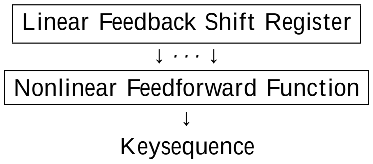

Linear Feedback Shift Register
↓ · · · ↓
Nonlinear Feedforward Function
↓
Keysequence

Fig. 1. Nonlinear Feedforward Generator

2 Attacks on Nonlinear FeedForward Generator

We summarize the few existing known results in this area. Correlation attacks are the most common class of attacks applied on stream ciphers. See [5] for overview of these type of attacks. Correlation attacks work when there is correlation between output sequence and input sequence to the nonlinear function. Anderson [1] describes how to pick up subsequences with optimum correlation properties for mounting the correlation attack on feedforward generators for known plaintexts. Correlation attacks work for ciphertext-only attack also. They need ciphertext length dependent on the value of correlation. Typical ciphertext length required would vary from 4000 bits (very optimistic case) to order of 10^6.

Among the known-plaintext attacks, the generalized inversion attack [2], [3] is very effective in cryptanalyzing nonlinear filter generators. It has a complexity which is exponential in the distance(d) between first and last tap points of the feedforward function. The generalized inversion attack will be practical if the feedforward function has its tap points bunched together or if it can be converted to this form (say by uniform decimation) This attack is based on theory of branching processes.

Our attack uses a totally different approach to this problem. We use the description of the system to setup nonlinear equations capturing relations between initial settings (unknown) and the bits of the output sequence (known part). Then we solve the system for the unknown initial variables.

3 Solving Nonlinear Boolean Equations

Solving an arbitrary set of Boolean equations (also known as the satisfiability problem) is an NP-Complete problem. Currently only exponential time algorithms are available for the general case. But if the system of equations is of a special kind, solutions can be obtained efficiently. We have used the technique of local reduction developed by Zakrevskij and Vasilkova [8] in our work. Local Reduction technique can solve large systems of nonlinear boolean equations efficiently when number of variables in each equation is small, (typically less than 10) and there are equations having large overlaps in their sets of variables.

3.1 Local Reduction Technique of Solving System of Nonlinear Equations

Given a system of nonlinear Boolean equations of the form $F_i = 1$, the local reduction technique takes two equations at a time and reduces the number of points in the solution space to be tried out.

Representation of the equations: For each equation, system will store the variables in the equation and the onset. (Onset of an equation is the set of points at which the equation is satisfied.) For e.g. $xy \quad yz = 1$ will be represented as the set of (binary) numbers $\{110, 011\}$ over the variables (x, y, z). This means that the equation is satisfied when $x = 1, y = 1, z = 0$ and when $x = 0, y = 1, z = 1$ and at no other points.

Given two equations, whose variable sets have a nonempty intersection, say

$$Eq1 : varlist_1 = (a, b, c, d, e), onset_1 = \{01101, 11010, 10011\}$$
$$Eq2 : varlist_2 = (c, d, e, f, g, h), onset_2 = \{101110, 001101, 010010\}$$

List of elements common to $varlist_1$ and $varlist_2 = (c, d, e) = \varphi$

From Eq1 we can infer that the 3-tuple (c,d,e) is 101 or 010 or 011 whenever Eq1 is true. The set $S_1 = \{101, 010, 011\}$ is called the projection of the onset of Eq1 on (c, d, e). From Eq2 we can infer that the 3-tuple (c, d, e) is 101 or 001 or 010 whenever Eq2 is true. We shall call $S_2 = \{101, 001, 010\}$ as projection of onset of Eq2 on (c, d, e). A correct solution to the system satisfies both the equations simultaneously. So all correct solutions to the system should have the 3-tupe (c,d,e) = 101 or 010 (a value from intersection of S_1 and S_2). Using this information we can delete all those elements from the $onset_1$ and $onset_2$ whose projection on (c, d, e) is not in the set $S_1 \cap S_2$ Using this reduction we can get an equivalent system of equations (i.e. solution set of the new system of equations is the same as the solution set for the old system). Applying this reduction on our system we get

$$NewEq1 : varlist_1 = (a, b, c, d, e), onset_1 = \{01101, 11010\}$$
$$NewEq2 : varlist_2 = (c, d, e, f, g, h), onset_2 = \{101110, 010010\}$$

We execute this operation sequentially on pairs where it can be applied. The procedure terminates when either some onset becomes empty (case when set of equations is inconsistent) or some reduced set of functions is obtained where the given operation cannot be applied on any pair. This process of reducing the size of onsets by considering a pair of equations at a time is called local reduction. Once a reduced set of functions is obtained, we can combine the onsets of each equation to get onset for the whole system. (Combining is done by doing a 'join' operation over the onsets.) For example, in the above case the solution set for the system of equations will be

$$(a, b, c, d, e, f, g, h) = \{01101110, 11010010\}$$

This complexity of local reduction is proportional to number of points in the onset or roughly exponential in number of variables per equation, which is why number of variables in each equation is to be kept small.

3.2 Removal of Common Factors

We also used the technique of removal of common factors. The basic idea behind this is very simple. Since the nonlinear equations are of the type $F_i = 1$, if some sub-term occurs as a common factor in all the terms of F_i, it can be factored out. In our case suppose we have

$$Eq_i : varlist_i = (a, b, c, d, e), onset_i = \{10100, 10101, 10110\}$$

We see that the triple (a,b,c) = (101) in all the points of the onset, so we can infer that a=1, b=0 and c=1 in the solution for the whole system of equations. We can substitute the values inferred to get reduced system of equations. In other words we are factoring out $a\bar{b}c$ and substituting the value in other equations. This operation of factoring out is linear in size of onset or roughly exponential in number of variables in the equation.

4 Description of the Algorithm

4.1 Making Equations

We try to make equations expressing relation between initial contents of the shift register and the bits of key sequence. Using system description such equations can be easily made. The initial contents will be the variables and the key sequence bits will be the constants. To ensure that the system of equations satisfy the criteria for Local reduction to be applicable, we had to make number of variables per equation less than 10.

This was achieved by introducing new variables for the bits generated by the LFSR. A set of linear equations based on the feedback polynomial giving relation between these new variables and the initial variables is added to the system of equations.

Given

1. Feedback polynomial of degree n,
2. Feedforward function of m variables and its tap points. Without loss of generality we assume that the tap points of the nonlinear function are in the last m consecutive stages of the shift register. (as taps are equidistant)
3. 2n consecutive bits of output sequence,

Our system of equations will be as follows:

- Variables involved : $x_1, ...x_{2n+m}$ corresponding to the 2n+m bits generated by LFSR while producing 2n bits of output sequence.
- 2n nonlinear equations, each equation over m variables, describing output bit in terms of m bits of the LFSR. These equations are formed using the feedforward function. We represent these equations in the form of onsets.

– n+m linear equations expressing each of the bits x_{n+1} to x_{2n+m} in terms of the previous n bits. These equations are formed using feedback function of LFSR. We represent these equations in the Algebraic Normal Form.

Representing the linear equations also as onsets would bring uniformity of representation to the system but would constrain the feedback polynomial to have a low density (number of taps less than 10).

4.2 Solving the System of Equations

In our case we had a system of nonlinear equations (2n equations), and a system of linear equations (n+m equations) over the same set of variables (2n+m variables). Each nonlinear equation had 10 or fewer variables. There was no constraint over the set of linear equations. We applied local reduction to the set of nonlinear equations. In case some onsets get reduced to singleton sets, we get the values of all the variables in that equation. For e.g.

$$varlist_i = (a, b, c), onset_i = \{011\}$$

Means that correct solution to the system has a=0, b=1 and c=1.

We substitute the values in both the nonlinear and linear system of equations. In case some linear equation gets reduced to an equation over a single variable, we can infer the value of the variable and substitute it back into the system of equations. We repeat this process local reduction followed by substitution again over the reduced system of equations. When no further reductions are possible, (and system is not yet solved) the reduced system of equations is saved and we perform tree search over the reduced onset space of the system. For this we first pick the equation with least number of points in the onset and then substitute values corresponding to each point in the onset. The new system obtained by the substitution is passed through the same process of local reduction followed by substitution. In case the substitution was correct, we get closer to our solution otherwise the system of equations leads to a contradiction. In that case we undo the substitution and try the next possible substitution. At every stage before applying local reduction we apply factoring out of common terms. The outline of this algorithm in pseudo code is given in Figure 2.

5 Results

The results of applying our algorithm to systems of nonlinear feedforward generators of different parameters are given in Table 1. The results have been obtained on P-II 300MHz system with 64MB RAM.

6 Conclusions

A novel method of cryptanalysis of nonlinear feedforward generator has been described. This method is a known-plaintext attack and works even when just

```
Input:  Feedback polynomial f of degree n,
        Feedforward function g of m variables and
        First 2n bits of the key sequence.
Output: The initial content of the Shift register

Making equations:
The first 2n+m bits of the sequence generated by LFSR are
considered as variables. We form two sets of equations.
* n+m linear equations are formed based on f, describing the
  relationship between the 2n+m variables.
* 2n nonlinear equations describing the given key sequence in
  terms of the variables
The n+m linear equations are represented in ANF
The nonlinear equations are taken in the form Fi = 1
Internal representation of the nonlinear equation is in the
form of sets. We store onsets, i.e. set of points at which
Fi evaluates to 1.

Pseudocode of the algorithm for solving the system of equations: -

numvarfound =0    /* number of variables whose values are found */
WHILE numvarfound < n
  Apply removal of common factors, local reduction on set of
     nonlinear eqns.
  WHILE some variables are found DO
    Substitute the values found in all equations
    Apply removal of common factors, local reduction on set of
        nonlinear eqns
  END WHILE
  IF  numvarfound > n  BREAK /* tree search over reduced eqns */
  Search for equations with least number of points in the onset.
  Try Substituting values as given by each point one by one.
    Check for consistency.
    IF not consistent, undo substitution.
    ELSE try solving the reduced system of equations
END WHILE

Make linear equations expressing the variables found in terms
of first n variables.
Solve system of the linear equations to get values of
the first n variables.
OUTPUT value of first n variables
```

Fig. 2. Algorithm for Cryptanalysis of Nonlinear feedforward generator

$2n$ bits of the keysequence is available, where n is the length of the shift register. The implementation took few minutes to find initial settings of the system with 128-degree polynomial and 10 variable feedforward function.

Table 1. Experimental Results

Number of taps for NLFF	Degree of Polynomial	Time taken
8	64	5.3s
8	64	6.5s
8	71	10.9s
8	128	29.5s
10	128	64.9s

The basic idea of the attack is to express the generator using a set of nonlinear boolean equations of a certain form so that they can be solved efficiently. We expressed the output of the feedforward generator using a system of equations with limited number of variables per equation and lot of common variables between equations. Zakreviskij's method was then used to solve the resulting system of Boolean equations.

We believe that this method can be extended to attack other building blocks of cryptosystems also.

Acknowledgements

The authors would like to thank Prof. A D Zakreviskij for kindly giving a copy of his paper and Prof. C E Veni Madhavan for useful discussions.

References

1. Ross Anderson: Searching for the Optimum Correlation Attack. Proc. Fast Software Encryption -Leuven '94, B. Preneel, ed., 1995
2. Jovan Dj. Golic: On the Security of Nonlinear Filter Generators FAst Software Encryption – Cambridge '96, D. Gollmann, ed., 1996.
3. Jovan Dj. Golic, Andrew Clark, Ed Dawson: Generalized Inversion Attack on Nonlinear Filter Generators IEEE Transactions on Computers, Vol. 49, No. 10, October 2000.
4. Rainer A. Rueppel: Analysis and Design of Stream Ciphers. Springer Verlag Communication and Control Engineering Series 1986.
5. Gustavus J. Simmons (Ed.): Contemporary Cryptology- The Science of Information Integrity. IEEE Press 1992.
6. Arkadij .D. Zakrevskij: Solving system of logical equations by the method of local reduction. Doklady NAN B, 1999, v. 43, No. 5, pp. 5-8. (in Russian).
7. Arkadij .D. Zakrevskij, Irina Vasilkova: Cryptanalysis of the Hagelin Machine by the method of spreading of constants Proc. of Third International Conference of Computer Aided Design of Discrete Devices (CAD DD 99) Minsk, November 10-12 (1999), Vol. 1, pp. 140-147.
8. Arkadij .D. Zakrevskij, Irina Vasilkova: Reducing Large Systems of Boolean Equations. Fourth International Workshop on Boolean Problems, 21-22 Sep. (2000).

Analysis of the GHS Weil Descent Attack on the ECDLP over Characteristic Two Finite Fields of Composite Degree

(Extended Abstract)

Markus Maurer, Alfred Menezes, and Edlyn Teske

Dept. of C&O, University of Waterloo, Canada,
{m2maurer,ajmeneze,eteske}@uwaterloo.ca

Abstract. We analyze the Gaudry-Hess-Smart (GHS) Weil descent attack on the elliptic curve discrete logarithm problem (ECDLP) for elliptic curves defined over characteristic two finite fields of composite extension degree. For each such field $\mathbb{F}_{2^N}$, $N \in [160, 600]$, we identify elliptic curve parameters such that (i) there should exist a cryptographically interesting elliptic curve E over $\mathbb{F}_{2^N}$ with these parameters; and (ii) the GHS attack is more efficient for solving the ECDLP in $E(\mathbb{F}_{2^N})$ than for any other cryptographically interesting elliptic curve over $\mathbb{F}_{2^N}$.

1 Introduction

Let E be an elliptic curve defined over a finite field $K = \mathbb{F}_{2^N}$. The elliptic curve discrete logarithm problem (ECDLP) in $E(K)$ is: given E, $P \in E(K)$, $r = \mathrm{ord}(P)$ and $Q \in \langle P \rangle$, find the integer $\lambda \in [0, r-1]$ such that $Q = \lambda P$. We write $\lambda = \log_P Q$. The ECDLP is of interest because its apparent intractability forms the basis for the security of elliptic curve cryptographic schemes.

The elliptic curve parameters have to be carefully chosen in order to circumvent some known attacks on the ECDLP. We say that an elliptic curve E over $\mathbb{F}_{2^N}$ is *cryptographically interesting* if: (i) $\#E(\mathbb{F}_{2^N})$ is almost prime—that is, $\#E(\mathbb{F}_{2^N}) = rd$ where r is prime and $d \in \{2, 4\}$—in order to avoid the Pohlig-Hellman [21] and Pollard's rho [22,19] attacks; and (ii) r does not divide $2^{Nj} - 1$ for each $j \in [1, J]$, where J is large enough so that it is computationally infeasible to find discrete logarithms in $\mathbb{F}_{2^{NJ}}$—in order to avoid the Weil pairing [17] and Tate pairing [7] attacks.

Frey [6] first proposed using Weil descent as a means to reduce the ECDLP in elliptic curves over $\mathbb{F}_{2^N}$ to the discrete logarithm problem in an abelian variety over a proper subfield $\mathbb{F}_{2^l}$ of $\mathbb{F}_{2^N}$. Frey's method, which we refer to as the *Weil descent attack methodology*, was further elaborated by Galbraith and Smart [9]. In 2000, Gaudry, Hess and Smart (GHS) [11] showed how Frey's methodology could be used (in most cases) to reduce any instance of the ECDLP to an instance of the discrete logarithm problem in the Jacobian of a hyperelliptic curve over $\mathbb{F}_{2^l}$. Since subexponential-time algorithms for the hyperelliptic curve discrete

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 195–213, 2001.

logarithm problem (HCDLP) are known, this could have important implications to the security of elliptic curve cryptographic schemes.

The GHS attack was analyzed in [11,18]. It was proven to fail for *all* cryptographically interesting elliptic curves over $\mathbb{F}_{2^N}$, where $N \in [160, 600]$ is prime. Namely, the hyperelliptic curves C produced either have genus too small (whence $J_C(\mathbb{F}_2)$ is too small to yield any non-trivial information about the ECDLP in $E(\mathbb{F}_{2^N})$), or have genus too large ($g \geq 2^{16} - 1$, whence the HCDLP in $J_C(\mathbb{F}_2)$ is infeasible). The purpose of this paper is to investigate the applicability of the GHS attack on the ECDLP for cryptographically interesting elliptic curves over $\mathbb{F}_{2^N}$ for composite $N \in [160, 600]$.

The paper is organized as follows. §2 provides a brief introduction to the relevant theory of hyperelliptic curves. The GHS Weil descent attack is outlined in §3, and an overview of the best methods known for solving the ECDLP and HCDLP is given in §4. Our analysis of the applicability of the GHS attack on the ECDLP over characteristic two finite fields of composite extension degree is presented in §5 and the Appendix. Our conclusions are stated in §6.

2 Hyperelliptic Curves

Hyperelliptic Curves. Let $k = \mathbb{F}_q$ denote the finite field of order q. The *algebraic closure* of $\mathbb{F}_q$ is $\bar{k} = \bigcup_{n \geq 1} \mathbb{F}_{q^n}$. A *hyperelliptic curve C of genus g over k* is defined by a non-singular equation $v^2 + h(u)v = f(u)$, where $h, f \in k[u]$, $\deg f = 2g + 1$, and $\deg h \leq g$. Let L be an extension field of k. The set of *L-rational points* on C is $C(L) = \{(x, y) : x, y \in L,\ y^2 + h(x)y = f(x)\} \cup \{\infty\}$. The *opposite* of $P = (x, y) \in C(L)$ is $\tilde{P} = (x, -y - h(x))$; we also define $\tilde{\infty} = \infty$. Note that $\tilde{P} \in C(L)$. Except for the case $g = 1$ (since a genus 1 hyperelliptic curve is precisely an elliptic curve), there is no natural group law on the set of points $C(L)$. Instead, one considers the Jacobian of C over k.

Jacobian of a Hyperelliptic Curve. The set D^0 of *degree zero divisors* of C is the set of formal sums $\sum_{P \in C(\bar{k})} m_P P$, where $m_P \in \mathbb{Z}$, $\sum m_P = 0$, and only a finite number of the m_P's are non-zero. D^0 is a group under the addition rule $\sum m_P P + \sum n_P P = \sum (m_P + n_P) P$. Let $\sigma : \bar{k} \to \bar{k}$ be the *Frobenius map* defined by $x \to x^q$. The map σ extends to $C(\bar{k})$ by $(x, y) \to (x^\sigma, y^\sigma)$ and $\infty^\sigma \to \infty$, and to D^0 by $\sum m_P P \to \sum m_P P^\sigma$. The set of zero divisors defined over k is $D_k^0 = \{D \in D^0 : D^\sigma = D\}$. The *function field* of C over k, denoted $k(C)$, is the field of fractions of the integral domain of polynomial functions $k[u, v]/(v^2 + h(u)v - f(u))$. For $f \in k(C)$, the *divisor of f* is $\mathrm{div}(f) = \sum_{P \in C(\bar{k})} v_P(f) P$, where $v_P(f)$ denotes the multiplicity of P as a root of f. Now the set $\mathrm{Prin}_k = \{\mathrm{div}(f) : f \in k(C)\}$ is a subgroup of D_k^0. The *Jacobian* of C (over k) is the quotient group $J_C(k) = D_k^0 / \mathrm{Prin}_k$.

Properties of the Jacobian. $J_C(k)$ is a finite group. A theorem of Weil's implies that $(\sqrt{q} - 1)^{2g} \leq \#J_C(k) \leq (\sqrt{q} + 1)^{2g}$. If D_1 and D_2 are in the same equivalence class of divisors in $J_C(k)$ we write $D_1 \sim D_2$. Each equivalence class has a unique divisor in *reduced form*, i.e., a divisor $\sum_{P\neq\infty} m_P P - (\sum_{P\neq\infty} m_P)\infty$ satisfying (i) $m_P \geq 0$ for all P; (ii) if $m_P \geq 1$ and $P \neq \tilde{P}$, then $m_{\tilde{P}} = 0$; (iii) $m_P = 0$ or 1 if $P = \tilde{P}$; and (iv) $\sum m_P \leq g$. Such a *reduced divisor* D can be uniquely represented by a pair of polynomials $a, b \in k[u]$ where (i) $\deg b < \deg a \leq g$; (ii) a is monic; and (iii) $a|(b^2 + bh - f)$. We write $D = \mathrm{div}(a, b)$ to mean $D = \gcd(\mathrm{div}(a), \mathrm{div}(b - v))$ where the gcd of two divisors $\sum_{P\neq\infty} m_P P - (\sum_{P\neq\infty} m_P)\infty$ and $\sum_{P\neq\infty} n_P P - (\sum_{P\neq\infty} n_P)\infty$ is defined to be $\sum_{P\neq\infty} \min(m_P, n_P)P - (\sum_{P\neq\infty} \min(m_P, n_P))\infty$. The *degree* of D is $\deg a$. Cantor's algorithm [2] can be used to efficiently compute the sum of two reduced divisors, and express the sum in reduced form.

Artin's Bound. In the above, we only considered the *imaginary* form of a hyperelliptic curve, and not the *real* form for which $\deg(f) = 2g + 2$ in the defining equation. Let C be a hyperelliptic curve (real or imaginary) of genus g over $k = \mathbb{F}_p$ with p an odd prime. Artin [1] showed that $\#J_C(k) = \sum_{\nu=0}^{2g} \chi_\nu$ if $\deg f = 2g+1$, and $\#J_C(k) = -\sum_{\nu=1}^{2g+1} \nu\chi_\nu$ if $\deg f = 2g+2$. Here, $\chi_\nu = \sum_{\deg F = \nu} [f/F]$, where the summation is over all degree-ν monic polynomials $F \in \mathbb{F}_p[u]$ coprime to f, and $[f/F]$ is the polynomial Legendre symbol. We trivially have that $|\chi_\nu| \leq p^\nu$, and Artin showed that $|\chi_\nu| \leq p^g$ $(0 \leq \nu \leq 2g)$ if $\deg f = 2g+1$, and $\chi_{2g+1} = -p^g$ and $|\chi_\nu| \leq 2p^g$ $(1 \leq \nu \leq 2g)$ if $\deg f = 2g + 2$. These results can be extended to the case $k = \mathbb{F}_q$, where $q = p^l$ and p is prime, by replacing the Artin character by the general quadratic character. Then $\#J_C(k) \leq gq^g + \sum_{\nu=0}^{g} q^\nu$ if $\deg f = 2g+1$, and $\#J_C(k) \leq ((2g + 1)^2 - g(g + 1))q^g + \sum_{\nu=1}^{g} \nu q^\nu$ if $\deg f = 2g + 2$. Since over constant fields of characteristic 2 the real case is strictly more general than the imaginary case (cf. [20]), we work with $B_2 := ((2g+1)^2 - g(g+1))q^g + \sum_{\nu=1}^{g} \nu q^\nu$ as an upper bound on the cardinality of the Jacobian. Notice that the larger q is, the larger is the smallest genus g for which the Artin bound B_2 is indeed smaller than the Hasse-Weil upper bound $B_1 := (\sqrt{q} + 1)^{2g}$.

3 Weil Descent Attack

Let l and n be positive integers, $N = ln$, $q = 2^l$, $k = \mathbb{F}_q$, and $K = \mathbb{F}_{q^n}$. Consider the elliptic curve E defined by $y^2 + xy = x^3 + ax^2 + b$, $a \in K$, $b \in K^*$. Gaudry, Hess and Smart [11] showed how Weil descent can be used to reduce the ECDLP in $E(K)$ to a discrete logarithm problem in the Jacobian $J_C(k)$ of a hyperelliptic curve C defined over k. One first constructs the Weil restriction $W_{E/k}$ of scalars of E, which is an n-dimensional abelian variety over k. Then, $W_{E/k}$ is intersected with $n-1$ hyperplanes to obtain the hyperelliptic curve C. We call their reduction algorithm the *GHS attack* on the ECDLP. The following is proven in [11].

Theorem 1 (Gaudry, Hess and Smart [11]) Let $q = 2^l$ and let $E : y^2 + xy = x^3 + ax^2 + b$ be an elliptic curve defined over $K = \mathbb{F}_{q^n}$. Let $\sigma : K \to K$ be the Frobenius automorphism defined by $\alpha \to \alpha^q$, and let $b_i = \sigma^i(b)$ for $0 \leq i \leq n - 1$. Let the *magic number for E relative to n* be $m = m(b) =$

$\dim_{\mathbb{F}_2}(\mathrm{Span}_{\mathbb{F}_2}\{(1, b_0^{1/2}), (1, b_1^{1/2}), \ldots, (1, b_{n-1}^{1/2})\})$. Assume that

$$n \text{ is odd}, \quad \text{or } m(b) = n, \quad \text{or } \mathrm{Tr}_{K/\mathbb{F}_2}(a) = 0. \tag{1}$$

Then the GHS attack constructs an explicit group homomorphism $\varphi : E(\mathbb{F}_{q^n}) \to J_C(\mathbb{F}_q)$, where C is a hyperelliptic curve over $\mathbb{F}_q$ of genus $g = 2^{m-1}$ or $2^{m-1} - 1$.

Remark 2 (*solving ECDLP instances in $E(\mathbb{F}_{q^n})$*) Assume now that $\#E(\mathbb{F}_{q^n})$ is almost prime, i.e., $\#E(\mathbb{F}_{q^n}) = rd$ where r is prime and d is small. In [11] it is argued that it is highly unlikely that the kernel of φ will contain the subgroup of order r of $E(\mathbb{F}_{q^n})$ unless E is defined over a proper subfield of $\mathbb{F}_{q^n}$. Thus, φ can be used to reduce instances of the ECDLP in $\langle P \rangle$, where P is a point of order r in $E(\mathbb{F}_{q^n})$, to instances of the HCDLP in $J_C(\mathbb{F}_q)$. Namely, given P and $Q \in \langle P \rangle$, then $\log_P Q = \log_{\varphi(P)} \varphi(Q)$.

Remark 3 (*efficiency of determining C and computing φ*) The running time complexity of the algorithm presented in [11] for finding the defining equation of C and for computing φ has not been determined. However, if ng is relatively small, say $ng \le 1000$, our extensive experiments suggest that Hess's KASH implementation [12,3] of the algorithm takes at most a few hours on a workstation.

The formula for m in Theorem 1 was analyzed in [18] and Theorem 5 was obtained. We first need to define the *type* of an element of $\mathbb{F}_{q^n}$.

Definition 4 Let $n = 2^e n_1$ where n_1 is odd. Let $h = 2^e$ and $x^n - 1 = (f_0 f_1 \cdots f_s)^h$ where $f_0 = x - 1$ and the f_i's are distinct irreducible polynomials over $\mathbb{F}_2$ with $\deg(f_i) = d_i$ and $1 = d_0 < d_1 \le d_2 \le \cdots \le d_s$. For $b \in \mathbb{F}_{q^n}$, let $\mathrm{Ord}_b(x)$ be the unique monic polynomial $f \in \mathbb{F}_2[x]$ of least degree such that $f(\sigma)b = 0$; we have $\mathrm{Ord}_b(x) | x^n - 1$. For each $i \in [0, s]$, let j_i be the largest power of f_i which divides $\mathrm{Ord}_b(x)$. The *type* of b is defined to be $(j_0, j_1, \ldots, j_s)$.

Theorem 5 ([18]) Let $b \in \mathbb{F}_{q^n}$ have type $(j_0, j_1, \ldots, j_s)$.

(i) Then $m(b) = \sum_{i=0}^{s} j_i d_i + c$, where $c = 1$ if $j_0 = 0$, and $c = 0$ if $j_0 \neq 0$.

(ii) There are $\prod_{i=0, j_i \neq 0}^{s} (q^{j_i d_i} - q^{(j_i - 1)d_i})$ elements of type $(j_0, j_1, \ldots, j_s)$ in $\mathbb{F}_{q^n}$.

Lemma 6 asserts that condition (1) of Theorem 1 can be weakened.

Lemma 6 Let $E/\mathbb{F}_{q^n}$ be an elliptic curve defined by the equation $y^2 + xy = x^3 + ax^2 + b$ where $b \in \mathbb{F}_{q^n}$ has type $(j_0, j_1, \ldots, j_s)$. In Theorem 1, condition (1) can be replaced by the following, weaker, condition:

$$n \text{ is odd}, \quad \text{or } 2^e = j_0, \quad \text{or } \mathrm{Tr}_{K/\mathbb{F}_2}(a) = 0. \tag{2}$$

Proof: Observe first that if n is even and $m(b) = n$, then b must be of type $(2^e, \ldots, 2^e)$ so that $2^e = j_0$. Thus, (1) indeed implies (2). Now, let $\overline{f} = (x - 1)^c \prod_{i=0}^{s} f_i^{j_i d_i}$, where $c = 1$ if $j_0 = 0$, and $c = 0$ if $j_0 \neq 0$. (This function has to replace the function f incorrectly defined in the proof of Lemma 11 in [11].) Let $\overline{h} = (x^n - 1)/\overline{f}$. From the proof of Lemma 11 in [11] it follows that Theorem 1 is true if $\mathrm{Tr}_{K/\mathbb{F}_2}(a) = 0$ or $\mathrm{Tr}_{K/\mathbb{F}_2}(a) + \overline{h}(1) = 0$. Thus, if $\mathrm{Tr}_{K/\mathbb{F}_2}(a) = 1$, Theorem 1 is true if $\overline{h}(1) = 1$. Since $x^n - 1 = (x^{n_1} - 1)^{2^e} = (x-1)^{2^e} \cdot \tilde{k}$ with $\tilde{k}(1) = 1$, we have $\overline{h}(1) = 1$ if and only if $(x-1)^{2^e}$ divides $\overline{f}$. Since the latter is true if and only if n is odd or $2^e = j_0$, the lemma is established. □

There are $2^{N+1} - 2$ isomorphism classes of elliptic curves over $\mathbb{F}_{2^N}$ with representatives $y^2 + xy = x^3 + b$, $y^2 + xy = x^3 + ax^2 + b$, where $b \in \mathbb{F}_{2^N}$ and $a \in \mathbb{F}_{2^N}$ is a fixed element with $\mathrm{Tr}_{\mathbb{F}_{2^N}/\mathbb{F}_2}(a) = 1$. The number I of isomorphism classes of elliptic curves over $\mathbb{F}_{2^N}$ with a given magic number m relative to n and satisfying (2) can be efficiently computed using the following.

Lemma 7 Let n and $m \in [1, n]$ be fixed. Let $c_{i,j} = q^{jd_i} - q^{(j-1)d_i}$ for $0 \le i \le s$ and $1 \le j \le h$. Let $F_0(z) = 2(z + \sum_{j=1}^{h} c_{0,j} z^j)$ if n is odd, and $F_0(z) = z + \sum_{j=1}^{h-1} c_{0,j} z^j + 2c_{0,h} z^h$ if n is even, $F_i(z) = 1 + \sum_{j=1}^{h} c_{i,j} z^{jd_i}$ for $1 \le i \le s$, and $F(z) = F_0(z) \prod_{i=1}^{s} F_i(z)$. Then the number of isomorphism classes of elliptic curves over $\mathbb{F}_{2^N}$ with magic number m relative to n and satisfying (2) is $I = [z^m]F(z)$ where [ ] denotes the coefficient operator.

Proof: Follows immediately from Theorem 5 and Lemma 6. □

4 Algorithms for the ECDLP and HCDLP

ECDLP. Let $E/\mathbb{F}_{2^N}$ be a cryptographically interesting elliptic curve, and let r be the large prime divisor of $\#E(\mathbb{F}_{2^N})$. Then Pollard's rho algorithm [22] for solving the ECDLP in the subgroup of order r of $E(\mathbb{F}_{2^N})$ has an expected running time of $(\sqrt{\pi r})/2$ elliptic curve additions. Since E is cryptographically interesting, $r \approx 2^{N-1}$ (taking into account that there is always a cofactor at least 2). We henceforth use $(\sqrt{\pi 2^{N-1}})/2$ to express the running time of Pollard's rho algorithm. Note that the algorithm can be effectively parallelized (see [19]) so that its expected running time on a network of S processors is $(\sqrt{\pi 2^{N-1}})/(2S)$.

HCDLP. Let C be a genus g hyperelliptic curve over $k = \mathbb{F}_q$. The HCDLP is the following: given C, $D_1 \in J_C(k)$, $r = \mathrm{ord}(D_1)$, and $D_2 \in \langle D_1 \rangle$, find the integer $\lambda \in [0, r-1]$ such that $D_2 = \lambda D_1$. We shall assume that r is prime. We describe the Enge-Gaudry (EG) index-calculus algorithm [10,4] for the HCDLP.

A reduced divisor $D = \mathrm{div}(a, b) \in J_C(k)$ is called a *prime divisor* if a is irreducible over k. Each reduced divisor $D = \mathrm{div}(a, b) \in J_C(k)$ can be expressed as a sum of prime divisors as follows: if $a = a_1^{e_1} a_2^{e_2} \cdots a_L^{e_L}$ is the factorization of a into monic irreducibles over k, then $D = \sum_{i=1}^{L} e_i \mathrm{div}(a_i, b_i)$ where $b_i = b \bmod a_i$ for all $i \in [1, L]$. Such a D is said to be *t-smooth* if $\max\{\deg a_i\} \le t$.

In the Enge-Gaudry algorithm, a *smoothness bound* t is first chosen. Next, the *factor base* $\{P_1, P_2, \ldots, P_w\}$ is constructed—for each prime divisor $D = \mathrm{div}(a, b)$ of degree $\le t$, exactly one of D and $-D$ is included in the factor base. Then, a random walk (á la Teske [24]) is performed in the set of reduced divisors equivalent to divisors of the form $\alpha D_1 + \beta D_2$ and the t-smooth divisors encountered in this walk are stored—each t-smooth divisor yields a relation $\alpha_i D_1 + \beta_i D_2 \mapsto R_i = \sum_j e_{ij} P_j$. When $w+5$ different relations have been found, one can find by linear algebra modulo r a non-trivial linear combination $\sum_{i=1}^{w+5} \gamma_i (e_{i1}, e_{i2}, \ldots, e_{iw}) = (0, 0, \ldots, 0)$. Thus $\sum_{i=1}^{w+5} \gamma_i R_i = 0$, whence $\sum \gamma_i (\alpha_i D_1 + \beta_i D_2) = 0$ and $\log_{D_1} D_2 = -(\sum \gamma_i \alpha_i)/(\sum \gamma_i \beta_i) \bmod r$.

The EG algorithm has a subexponential-time running time of $O(\exp((\sqrt{2} + o(1))\sqrt{\log q^g \log \log q^g}))$ bit operations for $g/\log q \to \infty$. In [14], the following non-asymptotic analysis of the running time for the relation gathering stage was given. A good approximation for the number A_l of prime divisors of degree l in the factor base is $A_l \approx \frac{1}{2}(\frac{1}{l}\sum_{d|l} \mu(l/d) q^d)$, where μ is the Möbius function. The factor base size w is therefore well approximated by $F(t) = \sum_{l=1}^{t} A_l$. By [14, Lemma 2], the number of t-smooth reduced divisors in $J_C(k)$ is $M(t) = \sum_{i=1}^{g}([x^i] \prod_{l=1}^{t}(\frac{1+x^l}{1-x^l})^{A_l})$, where [ ] denotes the coefficient operator. Under the heuristic assumption that the proportion of t-smooth divisors in $\langle D_1 \rangle$ is the same as the proportion of t-smooth divisors in the full group $J_C(k)$, the expected number of random walk iterations before a t-smooth divisor is encountered is $E(t) = \#J_C(k)/M(t)$. Finally, the expected number of random walk iterations before $F(t) + 5$ relations are generated is $T(t) = (F(t) + 5)E(t)$.

5 Analysis

For each composite $N \in [160, 600]$, we determine and compare the running times for solving the ECDLP in a (potentially) cryptographically interesting elliptic curve over $\mathbb{F}_{2^N}$ using the GHS attack and Pollard's rho method. We express the running times for Pollard's rho method and the GHS attack in terms of elliptic curve operations and in terms of random walk iterations in the Jacobian, respectively, as outlined in §4. In particular, we do not consider the different bit complexities of operations for elliptic and hyperelliptic curves since these are expected to be roughly the same. Furthermore, we do not take into account the time spent on mapping the ECDLP instance to a HCDLP instance, and the time spent on the linear algebra stage of the Enge-Gaudry index-calculus algorithm.

For each composite $N \in [160, 600]$, Algorithm 9 determines the elliptic curve parameters (in terms of n, m and g) such that (i) there should (cf. Remark 19) exist a cryptographically interesting elliptic curve E over $\mathbb{F}_{2^N}$ with these parameters; and (ii) the GHS attack is more efficient for solving the ECDLP in $E(\mathbb{F}_{2^N})$ than for solving the ECDLP on any other cryptographically interesting elliptic curve over $\mathbb{F}_{2^N}$. For each such set of parameters (n, m, g), we list the number I of isomorphism classes of elliptic curves over $\mathbb{F}_{2^N}$ that have magic number m relative to n and satisfy (2), the optimal smoothness bound t for the Enge-Gaudry algorithm, and the resulting estimates for the factor base size $F(t)$ and the (minimized) running time $T(t)$ in terms of random walk iterations.

Remark 8 (*EG1 versus EG2*) In Algorithm 9, two variants of the Enge-Gaudry algorithm are considered. The first variant, denoted by EG1, only works with a factor base whose size is upper bounded by $10^7 \approx 2^{23}$, while the second variant, denoted by EG2, does not assume any upper bound on the factor base size. Note that a factor base of size 10^7 is on the edge of what is considered feasible today [15,16]. If $A_1 = 2^{l-1} > 10^7$ for some hyperelliptic curve of genus g over $\mathbb{F}_{2^l}$, then, in order to achieve a factor base size $\le 10^7$, the Enge-Gaudry algorithm can be modified by selecting the factor base to consist of only a proportion $\frac{1}{}$ of all prime divisors of degree 1 [11]. However, the expected time to find a smooth divisor will be increased by a factor of ^{g}. Therefore, we decided not to consider this modification in our analysis. If the factor base size for EG2 is significantly larger than 10^7, then the EG2 algorithm is not currently practical. Nevertheless, we feel that listing the optimum times for EG2 is important because they will become relevant should improvements be made in the future to algorithms for solving sparse linear systems.

Algorithm 9 (*Computing optimal* (n, m, g, t, F, T))
INPUT: N, “EG1” or “EG2”.
OUTPUT: Parameters n, m, g for which there may exist an elliptic curve that is cryptographically interesting and whose ECDLP is most easily solved with the GHS attack; optimal smoothness bound t; (estimated) factor base size F; and (estimated) expected running time T in terms of random walk iterations.

```
1. For all divisors $n \ge 2$ of $N$ do the following:
   (a) Set $l \leftarrow N/n$ and $q \leftarrow 2^l$.
   (b) { For EG1: The $10^7$ bound on factor base size must be violated if $A_1 = 2^{l-1} > 10^7$. } Case EG1: If $l \ge 25$ then set $T_n \leftarrow \infty$ and go to step 1.
   (c) Write $n = n_1 h$ where $h = 2^e$ and $n_1$ is odd.
   (d) { Compute the degrees of the irreducible factors of $x^{n_1} - 1$ over $\mathbb{F}_2$. }
       Let the cyclotomic cosets of 2 modulo $n_1$ have sizes $1 = d_0 \le d_1 \cdots \le d_s$.
   (e) { Compute a lower bound $m$ on magic number $m$ relative to $n$ that yields a large enough Jacobian (cf. Remark 10). }
       For $m = 2, 3, \ldots, n$ do the following:
         i. Set $g \leftarrow 2^{m-1} - 1$. Compute $B_1$, $B_2$ as defined in §2.
         ii. If $\min\{\log_2 B_1, \log_2 B_2\} \ge N - 3$ then go to step 1(f).
         iii. Set $g \leftarrow 2^{m-1}$. Compute $B_1$, $B_2$ as defined in §2.
         iv. If $\min\{\log_2 B_1, \log_2 B_2\} \ge N - 3$ then go to step 1(f).
   (f) { Find the smallest admissible $m$ relative to $n$ (cf. Theorem 5). }
       For $m = m, m + 1, \ldots, n$ do the following:
       If $m$ can be written in the form $\sum_{i=0}^{s} d_i j_i$ with $0 \le j_i \le h$, $j_0 \ge 1$, then:
           { Check that the sufficient conditions of Lemma 12 (for every elliptic curve over $\mathbb{F}_{2^N}$ having magic number $m$ relative to $n$ to be defined over a proper subfield $\mathbb{F}_{2^\mu}$ of $\mathbb{F}_{2^N}$ for some $\mu \ge 3$) are violated. }
           If $n$ is a power of 2, set $d \leftarrow \infty$; else set $d \leftarrow d_1$.
           If $[m > d$ or $m \ge 2^e]$ or $[d = \infty$ and $m > 2^{e-1}]$ then go to step 1(g).
   (g) If $m > m$ then set $g \leftarrow 2^{m-1} - 1$.
   (h) { If the size of the Jacobian is not too large, i.e., if $gl \le 4096$ (cf. Remark 14), then find the optimum smoothness bound $t$ for the Enge-Gaudry algorithm using the formulas at the end of §4 to estimate the factor base size $F(t)$, the expected running time $E(t)$ to find a smooth divisor with $\#J_C(\mathbb{F}_q) = 2^{gl}$, and the expected running time $T(t)$. }
       If $gl \ge 4097$ then set $T_n \leftarrow \infty$. Else:
        i. Case EG1: Set $S \leftarrow \{1 \le t \le 120 \; : \; F(t) \le 10^7\}$.
           Case EG2: Set $S \leftarrow \{1, 2, \ldots, 120\}$.
       ii. Let $t$ be the index in $S$ which minimizes $T(t)$.
      iii. Set $m_n \leftarrow m$, $g_n \leftarrow g$, $t_n \leftarrow t$, $F_n \leftarrow F(t_n)$, $T_n \leftarrow T(t_n)$.
 2. If $T_n = \infty$ for all $n$, output "$gl \ge 4097$ for all $n$". Else, let $n$ be the index
    for which $T_n$ is a minimum and output "$(n, m_n, g_n, t_n, F_n, T_n)$".
```

Remark 10 (*explanation of the lower bound on* $\log_2 B_1$ *and* $\log_2 B_2$ *in step 1(e) of Algorithm 9*) If we restrict our attention to cryptographically interesting elliptic curves E over $\mathbb{F}_{2^N}$ with $\#E(\mathbb{F}_{2^N}) = dr$, where $d \in \{2, 4\}$ and r is prime, then $r \ge \#E(\mathbb{F}_{2^N})/4 \ge (2^{N/2} - 1)^2/4 > 2^{N-1}/4 = 2^{N-3}4$ for $N \ge 4$. Thus, if the hyperelliptic curve C over $\mathbb{F}_q$ generated by the GHS reduction has genus g, then a necessary condition for $J_C(\mathbb{F}_q)$ to have a subgroup of order r is $\min(B_1, B_2) \ge \#J_C(\mathbb{F}_q) \ge 2^{N-3}$.

Remark 11 (*explanation of step 1(f) of Algorithm 9*) There are some (N, l, g) parameters for which elliptic curves over $\mathbb{F}_{2^N}$ with parameters (l, g) do exist, but none of which are cryptographically interesting. For example, if $N = 160$, the ECDLP is most easily solved with the GHS attack if $(n, l, m, g) = (8, 20, 4, 8)$. Then, for the attack to work (cf. condition (2) in Lemma 6), we need $\mathrm{Tr}_{K/\mathbb{F}_2}(a) = 0$, i.e., without loss of generality, $a = 0$. Now, consider an elliptic curve $E : y^2 + xy = x^3 + b$ over $\mathbb{F}_{2^{160}}$ that yields magic number $m = 4$ on performing the GHS attack with $n = 8$. We have $x^n - 1 = (x - 1)^8$, and hence $(\sigma - 1)^4 b = 0$ where $\sigma : \mathbb{F}_{2^{160}} \to \mathbb{F}_{2^{160}}$ is defined by $\alpha \to \alpha^{2^{20}}$. That is, $b \in \mathbb{F}_{2^{80}}$, which implies that $\#E(\mathbb{F}_{2^{80}})$ divides $\#E(\mathbb{F}_{2^{160}})$. Hence E is not cryptographically interesting. The next easiest instance of an ECDLP over $\mathbb{F}_{2^{160}}$ for which a cryptographically interesting curve can exist is $(n, l, m, g) = (20, 8, 6, 31)$. Such a phenomenon always occurs when $(n, m) = (8, 4)$ are the GHS parameters for which the ECDLP is most easily solved, which is the case for $N = 176, 184, 192$ and many other N divisible by 8. But also for $N = 224$ where $(n, m) = (32, 6)$ would be best we find that $\#E(\mathbb{F}_{2^{56}})$ must divide $\#E(\mathbb{F}_{2^{224}})$ for any elliptic curve with these parameters. Another example is $N = 304$ where $(n, m) = (16, 5)$ would be optimal—here we find that $\#E(\mathbb{F}_{2^{304}})$ must be divisible by $\#E(\mathbb{F}_{2^{152}})$.

Lemma 12 generalizes the observations made in Remark 11.

Lemma 12 Let $E/\mathbb{F}_{q^n}$ be an elliptic curve defined by the equation $y^2 + xy = x^3 + ax^2 + b$, where $b \in \mathbb{F}_{q^n}$ has type $(j_0, j_1, \ldots, j_s)$. Suppose that (2) holds. Let $n = n_1 2^e$ where n_1 is odd. If n is a power of 2, then let $d = \infty$; otherwise, let $d = d_1 = \min\{d_i : 1 \le i \le s\}$. Let $m = m(b)$ be as in Theorem 1. Let $\mu = 2^{\lceil \log_2 m \rceil}$, i.e., the smallest power of 2 greater than or equal to m.

(i) If $m \le d$ and $m < 2^e$, then E is defined over $\mathbb{F}_{q^\mu}$ and hence $\#E(\mathbb{F}_{q^n})$ is divisible by $\#E(\mathbb{F}_{q^\mu})$.

(ii) If $m \le d$ and $m = 2^e$ and n/μ is odd, then E is (isomorphic to a curve) defined over $\mathbb{F}_{q^\mu}$ and hence $\#E(\mathbb{F}_{q^n})$ is divisible by $\#E(\mathbb{F}_{q^\mu})$.

Proof:

(i) Assume that $m \le d$ and $m < 2^e$. Then b must have type $(m, 0, \dots, 0)$, and n is even and $2^e = j_0$. The former implies that $b \in B = \{b \in \mathbb{F}_{q^n} : (\sigma + 1)^m(b) = 0\} \setminus \{b \in \mathbb{F}_{q^n} : (\sigma + 1)^{m-1}(b) = 0\}$. Let $m_- = \mu/2$, i.e., the largest power of 2 strictly less than m. Then $B \subseteq \mathbb{F}_{q^\mu} \setminus \mathbb{F}_{q^{m_-}}$. Since n is even and $2^e = j_0$, we require $\mathrm{Tr}_{K/\mathbb{F}_2}(a) = 0$ for Lemma 6 to hold. Thus, without loss of generality, $a = 0$. Thus, E is defined over $\mathbb{F}_{q^\mu}$ but not over any proper subfield of $\mathbb{F}_{q^\mu}$.

(ii) Now assume that $m \le d$ and $m = 2^e$ and n/μ is odd. Then, as before, $b \in \mathbb{F}_{q^\mu} \setminus \mathbb{F}_{q^{m_-}}$. Since $m = 2^e$, both $\mathrm{Tr}_{K/\mathbb{F}_2}(a) = 0, 1$ are possible. Now, $\mathrm{Tr}_{K/\mathbb{F}_2}(c) = (n/\mu)\mathrm{Tr}_{\mathbb{F}_{q^\mu}/\mathbb{F}_2}(c)$ for all $c \in \mathbb{F}_{q^\mu}$. Since n/μ is odd, $\mathrm{Tr}_{K/\mathbb{F}_2}(c) = \mathrm{Tr}_{\mathbb{F}_{q^\mu}/\mathbb{F}_2}(c)$, so that there exists $c \in \mathbb{F}_{q^\mu}$ such that $\mathrm{Tr}_{K/\mathbb{F}_2}(c) = 1$. Therefore, both for $\mathrm{Tr}_{K/\mathbb{F}_2}(a) = 0$ and $\mathrm{Tr}_{K/\mathbb{F}_2}(a) = 1$ there exists a curve isomorphic to E that is defined over $\mathbb{F}_{q^\mu}$ but not over any proper subfield of $\mathbb{F}_{q^\mu}$. □

Corollary 13 If n is a power of 2 and $n/4 < m \le n/2$, then E is defined over $\mathbb{F}_{q^{n/2}}$ and hence $\#E(\mathbb{F}_{q^n})$ is divisible by $\#E(\mathbb{F}_{q^{n/2}})$.

Remark 14 (*restriction on gl in step 1(h) of Algorithm 9*) For $g \ge 4097$ we were unable to compute the expected running time of EG1/EG2 because of computational limitations when computing Taylor series expansions needed to evaluate $M(t)$. We therefore ignore all instances (n, l, g) where $gl \ge 4097$. Notice that in this case the Jacobian $J_C(\mathbb{F}_q)$ has size at least 2^{4097} whence any (cryptographically interesting) HCDLP instance in $J_C(\mathbb{F}_q)$ is infeasible using the known index-calculus type algorithms. In particular, if $l = 1$ and $g = 4095$, the smallest running time for EG2 is with $t = 120$ and amounts to $\approx 2^{307}$ random walk iterations, which is more than the expected number of elliptic curve operations using Pollard's rho method for $N = 600$.

The outputs of Algorithm 9 with composite $N \in [160, 600]$ as inputs are listed in Appendix A. In these tables, the entries for I, F, T, and ρ are the *logarithms* (base 2, rounded to the nearest integer) of the number of isomorphism classes of elliptic curves with magic number m relative to n and satisfying (2), the factor base size, the expected number of random walk iterations in the Enge-Gaudry algorithm, and the number of elliptic curve operations in Pollard's rho method, respectively. $D1$ and $D2$ denote the differences $\rho - T$ (if positive) for EG1 and EG2, respectively. If for some N data is given for EG2 but not for EG1, we are in the situation that $gl \ge 4097$ for all divisors $l \le 24$ of N (such as for $N = 164$ and 166). If for some N data is given for neither EG1 nor EG2, we are in the situation that $gl \ge 4097$ for all l dividing N. The latter occurs for only 5 values of N: 289, 323, 361, 493 and 551.

Remark 15 (*further limitations of our analysis*) Our analysis yields the same running times whenever (g, l) are the same, independently of N (e.g., $T = 53$ when $(g, l) = (15, 13)$ for both $N = 130$ and $N = 195$—see Appendix A). This is because the running time of the Enge-Gaudry algorithm is computed under the assumption that $\#J_C(\mathbb{F}_q) \approx q^g = 2^{gl}$. However, we only expect that

$\#J_C(\mathbb{F}_q)$ is divisible by the large prime that divides $\#E(\mathbb{F}_{q^n})$. Hence if $gl \gg N$, it may well be the case that the Jacobian obtained from Weil descent is much smaller in size than q^g, which would then lead to a significantly smaller value $E(t) = \#J_C(k)/M(t)$, and hence also to a significantly smaller running time $T(t)$. This observation is particularly meaningful where $l = 1$, in which case the Hasse-Weil lower bound $(\sqrt{2} - 1)^{2g} \le \#J_C(\mathbb{F}_2)$ is trivial. For example, if $(l, g) = (1, 255)$, we have $T = 2^{54}$ for EG1 and $T = 2^{52}$ for EG2, for $N = 117, 153, 170, 171, 187, 190$, etc., $270, 273$. Thus, caution must be exercised when interpreting our data for those N where $gl \gg N$. Nevertheless, if $gl \approx N$, our running time estimates are precise.

Remark 16 (*success of the GHS attack*) There are some composite $N \in [160, 600]$ for which the GHS attack succeeds on some cryptographically interesting elliptic curves over $\mathbb{F}_{2^N}$. That is, Pollard's rho algorithm is infeasible for solving the ECDLP on these curves, and the GHS attack is successful in reducing instances of the ECDLP on these curves to instances of the HCDLP which are solvable using known algorithms and existing computer technology. Examples of such N are $N = 161, 180, 186, 217, 248, 300$[1].

Remark 17 (*failure of the GHS attack*) We can conclude that for those composite $N \in [160, 600]$ for which no values are entered for EG1, the GHS attack does not reduce the level of security offered—Pollard's rho method is the faster algorithm for *all* elliptic curves over $\mathbb{F}_{2^N}$. In particular, this is true for $N = 185$, which is of practical significance because a specific elliptic curve over $\mathbb{F}_{2^{185}}$ is listed in the IETF standard [13] for key establishment. We emphasize that our statements about the failure of the GHS attack for all elliptic curves over some field $\mathbb{F}_{2^N}$ are under the assumption that the Enge-Gaudry algorithm is essentially the best index-calculus algorithm for the HCDLP, and, in particular, that the linear algebra stage is intractable if the factor base size is greater than 10^7.

Remark 18 (*effectiveness of the GHS attack*) When $D1 > 0$ for some composite $N \in [160, 600]$, the level of security offered by some cryptographically interesting elliptic curves defined over $\mathbb{F}_{2^N}$ may be reduced due to the GHS attack. However, note that our data corresponds to elliptic curves with *least possible* magic numbers and genera, and only a small proportion of elliptic curves yield this minimal magic number. For example, if $N = 161$, then only $\approx 2^{94}$ out of $\approx 2^{162}$ elliptic curves over $\mathbb{F}_{2^{161}}$ have magic number $m = 4$ relative to $n = 7$. Correspondingly, for $N = 165$ the proportion of elliptic curves with magic number $m = 5$ relative to $n = 15$ is only $\approx 2^{58}$ out of 2^{166}, whereas for $N = 162$, the proportion of curves having magic number $m = 7$ relative to $n = 54$ is even smaller, namely $\approx 2^{21}$ out of 2^{163}. Galbraith, Hess and Smart [8] presented an algorithm with expected

[1] We have computed explicit ECDLP instances (i.e., the elliptic curve equations and points) over these six fields and the HCDLP instances (i.e., the hyperelliptic curve equations and divisors) they are mapped to under the GHS attack. These instances have not been included here due to space constraints.

average running time of $O(q^{n/4+\ })$ for explicitly computing an isogeny between two isogenous elliptic curve over $\mathbb{F}_{q^n}$. (Two elliptic curves $E_1/\mathbb{F}_{q^n}$ and $E_2/\mathbb{F}_{q^n}$ are said to be *isogenous* over $\mathbb{F}_{q^n}$ if $\#E_1(\mathbb{F}_{q^n}) = \#E_2(\mathbb{F}_{q^n})$.) They observed that this algorithm can be used to extend the effectiveness of the GHS attack. Namely, given an ECDLP instance on some cryptographically interesting elliptic curve $E_1/\mathbb{F}_{2^N}$, one can check if E_1 is isogenous to some elliptic curve $E_2/\mathbb{F}_{2^N}$ which yields an easier HCDLP than E_1, and then use an isogeny $\varphi : E_1 \to E_2$ to map the ECDLP instance to an instance of the ECDLP in $E_2(\mathbb{F}_{2^N})$. For example, in the case $N = 165$, we can expect that roughly 2^{135} out of 2^{166} elliptic curves over $\mathbb{F}_{2^{165}}$ are isogenous to one of the $\approx 2^{58}$ elliptic curves over $\mathbb{F}_{2^{165}}$ having magic number $m = 5$ relative to $n = 15$. Note, however, that finding a curve with $m = 5$ isogenous to a given elliptic curve over $\mathbb{F}_{2^{165}}$ (assuming that such an isogenous curve exists) may be difficult as one essentially has to search through the entire set of 2^{58} curves.

Remark 19 (*finding cryptographically interesting elliptic curves with given* (N, l, m) *parameters*) One can attempt to find a cryptographically interesting elliptic curve with given (N, l, m) parameters as follows. First select arbitrary b from the set $B = \{b \quad \mathbb{F}_{2^N} : m(b) = m\}$; that the elements of B can be efficiently enumerated can be seen from Theorem 5(i). Next, compute $H = \#E_b(\mathbb{F}_{2^N})$ where $E_b : y^2 + xy = x^3 + b$ using Satoh's algorithm [23,5], and test if either H or $2^{N+1} + 2 - H$ (the order of the twist of E_b) is almost a prime. Observe that if $b \quad B$, then $b^2 \quad B$. Moreover, E_b and E_{b^2} are isogenous over $\mathbb{F}_{2^N}$. Thus, if $b \quad B$ has already been tested, then one should not select b^{2^i} for any $1 \le i \le N - 1$. Now, it is known that the order of a randomly selected elliptic curve over $\mathbb{F}_{2^N}$ is roughly uniformly distributed over the even integers in the Hasse interval $[(2^{N/2} - 1)^2, (2^{N/2} + 1)^2]$. Thus, if the set B has sufficiently large cardinality (which can be determined from Lemma 7), then we can expect to quickly find an elliptic curve of almost prime order.

6 Conclusions

We analyzed the GHS Weil descent attack on the ECDLP for elliptic curves defined over characteristic two finite fields $\mathbb{F}_{2^N}$ of composite extension degree $N \quad [160, 600]$. For some such fields, there are cryptographically interesting elliptic curves over $\mathbb{F}_{2^N}$ where the ECDLP succumbs to the GHS attack. For other such fields $\mathbb{F}_{2^N}$, our results demonstrate that there are no cryptographically interesting elliptic curves over $\mathbb{F}_{2^N}$ for which the GHS attack yields an ECDLP solver that is faster than Pollard's rho method.

We stress that any statement we have made regarding the failure of the GHS attack on some elliptic curves over some field $\mathbb{F}_{2^N}$ is dependent on the assumption that the Enge-Gaudry algorithm cannot be significantly improved, and, in particular, that the linear algebra stage is intractable if the factor base size is greater than 10^7. Also, we stress that failure of the GHS attack does not imply failure of the Weil descent methodology—there may be other useful curves which lie on the Weil restriction $W_{E/k}$ that were not constructed by the GHS method. We thus hope that our work can serve as a stimulus for further work on the Weil descent method, on subexponential-time index-calculus methods for the HCDLP, and on algorithms for solving large systems of sparse linear equations.

Acknowledgements

We used several computer algebra systems in our work including KASH, Magma, Maple and NTL. We would like to thank Antoine Joux, Steven Galbraith, Florian Hess and Nigel Smart for their helpful advice and comments. We would also like to thank NSERC for providing partial funding for this research.

References

1. E. Artin. "Quadratische Körper im Gebiete der höheren Kongruenzen", *Mathematische Zeitschrift*, **19** (1924), 207-246.
2. D. Cantor, "Computing in the jacobian of a hyperelliptic curve", *Math. Comp.*, **48** (1987), 95-101.
3. M. Daberkow, C. Fieker, J. Klüners, M. Pohst, K. Roegner, M. Schörnig, K. Wildanger, "KANT V4", *J. Symbolic Computation*, **24** (1997), 267-283.
4. A. Enge, P. Gaudry, "A general framework for subexponential discrete logarithm algorithms", *Acta Arithmetica*, to appear.
5. M. Fouquet, P. Gaudry, R. Harley, "An extension of Satoh's algorithm and its implementation", *J. Ramanujan Mathematical Society*, **15** (2000), 281-318.
6. G. Frey, "How to disguise an elliptic curve (Weil descent)", Talk at ECC '98, Waterloo, 1998.
7. G. Frey, H. Rück, "A remark concerning *m*-divisibility and the discrete logarithm in the divisor class group of curves", *Math. Comp.*, **62** (1994), 865-874.
8. S. Galbraith, F. Hess, N. Smart, "Extending the GHS Weil descent attack", preprint, 2001.
9. S. Galbraith, N. Smart, "A cryptographic application of Weil descent", *Codes and Cryptography*, LNCS **1746**, 1999, 191-200.
10. P. Gaudry, "An algorithm for solving the discrete log problem on hyperelliptic curves", *Advances in Cryptology—Eurocrypt 2000*, LNCS **1807**, 2000, 19-34.
11. P. Gaudry, F. Hess, N. Smart, "Constructive and destructive facets of Weil descent on elliptic curves", preprint, January 2000.
12. F. Hess, KASH program for performing the GHS attack, 2000.
13. Internet Engineering Task Force, *The OAKLEY Key Determination Protocol*, IETF RFC 2412, November 1998.
14. M. Jacobson, A. Menezes, A. Stein, "Solving elliptic curve discrete logarithm problems using Weil descent", *J. Ramanujan Mathematical Society*, to appear.
15. A. Joux. Personal communication. June 2001.
16. A. Joux, R. Lercier, "Improvements on the general number field sieve for discrete logarithms in finite fields", *Math. Comp.*, to appear.
17. A. Menezes, T. Okamoto, S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", *IEEE Trans. Info. Th.*, **39** (1993), 1639-1646.
18. A. Menezes, M. Qu, "Analysis of the Weil descent attack of Gaudry, Hess and Smart", *Topics in Cryptology—CT-RSA 2001*, LNCS **2020**, 2001, 308-318.
19. P. van Oorschot, M. Wiener, "Parallel collision search with cryptanalytic applications", *J. Cryptology*, **12** (1999), 1-28.

20. S. Paulus, H. Rück, "Real amd imaginary quadratic representations of hyperelliptic function fields", *Math. Comp.*, **68** (1999), 1233-1241.
21. S. Pohlig, M. Hellman, "An improved algorithm for computing logs over $GF(p)$ and its cryptographic significance", *IEEE Trans. Info. Th.*, **24** (1978), 106-110.
22. J. Pollard, "Monte Carlo methods for index computation mod p", *Math. Comp.*, **32** (1978), 918-924.
23. T. Satoh, "The canonical lift of an ordinary elliptic curve over a finite field and its point counting", *J. Ramanujan Mathematical Society*, **15** (2000), 247-270.
24. E. Teske, "Speeding up Pollard's rho method for computing discrete logarithms", *Algorithmic Number Theory*, LNCS **1423**, 1998, 541-554.

A Results of our Analysis

Table 1.

	$EG1$										$EG2$									
N	n	l	m	g	l	t	F	T	ρ	$D1$	n	l	m	g	l	t	F	T	ρ	$D2$
160	20	8	6	31	48	3	21	52	79	27	4	40	3	4	120	1	39	44	79	35
161	7	23	4	7	94	1	22	34	80	46	7	23	4	7	94	1	22	34	80	46
162	54	3	7	63	21	9	23	42	80	38	6	27	4	7	109	1	26	38	80	42
164	–	–	–	–	–	–	–	–	–	–	4	41	3	4	123	1	40	45	81	36
165	15	11	5	15	58	2	20	37	82	45	15	11	5	15	58	2	20	37	82	45
166	–	–	–	–	–	–	–	–	–	–	2	83	2	2	167	1	82	83	82	–
168	7	24	4	7	98	1	23	35	83	48	7	24	4	7	98	1	23	35	83	48
169	169	1	13	4095	14	28	23	1208	84	–	169	1	13	4095	14	120	113	307	84	–
170	170	1	9	255	12	28	23	53	84	31	10	17	5	15	85	2	32	49	84	35
171	171	1	9	255	10	28	23	53	85	32	171	1	9	255	10	35	30	51	85	34
172	–	–	–	–	–	–	–	–	–	–	4	43	3	4	129	1	42	47	85	38
174	–	–	–	–	–	–	–	–	–	–	6	29	4	7	117	1	28	40	86	46
175	35	5	7	63	36	5	22	65	87	22	7	25	4	7	102	1	24	36	87	51
176	8	22	5	16	110	1	21	65	87	22	4	44	3	4	132	1	43	48	87	39
177	–	–	–	–	–	–	–	–	–	–	3	59	3	3	178	1	58	61	88	27
178	178	1	12	2047	15	28	23	529	88	–	2	89	2	2	179	1	88	89	88	–
180	15	12	5	15	63	2	22	39	89	50	15	12	5	15	63	2	22	39	89	50
182	14	13	5	15	67	1	12	52	90	38	7	26	4	7	106	1	25	37	90	53
183	–	–	–	–	–	–	–	–	–	–	3	61	3	3	184	1	60	63	91	28
184	8	23	5	16	115	1	22	66	91	25	4	46	3	4	138	1	45	50	91	41
185	–	–	–	–	–	–	–	–	–	–	5	37	5	15	186	1	36	76	92	16
186	31	6	6	31	40	4	21	41	92	51	31	6	6	31	40	5	27	41	92	51
187	187	1	9	255	11	28	23	53	93	40	187	1	9	255	11	35	30	51	93	42
188	–	–	–	–	–	–	–	–	–	–	4	47	3	4	141	1	46	51	93	42
189	63	3	7	63	25	9	23	42	94	52	7	27	4	7	110	1	26	38	94	56
190	190	1	9	255	9	28	23	53	94	41	190	1	9	255	9	35	30	51	94	43
192	24	8	6	31	50	3	21	52	95	43	6	32	4	7	129	1	31	43	95	52
194	–	–	–	–	–	–	–	–	–	–	2	97	2	2	195	1	96	97	96	–
195	15	13	5	15	68	1	12	52	97	45	15	13	5	15	68	2	24	41	97	56
196	28	7	6	31	43	3	18	49	97	48	7	28	4	7	114	1	27	39	97	58
198	198	1	9	255	9	28	23	53	98	45	6	33	4	7	133	1	32	44	98	54
200	200	1	9	255	9	28	23	53	99	46	100	2	8	127	17	17	29	51	99	48
201	–	–	–	–	–	–	–	–	–	–	3	67	3	3	202	1	66	69	100	31
202	–	–	–	–	–	–	–	–	–	–	2	101	2	2	203	1	100	101	100	–
203	–	–	–	–	–	–	–	–	–	–	7	29	4	7	118	1	28	40	101	61
204	204	1	9	255	12	28	23	53	101	48	6	34	4	7	137	1	33	45	101	56
205	–	–	–	–	–	–	–	–	–	–	5	41	5	15	206	1	40	80	102	22
206	–	–	–	–	–	–	–	–	–	–	2	103	2	2	207	1	102	103	102	–

Table 1. (continued)

	$EG1$										$EG2$									
N	n	l	m	g	l	t	F	T	ρ	$D1$	n	l	m	g	l	t	F	T	ρ	$D2$
207	207	1	9	255	10	28	23	53	103	50	207	1	9	255	10	35	30	51	103	52
208	208	1	13	4095	14	28	23	1208	103	–	4	52	3	4	156	1	51	56	103	47
209	209	1	11	1023	12	28	23	232	104	–	209	1	11	1023	12	77	71	124	104	–
210	30	7	6	31	45	3	18	49	104	55	7	30	4	7	122	1	29	41	104	63
212	–	–	–	–	–	–	–	–	–	–	4	53	3	4	159	1	52	57	105	48
213	–	–	–	–	–	–	–	–	–	–	3	71	3	3	214	1	70	73	106	33
214	–	–	–	–	–	–	–	–	–	–	2	107	2	2	215	1	106	107	106	–
215	–	–	–	–	–	–	–	–	–	–	5	43	5	15	216	1	42	82	107	25
216	216	1	9	255	11	28	23	53	107	54	6	36	4	7	145	1	35	47	107	60
217	31	7	6	31	46	3	18	49	108	59	7	31	4	7	126	1	30	42	108	66
218	–	–	–	–	–	–	–	–	–	–	2	109	2	2	219	1	108	109	108	–
219	219	1	10	511	14	28	23	105	109	4	3	73	3	3	220	1	72	75	109	34
220	220	1	9	255	9	28	23	53	109	56	220	1	9	255	9	35	30	51	109	58
221	221	1	9	255	11	28	23	53	110	57	221	1	9	255	11	35	30	51	110	59
222	–	–	–	–	–	–	–	–	–	–	6	37	4	7	149	1	36	48	110	62
224	28	8	6	31	49	3	21	52	111	59	7	32	4	7	130	1	31	43	111	68
225	225	1	9	255	12	28	23	53	112	59	15	15	5	15	78	2	28	45	112	67
226	–	–	–	–	–	–	–	–	–	–	2	113	2	2	227	1	112	113	112	–
228	228	1	9	255	9	28	23	53	113	60	6	38	4	7	153	1	37	49	113	64
230	230	1	9	255	9	28	23	53	114	61	230	1	9	255	9	35	30	51	114	63
231	231	1	9	255	11	28	23	53	115	62	7	33	4	7	134	1	32	44	115	71
232	–	–	–	–	–	–	–	–	–	–	4	58	3	4	174	1	57	62	115	53
234	234	1	9	255	9	28	23	53	116	63	6	39	4	7	157	1	38	50	116	66
235	–	–	–	–	–	–	–	–	–	–	5	47	5	15	236	1	46	86	117	31
236	–	–	–	–	–	–	–	–	–	–	4	59	3	4	177	1	58	63	117	54
237	–	–	–	–	–	–	–	–	–	–	3	79	3	3	238	1	78	81	118	37
238	238	1	9	255	10	28	23	53	118	65	7	34	4	7	138	1	33	45	118	73
240	30	8	6	31	51	3	21	52	119	67	15	16	5	15	83	2	30	47	119	72
242	242	1	11	1023	11	28	23	232	120	–	2	121	2	2	243	1	120	121	120	–
243	243	1	9	255	10	28	23	53	121	68	243	1	9	255	10	35	30	51	121	70
244	–	–	–	–	–	–	–	–	–	–	4	61	3	4	183	1	60	65	121	56
245	49	5	7	63	36	5	22	65	122	57	7	35	4	7	142	1	34	46	122	76
246	–	–	–	–	–	–	–	–	–	–	6	41	4	7	165	1	40	52	122	70
247	247	1	13	4095	14	28	23	1208	123	–	247	1	13	4095	14	120	113	307	123	–
248	31	8	6	31	52	3	21	52	123	71	31	8	6	31	52	4	29	49	123	74
249	–	–	–	–	–	–	–	–	–	–	3	83	3	3	250	1	82	85	124	39
250	250	1	9	255	9	28	23	53	124	71	250	1	9	255	9	35	30	51	124	73
252	252	1	9	255	13	28	23	53	125	72	7	36	4	7	146	1	35	47	125	78
253	253	1	11	1023	12	28	23	232	126	–	253	1	11	1023	12	77	71	124	126	2
254	254	1	9	255	13	28	23	53	126	73	127	2	8	127	21	17	29	51	126	75
255	255	1	9	255	15	28	23	53	127	74	15	17	5	15	88	2	32	49	127	78
256	256	1	9	255	8	28	23	53	127	74	256	1	9	255	8	35	30	51	127	76
258	–	–	–	–	–	–	–	–	–	–	6	43	4	7	173	1	42	54	128	74
259	–	–	–	–	–	–	–	–	–	–	7	37	4	7	150	1	36	48	129	81
260	260	1	9	255	9	28	23	53	129	76	260	1	9	255	9	35	30	51	129	78
261	261	1	9	255	10	28	23	53	130	77	261	1	9	255	10	35	30	51	130	79
262	–	–	–	–	–	–	–	–	–	–	2	131	2	2	263	1	130	131	130	–
264	264	1	9	255	10	28	23	53	131	78	132	2	8	127	17	17	29	51	131	80
265	–	–	–	–	–	–	–	–	–	–	5	53	5	15	266	1	52	92	132	40
266	14	19	5	15	97	1	18	58	132	74	7	38	4	7	154	1	37	49	132	83
267	267	1	12	2047	16	28	23	529	133	–	3	89	3	3	268	1	88	91	133	42
268	–	–	–	–	–	–	–	–	–	–	4	67	3	4	201	1	66	71	133	62
270	270	1	9	255	12	28	23	53	134	81	15	18	5	15	93	2	34	51	134	83
272	272	1	9	255	10	28	23	53	135	82	272	1	9	255	10	35	30	51	135	84
273	273	1	9	255	11	28	23	53	136	83	7	39	4	7	158	1	38	50	136	86
274	–	–	–	–	–	–	–	–	–	–	2	137	2	2	275	1	136	137	136	–

Table 1. (continued)

	$EG1$										$EG2$									
N	n	l	m	g	l	t	F	T	ρ	$D1$	n	l	m	g	l	t	F	T	ρ	$D2$
275	275	1	11	1023	12	28	23	232	137	–	5	55	5	15	276	1	54	94	137	43
276	276	1	9	256	9	28	23	53	137	84	276	1	9	256	9	35	30	51	137	86
278	–	–	–	–	–	–	–	–	–	–	2	139	2	2	279	1	138	139	138	–
279	31	9	6	31	58	2	16	67	139	72	31	9	6	31	58	4	33	53	139	86
280	14	20	5	15	102	1	19	59	139	80	7	40	4	7	162	1	39	51	139	88
282	–	–	–	–	–	–	–	–	–	–	6	47	4	7	189	1	46	58	140	82
284	–	–	–	–	–	–	–	–	–	–	4	71	3	4	213	1	70	75	141	66
285	15	19	5	15	98	1	18	58	142	84	15	19	5	15	98	2	36	53	142	89
286	286	1	11	1023	11	28	23	232	142	–	286	1	11	1023	11	77	71	124	142	18
287	–	–	–	–	–	–	–	–	–	–	7	41	4	7	166	1	40	52	143	91
288	12	24	5	15	121	1	23	63	143	80	6	48	4	7	193	1	47	59	143	84
289	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
290	290	1	10	511	10	28	23	105	144	39	10	29	5	15	145	1	28	68	144	76
291	–	–	–	–	–	–	–	–	–	–	3	97	3	3	292	1	96	99	145	46
292	292	1	10	511	13	28	23	105	145	40	4	73	3	4	219	1	72	77	145	68
294	14	21	5	15	107	1	20	60	146	86	7	42	4	7	170	1	41	53	146	93
295	–	–	–	–	–	–	–	–	–	–	5	59	5	15	296	1	58	98	147	49
296	–	–	–	–	–	–	–	–	–	–	4	74	3	4	222	1	73	78	147	69
297	27	11	7	63	78	2	20	157	148	–	27	11	7	63	78	5	52	95	148	53
298	–	–	–	–	–	–	–	–	–	–	2	149	2	2	299	1	148	149	148	–
299	299	1	12	2047	14	28	23	529	149	–	299	1	12	2047	14	114	107	189	149	–
300	15	20	5	15	103	1	19	59	149	90	15	20	5	15	103	2	38	55	149	94
301	–	–	–	–	–	–	–	–	–	–	7	43	4	7	174	1	42	54	150	96
302	–	–	–	–	–	–	–	–	–	–	2	151	2	2	303	1	150	151	150	–
303	–	–	–	–	–	–	–	–	–	–	3	101	3	3	304	1	100	103	151	48
304	–	–	–	–	–	–	–	–	–	–	4	76	3	4	228	1	75	80	151	71
305	–	–	–	–	–	–	–	–	–	–	5	61	5	15	306	1	60	100	152	52
306	306	1	10	511	13	28	23	105	152	47	6	51	4	7	205	1	50	62	152	90
308	14	22	5	15	112	1	21	61	153	92	7	44	4	7	178	1	43	55	153	98
309	–	–	–	–	–	–	–	–	–	–	3	103	3	3	310	1	102	105	154	49
310	62	5	7	63	39	5	22	65	154	89	31	10	6	31	64	4	37	57	154	97
312	312	1	10	511	11	28	23	105	155	50	6	52	4	7	209	1	51	63	155	92
314	–	–	–	–	–	–	–	–	–	–	2	157	2	2	315	1	156	157	156	–
315	15	21	5	15	108	1	20	60	157	97	7	45	4	7	182	1	44	56	157	101
316	–	–	–	–	–	–	–	–	–	–	4	79	3	4	237	1	78	83	157	74
318	–	–	–	–	–	–	–	–	–	–	6	53	4	7	213	1	52	64	158	94
319	319	1	11	1023	12	28	23	232	159	–	319	1	11	1023	12	77	71	124	159	35
320	320	1	10	511	11	28	23	105	159	54	10	32	5	15	160	1	31	71	159	88
321	–	–	–	–	–	–	–	–	–	–	3	107	3	3	322	1	106	109	160	51
322	14	23	5	15	117	1	22	62	160	98	7	46	4	7	186	1	45	57	160	103
323	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
324	108	3	8	127	26	9	23	77	161	84	6	54	4	7	217	1	53	65	161	96
325	325	1	13	4095	16	28	23	1208	162	–	5	65	5	15	326	1	64	104	162	58
326	–	–	–	–	–	–	–	–	–	–	2	163	2	2	327	1	162	163	162	–
327	–	–	–	–	–	–	–	–	–	–	3	109	3	3	328	1	108	111	163	52
328	–	–	–	–	–	–	–	–	–	–	8	41	5	16	205	1	40	84	163	79
329	–	–	–	–	–	–	–	–	–	–	7	47	4	7	190	1	46	58	164	106
330	15	22	5	15	113	1	21	61	164	103	15	22	5	15	113	2	42	59	164	105
332	–	–	–	–	–	–	–	–	–	–	4	83	3	4	249	1	82	87	165	78
333	–	–	–	–	–	–	–	–	–	–	3	111	3	3	334	1	110	113	166	53
334	–	–	–	–	–	–	–	–	–	–	2	167	2	2	335	1	166	167	166	–
335	–	–	–	–	–	–	–	–	–	–	5	67	5	15	336	1	66	106	167	61
336	14	24	5	15	122	1	23	63	167	104	7	48	4	7	194	1	47	59	167	108
338	338	1	13	4095	13	28	23	1208	168	–	2	169	2	2	339	1	168	169	168	–
339	–	–	–	–	–	–	–	–	–	–	3	113	3	3	340	1	112	115	169	54
340	340	1	10	511	12	28	23	105	169	64	10	34	5	15	170	1	33	73	169	96

Table 1. (continued)

	$EG1$										$EG2$									
N	n	l	m	g	l	t	F	T	ρ	$D1$	n	l	m	g	l	t	F	T	ρ	$D2$
341	31	11	6	31	70	2	20	71	170	99	31	11	6	31	70	3	30	61	170	109
342	342	1	10	511	10	28	23	105	170	65	6	57	4	7	229	1	56	68	170	102
343	49	7	7	63	50	3	18	103	171	68	7	49	4	7	198	1	48	60	171	111
344	–	–	–	–	–	–	–	–	–	–	8	43	5	16	215	1	42	86	171	85
345	15	23	5	15	118	1	22	62	172	110	15	23	5	15	118	2	44	61	172	111
346	–	–	–	–	–	–	–	–	–	–	2	173	2	2	347	1	172	173	172	–
348	348	1	10	511	10	28	23	105	173	68	12	29	5	15	146	1	28	68	173	105
350	350	1	10	511	11	28	23	105	174	69	7	50	4	7	202	1	49	61	174	113
351	117	3	9	255	28	9	23	165	175	10	117	3	9	255	28	22	61	103	175	72
352	352	1	11	1023	12	28	23	232	175	–	8	44	5	16	220	1	43	87	175	88
354	–	–	–	–	–	–	–	–	–	–	6	59	4	7	237	1	58	70	176	106
355	–	–	–	–	–	–	–	–	–	–	5	71	5	15	356	1	70	110	177	67
356	356	1	12	2047	15	28	23	529	177	–	4	89	3	4	267	1	88	93	177	84
357	357	1	10	511	13	28	23	105	178	73	7	51	4	7	206	1	50	62	178	116
358	–	–	–	–	–	–	–	–	–	–	2	179	2	2	359	1	178	179	178	–
360	15	24	5	15	123	1	23	63	179	116	15	24	5	15	123	2	46	63	179	116
361	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
362	–	–	–	–	–	–	–	–	–	–	2	181	2	2	363	1	180	181	180	–
363	363	1	11	1023	14	28	23	232	181	–	3	121	3	3	364	1	120	123	181	58
364	364	1	10	511	13	28	23	105	181	76	7	52	4	7	210	1	51	63	181	118
365	365	1	10	511	14	28	23	105	182	77	365	1	10	511	14	52	46	80	182	102
366	–	–	–	–	–	–	–	–	–	–	6	61	4	7	245	1	60	72	182	110
368	368	1	12	2047	13	28	23	529	183	–	8	46	5	16	230	1	45	89	183	94
369	–	–	–	–	–	–	–	–	–	–	3	123	3	3	370	1	122	125	184	59
370	370	1	10	511	10	28	23	105	184	79	10	37	5	15	185	1	36	76	184	108
371	–	–	–	–	–	–	–	–	–	–	7	53	4	7	214	1	52	64	185	121
372	31	12	6	31	76	2	22	73	185	112	31	12	6	31	76	3	33	64	185	121
374	374	1	10	511	11	28	23	105	186	81	187	2	9	255	20	26	47	80	186	106
375	375	1	11	1023	13	28	23	232	187	–	15	25	5	15	128	1	24	64	187	123
376	–	–	–	–	–	–	–	–	–	–	8	47	5	16	235	1	46	90	187	97
377	377	1	13	4095	14	28	23	1208	188	–	377	1	13	4095	14	120	113	307	188	–
378	126	3	8	127	29	9	23	77	188	111	7	54	4	7	218	1	53	65	188	123
380	380	1	10	511	9	28	23	105	189	84	10	38	5	15	190	1	37	77	189	112
381	127	3	8	127	29	9	23	77	190	113	127	3	8	127	29	14	37	66	190	124
382	–	–	–	–	–	–	–	–	–	–	2	191	2	2	383	1	190	191	190	–
384	384	1	10	511	11	28	23	105	191	86	12	32	5	15	161	1	31	71	191	120
385	35	11	7	63	78	2	20	157	192	35	7	55	4	7	222	1	54	66	192	126
386	–	–	–	–	–	–	–	–	–	–	2	193	2	2	387	1	192	193	192	–
387	–	–	–	–	–	–	–	–	–	–	3	129	3	3	388	1	128	131	193	62
388	–	–	–	–	–	–	–	–	–	–	4	97	3	4	291	1	96	101	193	92
390	390	1	10	511	13	28	23	105	194	89	15	26	5	15	133	1	25	65	194	129
391	391	1	12	2047	14	28	23	529	195	–	391	1	12	2047	14	114	107	189	195	6
392	56	7	7	63	52	3	18	103	195	92	7	56	4	7	226	1	55	67	195	128
393	–	–	–	–	–	–	–	–	–	–	3	131	3	3	394	1	130	133	196	63
394	–	–	–	–	–	–	–	–	–	–	2	197	2	2	395	1	196	197	196	–
395	–	–	–	–	–	–	–	–	–	–	5	79	5	15	396	1	78	118	197	79
396	132	3	8	127	25	9	23	77	197	120	132	3	8	127	25	14	37	66	197	131
398	–	–	–	–	–	–	–	–	–	–	2	199	2	2	399	1	198	199	198	–
399	399	1	10	511	13	28	23	105	199	94	7	57	4	7	230	1	56	68	199	131
400	400	1	10	511	11	28	23	105	199	94	10	40	5	15	200	1	39	79	199	120
402	–	–	–	–	–	–	–	–	–	–	6	67	4	7	269	1	66	78	200	122
403	31	13	6	31	82	1	12	125	201	76	31	13	6	31	82	3	36	67	201	134
404	–	–	–	–	–	–	–	–	–	–	4	101	3	4	303	1	100	105	201	96
405	45	9	7	63	66	2	16	153	202	49	15	27	5	15	138	1	26	66	202	136
406	406	1	10	511	11	28	23	105	202	97	14	29	5	15	147	1	28	68	202	134
407	407	1	11	1023	12	28	23	232	203	–	407	1	11	1023	12	77	71	124	203	79

Table 1. (continued)

	$EG1$										$EG2$									
N	n	l	m	g	l	t	F	T	ρ	$D1$	n	l	m	g	l	t	F	T	ρ	$D2$
408	408	1	10	511	12	28	23	105	203	98	12	34	5	15	171	1	33	73	203	130
410	410	1	10	511	10	28	23	105	204	99	10	41	5	15	205	1	40	80	204	124
411	–	–	–	–	–	–	–	–	–	–	3	137	3	3	412	1	136	139	205	66
412	–	–	–	–	–	–	–	–	–	–	4	103	3	4	309	1	102	107	205	98
413	–	–	–	–	–	–	–	–	–	–	7	59	4	7	238	1	58	70	206	136
414	414	1	10	511	10	28	23	105	206	101	6	69	4	7	277	1	68	80	206	126
415	–	–	–	–	–	–	–	–	–	–	5	83	5	15	416	1	82	122	207	85
416	416	1	13	4095	14	28	23	1208	207	–	8	52	5	16	260	1	51	95	207	112
417	–	–	–	–	–	–	–	–	–	–	3	139	3	3	418	1	138	141	208	67
418	418	1	11	1023	11	28	23	232	208	–	418	1	11	1023	11	77	71	124	208	84
420	60	7	7	63	52	3	18	103	209	106	15	28	5	15	143	1	27	67	209	142
422	–	–	–	–	–	–	–	–	–	–	2	211	2	2	423	1	210	211	210	–
423	–	–	–	–	–	–	–	–	–	–	3	141	3	3	424	1	140	143	211	68
424	–	–	–	–	–	–	–	–	–	–	8	53	5	16	265	1	52	96	211	115
425	85	5	9	255	49	5	22	315	212	–	5	85	5	15	426	1	84	124	212	88
426	–	–	–	–	–	–	–	–	–	–	6	71	4	7	285	1	70	82	212	130
427	–	–	–	–	–	–	–	–	–	–	7	61	4	7	246	1	60	72	213	141
428	–	–	–	–	–	–	–	–	–	–	4	107	3	4	321	1	106	111	213	102
429	429	1	11	1023	14	28	23	232	214	–	429	1	11	1023	14	77	71	124	214	90
430	430	1	10	511	10	28	23	105	214	109	430	1	10	511	10	52	46	80	214	134
432	432	1	10	511	12	28	23	105	215	110	12	36	5	15	181	1	35	75	215	140
434	62	7	7	63	53	3	18	103	216	113	14	31	5	15	157	1	30	70	216	146
435	435	1	11	1023	13	28	23	232	217	–	15	29	5	15	148	1	28	68	217	149
436	–	–	–	–	–	–	–	–	–	–	4	109	3	4	327	1	108	113	217	104
437	437	1	12	2047	14	28	23	529	218	–	437	1	12	2047	14	114	107	189	218	29
438	438	1	10	511	13	28	23	105	218	113	438	1	10	511	13	52	46	80	218	138
440	440	1	10	511	10	28	23	105	219	114	220	2	9	255	18	26	47	80	219	139
441	63	7	7	63	53	3	18	103	220	117	63	7	7	63	53	6	38	72	220	148
442	442	1	10	511	11	28	23	105	220	115	221	2	9	255	20	26	47	80	220	140
444	444	1	10	511	10	28	23	105	221	116	12	37	5	15	186	1	36	76	221	145
445	445	1	12	2047	16	28	23	529	222	–	5	89	5	15	446	1	88	128	222	94
446	–	–	–	–	–	–	–	–	–	–	2	223	2	2	447	1	222	223	222	–
447	–	–	–	–	–	–	–	–	–	–	3	149	3	3	448	1	148	151	223	72
448	448	1	10	511	13	28	23	105	223	118	14	32	5	15	162	1	31	71	223	152
450	450	1	10	511	13	28	23	105	224	119	15	30	5	15	153	1	29	69	224	155
451	451	1	11	1023	12	28	23	232	225	–	451	1	11	1023	12	77	71	124	225	101
452	–	–	–	–	–	–	–	–	–	–	4	113	3	4	339	1	112	117	225	108
453	–	–	–	–	–	–	–	–	–	–	3	151	3	3	454	1	150	153	226	73
454	–	–	–	–	–	–	–	–	–	–	2	227	2	2	455	1	226	227	226	–
455	455	1	11	1023	12	28	23	232	227	–	7	65	4	7	262	1	64	76	227	151
456	456	1	10	511	11	28	23	105	227	122	12	38	5	15	191	1	37	77	227	150
458	–	–	–	–	–	–	–	–	–	–	2	229	2	2	459	1	228	229	228	–
459	153	3	9	255	31	9	23	165	229	64	153	3	9	255	31	22	61	103	229	126
460	460	1	10	511	9	28	23	105	229	124	230	2	9	255	18	26	47	80	229	149
462	462	1	10	511	13	28	23	105	230	125	14	33	5	15	167	1	32	72	230	158
464	–	–	–	–	–	–	–	–	–	–	8	58	5	16	290	1	57	101	231	130
465	465	1	10	511	15	28	23	105	232	127	15	31	5	15	158	1	30	70	232	162
466	–	–	–	–	–	–	–	–	–	–	2	233	2	2	467	1	232	233	232	–
468	468	1	10	511	11	28	23	105	233	128	12	39	5	15	196	1	38	78	233	155
469	–	–	–	–	–	–	–	–	–	–	7	67	4	7	270	1	66	78	234	156
470	470	1	10	511	10	28	23	105	234	129	470	1	10	511	10	52	46	80	234	154
471	–	–	–	–	–	–	–	–	–	–	3	157	3	3	472	1	156	159	235	76
472	–	–	–	–	–	–	–	–	–	–	8	59	5	16	295	1	58	102	235	133
473	473	1	11	1023	12	28	23	232	236	4	473	1	11	1023	12	77	71	124	236	112
474	–	–	–	–	–	–	–	–	–	–	6	79	4	7	317	1	78	90	236	146
475	–	–	–	–	–	–	–	–	–	–	5	95	5	15	476	1	94	134	237	103

Table 1. (continued)

	$EG1$										$EG2$									
N	n	l	m	g	I	t	F	T	ρ	$D1$	n	l	m	g	I	t	F	T	ρ	$D2$
476	476	1	10	511	13	28	23	105	237	132	14	34	5	15	172	1	33	73	237	164
477	–	–	–	–	–	–	–	–	–	–	3	159	3	3	478	1	158	161	238	77
478	–	–	–	–	–	–	–	–	–	–	2	239	2	2	479	1	238	239	238	–
480	480	1	10	511	13	28	23	105	239	134	15	32	5	15	163	1	31	71	239	168
481	481	1	13	4095	14	28	23	1208	240	–	481	1	13	4095	14	120	113	307	240	–
482	–	–	–	–	–	–	–	–	–	–	2	241	2	2	483	1	240	241	240	–
483	483	1	10	511	13	28	23	105	241	136	7	69	4	7	278	1	68	80	241	161
484	484	1	11	1023	11	28	23	232	241	9	484	1	11	1023	11	77	71	124	241	117
485	–	–	–	–	–	–	–	–	–	–	5	97	5	15	486	1	96	136	242	106
486	486	1	10	511	10	28	23	105	242	137	243	2	9	255	19	26	47	80	242	162
488	–	–	–	–	–	–	–	–	–	–	8	61	5	16	305	1	60	104	243	139
489	–	–	–	–	–	–	–	–	–	–	3	163	3	3	490	1	162	165	244	79
490	490	1	10	511	11	28	23	105	244	139	14	35	5	15	177	1	34	74	244	170
492	492	1	10	511	10	28	23	105	245	140	12	41	5	15	206	1	40	80	245	165
493	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
494	494	1	13	4095	13	28	23	1208	246	–	2	247	2	2	495	1	246	247	246	–
495	45	11	7	63	80	2	20	157	247	90	15	33	5	15	168	1	32	72	247	175
496	496	1	10	511	12	28	23	105	247	142	31	16	6	31	100	3	45	76	247	171
497	–	–	–	–	–	–	–	–	–	–	7	71	4	7	286	1	70	82	248	166
498	–	–	–	–	–	–	–	–	–	–	6	83	4	7	333	1	82	94	248	154
500	500	1	10	511	9	28	23	105	249	144	250	2	9	255	18	26	47	80	249	169
501	–	–	–	–	–	–	–	–	–	–	3	167	3	3	502	1	166	169	250	81
502	–	–	–	–	–	–	–	–	–	–	2	251	2	2	503	1	250	251	250	–
504	504	1	10	511	15	28	23	105	251	146	14	36	5	15	182	1	35	75	251	176
505	–	–	–	–	–	–	–	–	–	–	5	101	5	15	506	1	100	140	252	112
506	506	1	11	1023	11	28	23	232	252	20	506	1	11	1023	11	77	71	124	252	128
507	507	1	13	4095	16	28	23	1208	253	–	3	169	3	3	508	1	168	171	253	82
508	508	1	10	511	13	28	23	105	253	148	127	4	8	127	37	13	47	80	253	173
510	510	1	10	511	15	28	23	105	254	149	15	34	5	15	173	1	33	73	254	181
511	511	1	10	511	17	28	23	105	255	150	511	1	10	511	17	52	46	80	255	175
512	–	–	–	–	–	–	–	–	–	–	8	64	5	16	320	1	63	107	255	148
513	171	3	9	255	28	9	23	165	256	91	171	3	9	255	28	22	61	103	256	153
514	–	–	–	–	–	–	–	–	–	–	2	257	2	2	515	1	256	257	256	–
515	–	–	–	–	–	–	–	–	–	–	5	103	5	15	516	1	102	142	257	115
516	516	1	10	511	10	28	23	105	257	152	516	1	10	511	10	52	46	80	257	177
517	517	1	11	1023	12	28	23	232	258	26	517	1	11	1023	12	77	71	124	258	134
518	518	1	10	511	11	28	23	105	258	153	14	37	5	15	187	1	36	76	258	182
519	–	–	–	–	–	–	–	–	–	–	3	173	3	3	520	1	172	175	259	84
520	520	1	10	511	10	28	23	105	259	154	260	2	9	255	18	26	47	80	259	179
522	522	1	10	511	10	28	23	105	260	155	261	2	9	255	19	26	47	80	260	180
524	–	–	–	–	–	–	–	–	–	–	4	131	3	4	393	1	130	135	261	126
525	525	1	10	511	14	28	23	105	262	157	15	35	5	15	178	1	34	74	262	188
526	–	–	–	–	–	–	–	–	–	–	2	263	2	2	527	1	262	263	262	–
527	31	17	6	31	106	1	16	129	263	134	31	17	6	31	106	3	48	79	263	184
528	528	1	10	511	11	28	23	105	263	158	132	4	8	128	33	13	47	80	263	183
529	529	1	12	2047	14	28	23	529	264	–	529	1	12	2047	14	114	107	189	264	75
530	530	1	10	511	10	28	23	105	264	159	530	1	10	511	10	52	46	80	264	184
531	–	–	–	–	–	–	–	–	–	–	3	177	3	3	532	1	176	179	265	86
532	532	1	10	511	13	28	23	105	265	160	14	38	5	15	192	1	37	77	265	188
533	533	1	13	4095	14	28	23	1208	266	–	533	1	13	4095	14	120	113	307	266	–
534	534	1	12	2048	15	28	23	529	266	–	6	89	4	7	357	1	88	100	266	166
535	–	–	–	–	–	–	–	–	–	–	5	107	5	15	536	1	106	146	267	121
536	–	–	–	–	–	–	–	–	–	–	8	67	5	16	335	1	66	110	267	157
537	–	–	–	–	–	–	–	–	–	–	3	179	3	3	538	1	178	181	268	87
538	–	–	–	–	–	–	–	–	–	–	2	269	2	2	539	1	268	269	268	–
539	49	11	7	63	78	2	20	157	269	112	7	77	4	7	310	1	76	88	269	181

Table 1. (continued)

	$EG1$										$EG2$									
N	n	l	m	g	I	t	F	T	ρ	$D1$	n	l	m	g	I	t	F	T	ρ	$D2$
540	30	18	6	31	111	1	17	130	269	139	15	36	5	15	183	1	35	75	269	194
542	–	–	–	–	–	–	–	–	–	–	2	271	2	2	543	1	270	271	270	–
543	–	–	–	–	–	–	–	–	–	–	3	181	3	3	544	1	180	183	271	88
544	544	1	11	1023	12	28	23	232	271	39	8	68	5	16	340	1	67	111	271	160
545	–	–	–	–	–	–	–	–	–	–	5	109	5	15	546	1	108	148	272	124
546	546	1	11	1023	14	28	23	232	272	40	14	39	5	15	197	1	38	78	272	194
548	–	–	–	–	–	–	–	–	–	–	4	137	3	4	411	1	136	141	273	132
549	–	–	–	–	–	–	–	–	–	–	3	183	3	3	550	1	182	185	274	89
550	550	1	11	1023	11	28	23	232	274	42	10	55	5	15	275	1	54	94	274	180
551	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
552	24	23	6	31	140	1	22	135	275	140	12	46	5	15	231	1	45	85	275	190
553	–	–	–	–	–	–	–	–	–	–	7	79	4	7	318	1	78	90	276	186
554	–	–	–	–	–	–	–	–	–	–	2	277	2	2	555	1	276	277	276	–
555	555	1	11	1023	13	28	23	232	277	45	15	37	5	15	188	1	36	76	277	201
556	–	–	–	–	–	–	–	–	–	–	4	139	3	4	417	1	138	143	277	134
558	31	18	6	31	112	1	17	130	278	148	31	18	6	31	112	3	51	82	278	196
559	559	1	13	4095	14	28	23	1208	279	–	559	1	13	4095	14	120	113	307	279	–
560	28	20	6	31	121	1	19	132	279	147	14	40	5	15	202	1	39	79	279	200
561	187	3	9	255	29	9	23	165	280	115	187	3	9	255	29	22	61	103	280	177
562	–	–	–	–	–	–	–	–	–	–	2	281	2	2	563	1	280	281	280	–
564	564	1	11	1023	10	28	23	232	281	49	12	47	5	15	236	1	46	86	281	195
565	–	–	–	–	–	–	–	–	–	–	5	113	5	15	566	1	112	152	282	130
566	–	–	–	–	–	–	–	–	–	–	2	283	2	2	567	1	282	283	282	–
567	63	9	7	63	67	2	16	153	283	130	63	9	7	63	67	6	50	84	283	199
568	–	–	–	–	–	–	–	–	–	–	8	71	5	16	355	1	70	114	283	169
570	30	19	6	31	117	1	18	131	284	153	15	38	5	15	193	1	37	77	284	207
572	572	1	11	1023	11	28	23	232	285	53	572	1	11	1023	11	77	71	124	285	161
573	–	–	–	–	–	–	–	–	–	–	3	191	3	3	574	1	190	193	286	93
574	574	1	11	1023	12	28	23	232	286	54	14	41	5	15	207	1	40	80	286	206
575	575	1	12	2047	14	28	23	529	287	–	5	115	5	15	576	1	114	154	287	133
576	24	24	6	31	146	1	23	136	287	151	12	48	5	15	241	1	47	87	287	200
578	–	–	–	–	–	–	–	–	–	–	2	289	2	2	579	1	288	289	288	–
579	–	–	–	–	–	–	–	–	–	–	3	193	3	3	580	1	192	195	289	94
580	580	1	11	1023	10	28	23	232	289	57	10	58	5	15	290	1	57	97	289	192
581	–	–	–	–	–	–	–	–	–	–	7	83	4	7	334	1	82	94	290	196
582	–	–	–	–	–	–	–	–	–	–	6	97	4	7	389	1	96	108	290	182
583	583	1	11	1023	12	28	23	232	291	59	583	1	11	1023	12	77	71	124	291	167
584	584	1	11	1023	13	28	23	232	291	59	8	73	5	16	365	1	72	116	291	175
585	195	3	9	255	30	9	23	165	292	127	15	39	5	15	198	1	38	78	292	214
586	–	–	–	–	–	–	–	–	–	–	2	293	2	2	587	1	292	293	292	–
588	28	21	6	31	127	1	20	133	293	160	14	42	5	15	212	1	41	81	293	212
589	31	19	6	31	118	1	18	131	294	163	31	19	6	31	118	3	54	85	294	209
590	–	–	–	–	–	–	–	–	–	–	10	59	5	15	295	1	58	98	294	196
591	–	–	–	–	–	–	–	–	–	–	3	197	3	3	592	1	196	199	295	96
592	–	–	–	–	–	–	–	–	–	–	8	74	5	16	370	1	73	117	295	178
594	54	11	7	63	77	2	20	157	296	139	54	11	7	63	77	5	52	95	296	201
595	595	1	11	1023	12	28	23	232	297	65	7	85	4	7	342	1	84	96	297	201
596	–	–	–	–	–	–	–	–	–	–	4	149	3	4	447	1	148	153	297	144
597	–	–	–	–	–	–	–	–	–	–	3	199	3	3	598	1	198	201	298	97
598	598	1	12	2047	13	28	23	529	298	–	598	1	12	2047	13	114	107	189	298	109
600	30	20	6	31	123	1	19	132	299	167	15	40	5	15	203	1	39	79	299	220

Cryptanalysis of Imai and Matsumoto Scheme B Asymmetric Cryptosystem

Amr Youssef[1] and Guang Gong[2]

[1] Center for Applied Cryptographic Research,
Department of Combinatorics & Optimization,
University of Waterloo, Waterloo, Ontario N2L 3G1, Canada,
[2] Center for Applied Cryptographic Research,
Department of Electrical and Computer Engineering,
University of Waterloo, Waterloo, Ontario N2L 3G1, Canada,
{a2youssef,ggong}@cacr.math.uwaterloo.ca

Abstract. Imai and Matsumoto introduced alternative algebraic methods for constructing public key cryptosystems. An obvious advantage of theses public key cryptosystems is that the private side computations can be made very efficient with a simple hardware. Almost all of these proposals and variants of them were broken. However, scheme "B" in [3] is still unbroken. In this paper we show some statistical weaknesses of this scheme. In particular, we show that trying to minimize the size of the public key facilitates a cryptanalytic attack that enables the cryptanalyst to decrypt, with high probability of success, a given ciphertext by performing a very limited number of encryption operations using the public encryption function.

Keywords: Public-key cryptosystems , cryptanalysis, Imai and Matsumoto asymmetric cryptosystems

1 Introduction

Public key cryptosystems based on integer factorization and discrete log problem, such as RSA and ElGamal [7], need to perform a large amount of arithmetic operations, so they are not very efficient compared to symmetric key cryptosystems such as DES. Imai and Matsumoto [3] [6] and Matsumoto *et. al.* [5] introduced alternative algebraic methods for constructing public key cryptosystems. An obvious advantage of theses public key cryptosystems is that the private side computations (decrypting and signing) can be made very efficient with a simple hardware. Almost all of these proposals and variants of them were broken (see [1], [2], [8], [9] [10] [11] [12]). However, as noted in [2], scheme "B" in [3], which was originally proposed by Matsumoto *et. al.* in [5] is still unbroken. In this paper we introduce a piecewise affine approximation attack on this scheme and show that it is insecure.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 214–222, 2001.

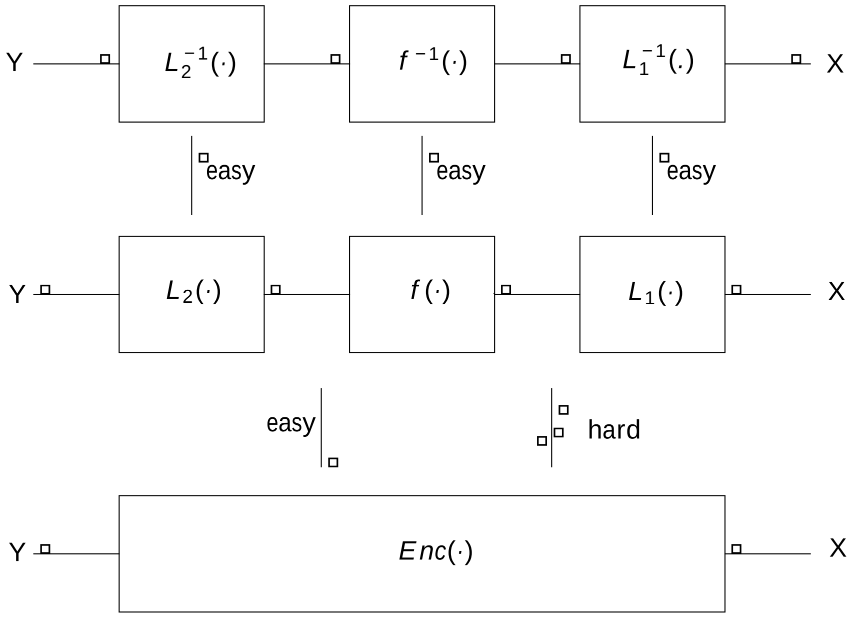


Fig. 1. The basic idea of Imai and Matsumoto Scheme B

2 Description of Scheme "B" in [3]

For a given block length n, the encryption function of Scheme "B" in [3] is composed of

$$L_1 \circ f \circ L_2 \tag{1}$$

where L_1 and L_2 are two secret bijective linear mappings over $GF(2)^n$ and

$$f(x) = \begin{cases} (x + c - 1) \bmod (2^n - 1) + 1, & x = 0 \\ 0, & x = 0, \end{cases} \tag{2}$$

where c is a secret positive integer whose binary representation has *small* Hamming weight, wt(c). The main reason to choose c with a small Hamming weight is to reduce the size of the public key [3]. The encryption of x is given by

$$Enc(x) = L_2(f(L_1(x))).$$

The private key is L_1, L_2 and c. The public key is an AND-XOR array pattern for the m-tuple of m-variate sparse polynomials over $GF(2)$ representing the composite function $Enc(\cdot)$. As mentioned above, small values for $wt(c)$ is required to reduce the public key size. This restriction is the basic motive for our attack. The security of this scheme (see Figure 1) relies on the fact that the transformations L_1, L_2 and f operate on two different algebraic structures ($GF(2)^n$ and the non-negative set of integers $< 2^n$) . Thus the $Enc^{-1}(\cdot)$ is assumed to have a complex representation when considered as a mapping over only one of these two structures. In other words, it is assumed that it is difficult to obtain any simple algebraic description for the function $Enc^{-1}(\cdot)$ given only the AND-XOR array of the function $Enc(\cdot)$. The size of the public and secret key bits and the complexity of the encryption and decryption operations are all $O(n^2)$.

3 Observations

Our attack is based on the following observation

Observation 1 *For a small Hamming weight of c, the piecewise affine approximation of the function f in equation (2) has small number of affine segments over $GF(2)^n$ compared to that of a randomly selected bijective mapping. Moreover, most of the points belong to a small number of segments, i.e., a small number of segments is enough to achieve a good approximation accuracy.*

Example 1. Let $n = 8$ and $c = 3$ with Hamming weight 2. Then for $x = 0$, the binary representation of f belong to one of the following piecewise affine functions

$$l_i(x) = x \quad d_i$$

where $d_i \quad \{3, 5, 7, 13, 15, 29, 31, 61, 63, 125, 127, 252, 253, 255\}$. The number of points on each segment is shown in Table 1. It is clear that the approximation accuracy using the first two constants is about 50%. Using the first 8 constants the accuracy increases to about 94%.

Table 1. The affine constants for Example 1

c	5	3	7	13	29	15	61	31	63	125	252	253	127	255
No. of points	64	63	32	32	16	16	8	8	4	4	3	2	2	1

Example 2. Let $n = 16$ and $c = 1056$ with Hamming weight 2, then more than 90% of the points corresponding to the binary representation of the function belong to one of the following the affine segments

$$l_i(x) = x \quad d_i,$$

where

$$d_i \quad \{15456, 2016, 3040, 31776, 3552, 7392, 1504,$$
$$15392, 3296, 7264, 1248, 3168, 7200, 1120, 3104, 1056\}.$$

Table 2 shows the expected number of segments for $n = 8, 10, 12$. The average number of segments for a randomly selected bijective mapping (obtained by our experiments) is around 176, 690 and 2778 for $n = 8, 10$ and 12 respectively. In

Table 2. Average Number of Segments in the piecewise approximation of f

wt(c)	1	2	3	4	5	6	7	8	9	10	11	12
n=8	9	19	28	31.63	28	19	9	1				
n=10	11	27.33	47.5	64.43	71.06	64.43	47.5	27.33	11	1		
n=12	13	37	73.4	114.4	147.17	159.72	147.17	114.4	73.4	37	13	1

all of our experiments with small values of $wt(c)$, there always exists an affine relation satisfied by $\geq 2^{(n-c)} - 1$ points.

Using Observation 1 the Encryption function (and consequently the decryption function) can be divided into M affine segments $Enc_i(\cdot)$, $i = 1, 2, \cdots M$ where $Enc_i(x)$ given by

$$Enc_i(x) = L_2(L_1(x) \oplus d_i) = L_2L_1(x) \oplus b_i,$$

and $b_i = L_2(d_i)$, $i = 1, 2, \cdots M$. Thus for any specific L_1, L_2, c, the input (plaintext) space can be partitioned into M sets such that the ciphertext Y of each set is related to the plaintext X by an affine relation

$$Y = AX \oplus b_i,$$

where $A = L_2 \circ L1$. The expected value of M is small for c with small Hamming weight.

Remark 1. Probabilistic interpolation attacks [4] based on Sudan's algorithm [13], which operates over $GF(2^n)$, cannot be used to recover these affine segments since the affine function over $GF(2)^n$ will have a high degree when considered as a function over $GF(2^n)$ [14].

In the following, we will describe the basic step in the attack. We use a differential-like attack to group pairs that belong to the same segment. Figure 2 shows the algorithm that enables us to do so. We pick random triples R_1, R_2 and R_3 and test for the condition

$$Enc(R_1) \oplus Enc(R_2) \oplus Enc(R_3) \oplus Enc((R_3 \oplus (R_1 \oplus R_2))) = 0.$$

This condition is satisfied if $R1, R2, R3$ and $R1 \oplus R2 \oplus R3$ are all on one affine segment. Since there is no guarantee that $R_3, (R_3 \oplus (R_1 \oplus R_2))$ will belong to segment S_i even if R_1 and R_2 do, we repeat the test for different values of R_3 ($Trials$ in Figure 2). We decide that R_1 and R_2 belong to the same segment if the equation above is satisfied for a large number of times ($Threshold$ in Figure 2). To prevent the algorithm from accepting wrong pairs we may increase the value of $Trials$ and make the value of $Threshold$ very close to $Trials$. However, very large values for $Trials$ increases the number of plaintext-ciphertext pairs required to break the algorithm. Throughout our experiments, we set $Threshold = Trials$.

One can prove that the plaintext that belong to the same linear segment are not linearly independent and hence the matrix A cannot be uniquely determined

```
$R_1 = Random()$
$R_2 = Random()$
$pass = 0$
$\delta_x = R_1 \quad R_2$
for $i = 1$ to $i = Trials$
{
    $R_3 = Random()$
    $R_4 = R_3 \quad \delta_x$
    $\delta_y = Enc(R_1) \quad Enc(R_2) \quad Enc(R_3) \quad Enc(R_4)$
    if ($\delta_y = 0$) increment pass
}
if(pass $\geq Threshold$) Declare $R_1$ and $R_2$ same set
```

Fig. 2. The Basic Step in the Attack

by collecting plaintext-ciphertext pairs on one segment. In fact, our experiments show that the matrix A cannot be uniquely determined by any reasonable number of queries to the encryption function. So our attack doesn't try to find such unique solution for A.

4 The Attack

Let x_1, x_2 be on the same affine segment S_i. Then

$$Enc(x_1) \quad Enc(x_2) = Ax_1 \quad b_i \quad Ax_2 \quad b_i = A(x_1 \quad x_2)$$

which is independent of the segment they belong to and depends only on the difference $(x_1 \quad x_2)$. Our attack proceeds as follows:

1. Use the basic step in Figure 2 to pick any two plaintext points $(x, x \quad \delta_x)$ that are on the same segment. Collect enough number of (δ_x, δ_y) pairs for linearly independent δ_x's.
2. Solve for the matrix B that satisfy the linear relation

$$\delta_x = B \times \delta_y$$

   The coverage of the attack (i.e., the probability of being able to decrypt a random ciphertext) increases exponentially with the number of pairs collected in step 1. (*Remark.* Note that $(L_2L_1)^{-1}$ is not the only valid solution for B).

After determining this linear relation between δx's and δ_y's we can decrypt any given ciphertext u as follows:

1. Pick random x and assume that it is on the same segment with $Dec(u)$.

2. Calculate
$$TrialDec(u) = x \quad B \times (u \quad Enc(x)) \qquad (3)$$
3. Using the public encryption function, verify if $Enc(TrialDec(u)) = u$.

If yes, then we have found $Dec(u)$. If no, then pick a different x and repeat the steps above.

Relation 3 holds if x and $Dec(u)$ belong to the same segment and this happens with high probability because we have a small number of segments.

One should note that deriving an accurate theoretical estimate for the number of encryption operations required to achieve certain coverage is difficult because the $Enc(\cdot)$ function doesn't behave like a random function. Table 3 and Table 4 show the result of some of our experiments for $wt(c) = 1, 2$ and $n = 16, 18, 20$. The tables show the number of queries (to the public encryption function) that are required to successfully decrypt more than 50% of a random sample of 100 ciphertext. Increasing the coverage close to 99% requires a slight increase in the number of collected pairs. For example, for $n = 20, wt(c) = 1$, only a total of 866 and 900 encryption operations were required to increase the coverage to 92% and 98% respectively. Let the fraction P denote the number of chosen plaintext-ciphertext pairs required to achieve certain coverage. Then , our experimental results show that, on average and for a fixed small Hamming weight of c, $P/2^n$ decreases dramatically with n.

Table 3. Experimental Results for $wt(c) = 1$

n	Number of Encryption Operations	Coverage
16	813	56%
18	670	66%
20	630	64%

Table 4. Experimental Results for $wt(c) = 2$

n	Number of Encryption Operations	Coverage
16	2418	61%
18	2605	51%
20	3525	61%

5 Conclusion

For some selections of the algorithm parameter c, the encryption and decryption operations in Scheme B proposed by *Imai et. al.* can be approximated by a piecewise affine function over $GF(2)^n$ with small number of affine segments. Trying to minimize the size of the public key by using a very small Hamming weight for the algorithm parameter c reduces the number of theses affine segments and may compromise the security of the algorithm. It should be noted that avoiding such selections for c, while may increase the size of the public key, makes the algorithm totally secure against our attack and the security of this scheme still remains an open problem.

References

1. E. Biham *Cryptanalysis of Patarin's 2-Round Public Key System with S Boxes (2R)*, Advances in Cryptology, Proceedings of EUROCRYPT'2000, LNCS 1807, pp. 408-416, Springer-Verlag, 2000.
2. Y. Feng, L. Yan and D. Duo *Cryptanalysis of "2R" Schemes*, Advances in Cryptology, Proceedings of CRYPTO'99, LNCS1666 , pp. 315-325, Springer-Verlag, 1999.
3. H. Imai and T. Matsumoto, *Algebraic methods for constructing asymmetric cryptosystems*, Proceedings of Algebraic Algorithms and error-correcting codes (AAECC-3), Springer-Verlag, LNCS 229 , pp. 108-119
4. T. Jakobsen, *Cryptanalysis of Block Ciphers with Probabilistic Non-linear Relations of Low Degree*, Proceedings of CRYPTO'99, LNCS 1462, pp. 213-222, 1999.
5. T. Matsumoto, H. Imai, H. Harashima and H. Miyakawa *A high-speed asymmetric cryptosystem with obscure public-keys*, Paper of Technical group, TGAL84-83, Mar. 1985. (in Japanese)
6. T. Matsumoto and H. Imai *Public quadratic polynomial tuples for efficient signature verification and message encryption*, Advances in Cryptology, Proceedings of EUROCRYPT'88, LNCS330 , pp. 419-453, Springer-Verlag, 1988.
7. A J. Menezes, P. C. van Oorschot and S A. Vanstone *Handbook of Applied Cryptographic Research*, CRC Press, 1996.
8. J. Patarin, L. Goubin, and N. Courtois, *C*-+ and HM: Variations around two schemes of T. Matsumoto and H. Imai*, Advances in Cryptology, Proceedings of ASIACRYPT'98, Springer-Verlag, LNCS 1514, pp. 35-49, 1998.
9. J. Patarin and L. Goubin, *Asymmetric Cryptography with S-Boxes*, Proceedings of ICICS'97, Springer-Verlag, LNCS 1334, pp. 369-380, 1997.
10. J. Patarin, *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88*, Advances in Cryptology, Proceedings of CRYPTO '95, Springer-Verlag, LNCS 963, pp. 248-261, 1995.
11. J. Patarin, *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Asymmetric Algorithms*, Advances in Cryptology, Proceedings of EUROCRYPT'96, Springer-Verlag, LNCS 1070, pp. 33-48, 1996
12. J. Patarin and L. Goubin, *Trapdoor one-way permutations and multivariate polynomials*, Advances in Cryptology, Proceedings of ICICS'97, Springer-Veralg, LNCS 1334, pp. 356-368, 1997.
13. M. Sudan, *Decoding Reed Solomon Codes beyond the error-correction bound*, Journal of Complexity, Vol. 13, no 1, pp180-193, March, 1997.
14. A. M. Youssef and G. Gong, On the interpolation attacks on block ciphers, Proceedings of *Fast Software Encryption 2000* , Springer-Veralg, LNCS 1978, pp. 109-120, 2001 .

Appendix: A Detailed Example

In this section we give a detailed example for our attack on a toy version with $n = 20, C = 4096, wt(C) = 1$. L_1 and L_2 are given by

$$\begin{pmatrix}
1&0&0&0&0&1&1&1&1&1&1&1&0&1&1&0&1&0&0&0\\
0&0&1&0&0&0&1&0&1&0&1&1&0&0&0&1&0&1&0&0\\
0&1&1&1&0&1&1&1&1&1&0&1&1&1&0&1&1&0&1&1\\
0&1&0&0&0&0&1&0&0&0&0&1&1&1&1&0&1&1&0&1\\
0&0&1&1&1&1&0&0&1&0&1&0&0&0&1&1&1&1&0&1\\
1&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0&1&0&1&1\\
0&1&1&1&0&0&0&0&1&0&1&1&1&1&1&0&0&0&0&0\\
0&0&0&1&1&1&0&1&1&1&0&1&0&0&0&1&1&1&1&0\\
1&1&1&0&0&1&0&0&0&0&0&0&0&1&1&0&0&1&1&1\\
1&1&0&0&1&0&1&0&0&0&0&1&1&1&1&0&0&0&1&0\\
0&0&0&0&1&1&1&0&1&1&0&1&0&1&1&0&1&0&0&1\\
0&1&1&1&1&0&1&0&1&0&0&0&0&0&1&1&1&0&1&0\\
1&0&1&0&0&0&0&1&0&1&0&1&0&1&1&0&0&1&1&1\\
0&0&1&0&0&1&1&1&0&1&0&0&0&1&0&0&1&1&0&1\\
0&0&0&0&1&0&1&0&0&0&0&1&0&0&0&0&0&0&1&0\\
1&0&0&0&0&1&0&0&0&1&1&0&1&1&0&0&1&1&0&1\\
1&0&1&1&1&0&1&0&1&0&1&0&0&0&1&1&0&0&1&1\\
1&1&0&0&1&1&0&1&0&1&0&1&0&0&0&1&1&1&0&1\\
0&0&0&1&1&0&0&0&0&0&1&1&0&0&0&1&0&1&1&1\\
0&0&1&0&1&1&0&1&1&0&1&0&1&0&1&0&1&0&0&1
\end{pmatrix},
\begin{pmatrix}
1&1&1&0&1&1&1&0&1&1&0&1&0&1&1&0&0&0&1&0\\
1&0&0&0&1&1&0&1&0&0&1&1&1&1&1&1&1&1&0&0\\
1&1&0&0&0&1&0&0&1&1&0&1&0&0&0&1&1&1&1&1\\
0&1&1&1&0&1&1&1&0&0&0&0&0&1&1&1&0&0&1&1\\
1&0&1&1&1&0&0&0&0&1&1&1&1&0&1&0&1&0&0&0\\
0&0&0&1&0&1&1&0&1&0&0&0&1&0&1&0&0&0&1&0\\
0&0&1&1&1&0&1&0&0&1&0&0&1&1&0&0&1&0&1&0\\
1&1&0&1&1&0&1&0&0&0&0&1&1&1&1&0&0&1&1&1\\
1&0&0&1&0&0&0&1&0&1&0&1&0&0&0&1&0&1&1&1\\
1&1&1&1&1&1&0&0&1&0&0&0&1&0&1&0&0&1&0&1\\
0&1&1&0&0&1&0&1&0&0&0&0&1&0&0&0&0&1&1&1\\
1&0&1&0&0&0&1&0&0&1&0&1&1&1&0&0&1&1&1&1\\
1&0&1&0&1&1&0&0&0&1&0&1&0&0&1&0&0&1&1&1\\
0&1&0&0&1&0&1&1&0&0&0&1&0&0&0&1&1&0&0&1\\
1&0&0&0&0&1&0&1&0&0&1&0&0&0&1&1&0&1&1&1\\
0&1&0&0&1&0&0&0&1&0&1&1&1&1&0&0&1&0&0&1\\
1&0&0&1&0&1&0&0&0&1&1&1&1&0&1&0&0&1&0&1\\
0&1&1&1&1&1&0&1&0&0&0&0&1&0&1&0&0&1&0&0\\
1&1&1&1&1&1&1&0&0&0&0&0&0&0&1&1&0&1&0&0\\
0&1&1&1&0&1&0&1&1&1&1&1&1&1&0&0&0&0&1&0
\end{pmatrix}$$

respectively. By setting $Threshold = Trials = 8$ (See Figure 2) we were able to collect the following 19 (δ_x, δ_y) pairs:

(624503, 241984) (776771, 695001) (327753, 131087) (55169, 514545)
(202272, 445310) (602355, 656872) (917362, 320210) (58440, 623796)
(974042, 35345) (715678, 214754) (383370, 531929) (204095, 609811)
(653178, 824812) 108979, 97871) (174443, 861123) (469759, 1002664)
(741723, 572238) (671505, 841867) 928012, 475934)

by performing 1736 encryption operations. We chose δ_x's to be linearly independent. Using the above pairs, we formed the following linear relation for any points on the same segment:

$$\delta_x = B\delta_y,$$

where B is given by

$$\left(\begin{array}{ccccccccccccccccccccccccccccccc}
1 & t_1 & 1 & t_1 & & 1 & t_1 & 0 & 1 & t_1 & t & & 1 & 1 & t_1 & t_1 & & 1 & t_1 & & 1 & 0 & 0 & 1 & t_1 & 0 & t & & 1 & t_1 & t_1 \\
1 & t_2 & 0 & 1 & t_2 & 1 & t_2 & 0 & 1 & t_2 & t_2 & & 1 & 1 & t_2 & 1 & t_2 & 0 & t_2 & & 1 & 1 & 1 & 1 & t_2 & 0 & 1 & t_2 & t_2 & & t_2 \\
1 & t_3 & 1 & t_3 & & t_3 & & 1 & 1 & t_3 & t_3 & & 1 & 1 & t_3 & t_3 & & 1 & t_3 & & 1 & 1 & 1 & t_3 & & 0 & t_3 & & 1 & t_3 & t_3 \\
t_4 & & 1 & t & & t & & 0 & t_4 & & t_4 & & 0 & t_4 & & 1 & t_4 & 0 & t_4 & & 1 & 0 & 0 & t_4 & & 1 & t_4 & & t_4 & & t_4 \\
t_5 & & 1 & 1 & t_5 & t & & 1 & 1 & t_5 & t_5 & & 1 & 1 & t_5 & t & & 0 & 1 & t_5 & 0 & 0 & 1 & 1 & t_5 & 1 & t_5 & & t_5 & & t_5 \\
1 & t_6 & 0 & 1 & t_6 & 1 & t_6 & 0 & t_6 & & t_6 & & 1 & 1 & t_6 & 1 & t_6 & 1 & t_6 & & 1 & 1 & 1 & t_6 & & 1 & 1 & t_6 & 1 & t_6 & t_6 \\
t_7 & & 1 & 1 & t_7 & 1 & t_7 & 0 & 1 & t_7 & 1 & t_7 & 0 & 1 & t_7 & t_7 & & 1 & t_7 & & 0 & 1 & 1 & 1 & t_7 & 1 & t_7 & & t_7 & & t_7 \\
1 & t_8 & 0 & 1 & t_8 & t_8 & & 0 & t_8 & & t_8 & & 1 & 1 & t_8 & t_8 & & 1 & 1 & t_8 & 0 & 0 & 1 & 1 & t_8 & 0 & t_8 & & 1 & t_8 & t_8 \\
t_9 & & 1 & t & & 1 & t_9 & 0 & 1 & t_9 & 1 & t_9 & 1 & t_9 & & 1 & t_9 & 0 & 1 & t_9 & 1 & 1 & 0 & t_9 & & 0 & t_9 & & t_9 & & t_9 \\
1 & t_{10} & 0 & 1 & t_{10} & 1 & t_{10} & 1 & 1 & t_{10} & 1 & t_{10} & 1 & 1 & t_{10} & 1 & t_{10} & 0 & 1 & t_{10} & 1 & 1 & 0 & t_{10} & & 0 & 1 & t_{10} & t_{10} & & t_{10} \\
1 & t_{11} & 0 & 1 & t & 1 & t_{11} & 1 & t_{11} & & 1 & t_{11} & 0 & 1 & t_{11} & t_{11} & & 0 & 1 & t_{11} & 0 & 1 & 1 & t & & 1 & 1 & t_{11} & 1 & t_{11} & t_{11} \\
t_{12} & & 0 & 1 & t_{12} & t_{12} & & 1 & 1 & t_{12} & t_{12} & & 1 & 1 & t_{12} & 1 & t_{12} & 0 & t_{12} & & 0 & 0 & 0 & 1 & t_{12} & 1 & 1 & t_{12} & 1 & t_{12} & t_{12} \\
1 & t_{13} & 1 & 1 & t_{13} & 1 & t_{13} & 1 & 1 & t_{13} & t_{13} & & 1 & 1 & t_{13} & t_{13} & & 1 & 1 & t_{13} & 0 & 0 & 1 & t_{13} & & 1 & t_{13} & & 1 & t_{13} & t_{13} \\
t_{14} & & 1 & 1 & t_{14} & t & & 0 & 1 & t_{14} & 1 & t_{14} & 0 & t_{14} & & 1 & t_{14} & 1 & 1 & t_{14} & 1 & 0 & 0 & t_{14} & & 0 & t_{14} & & t_{14} & & t_{14} \\
t_{15} & & 0 & t_{15} & & t_{15} & & 0 & t_{15} & & 1 & t_{15} & 1 & t_{15} & & t_{15} & & 0 & 1 & t_{15} & 0 & 1 & 1 & t_{15} & & 0 & t_{15} & & 1 & t_{15} & t_{15} \\
t_{16} & & 1 & 1 & t_{16} & {}_{16}t & & 0 & t_{16} & & 1 & t_{16} & 1 & 1 & t_{16} & t_{16} & & 1 & 1 & t_{16} & 0 & 1 & 1 & t_{16} & & 0 & t_{16} & & t_{16} & & t_{16} \\
1 & t_{17} & 1 & 1 & t_{17} & t_{17} & & 1 & t_{17} & & t_{17} & & 0 & 1 & t_{17} & t & & 0 & 1 & t_{17} & 1 & 1 & 0 & 1 & t_{17} & 1 & t & & 1 & t_{17} & t_{17} \\
t_{18} & & 0 & t_{18} & & t_{18} & & 1 & 1 & t_{18} & 1 & t_{18} & 0 & 1 & t_{18} & 1 & t_{18} & 1 & 1 & t_{18} & 1 & 0 & 0 & t_{18} & & 1 & t_{18} & & 1 & t_{18} & t_{18} \\
t_{19} & & 1 & t_{19} & & 1 & t_{19} & 0 & 1 & t_{19} & t_{19} & & 0 & t_{19} & & 1 & t_{19} & 1 & t & & 1 & 0 & 1 & t_{19} & & 0 & 1 & t_{19} & t_{19} & & t_{19} \\
t_{20} & & 0 & t_{20} & & 1 & t_{20} & 0 & 1 & t_{20} & t_{20} & & 0 & t_{20} & & 1 & t_{20} & 0 & 1 & t_{20} & 0 & 1 & 0 & 1 & t_{20} & 0 & 1 & t_{20} & t_{20} & & t_{20}
\end{array}\right)$$

for any $t_i \quad \{0,1\}$, $i = 1, 2, \cdots 20$. Using this relation and the pool of already encrypted 1736 plaintext-ciphertext pairs, we were able to decode correctly 99.946% of a random 100,000 ciphertext.

Robust and Secure Broadcasting

Yi Mu and Vijay Varadharajan

Department of Computing, Macquarie University,
Sydney, Australia
{ymu,vijay}@ics.mq.edu.au

Abstract. This paper describes a secure Pay TV protocol based on a public-key distributed encryption scheme that enables the Pay TV broadcaster to robustly add or remove any subscriber without changing private decryption keys of other subscribers. In other words, the updating process is transparent to the subscribers. This feature exhibits a distinct advantage over a symmetric key based system where all subscribers share a single key and therefore it is impossible to dynamically remove a subscriber from the system.

1 Introduction

A typical Pay TV system consists of a broadcaster and a number of subscribers. The broadcaster broadcasts TV programs to its subscribers. When a Pay TV program is transmitted through an optical fibre or a microwave network, the protection of the program must be enforced against non-subscribers. This can be done using a symmetric-key cryptographic algorithm. That is, the Pay TV broadcaster and all its subscribers share a secret key that is used by the broadcaster to encrypt the TV signal and is then used by subscribers to decrypt the signal. The major disadvantage of such a scheme is that it is difficult for the broadcaster to stop an illegal user who has the secret key to receive Pay TV programs, since the secret key is shared by all subscribers. Changing the secret shared key requires updating all decoding boxes of subscribers. This is infeasible and costly.

In this paper, we propose an asymmetric key based Pay TV system tackling this problem. The proposed system meets the following criteria:

1. The broadcaster can arbitrarily add or remove a subscriber to or from the system without changing the decryption keys of other subscribers.
2. Subscribers are required to have the minimum computational power.

The major task of constructing a public key broadcasting system is to find an algorithm where a public key maps to several associated private keys. This kind of mappings was first introduced by Desmedt[1]. Group signature[2,3,4,5] and distributed encryption[6,7] can be considered as examples of this kind of mappings. Unfortunately, the existing distributed encryption schemes cannot be used in a Pay TV system, due to the computational complexity. Moreover, in

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 223–231, 2001.

these schemes, the Pay TV broadcaster is not able to remove a subscriber from the system without the involvement of all other subscribers.

The proposed scheme in this paper is a variant of [6], but it has some significant breakthrough in the revocation of decryption keys used to a secure broadcasting service and in computational efficiency. The major computation in the system is shifted from its subscribers to the broadcaster and can be done in the setup phase; therefore the actual encryption and decryption computational complexity is thus minimised.

The rest of this paper is arranged as follows. Section 2 describe the basic model. Section 3 gives the preliminary knowledge required for the new scheme, where we will review the basic algorithm to be used in our system. Section 4 introduces the system setup. Section 5 describes the secure Pay TV protocol. Section 6 shows how to update the system by adding or removing a subscriber. The last section is our concluding remarks.

2 Model

A typical Pay TV system consists of a broadcaster and a number of subscribers. The broadcaster broadcasts TV programs to its subscribers through a secure channel.

Let us consider the situation that the secure transmission of Pay TV signals is based on the public-key distributed encryption[6]. A distributed encryption system consists of a manager and several users who form a group managed by the manager. The group manager has two major tasks: constructing a unique group public key and doing revocation. In the group, all group members or users have a private decryption key that can be used for decryption. The group possesses a unique public key or a group public key that maps to all private keys owned by its members. A message encrypted using the group public key can be decrypted using any one of these private keys.

The broadcaster in our Pay TV system acts as a group manager who uses the group public key or the encryption key to encrypt TV signals that are then transmitted to its subscribers or members who respectively possess a private key or decryption key that can be used to decrypt the TV signals.

Our system differs from a normal distributed encryption system in that it allows the Pay TV broadcaster to remove or add a decryption key from or to the public key such that it is easy for the broadcaster to perform the basic management over the system without any involvement of subscribers. The difference also lies in the fact that our scheme has very low computational complexity.

3 Preliminaries

In this section, we describe the distributed encryption algorithm, a variant of that proposed in Ref. [6].

The security of this system relies on difficulty of computing discrete logarithm. The protocols are based on a polynomial function and a set of exponentials. Let p be a large prime, $\mathbb{Z}_p$ be a multiplicative group of order q for $q|p-1$, and $g \in \mathbb{Z}_p$ be a generator. Let $x_i \in_R \mathbb{Z}_q$ for $i = 0, 1, 2..., n$ be a set of integers. The polynomial function of order n is constructed as follows.

$$f(x) = \prod_{i=1}^{n} (x - x_i) \equiv \sum_{i=0}^{n} a_i x^i \ (\bmod q),$$

where $\{a_i\}$ are coefficients: $a_0 = \prod_{j=1}^{n}(-x_j)$, $a_1 = \sum_{i=1}^{n}\prod_{i=j}^{n}(-x_j)$, $\cdots$, $a_{n-2} = \sum_{i=j}^{n}(-x_i)(-x_j)$, $a_{n-1} = \sum_{i=1}^{n}(-x_j)$, $a_n = 1$. It is noted that $\sum_{i=0}^{n} a_i x_j^i = 0$. This property is important for us to construct the distributed encryption system.

Having the set $\{a_i\}$, we can then construct the corresponding exponential functions, $\{g^{a_0}, g^{a_1}, \cdots, g^{a_n}\} \equiv \{g_0, g_1, \cdots, g_n\}$. All elements are computed under modulo p. For convenience, we will omit modulo p in the rest of this paper. Note that $\prod_{i=0}^{n} g_i^{x_i} = 1$.

Now we are ready to construct an asymmetric-key system where the encryption key is the tuple $\{g_0, g_1, \cdots, g_n\}$ mapping to n decryption keys $\{x_i\}$. The encryption key should not be made public; otherwise it needs some additional algorithm to make it secure[6]. Actually, in a Pay TV system, the encryption key is never made public.

Let H be a strong one-way hash function and M be the message to be sent to a recipient by the sender who possesses the encryption key,. The idea of this protocol is for the sender to encrypt a message M using the encryption key and produce the corresponding ciphertext that can be decrypted by any one who has a private key, $x_i \in \{x_1, x_2, \cdots, x_n\}$

To encrypt a message M, the sender picks a random number $k \in_R \mathbb{Z}_q$, computes $k = H(m)$ and encrypts M using the encryption key to obtain the ciphertext $c \leftarrow (c_1, c_2)$, where $c_1 \leftarrow (g^k g_0^k, g_i^k)$, $i = 1, \cdots, n$, and $c_2 = Mg^k$. c is sent to recipients.

Since all recipients have their private decryption keys, they can obtain M by decrypting the ciphertext c_2. The process is as follows. For recipient j,

$$\begin{aligned} d \leftarrow g^k g_0^k \prod_{i=1}^{n} g_i^{kx_j^i} &= g^k \prod_{i=0}^{n} g_i^{kx_j^i} = g^k \prod_{i=0}^{n} g^{a_i k x_j^i} \\ &= g^k g^{k\sum_{i=0}^{n} a_i x_j^i} = g^k . \end{aligned}$$

The last equality holds because $\sum_{i=0}^{n} a_i x_j^i = 0$. The message can be recovered by computing $M = c_2/d$.

This scheme is not suitable for a Pay TV system, because the recipients share considerable computational overhead in order to decrypt a message, especially when the size of the recipient group is large. However, the properties of the polynomial function given here are useful for our new protocols.

4 System Setup

The broadcaster needs to prepare the encryption key and all decryption keys for its subscribers. It is assumed that the broadcaster has sufficient computational power, whereas its subscribers have very limited computational power.

We assume that the system has the upper limit, n, for the maximum number of subscribers. The actual number of subscribers is m for $0 < m \leq n$. The difference between n and m will be used for adding new subscribers to the system.

The construction of the encryption key and decryption keys follows the steps below:

- Select n distinct random numbers $x_i \in_R \mathbb{Z}_q$ for $i = 1, 2, \cdots, n$, which form a set X_n and a subset $X_m \subseteq X_n$.
- Compute $A = \prod_{j=1}^{n}(\prod_{i=0}^{n-1} g_i^{x_j^i})$. We will see later that the broadcaster needs only to compute A *once*.
- Select an integer $b \in_R \mathbb{Z}_q$ and compute its multiplicative inverse b^{-1} such that $bb^{-1} = 1 \bmod q$.
- Compute $\bar{x}_j = b^{-1} \prod_{i=j}^{n} x_i^n \bmod q$, for $j = 1, 2, \cdots, n$.
- Compute $\hat{x}_j = s_j x_j^n \bmod q$, where $s = s_1 s_2 \cdots s_n$, and $s_i s_i \bmod q = 1$, $\forall s_i \in \mathbb{Z}_q$, i.e., the multiplicative inverse is itself. It is easy to see that it can be realised by simply setting $q = s_i(s_i - 1)/k$ for an integer k such that $q + 1$ is still a prime. The solutions of s_i can be found if $1 + 4kq = X^2$ where X is an odd number. It is not difficult to see that there are infinite solutions when we let k be $k (1 + k q)$ for an integer k .

These values satisfy the equality:

$$A^s g^{sb\bar{x}_j} g^{s\hat{x}_j} = 1, \quad \forall j \in \{1, 2, \cdots, n\}.$$

A is kept by the broadcaster and will be used as the encryption key for broadcasting Pay TV signals. Since the encryption key is not public, there is no need for us to protect it against any illegal modification.

$\bar{x}_j$ and $\hat{x}_j$ are given to subscriber j as its secret decryption key during the process of its registration. The private decryption key doublet, $(\bar{x}_j, \hat{x}_j)$, is embedded in a tamper proof box in the Pay TV decoding device for subscriber j, because there is no need for the subscriber to know it. This assumption seems necessary, otherwise it would be possible to illegally forge a decoding device. However, we assume that subscribers know their decryption keys and the algorithm meets the condition that it is secure against any collusion attacks. This assumption makes our scheme to have better applicability in other secure broadcasting services.

5 Broadcasting Protocol

The unique encryption key A is used for the encryption of Pay TV signals. Any subscriber who has a legitimate private decryption key can decrypt them. Assuming that M is the TV signal to be transmitted, the broadcasting protocol is given as follows:

– Select an integer $k \in_R \mathbb{Z}_q$.
– Compute $\bar{g} = g^{sk}$ and $\hat{g} = g^{sbk}$.
– Compute the ciphertext $c = MA^{sk}$.
– Broadcast the triplet $(\bar{g}, \hat{g}, c)$ to all subscribers.

To decrypt it, the subscriber j computes

$$c\hat{g}^{\bar{x}_j}\bar{g}^{\hat{x}_j} = M.$$

Remark 1: In consideration of further efficiency, the system should be made hybrid, e.g., the broadcaster encrypts the a symmetric session key using the encryption key and then uses the session key to encrypt the Pay TV signals. That is, the symmetric key system is used for secure transmission of Pay TV signals and the public key system is used for secure transmission of the secret session key only. The subscribers obtain the session key using their private decryption keys and decrypt the Pay TV signals using the session key.

The completeness of this protocol is obvious:

Lemma 1. *For a given ciphertext c, if the broadcaster follows the correct encryption procedure, any registered subscriber can correctly decrypt the ciphertext to obtain M.*

Proof: Obvious. This is based on the following: when a registered subscriber j decrypts the signals, he does not change the value of s, i.e. $ss_j = s \bmod q$. Therefore, in the decryption, he can remove the A from c,.

$$\begin{aligned} c\hat{g}^{\bar{x}_j}\bar{g}^{\hat{x}_j} &= MA^{sk}(g^{sbk})^{b^{-1}\sum_{i=1,i=j}^{n} x_i^n}(g^{sk})^{s_j x_j^n} \\ &= MA^{sk}(g^{sk})^{\sum_{i=1}^{n} x_i^n} = M\prod_{j=1}^{n}(\prod_{i=0}^{n} g_i^{x_j^i}) = M \end{aligned}$$

The last equality is based on $\prod_{j=1}^{n}(\prod_{i=0}^{n} g_i^{x_j^i}) = 1$. $\square$

The soundness of the protocol is twofold. The basic security of the system is based on the trustworthiness of the broadcaster. This makes sense, since in practice the broadcaster should have the ultimate control over the system. On the other hand, it is also quite clear that any party who has not registered with the system cannot decrypt the Pay TV signal. Finding a correct decryption key is equivalent to computing discrete logarithm, which is infeasible in a polynomial time frame. We can actually see that this encryption scheme is a variant of El-Gamal's encryption scheme[8]. The only difference is that the public encryption key A is constructed differently.

6 Update Protocols

The broadcaster has the complete control over who can decode the TV program. In other words, the broadcaster can remove or add any subscriber when needed. We now present two schemes to show that this updating process does not require any cooperation of the subscribers.

6.1 Removing a Subscriber – Scheme 1

To remove a subscriber y from the list or x_y from X_m, the broadcaster needs to

- Recompute A:

$$A = \prod_{j=1, j=y}^{n} \Big(\prod_{i=0}^{n-1} g_i^{x_j^i} \Big).$$

- Compute a new parameter $d = g^{-\hat{x}_y}$.

To broadcast a TV signal, the broadcaster now needs to compute the ciphertext in terms of

$$c = MA^{s k} d^{s k},$$

where $s = \sum_{i=1, i=y}^{n} s_i \bmod q$. The broadcasted token consists of $(\bar{g}, \hat{g}, c)$, where $\bar{g} = g^{s k}$ and $\hat{g} = g^{s bk}$.

To decrypt it, the subscriber j computes

$$c \, \hat{g}^{\bar{x}_j} \bar{g}^{\hat{x}_j} = M.$$

The completeness of this protocol is given in the lemma below.

Lemma 2. *A registered subscriber can decrypt the PayTV signals using his secret key.*

Proof: Obvious. This is based on the following: when a registered subscriber j decrypts the signals, he does not change the value of s, i.e. $s s_j = s \bmod q$. Therefore, in the decryption, he can remove the A from c,.

$$\begin{aligned} c \, \hat{g}^{\bar{x}_j} \bar{g}^{\hat{x}_j} &= MA^{ks} d^{s k} (g^{s bk})^{b^{-1} \sum_{i=1, i=j}^{n} x_i^n} (g^{s k})^{s_j x_j^n} \\ &= MA^{ks} (g^{s k})^{\sum_{i=1, i=y}^{n} x_i^n} = M \prod_{j=1, j=y}^{n} \Big(\prod_{i=0}^{n} g_i^{x_j^i} \Big) = M \end{aligned}$$

The last equality is due to $\prod_{j=1, j=y}^{n} \big(\prod_{i=0}^{n} g_i^{x_j^i} \big) = 1$. □

The soundness of the protocol is shown in the following lemma.

Lemma 3. *After a subscriber has been removed from the system, he can not decrypt the PayTV signal using his secret decryption key doublet $(\bar{x}_y, \hat{x}_y)$.*

Proof: This is also obvious. It is based on: $s = s s_y \bmod q$ for the removed subscriber y. □

We have assumed that the secret decryption keys of all subscribers have been embedded in a tamper-resistant decoding box that should not be seen by the subscribers. However, let us consider the case where the subscribers including the one who has been just removed from the system know about their secret decryption keys. In the following lemma we show that the updating protocol is still sound against collusion attacks.

Lemma 4. *The broadcasting protocol is secure against collusion attacks of the subscribers.*

Proof: It is clear that if a subscriber i finds the value of $b^{-1}x_i^n$, he can then decrypt the Pay TV signal easily. We first examine whether or not several subscribers can compute $b^{-1}x_i^n$, based on their secret keys. Assume there are two subscribers, i and j and they have private key doublets $(\bar{x}_i, \hat{x}_i)$ and $(\bar{x}_j, \hat{x}_j)$ respectively. By subtracting $\bar{x}_i$ from $\bar{x}_j$, we have $\bar{x}_i - \bar{x}_j = b^{-1}(x_i^n - x_j^n)$. If there is another subscriber k who is also involved in the computation, we then have an additional equation $\bar{x}_i - \bar{x}_k = b^{-1}(x_i^n - x_k^n)$. Any additional subscriber has to introduce a new unknown variable to the computation. Also recall that $\hat{x}_j = s_j x_j^n \bmod q$ and $s_j = s_j = s_k$. Therefore, this kind of collusion attack is not feasible. □

Remark 2: The broadcaster does not need to recompute A if all the data in the setup phase have been stored. Only thing needed is to remove the part related to the subscriber y from A.

6.2 Removing a Subscriber – Scheme 2

This scheme is much simpler than the first scheme. The broadcaster does not need to reconstruct the encryption key A. Instead, the broadcaster just recomputes s such that the s_y for the subscriber to be removed is moved from the computation, i.e., $s = \prod_{i=1,i=y}^{n} s_i$. Under this scheme, the broadcasting protocol given in Section 5 can still be used without any modification. The removed subscriber cannot decrypt the Pay TV signal since $s_y s = s \bmod q$, while other subscribers can still decrypt the Pay TV signals as usual.

6.3 Adding a Subscriber

There are two ways for the broadcaster to add a new subscriber to the system.

1. The first approach makes use of an element in the spare set $X_n - X_m$. Recall that we have assumed that the actual number of subscribers is less than n, i.e., $m < n$ or $X_m \quad X_n$. To add a new subscriber, the broadcaster just simply moves one unused element from $X_n - X_m$ to X_m. For convenience, we still denote by X_m the new set. Suppose that the new subscriber is denoted by subindex and is given $\bar{x}$ and $\hat{x}$ as his decryption key doublet stored in his decoding box.
2. The second approach reuses x_y, after $x_y \quad X_m$ has been removed from the system. Once x_y has been removed from the system, the corresponding party y is no longer able to decrypt Pay TV programs. However, because x_y still exists in X_m, it is perfectly legal for the broadcaster to reuse it, provided that the removed subscriber y cannot gain anything from it. The algorithm for the broadcaster to reuse x_y is actually very simple, namely for a new subscriber r the broadcaster needs only to generate a new "s_r" and compute a new "s" by replacing s_y with s_r, i.e., $s = s_1 s_2 \cdots s_r \cdots s_n$. The new decryption

key is then constructed in the same manner: $\bar{x}_r = b^{-1} \prod_{i=r}^{n} x_r^n \bmod q$ and $\hat{x}_r = s_r x_r^n \bmod q$. The encryption protocol in Section 5 still stands. The soundness of this scheme is based on the following: $s\hat{x}_r = ss_r x_r^n = sx_r^n \bmod q$ and $s\hat{x}_y = ss_y x_y^n = sx_y^n \bmod q$.

7 Concluding Remarks

We have proposed a new scheme for a public-key based broadcasting system, where the broadcaster can easily add or remove any subscriber when needed. More importantly, the updating process is transparent to subscribers, since they are not involved in the update computation. This provides an elegant solution to the revocation problem in secure broadcasting service. It is clear that our scheme has a distinct advantage over a symmetric-key based method. In a symmetric key based system, all the subscribers share the same secret key in order to decrypt the Pay TV signals. This provides opportunities for forging Pay TV decoding boxes. This also makes it difficult for the broadcaster to exclude anyone from receiving the Pay TV signals (say if a member does not pay his subscription).

On the other hand, our algorithm is designed in such a way that the subscribers have limited computational power, while most computations are done on the side of the broadcaster. From the broadcaster point of view the major computation involves the calculation of the encryption key A. It is noted that such computation can be done prior to broadcasting and is done once only. The encryption process involves only three exponentials. On the subscriber's side, the decryption needs only two exponential computations. Furthermore, with the hybrid model such computations are for secure session key transmission only and the real signal encryption/decryption is done under a symmetric key algorithm.

References

1. Y. Desmedt, "Society and group oriented cryptography: A new concept," in *Advances in Cryptology, Proc. CRYPTO 87,* LNCS 293, pp. 120–127, Springer Verlag, 1987.
2. D. Chaum and E. van Heijst, "Group signatures," in *Advances in Cryptology, Proc. EUROCRYPT 91,* LNCS 547, pp. 257–265, Springer-Verlag, 1991.
3. L. Chen and T. P. Pedersen, "New group signature schemes," in *Adances in cryptology - EUROCRYPT'94, Lecture Notes in Computer Secience 950*, pp. 171–181, Springer-Verlag, Berlin, 1994.
4. J. Camenisch, "Efficient and generalized group signatures," in *Adances in cryptology - EUROCRYPT'97, Lecture Notes in Computer Secience 1233*, pp. 465–479, Springer-Verlag, Berlin, 1997.
5. J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *Advances in Cryptology, Proc. CRYPTO 97,* LNCS 1296, pp. 410–424, Springer-Verlag, Berlin, 1997.
6. Y. Mu, V. Varadharajan, and K. Q. Nguyen, "Delegated decryption," in *Procedings of Cryptography and Coding, Lecture Notes in Computer Science*, Springer Verlag, 1999.

7. D. Boneh and M. Franklin, "An efficient public key traitor tracing scheme," in *Adances in cryptology - CRYPTO '99, Lecture Notes in Computer Secience 1666*, Springer Verlag, 1999.
8. T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, pp. 469–472, 1985.

Toward Optimal Player Weights in Secure Distributed Protocols

K. Srinathan, C. Pandu Rangan, and V. Kamakoti

Department of Computer Science and Engineering, Indian Institute of Technology,
Madras, Chennai - 600036, India,
ksrinath@cs.iitm.ernet.in, {rangan,kama}@iitm.ernet.in

Abstract. A secure threshold protocol for n players tolerating an adversary structure $\mathcal{A}$ is feasible iff $\max_{a \in \mathcal{A}} |a| < \frac{n}{c}$, where $c = 2$ or $c = 3$ depending on the adversary being eavesdropping (passive) or Byzantine (active) respectively [1]. However, there are situations where the threshold protocol Π for n players tolerating an adversary structure $\mathcal{A}$ may not be feasible but by letting each player P_i to act for a number of similar players, say w_i, a new secure threshold protocol Π' tolerating $\mathcal{A}$ may be devised. Note that the new protocol Π' has $N = \sum_{i=1}^{n} w_i$ players and works with the same adversary structure $\mathcal{A}$ used in Π. The integer quantities w_i's are called weights and we are interested in computing w_i's so that

1. Π' tolerates $\mathcal{A}$ even if Π does not tolerate $\mathcal{A}$.
2. $N = \sum_{i=1}^{n} w_i$ is minimum.

Since the best known secure threshold protocol over N players has a communication complexity of $O(mN^2 \lg |\mathbb{F}|)$ bits [9], where m is the number of multiplication gates in the arithmetic circuit, over the finite field $\mathbb{F}$, that describes the functionality of the protocol, it is evident that the weights assigned to the players have a direct influence on the complexity of the resulting secure weighted threshold protocol. In this work, we focus on computing the optimum N. We show that computing the optimum N is NP-Hard. Furthermore, we prove that the above problem of computing the optimum N is inapproximable within $(1-\epsilon)\ln\left(\frac{|\mathcal{A}|}{c}\right) + \frac{\left(\ln\left(\frac{|\mathcal{A}|}{c}\right)\right)^{(1-\epsilon)} - 1}{N^*}(c-1)$, for any $\epsilon > 0$ (and hence inapproximable within $\Omega(\lg |\mathcal{A}|)$), unless $NP \subseteq DTIME(n^{\log \log n})$, where N^* is the optimum solution.

1 Motivating Example

Consider a set of five players $\mathcal{P} = \{P_1, P_2, P_3, P_4, P_5\}$ involved in a secure distributed protocol wanting to tolerate the (passive) adversary structure $\mathcal{A}$ given by

$$\mathcal{A} = \{(1,2,3), (1,2,4), (1,5), (2,5), (3,4)\}$$

The first author would like to thank Infosys Technologies Ltd., India for financial support.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 232–241, 2001.

From the results of [1], it is clear that A cannot be tolerated by any threshold protocol. Nevertheless, the above adversary can be tolerated by a threshold-type protocol among nine players where players P_1, P_2, P_3, P_4 and P_5 act for one, one, two, two and three players respectively. This is indeed so because the corruption of any one set in the adversary structure leads to the corruption of at the most four out of the nine players which is tolerable[1]. In this example, $n = 5, w_1 = w_2 = 1, w_3 = w_4 = 2, w_5 = 3, c = 2, N = \sum_{i=1}^{n} w_i = 9$.

2 Basic Definitions and Model

2.1 Secure Multiparty Computation

Consider a fully connected synchronous network of n players (processors), $P = \{P_1, P_2, \ldots, P_n\}$, who do not trust each other. Nevertheless they want to compute some agreed function of their inputs in a secure way. Security here means maintaining correctness of the output while keeping the players' inputs as private as possible, even if some of the players are faulty. This task can be easily accomplished if there exists a trusted third party. But assuming the existence of such a trusted third party is quite unrealistic. The goal of *secure multiparty computation* is to transform a given protocol involving a trusted third party into a protocol without need for the trusted third party, by simulating the trusted third party among the n players.

The players' distrust in each other and in the underlying network is usually modeled via an *adversary* that has control over some of the players and communication channels. Many different adversary models have been considered, each modeling different problems, or addressing a different setting. These approaches can be classified according to a number of criteria that are briefly discussed below. Adversaries are classified according to their *computational resources* (limited (cryptographic) or unlimited (information theoretic)), their control over communication (secure, insecure, or unauthenticated channels), their control over corrupted players (eavesdropping (passive), fail-stop, or Byzantine (active)), their mobility (static, adaptive, or mobile) and their corruption capacity (threshold or non-threshold). In the information theoretic model one can distinguish between protocols with small (*unconditional*) or zero (*perfect*) failure probability.

In the information theoretic setting, [1] gave a perfect protocol for the general secure multiparty computation problem in the synchronous secure channels model without broadcast and proved tight bounds on the number of corrupted players that can be tolerated.

Theorem 1 ([1]). *For every $n \geq 2$, there exist Boolean functions f such that there is no synchronous $\lceil \frac{n}{2} \rceil$-secure protocol for n players that computes f. For every $n \geq 3$, there exist functions f such that no synchronous protocol for n players $\lceil \frac{n}{3} \rceil$-securely computes f, if Byzantine adversaries are allowed.* ■

2.2 The Adversary Model

In this section, we formally define the *weighted threshold adversaries*. We begin with a brief look at the *threshold* adversaries.

Threshold Adversaries. A threshold adversary, A, is a probabilistic strategy, that can *corrupt* up to $t < n$ among the n players involved in the protocol. The *corruption* may be either *active* or *passive*, by which we mean the following:

1. *Passive Corruption:* The adversary in this case behaves like an *eavesdropper*; that is, the adversary can gather all the information present with the corrupted players and can also perform any arbitrary computation on these gathered data.
2. *Active Corruption:* The adversary here is also referred to as a *Byzantine* adversary. They can do all what an eavesdropping adversary can and in addition can also take complete control of the corrupted players and alter the behaviour of the corrupted players in an arbitrary and coordinated fashion.

Tolerable Threshold Adversaries: It is known that all the passive threshold adversaries such that $t \leq \lfloor \frac{n-1}{2} \rfloor$ can be tolerated. That is, it is possible to construct multiparty computation protocols that are *secure* against such an adversary. By a security against an adversary A, we mean, whatever A does in the protocol, the same effect (on the output) could be achieved by an adversary (may be different from A but similar to it in costs) in the ideal protocol (that assumes the existence of a trusted third party to whom all the inputs can be sent and outputs received). For more formal and “correct” definitions of security, we refer the readers to [2,6,10]. Similarly, in the case of active adversaries, we require that $t \leq \lfloor \frac{n-1}{3} \rfloor$.

Generalized Adversaries. In contrast to the threshold adversaries, [7,8] introduced a more general adversary characterized by a monotone adversary structure which is a set of subsets of the player set, wherein the adversary may corrupt the players of one set in the structure. An adversary structure is said to satisfy the $Q^{(c)}$ property if no c sets in the adversary structure cover the full set of players. It is proved that in the passive model, every function can be computed securely with respect to a given adversary structure if and only if the adversary structure satisfies the $Q^{(2)}$ property. Similarly, in the active model, a secure computation of a function is possible if and only if the adversary structure satisfies the $Q^{(3)}$ property.

Weighted Threshold Adversaries. The weighted threshold adversaries are somewhere in between the threshold and the generalized adversaries. These adversaries are characterized by adversary structures that possess the following addition property so that they are tolerable: for each player P_i, $1 \leq i \leq n$, there exists a non-negative weight w_i, such that the adversary structure is tolerated in a threshold-type protocol with $N = \sum_{i:P_i \in P} w_i$ players. Hereafter in the sequel, unless explicitly specified, we will use the term adversary structure to mean the maximal basis[1].

In the weighted threshold adversary setting, one of the ways to improve the complexity of the resulting secure protocol is to assign weights to each of the players so that the adversary can be tolerated with the sum of the weights kept at a minimum, since, a larger sum of weights calls for larger number of secret shares (essentially of the same size) and hence an increase in the computation and communication complexities.

3 The Optimal Player Weights Problem

In this section, we define the problem of assigning optimum weights to the players in a secure multiparty protocol tolerating weighted threshold adversaries.

Definition 1 (Optimum Assignment of Player Weights(OAPW)). *Given the player set $P = \{P_1, P_2, \ldots, P_n\}$, the adversary structure $\mathcal{A} \subseteq 2^P$, and a constant c,*[2] *a valid assignment of player weights (if it exists) is a function $f : P \rightarrow Z^+ \cup \{0\}$ such that for all sets $z \in \mathcal{A}$, $\sum_{P_i \in z} f(P_i) < \frac{\sum_{P_i \in P} f(P_i)}{c}$. A valid assignment of player weights, f, is said to be an optimum assignment of player weights if there does not exist any valid assignment of player weights, f', such that $\sum_{P_i \in P} f'(P_i) < \sum_{P_i \in P} f(P_i)$.*

Definition 2 (Decision Version of the OAPW problem).

INSTANCE: *A finite set P, a collection $\mathcal{A}$ of subsets of P, a constant c, and a positive integer k.*
QUESTION: *Does there exist a valid assignment of player weights $f : P \rightarrow Z^+ \cup \{0\}$ such that $\sum_{P_i \in P} f(P_i) > ck$ and for all sets $z \in \mathcal{A}$, $\sum_{P_i \in z} f(P_i) \leq k$?*

From the (refer Definition 2) constraint that for all sets $z \in \mathcal{A}$, $\sum_{P_i \in z} f(P_i) \leq k$, it is obvious that we can restrict the range of the function f to $\{0, \ldots, k\}$. We denote an instance to the OAPW problem by the ordered list $< P, \mathcal{A}, c, k, r >$, where $f : P \rightarrow \{0, \ldots, r\}$, and the *size of the solution* to the above instance is $ck + 1$.

4 Hardness of the OAPW Problem

Definition 3 (Density Index Number of a Graph G). *: Given a simple undirected graph $G = (V, E)$, the c-Density Index Number of G is defined as*

$$c\text{-}DIN(G) = \min \left\{ ck+1 \;\middle|\; \begin{array}{l} \text{there exists } V' \subseteq V, |V'| = ck+1 \text{ such that there does not} \\ \text{exist a vertex } u \in V' \text{ adjacent to } \geq k+1 \text{ vertices in } V' \end{array} \right\}.$$

[1] Given the adversary structure $\mathcal{A}$, the maximal basis $\mathcal{A}_{basis} = \{z \in \mathcal{A} \mid \nexists z' \supset z, z' \in \mathcal{A}\}$.

[2] Note that $c = 2$ if the adversary is passive and $c = 3$ if the adversary is active.

Theorem 2. *Given a simple undirected graph $G = (V, E)$, the size of the minimum dominating set (*MDS*) of G is equal to* 1-DIN(G^c)*, where G^c is the complement of G.*

Proof: Let 1-DIN$(G^c) = k + 1$ and $V \quad V$ be the subset satisfying the property as defined in Definition 3 with $|V \;| = k + 1$. From definition of 1-DIN we see that every vertex in V (and hence in $V - V$) in G^c is *not* adjacent to at least one vertex in V . This implies that in G, every vertex in $V - V$ is adjacent to at least one vertex in V . Hence, V is a dominating set of G. If V is not the minimum dominating set, then let U be the minimum dominating set of G. Then, $|U| \leq k$ and any vertex $u \quad V - U$ is adjacent to at least one vertex in U. Therefore in G^c, any vertex $u \quad V - U$ is not adjacent to all the vertices in U. Also, since no vertex in U can be adjacent to all the vertices in U (due to the fact that a vertex cannot be adjacent to itself), the 1-DIN(G^c)=k which is a contradiction. Thus, the *minimum* constraint in the definition of 1-DIN implies that V is indeed a MDS of G. ■

The fact that computing the size of a MDS of a graph is NP-complete [5] and Theorem 2 imply the following theorem.

Theorem 3. *Given a simple undirected graph $G = (V, E)$, computing* 1-DIN(G) *is NP-complete.*

Theorem 4. *c*-DIN(G) *for any fixed constant c, where $G = (V, E)$ is a simple undirected graph, is NP-Hard.*

Proof: We reduce the problem 1-DIN(G) to the problem c-DIN(G). Given an instance of the problem 1-DIN(G), construct $G \;= (V \;, E \;)$ containing c copies of $G = (V, E)$; $V \;= V_1 \quad \cdots \quad V_c$ such that $i \quad V_j$ is relabelled $< i, j >$. Similarly, E containing c copies of E; $E \;= E_1 \quad \cdots \quad E_c$ such that for every pair of vertices $(i, k) \quad E_j$, is relabelled $(< i, j >, < k, j >)$. Solve the c-DIN problem on G . By the Pigeonhole principle we see that, there exists a solution to $c{-}DIN(G \;)$ if and only if there exists a solution to the problem $1{-}DIN(G)$. ■

Theorem 5. *Problem OAPW is NP-Hard.*

Proof: Given a simple undirected graph $G = (V, E)$, we suggest the following method for computing c-DIN(G). Without loss of generality let us assume that the vertices of G are numbered $\{\, 1, \ldots, n \,\}$, $|V| = n$. Let the set $P = V$. Construct the set $A = \{V_1, \ldots, V_n\}$, such that, V_i is a set containing all vertices adjacent to vertex i in G. Solve the problem OAPW with the sets P and A as defined above. We now show that the function f does not exist for $ck + 1 <$ c-DIN(G) $- c$. Let us assume that f exists and let $V \quad P$ be the set of vertices that are assigned non-zero values by f.

Case 1: ($|V \;| \leq k + 1$). From Definition 3 (of c-DIN), there exist an $a \quad A$ such that $V \quad a$. This implies that $\;_{i \;\; a} f(i) > k$, a contradiction.

Case 2: $(|V\ \ | > k + 1)$. Consider a $V\,”\ \ \ \ V\ $, $|V\,”| = k + 1$. From Definition 3 (of c-DIN), there exist an $a\ \ \ \ A$ such that $V\ \ \ \ \ \ a$. This implies that □ $_{i\ \ a} f(i) > k$, a contradiction.

For $ck + 1$ = c-DIN(G)$-c$, consider any $V\ \ \ \ \ P$, $|V\ \ | = ck + 1$. Assign $f(i) \leftarrow 1,\ \ \ i\ \ \ \ V$ and $f(i) \leftarrow 0,\ \ \ i\ /\ V$. It is easy to see that f is a solution to the $OAPW$ problem. The same concept can be extended to show that for $ck + 1 >$ c-DIN(G)$-c$ there exist a function f satisfying the constraints specified by the $OAPW$ problem. Hence, given an algorithm for the $OAPW$ problem that takes $O(g(n))$ time, one can compute the c-DIN of a simple undirected graph G in $O(g(n) \log n)$ time. Since c-DIN(G) $\le n$ it is enough to consider $ck + 1 \le n$. The above discussion and Theorem 4 imply the proof of this theorem. ■

5 An Approximate Algorithm for the OAPW Problem

In this section, we first reduce the $OAPW(P, A, c, k, k)$ to $OAPW(P, A, c, k, 1)$. We then provide an (exponential) algorithm to solve the $OAPW(P, A, c, k, 1)$ exactly followed by a (polynomial) approximate algorithm for the same. Finally, we design an approximate algorithm for $OAPW(P, A, c, k, k)$ and analyse its performance.

Theorem 6. *A solution to OAPW in which* $f : P \to \{0, 1\}$ *implies a solution to OAPW in which* $f : P \to \{0, ..., k\}$*.*

Proof: Given an instance $I = (P, A, c, k, k)$ of OAPW, construct an instance $I\ \ = (P\ \ , A, c, k, 1)$ in which P is k copies[3] of P. Solving OAPW on I , each $P_i\ \ \ \ P$ would have been assigned *at most* k 1s (at most once in each copy of P). Computing $f(i)$ to be the number of 1s assigned to P_i in P , gives a solution for the OAPW on I. ■

5.1 Solving $OAPW(P, A, c, k, 1)$

Exact Solution Given the instance $I = (P, A, c, k, 1)$ of the OAPW problem, construct a bipartite graph $G = (X, Y, E_g)$ with $X = P$, $Y = A$. Add (x, y) to E_g if and only if $x\ \ \ \ X$, $y\ \ \ \ Y$, and $x\ \ \ \ y$ (that is, the player x is present in the set y). Now, the problem of $OAPW(P, A, c, k, 1)$ stated graph theoretically is to find $ck + 1$ vertices in X such that the degree of each vertex $y\ \ \ \ Y$ in the subgraph induced by these $ck + 1$ vertices union Y on G is $\le k$.

Consider the bipartite graph $H = (X, Y, E_h)$ where $E_h = \{(x, y)|x\ \ \ \ X, y\ \ \ \ Y, (x, y)\ \ \ \ E_g\}$. The $OAPW(P, A, c, k, 1)$ problem can now be rephrased as to find $ck + 1$ vertices in X such that the degree of each vertex $y\ \ \ \ Y$ in the subgraph induced by these $ck + 1$ vertices union Y on H is at least $(c - 1)k + 1$.

From the bipartite graph $H = (X, Y, E_h)$, we construct the following instance of a set multi-cover problem that solves the $OAPW(P, A, c, k, 1)$ problem: Let

[3] Since the search space of k is bounded by a polynomial in $(|P| + |A|)$, the given construction is feasible.

Algorithm for $OAPW(P, A, c, k, 1)$

1. Given the instance $I_{oapw} = (P, A, c, k, 1)$ of the OAPW problem, construct the instance $I_{smc} = (U, F, (c-1)k + 1, ck + 1)$ of the set multi-cover problem as illustrated in Subsection 5.1.
2. Solve the instance I_{smc} using the approximate set multi-cover algorithm of [3], which is a natural extension to the greedy approximate algorithm for the set cover problem.
3. The solution to the instance I_{smc} (if it exists) gives rise to the set of vertices in X (i.e. the set of players in P) that are to be given the weight 1. The rest of the players are given the weight 0.
4. If the instance I_{smc} has no solution then the instance I_{oapw} has no solution as well.

Fig. 1. The Approximate Algorithm for $OAPW(P, A, c, k, 1)$

the set $U = Y$ and the family of subsets of U be $F = \{X_i | i = 1, 2, \ldots, |X|\}$, where X_i denotes the set of all elements in Y that are adjacent to the i^{th} element in X in the bipartite graph H. The decision version of the $OAPW(P, A, c, k, 1)$ problem now reads as follows: *Does there exist $ck + 1$ or less number of sets from F such that their union covers each element of U at least $(c-1)k + 1$ times?*

The above problem can be solved using the solution to the set multi-cover problem which is as follows.

Definition 4 (Set Multi-Cover Problem).
INSTANCE: *A set U, a family F of set of subsets of U, positive integers m and k.*
QUESTION: *Does there exist $\leq k$ sets from F such that they together cover each element of U at least m times?*

Thus, based on the (exponential) algorithm of finding the minimum set multi-cover, we now have an (exponential) algorithm to solve the $OAPW(P, A, c, k, 1)$ problem.

Approximating $OAPW(P, A, c, k, 1)$. We proceed by "replacing" the exponential algorithm of finding the minimum set multi-cover by its corresponding approximate greedy algorithm as proposed by [3]. Thus, the resulting approximate algorithm for $OAPW(P, A, c, k, 1)$ is as given in Fig. 1.

Theorem 7. *The algorithm presented in Fig. 1 runs in time polynomial in the size of the input and correctly solves the $OAPW(P, A, c, k, 1)$ problem.*

Proof: Since each of the four steps in the algorithm runs in time polynomial in $(|P| + |A|)$, it is evident that the overall algorithm runs in time polynomial in the input size.

From the construction of Subsection 5.1, it is clear that a solution to the instance I_{oapw} exists if and only if the instance I_{smc} has a solution. We now show that every solution to the instance I_{smc} leads to a solution to the instance I_{oapw}, thereby proving the theorem.

Let $(X_{i_1}, X_{i_2}, \ldots, X_{i_{ck+1}})$, $X_j \ \ F$ be a set multi-cover of U such that their union covers U at least $(c-1)k+1$ times. We stress that the value of k here may be much larger than what the minimum set multi-cover requires it to be. From this (approximate) set multi-cover, we obtain the corresponding vertices in X that along with the vertices in Y induce a subgraph H_{sub} on H such that each vertex in Y in H_{sub} has a degree of at least $(c-1)k+1$. Therefore, since E_h and E_g compliment each other, there exist $ck+1$ vertices in X such that the degree of every vertex in Y is bounded by $\leq k$ in the subgraph G_{sub} induced by the $ck+1$ vertices of X along with Y on G: thus providing a solution to the instance I_{oapw}. ■

Corollary 1. *The approximate algorithm for $OAPW(P, A, c, k, k)$ (see Fig. 2) follows from the Theorems 6 and 7.*

6 Inapproximability Results regarding the OAPW Problem

We begin with the known inapproximablity result of the set cover problem.

Theorem 8 ([4]). *The minimum set cover problem with the instance (U, F) is inapproximable within $(1-\ )\ln|U|$ for any $\ > 0$, unless $NP \ \ DTIME(n^{\log\log n})$.*

Using the above result, we show that the $OAPW(P, A, c, k, k)$ problem is inapproximable within $\Omega(\lg|A|)$ unless $NP \ \ DTIME(n^{\log\log n})$.

Theorem 9. *The problem of computing the optimum player weights is inapproximable within $(1-\ )\ln\left(\frac{|A|}{c}\right) + \frac{\ln\left(\left(\frac{|A|}{c}\right)^{(1-\ )}\right)-1}{N}(c-1)$, for any $\ > 0$ (and hence inapproximable within $\Omega(\lg|A|)$), unless $NP \ \ DTIME(n^{\log\log n})$, where N denotes the sum of the optimum player weights, and $c = 2$ for eavesdropping adversary and $c = 3$ if the adversary is Byzantine.*

Algorithm for $OAPW(P, A, c, k, k)$

1. Given the instance $I_k = (P, A, c, k, k)$ of the OAPW problem, construct the instance $I_1 = (P^{(k)}, A, c, k, 1)$ of the OAPW problem as illustrated in the proof of Theorem 6.
2. Solve the instance I_1 using the approximate algorithm given in Fig. 1.

Fig. 2. The Approximate Algorithm for $OAPW(P, A, c, k, k)$

Proof: Given a set U and a set F containing subsets of U, the *Set Cover problem* is to find the minimum number of sets of F that covers U. We show that the Set Cover problem could be solved using an algorithm for the $OAPW$ problem. Given U and F, construct the instance $OAPW(P, A, c, k, k)$ as follows:

1. Construct a bipartite graph $H = (X, Y, E)$ such that $X = F$ and $Y = U$ and $(x, y) \quad E$ if and only if $x \quad X, y \quad Y$ and $y \quad x$.
2. Let $U = \{u_1, u_2, \ldots u_{|U|}\}$. Construct the set A of $|U|$ elements, such that the i^{th} element of A , is the set of elements in X that are adjacent to u_i in H. Let $P \ = X$.
3. Let P be c copies of P and A be c copies of A .

From the Theorems 2 and 4, it is straightforward to observe that a solution to the instance $OAPW(P, A, c, k, k)$ gives a set cover of size $k + 1$, and we minimize k to get the Minimum Set Cover.

Let $N \quad = ck \quad + 1$ be the size of the optimal solution to $OAPW(P, A, c, k \ , k \ )$ and let $N = ck + 1$ be the size of the solution yielded by our algorithm. This implies that the minimum set cover is of size $k \quad + 1$ and the solution got by application of our algorithm is a set cover of size $k + 1$.

From Theorem 8 we see that,

$$k + 1 > (k \quad + 1)R,$$

where $R = (1 - \quad) \ln \frac{|A|}{c}$.

Note that $|U| = \frac{|A|}{c}$. Therefore we get,

$$\frac{N - 1 + c}{c} > \frac{N \quad - 1 + c}{c} R,$$

and thus,

$$\frac{N}{N} > R + \frac{(R - 1)(c - 1)}{N}$$

Hence the proof. ■

7 Conclusion

The bottleneck in secure distributed protocols is, in general, the communication/round complexity rather than the computation complexity. In the weighted threshold adversary setting, the communication complexity of the (resulting) protocol can be improved by two (independent) methods, viz., optimizing the players' weights, and developing/adapting the techniques of the threshold setting to the weighted threshold setting. In this work, we studied the former method and examined its complexity. We also presented an approximation algorithm for the Optimal Assignment of Player Weights ($OAPW$) problem and proved an inapproximability bound using the well-known set cover problem. Analyzing the quality of approximation is left open and is attempted in the full version of this paper.

References

1. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of 20th ACM Symposium on Theory of Computing (STOC)*, pages 1–10, 1988.
2. R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.
3. G. Dobson. Worst-case analysis of greedy heuristics for integer programming with non-negative data. *Math. Oper. Res.*, 7:515–531, 1982.
4. U. Feige. A threshold of $\ln n$ for approximating set cover. In *In Proceedings of 28th ACM Symposium on Theory of Computing (STOC)*, pages 314–318, 1996.
5. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
6. O. Goldreich. Secure multiparty computation, 1998. First draft available at http://theory.lcs.mit.edu/~ oded.
7. M. Hirt and U. Maurer. Complete characterization of adversaries tolerable in secure multiparty computation. In *16th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 25–34, August 1997.
8. M. Hirt and U. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *Journal of Cryptology*, 13(1):31–60, April 2000.
9. Martin Hirt and Ueli Maurer. Robustness for free in unconditional multi-party computation. In *CRYPTO '01*, Lecture Notes in Computer Science (LNCS). Springer-Verlag, 2001.
10. S. Micali and P. Rogaway. *Secure Computation: The information theoretic case*, 1998. Former version: Secure Computation, In *Advances in Cryptology CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 392-404, Springer-Verlag, 1991.

Autocorrelation Properties of Correlation Immune Boolean Functions

Subhamoy Maitra

Computer and Statistical Service Centre, Indian Statistical Institute,
203, B.T. Road, Calcutta 700 035, India,
subho@isical.ac.in

Abstract. In this paper we study the autocorrelation values of correlation immune and resilient Boolean functions. We provide new lower bounds and related results on absolute indicator and sum of square indicator of autocorrelation values for low orders of correlation immunity. Recently it has been identified that the nonlinearity and algebraic degree of the correlation immune and resilient functions are optimized simultaneously. Our analysis shows that in such a scenario the sum of square indicator attains its minimum value too. We also point out the weakness of two recursive construction techniques for resilient functions in terms of autocorrelation values.

Keywords: Autocorrelation, Boolean Function, Correlation Immunity, Global Avalanche Characteristics, Resiliency.

1 Introduction

Here we concentrate on the autocorrelation values for correlation immune and resilient (balanced correlation immune) Boolean functions. We provide the currently best known lower bounds on Δ_f (the absolute indicator) and σ_f (the sum of square indicator) for low orders of correlation immunity. Very recently autocorrelation properties of correlation immune and resilient Boolean functions were presented in [22] and we provide better results here. The autocorrelation property of higher order correlation immune functions has been considered in [18]. It has been shown in [18] that for an n-variable, m-resilient function $\Delta_f \geq 2^{n \frac{2m-n+3}{n+1}}$. However, the result is not applicable for low order of correlation immunity.

For high order of correlation immunity we provide sharper result for an important subclass of correlation immune and resilient functions which attain the maximum possible nonlinearity. It has currently been noticed that given certain order of correlation immunity, the nonlinearity and algebraic degree of the correlation immune and resilient functions are optimized simultaneously [17,2,3]. We here extend this analysis in terms of the sum of square indicator of autocorrelation values. We show that when the nonlinearity and algebraic degree are maximized, the sum of square indicator attains its minimum value.

In [23], it has been discussed that the propagation property goes against correlation immunity. We here explicitly show that the Δ_f value goes against the

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 242–253, 2001.

order of correlation immunity. We also point out the limitation of two recursive construction methods of resilient Boolean functions in terms of autocorrelation values.

Next we introduce a few definitions and notations. Let s, s_1, s_2 be binary strings of same length λ. The bitwise complement of s is denoted by s^c. We denote by $\#(s_1 = s_2)$ (respectively $\#(s_1 \neq s_2)$), the number of places where s_1 and s_2 are equal (respectively unequal). The Hamming distance between s_1, s_2 is denoted by $d(s_1, s_2)$, i.e., $d(s_1, s_2) = \#(s_1 \neq s_2)$. Another measure $wd(s_1, s_2)$ between s_1 and s_2, is defined as, $wd(s_1, s_2) = \#(s_1 = s_2) - \#(s_1 \neq s_2)$. Note that, $wd(s_1, s_2) = \lambda - 2\,d(s_1, s_2)$. The Hamming weight or simply the weight of s is the number of ones in s and is denoted by $wt(s)$.

By Ω_n we mean the set of all n-variable Boolean functions. We represent an n-variable Boolean function as a bit string of length 2^n, which is the output column of its truth table. An n-variable function f is said to be balanced if its output column in the truth table contains equal number of 0's and 1's (i.e., $wt(f) = 2^{n-1}$).

Note that we denote the addition operator over $GF(2)$ by $\oplus$. An n-variable Boolean function can be uniquely represented by a multivariate polynomial over $GF(2)$. We can write $f(X_n, \ldots, X_1)$ as

$$a_0 \oplus \left(\bigoplus_{i=1}^{i=n} a_i X_i\right) \oplus \left(\bigoplus_{1 \le i \neq j \le n} a_{ij} X_i X_j\right) \oplus \ldots \oplus a_{12\ldots n} X_1 X_2 \ldots X_n,$$

where the coefficients $a_0, a_i, a_{ij}, \ldots, a_{12\ldots n} \in \{0, 1\}$. This representation of f is called the algebraic normal form (ANF) of f. The number of variables in the highest order product term with nonzero coefficient is called the algebraic degree, or simply degree of f. Functions of degree at most one are called affine functions. An affine function with constant term equal to zero is called a linear function. The set of all n-variable affine (respectively linear) functions is denoted by $A(n)$ (respectively $L(n)$). The nonlinearity $nl(f)$ of an n-variable function f is defined as

$$nl(f) = \min_{g \in A(n)} (d(f, g)),$$

i.e., $nl(f)$ is the distance of f from the set of all n-variable affine functions.

In this document we will use concatenation of Boolean functions. Consider $f_1, f_2 \in \Omega_{n-1}$ and $f \in \Omega_n$. Then by concatenation of f_1 and f_2, we mean that the output columns of truth table of f_1, f_2 will be concatenated to provide the output column of the truth table of an n-variable function. We denote the concatenation of f_1, f_2 by $f_1 f_2$. Thus, $f = f_1 f_2$ means that in algebraic normal form, $f = (1 \oplus X_n) f_1 \oplus X_n f_2$.

Now we define an important tool for analysing Boolean functions. Let $\overline{X} = (X_n, \ldots, X_1)$ and $\overline{\omega} = (\omega_n, \ldots, \omega_1)$ be n-tuples on GF(2) and $\overline{X}.\overline{\omega} = X_n\omega_n \oplus \ldots \oplus X_1\omega_1$. Let $f(\overline{X})$ be a Boolean function whose domain is the vector space over GF$(2)^n$. Then the Walsh transform of $f(\overline{X})$ is a real valued function over

$GF(2)^n$ that can be defined as

$$W_f(\overline{\omega}) = \sum_{\overline{X}} (-1)^{f(\overline{X}) \oplus \overline{X}\cdot\overline{\omega}},$$

where the sum is over all $\overline{X}$ in $GF(2)^n$. For a function f, we define $F_f = |\{\overline{\omega} \in \{0,1\}^n \mid W_f(\overline{\omega}) \neq 0\}|$. This is the number of nonzero coefficients in the Walsh spectra.

Propagation Characteristic (PC) and Strict Avalanche Criteria (SAC) [11] are important properties of Boolean functions to be used in S-boxes. However, Zhang and Zheng [19] justified that SAC and PC have some limitations in identifying certain desirable cryptographic properties of a Boolean function. In this direction they have proposed the idea of Global Avalanche Characteristics (GAC). Next we state two important indicators of GAC.

Let $\overline{X} \in \{0,1\}^n$ be an n tuple $X_n, \ldots, X_1$ and $\overline{\alpha} \in \{0,1\}^n$ be an n tuple $\alpha_n, \ldots, \alpha_1$. Let $f \in \Omega_n$ and $\Delta_f(\overline{\alpha}) = wd(f(\overline{X}), f(\overline{X} \oplus \overline{\alpha}))$, the autocorrelation value of f with respect to the vector $\overline{\alpha}$. The sum-of-square indicator

$$\sigma_f = \sum_{\overline{\alpha} \in \{0,1\}^n} \Delta_f^2(\overline{\alpha}), \text{ and the absolute indicator } \Delta_f = \max_{\overline{\alpha} \in \{0,1\}^n, \overline{\alpha} \neq \overline{0}} |\Delta_f(\overline{\alpha})|.$$

We here concentrate on the autocorrelation spectra of correlation immune and resilient Boolean functions. In [6], the following characterization of correlation immunity is provided. A function $f(X_n, \ldots, X_1)$ is m-th order correlation immune (CI) iff its Walsh transform W_f satisfies $W_f(\overline{\omega}) = 0$, for $1 \leq wt(\overline{\omega}) \leq m$. If f is balanced then $W_f(\overline{0}) = 0$. Balanced m-th order correlation immune functions are called m-resilient functions. Thus, a function $f(X_n, \ldots, X_1)$ is m-resilient iff its Walsh transform W_f satisfies

$$W_f(\overline{\omega}) = 0, \text{ for } 0 \leq wt(\overline{\omega}) \leq m.$$

By (n, m, d, x) we denote an n-variable resilient function of order m, nonlinearity x and degree d.

It may very well happen that correlation immune or resilient functions, which are good in terms of order of correlation immunity, algebraic degree and nonlinearity, may not be good in terms of SAC or PC properties. Also getting good SAC or PC properties may not be sufficient for cryptographic purposes. There may be a function f which possesses good SAC or PC properties, but $f(\overline{X}) \oplus f(\overline{X} \oplus \overline{\alpha})$ is constant for some nonzero $\overline{\alpha}$, which is a weakness. It is important to get good autocorrelation properties for such functions. That is why, we here look into the autocorrelation properties of correlation immune and resilient functions.

For a linear function f, $\Delta_f = 2^n$, and $\sigma_f = 2^{3n}$. For functions f, on even number of variables, we have $\Delta_f = 0$ ($\sigma_f = 2^{2n}$) iff f is a bent function [9,19]. However, bent functions are not balanced. In fact, for a function f of even weight $\Delta_f \equiv 0 \bmod 8$ and for a function f of odd weight $\Delta_f \equiv 4 \bmod 8$ [5]. For balanced function f, $\sigma_f \geq 2^{2n} + 2^{n+3}$ [15] for both odd and even number of variables. A comparatively sharper result in this direction has been proposed in [16] which we will discuss shortly.

Note that the properties Δ_f, σ_f are invariant under nonsingular linear transformation on input variables of the function f. Thus, it is easy to see that the σ_f results of the papers [15,16] are valid for any Boolean function f whose Walsh spectrum contains at least one zero.

2 Lower Bounds on Sum-of-Square Indicator

We start this section with a result from [20, Theorem 3].

Theorem 1. *Let $f \in \Omega_n$. Then $\sigma_f \geq \frac{2^{3n}}{F_f}$. Moreover, if f has a three valued Walsh spectra $0, \pm 2^x$, then $\sigma_f = \frac{2^{3n}}{F_f}$.*

Next we have the following result which follows directly from the definition of correlation immunity and F_f.

Proposition 1. *Let $f \in \Omega_n$ be an m-th order correlation immune function. Then $F_f \leq 2^n - \sum_{i=1}^{m} \binom{n}{i}$. Moreover, if f is m-resilient, then $F_f \leq 2^n - \sum_{i=0}^{m} \binom{n}{i}$.*

The next result follows from Theorem 1 and Proposition 1.

Lemma 1. *Let $f \in \Omega_n$ be an m-th order correlation immune function. Then, $\sigma_f \geq \frac{2^{3n}}{2^n - \sum_{i=1}^{m} \binom{n}{i}}$. Moreover, if f is m-resilient, then $\sigma_f \geq \frac{2^{3n}}{2^n - \sum_{i=0}^{m} \binom{n}{i}}$.*

To identify important consequences of this result we need to get an approximate result which will provide a σ_f value of the form $2^{2n} + 2^{n+q}$, where q is a function of n, m.

Theorem 2. *Let $f \in \Omega_n$ be an m-th order correlation immune function. Then, $\sigma_f > 2^{2n} + 2^{n + \log_2 \left(\sum_{i=1}^{m} \binom{n}{i}\right)}$. Similarly, if f is m-resilient, then $\sigma_f > 2^{2n} + 2^{n + \log_2 \left(\sum_{i=0}^{m} \binom{n}{i}\right)}$.*

Proof. Note that $\frac{2^{3n}}{2^n - \sum_{i=1}^{m} \binom{n}{i}} > 2^{2n} + 2^n \left(\sum_{i=1}^{m} \binom{n}{i}\right)$. Thus the result follows for correlation immune functions. Similar result follows for resilient functions.

Note that, in our analysis, there is no significant difference in the result of correlation immune and resilient functions in terms of numerical values.

Currently there is no result on lower bound of σ_f values for correlation immune and resilient functions. The only known results are for balanced functions which are given in [15,16]. The lower bound for balanced functions given in [15] is $2^{2n} + 2^{n+3}$. The result in [16] is as follows. For a balanced function f,

$$\begin{aligned}
\sigma_f \geq\ & 2^{2n} + 2^6(2^n - t - 1), && \text{if } 0 \leq t \leq 2^n - 2^{n-3} - 1,\ t \text{ odd}, && \text{(i)}\\
& 2^{2n} + 2^6(2^n - t + 2), && \text{if } 0 \leq t \leq 2^n - 2^{n-3} - 1,\ t \text{ even}, && \text{(ii)}\\
& \left(1 + \frac{1}{2^n - 1 - t}\right)2^{2n}, && \text{if } 2^n - 2^{n-3} - 1 < t \leq 2^n - 2, && \text{(iii)}
\end{aligned}$$

if f satisfies propagation characteristics with respect to t vectors. Note that for case (i) and (ii), even if we overestimate this lower bound, it is $2^{2n} + 2^{n+6}$. For the case (iii) the lower bound varies from $2^{2n} + 2^{n+3}$ to 2^{2n+1} and also this depends on the propagation characteristics of the function.

Now we enumerate the consequences of our result.

- In our result the lower bound depends directly on the order m of correlation immunity and this is the first nontrivial result in this direction.
- Note that for $m > \frac{n}{2}$, $\log_2 \left(\sum_{i=1}^{m} \binom{n}{i}\right) > n - 1$. Thus for all m-th order correlation immune functions with $m > \frac{n}{2}$, $\sigma_f > 2^{2n} + 2^{2n-1}$. The result is true for m-resilient functions also. This provides a strong lower bound on sum-of-square indicator for m-th order correlation immune and m-resilient functions.
- Given any value r $(1 \leq r < n)$, it is possible to find an m-th order correlation immune or m-resilient function f such that $\sigma_f > 2^{2n} + 2^{n+r}$ by properly choosing m.

3 Lower Bounds on Absolute Indicator

Now we concentrate on the absolute indicator of GAC. We have the result on sum-of-square indicator for correlation immune and resilient functions. We use the result in this direction.

Lemma 2. *For an n-variable m-th order correlation immune function f, $\Delta_f \geq \sqrt{\frac{1}{2^n-1}\frac{2^{2n}\left(\sum_{i=1}^{m}\binom{n}{i}\right)}{2^n - \sum_{i=1}^{m}\binom{n}{i}}}$. Similarly, $\Delta_f \geq \sqrt{\frac{1}{2^n-1}\frac{2^{2n}\left(\sum_{i=0}^{m}\binom{n}{i}\right)}{2^n - \sum_{i=0}^{m}\binom{n}{i}}}$ for an n-variable m-resilient function f.*

Proof. We know, $\sigma_f = \sum_{\overline{\alpha} \in \{0,1\}^n} \Delta_f^2(\overline{\alpha})$. Thus, the absolute value of each $\Delta_f(\overline{\alpha})$ will be minimum only when they all possess equal values. Hence, the minimum value of Δ_f will be $\sqrt{\frac{\sigma_f - 2^{2n}}{2^n-1}}$. This gives the result using the value of σ_f from Lemma 1.

In [22], it has been shown that $\Delta_f \geq 2^{m-1} \sum_{i=0}^{+\infty} 2^{i(m-1-n)}$ for an unbalanced n-variable m-th order correlation immune function for the range $2 \leq m \leq n$. Note that, $\Delta_f \geq 2^{m-1} \sum_{i=0}^{+\infty} 2^{i(m-1-n)} = 2^{m-1}\frac{1}{1-2^{m-1-n}}$. Also $\Delta_f \geq 2^m \sum_{i=0}^{+\infty} 2^{i(m-n)}$ for an n-variable m-resilient function for the range $1 \leq m \leq n-1$. This gives, $\Delta_f \geq 2^m \sum_{i=0}^{+\infty} 2^{i(m-n)} = 2^m \frac{1}{1-2^{m-n}}$.

For lower order of correlation immunity $(m \leq \frac{n}{2} - 1)$, and lower order of resiliency $(m \leq \frac{n}{2} - 2)$, our result in Lemma 2 is better than the result of [22]. Since the expressions are too complicated to compare, we provide a table below to substantiate our claim. We present the comparison for n-variable, m-resilient functions. Note that the Δ_f values for balanced functions are divisible by 8. Thus after calculating the expressions we increase the values to the closest integer divisible by 8. In each row we first provide the Δ_f values from Lemma 2 and

Table 1. Comparison of our results with that of [22].

n ^{m}	1	2	3	4	5	6	7	8	9	10	11	12	13	14
8	8	8												
	8	8												
9	8	8												
	8	8												
10	8	8	16											
	8	8	16											
11	8	16	24											
	8	8	16											
12	8	16	24	32										
	8	8	16	24										
13	8	16	24	40										
	8	8	16	24										
14	8	16	24	48	72									
	8	8	16	24	40									
15	8	16	32	48	80									
	8	8	16	24	40									
16	8	16	32	56	88	144								
	8	8	16	24	40	72								
17	8	16	32	64	104	168								
	8	8	16	24	40	72								
18	8	16	32	72	120	192	288							
	8	8	16	24	40	72	136							
19	8	16	40	72	136	224	344							
	8	8	16	24	40	72	136							
20	8	16	40	80	152	256	400	600						
	8	8	16	24	40	72	136	264						
21	8	16	40	88	176	296	472	712						
	8	8	16	24	40	72	136	264						
22	8	16	48	96	192	344	552	840	1224					
	8	8	16	24	40	72	136	264	520					
23	8	24	48	112	216	392	648	1000	1464					
	8	8	16	24	40	72	136	264	520					
24	8	24	56	120	240	440	752	1176	1752	2496				
	8	8	16	24	40	72	136	264	520	1032				
25	8	24	56	128	264	504	864	1384	2088	3008				
	8	8	16	24	40	72	136	264	520	1032				
26	8	24	56	136	296	568	1000	1624	2488	3624	5096			
	8	8	16	24	40	72	136	264	520	1032	2056			
27	8	24	64	152	320	632	1144	1904	2960	4360	6176			
	8	8	16	24	40	72	136	264	520	1032	2056			
28	8	24	64	160	352	712	1304	2216	3504	5232	7480	10368		
	8	8	16	24	40	72	136	264	520	1032	2056	4104		
29	8	24	64	168	384	792	1488	2560	4128	6264	9056	12640		
	8	8	16	24	40	72	136	264	520	1032	2056	4104		
30	8	24	72	184	424	880	1680	2960	4848	7472	10944	15400	21064	
	8	8	16	24	40	72	136	264	520	1032	2056	4104	8200	
31	8	24	72	192	456	976	1896	3400	5672	8880	13184	18744	25800	
	8	8	16	24	40	72	136	264	520	1032	2056	4104	8200	
32	8	24	80	208	496	1080	2128	3888	6600	10512	15832	22768	31592	42736
	8	8	16	24	40	72	136	264	520	1032	2056	4104	8200	16392

then under that we provide the result from [22]. It is clear that our result provide a considerable improvement over that of [22] as both n, m increases.

Using simplification in Lemma 2 we get the following result.

Theorem 3. *For an n-variable m-th order correlation immune function f, $\Delta_f > 2^{\frac{n}{2}} \sqrt{\frac{\sum_{i=1}^{m} \binom{n}{i}}{2^n - \sum_{i=1}^{m} \binom{n}{i}}}$. Similarly, $\Delta_f > 2^{\frac{n}{2}} \sqrt{\frac{\sum_{i=0}^{m} \binom{n}{i}}{2^n - \sum_{i=0}^{m} \binom{n}{i}}}$ for an n-variable m-resilient function f.*

Proof. The result follows from overestimating $2^n - 1$ by 2^n.

It is known that, for a function f of even weight, $\Delta_f \equiv 0 \bmod 8$. Since the correlation immune functions and resilient functions are all of even weight, the Δ_f values will be the value greater than the values given in Theorem 3, which are divisible by 8. Our result has the following consequences.

- The value Δ_f is a function of n, m.
- For small values of m, $\Delta_f > \sqrt{\sum_{i=1}^{m} \binom{n}{i}} > \sqrt{\binom{n}{m}}$.
- For $m = 1$, $\Delta_f > \sqrt{n}$.

4 Lower Bounds Using Weight Divisibility Results

Here we use the weight divisibility results of correlation immune and resilient Boolean functions [12]. It is known that the values in the Walsh spectrum of an m-th order correlation immune function is divisible by 2^{m+1}. Similarly for m-resilient functions, the Walsh spectrum values are divisible by 2^{m+2}.

Let us now find out the sum of square indicators of such functions. We once again refer to Theorem 1. For $f \in \Omega_n$, $\sigma_f \geq \frac{2^{3n}}{\mathcal{F}_f}$.

- For an n-variable, m-th order correlation immune function the values in Walsh spectra are $0, \pm i 2^{m+1}, i = 1, 2, \ldots$. From Parseval's relation [4] we get $\sum_{\overline{\omega} \in \{0,1\}^n} W_f^2(\overline{\omega}) = 2^{2n}$. Hence, we get that for such a function f, $\mathcal{F}_f \leq 2^{2n-2m-2}$.
- For an n-variable, m-resilient function the Walsh spectra contain the values $0, \pm i 2^{m+2}, i = 1, 2, \ldots$. Using Parseval's relation, we get that for such a function f, $\mathcal{F}_f \leq 2^{2n-2m-4}$.

Theorem 4. *For an n-variable, m-th order correlation immune function f, $\sigma_f \geq 2^{n+2m+2}$ and for an n-variable, m-resilient function f, $\sigma_f \geq 2^{n+2m+4}$.*

Proof. It is known from [12] that for m-th order correlation immune (respectively m-resilient) function the nonzero values of the Walsh spectra will always be divisible by 2^{m+1} (respectively 2^{m+2}). Thus, using Parseval's relation we get that for correlation immune (respectively resilient) function $\mathcal{F}_f \leq 2^{2n-2m-2}$ (respectively $\mathcal{F}_f \leq 2^{2n-2m-4}$). Hence the result follows from Theorem 1.

Note that the trivial lower bound on the sum of square indicator is 2^{2n}. Hence, for correlation immune functions, this bound is nontrivial, when $n+2m+2 > 2n$, i.e, $m > \frac{n}{2} - 1$. Similarly for resilient functions, this bound is nontrivial for $m > \frac{n}{2} - 2$.

The weight divisibility results using algebraic degree of the functions have been presented in [2,3]. These results can be used to provide a sharper lower bound on σ_f involving algebraic degree. From [2,3], it is clear that for an n-variable, m-th order correlation immune function with algebraic degree d, the values of the Walsh spectra will be divisible by $2^{m+1+\frac{n-m-1}{d}}$. Similarly for an n-variable, m-resilient function with algebraic degree d, the values of the Walsh spectra will be divisible by $2^{m+2+\frac{n-m-2}{d}}$. Using these results we can update Theorem 4 involving algebraic degree as follows.

Theorem 5. *For an n-variable, m-th order ($m > \frac{n}{2}-1$) correlation immune (respectively resilient) function f with algebraic degree d, $\sigma_f \geq 2^{n+2m+2+2\frac{n-m-1}{d}}$ (respectively $\sigma_f \geq 2^{n+2m+4+2\frac{n-m-2}{d}}$).*

Next we concentrate on a very important subset of correlation immune and resilient functions which possess maximum possible nonlinearity. Importantly the resilient functions have direct application in stream cipher systems. Now the clear benchmark in selecting the resilient functions is the functions which possess the best possible trade-off among the parameters nonlinearity, algebraic degree and the order of resiliency. However, we point out that we should consider one more important criteria in the selection process. In fact we find functions with best possible trade-off having same values of nonlinearity, algebraic degree and order of resiliency but having different autocorrelation properties. Thus, it is important to select the one with better Δ_f values. It is also interesting to note that any two functions with this best possible trade-off must possess the same σ_f values, which we will show shortly. For this we concentrate on definition of plateaued functions [20, Definition 9]. Apart from the bent and linear functions, the other plateaued functions have the property that they have three valued Walsh spectra $0, \pm 2^x$. Next we once again concentrate on Theorem 1 (the result from [20, Theorem 3]). Let $f \in \Omega_n$ and f has a three valued Walsh spectra $0, \pm 2^x$. Then $\sigma_f = \frac{2^{3n}}{\mathrm{F}_f}$. We present the following known [12] results.

- For an n-variable, m-th order correlation immune function with $m > \frac{n}{2} - 1$, the maximum possible nonlinearity that can be achieved is $2^{n-1} - 2^m$ and these functions possess three valued Walsh spectra $0, \pm 2^{m+1}$. Thus from Parseval's relation [4] $\sum_{\varpi \in \{0,1\}^n} W_f^2(\varpi) = 2^{2n}$. Hence, we get that for such a function f, $\mathrm{F}_f = 2^{2n-2m-2}$.
- For an n-variable, m-resilient function with $m > \frac{n}{2}-2$, the maximum possible nonlinearity that can be achieved is $2^{n-1} - 2^{m+1}$ and these functions possess three valued Walsh spectra $0, \pm 2^{m+2}$. Using Parseval's relation, we get that for such a function f, $\mathrm{F}_f = 2^{2n-2m-4}$.

Hence we get the following result.

Theorem 6. *For an n-variable, m-th* ($m > \frac{n}{2} - 1$) *order correlation immune function f with maximum possible nonlinearity,* $\sigma_f = 2^{n+2m+2}$. *Similarly, for an n-variable, m-resilient* ($m > \frac{n}{2} - 2$) *function f with maximum possible nonlinearity,* $\sigma_f = 2^{n+2m+4}$.

Proof. The result for correlation immune (respectively resilient) function follows from Theorem 1 and $F_f = 2^{2n-2m-2}$ (respectively $F_f = 2^{2n-2m-4}$) [12].

Current results [17,2,3] clearly identify that the nonlinearity and algebraic degree of the correlation immune and resilient functions are optimized simultaneously. Theorem 6 provides the result when the nonlinearity is maximized. Thus, the algebraic degree is also maximized in this case. Here we show that at this situation, the sum of square indicator attains its minimum value too. This gives that for an n-variable, m-resilient function the nonlinearity, algebraic degree and sum of square indicator of autocorrelation values are optimized simultaneously.

5 Construction Results

Resilient Boolean functions, which are provably optimized in terms of order of resiliency, algebraic degree and nonlinearity [12], have immediate applications in stream cipher systems. Unfortunately, the general construction techniques does not provide good autocorrelation properties. First we will talk about some specific resilient functions and their Δ_f values. Then we will analyze some of the well known constructions and calculate the autocorrelation values.

Let us consider the $(5, 1, 3, 12)$ functions. We initially consider such a function f constructed using linear concatenation, which is $(1 \quad X_5)(1 \quad X_4)(X_1 \quad X_2) \quad (1 \quad X_5)X_4(X_1 \quad X_3) \quad X_5(1 \quad X_4)(X_2 \quad X_3) \quad X_5X_4(X_1 \quad X_2 \quad X_3)$. This function has $\Delta_f = 16$. However, using search techniques, it is possible to get a $(5, 1, 3, 12)$ function g, such that $\Delta_g = 8$. The truth table of the function is 00001011110110011110010100111000. *This function achieves the best possible trade-off among order of resiliency, nonlinearity, algebraic degree and autocorrelation.*

Recently (7, 2, 4, 56) [10] and (8, 1, 6, 116) [7] functions have been found by computer search. It has been reported that the minimum Δ_f values for these two cases (so far found by computer search) are 32, 80 respectively. However, the existing generalized recursive construction results are not very good in terms of the autocorrelation values. We now discuss the absolute indicator values of autocorrelation for some of these constructions.

5.1 Recursive Construction I

Here we consider the recursive construction which has been discussed in [1,8] in different forms. We consider the notation in [8] here for constructing an $(n + 1)$-variable function F from two n-variable functions f, g.

$$Q_i(f(X_n, \ldots, X_1), g(X_n, \ldots, X_1)) = F(X_{n+1}, \ldots, X_1)$$
$$= (1 \quad X_i)f(X_n, \ldots, X_{i+1}, X_{i-1}, \ldots, X_1) \quad X_i g(X_n, \ldots, X_{i+1}, X_{i-1}, \ldots, X_1).$$

Let f be an n-variable, m-resilient degree d function having nonlinearity x. Define $F(X_{n+1},\ldots,X_1)$ to be an $(n+1)$-variable function as $F(X_{n+1},\ldots,X_1) = Q_i(f(X_n,\ldots,X_1), a \quad f(b \quad X_n,\ldots,b \quad X_1))$. Here $a, b \quad \{0,1\}$ and if m is even $a = b$ and if m is odd, $a = 1$ and b can be either 0 or 1. Then $F(X_{n+1}, X_n,\ldots,X_1)$ is an $(m+1)$-resilient, degree d function having nonlinearity $2x$ [8].

Note that, any of the operators Q_i can be expressed as a composition of Q_{n+1} and a suitable permutation of the input variables. The permutation of input variables preserves the autocorrelation property, resiliency, algebraic degree and nonlinearity. So it is enough to look into the construction function as $F(X_{n+1},\ldots,X_1) = Q_{n+1}(f(X_n,\ldots,X_1), a \quad f(b \quad X_n,\ldots,b \quad X_1))$, *i.e.*, $F(X_{n+1},\ldots,X_1) = (1 \quad X_{n+1})f(X_n,\ldots,X_1) \quad X_{n+1}(a \quad f(b \quad X_n,\ldots,b \quad X_1))$.

First consider the case when m is even. Then $a = b$. Let us consider, $a = 1, b = 0$, then $F(X_{n+1},\ldots,X_1) = (1 \quad X_{n+1})f(X_n,\ldots,X_1) \quad X_{n+1}(1 \quad f(X_n,\ldots,X_1)) = X_{n+1} \quad f(X_n,\ldots,X_1)$. It is clear that $\Delta_f(1,0,\ldots,0) = -2^{n+1}$.

If we consider $a = 0, b = 1$, $F(X_{n+1},\ldots,X_1) = (1 \quad X_{n+1})f(X_n,\ldots,X_1) \quad X_{n+1}f(1 \quad X_n,\ldots,1 \quad X_1)$. Then, $\Delta_f(1,1,\ldots,1) = 2^{n+1}$.

Similarly it can be shown that for the case m odd, there will be linear structures in this construction. Thus, for this recursive construction, for an n variable function, the absolute indicator value is 2^n.

5.2 Recursive Construction II

Now we consider the construction [17] which was later modified in [10]. An (n,m,d,x) function f is said to be in *desired* form [10] if it is of the form $(1 \quad X_n)f_1 \quad X_nf_2$, where f_1, f_2 are $(n-1, m, d-1, x-2^{n-2})$ functions. This means that the nonzero values of the Walsh spectra of f_1, f_2 do not intersect, i.e., if $W_{f_1}(\varpi) = 0$, then $W_{f_2}(\varpi) = 0$, and vice versa. Let f be an (n,m,d,x) function in the *desired* form, where f_1, f_2 are both $(n-1, m, d-1, x-2^{n-2})$ functions. Let $F = X_{n+2} \quad X_{n+1} \quad f$ and $G = (1 \quad X_{n+2} \quad X_{n+1})f_1 \quad (X_{n+2} \quad X_{n+1})f_2 \quad X_{n+2} \quad X_n$. Note that in the language of [17], the function G above is said to depend quasilinearly on the pair of variables (X_{n+2}, X_{n+1}). Also, $F_1 = (1 \quad X_{n+3})F \quad X_{n+3}G$. The function F_1 constructed from f above is an $(n+3, m+2, d+1, 2^{n+1}+4x)$ function in the desired form.

Consider the case $\alpha_{n+3} = 0, \alpha_{n+2} = \alpha_{n+1} = 1$ and any pattern for $\alpha_n,\ldots,\alpha_1$. In this case, $F(X_{n+2},\ldots,X_1) = F(X_{n+2} \quad \alpha_{n+2},\ldots,X_1 \quad \alpha_1)$ and hence we get $\Delta_F(\alpha_{n+2},\ldots,\alpha_1) = 2^{n+2}$. On the other hand, $G(X_{n+2},\ldots,X_1) \quad G(X_{n+2} \quad \alpha_{n+2},\ldots,X_1 \quad \alpha_1) = f_1 \quad f_2 \quad 1$. Note that, if the nonzero values of the Walsh spectra of f_1, f_2 do not intersect, then $f_1 \quad f_2$ is balanced [13], i.e., $f_1 \quad f_2 \quad 1$ is also balanced. Hence, $\Delta_G(\alpha_{n+2},\ldots,\alpha_1) = 0$. This gives that $\Delta_{F_1}(\alpha_{n+3},\ldots,\alpha_1) = \Delta_F(\alpha_{n+2},\ldots,\alpha_1) + \Delta_G(\alpha_{n+2},\ldots,\alpha_1) = 2^{n+2} + 0 = 2^{n+2}$. So, $\Delta_{F_1} \geq 2^{n+2}$.

Thus, for this recursive construction, for an n variable function the absolute indicator value is greater than or equal to 2^{n-1}.

It will be interesting to find out a construction which provides good Δ_f value for resilient functions f with best possible nonlinearity, algebraic degree and σ_f values.

References

1. P. Camion, C. Carlet, P. Charpin, and N. Sendrier. On correlation immune functions. In *Advances in Cryptology - CRYPTO'91*, pages 86–100. Springer-Verlag, 1992.
2. C. Carlet. On the coset weight divisibility and nonlinearity of resilient and correlation immune functions. In *Sequences and Their Applications*, SETA 2001.
3. C. Carlet and P. Sarkar. Spectral domain analysis of correlation immune and resilient Boolean functions. Accepted in *Finite Fields and Its Applications*, 2001.
4. C. Ding, G. Xiao, and W. Shan. *The Stability Theory of Stream Ciphers*. Number 561 in Lecture Notes in Computer Science. Springer-Verlag, 1991.
5. C. Ding and P. Sarkar. *Personal Communications*, 2000.
6. X. Guo-Zhen and J. Massey. A spectral characterization of correlation immune combining functions. *IEEE Transactions on Information Theory*, 34(3):569–571, May 1988.
7. S. Maitra and E. Pasalic. Further constructions of resilient Boolean functions with very high nonlinearity. In *Sequences and Their Applications*, SETA 2001.
8. S. Maitra and P. Sarkar. Cryptographically significant Boolean functions with five valued Walsh spectra. *Accepted in Theoretical Computer Science*, 2001.
9. W. Meier and O. Staffelbach. Nonlinearity criteria for cryptographic functions. In *Advances in Cryptology - EUROCRYPT'89*, pages 549–562. Springer-Verlag, 1990.
10. E. Pasalic, T. Johansson, S. Maitra, and P. Sarkar. New constructions of resilient and correlation immune Boolean functions achieving upper bounds on nonlinearity. In *Workshop on Coding and Cryptography*, Electronic Notes in Discrete Mathematics. Elsevier, January 2001.
11. B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle. Propagation characteristics of Boolean functions. In *Advances in Cryptology - EUROCRYPT'90*, Lecture Notes in Computer Science, pages 161–173. Springer-Verlag, 1991.
12. P. Sarkar and S. Maitra. Nonlinearity bounds and constructions of resilient Boolean functions. In *Advances in Cryptology - CRYPTO 2000*, number 1880 in Lecture Notes in Computer Science, pages 515–532. Springer Verlag, 2000.
13. P. Sarkar and S. Maitra. Cross-correlation analysis of cryptographically useful Boolean functions and S-boxes. *Preprint*, 2001.
14. T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, IT-30(5):776–780, September 1984.
15. J. J. Son, J. I. Lim, S. Chee, and S. H. Sung. Global avalanche characteristics and nonlinearity of balanced Boolean functions. *Information Processing Letters*, 65:139–144, 1998.
16. S. H. Sung, S. Chee, and C. Park. Global avalanche characteristics and propagation criterion of balanced Boolean functions. *Information Processing Letters*, 69:21–24, 1999.
17. Y. V. Tarannikov. On resilient Boolean functions with maximum possible nonlinearity. In *Progress in Cryptology - INDOCRYPT 2000*, number 1977 in Lecture Notes in Computer Science, pages 19–30. Springer Verlag, 2000.
18. Y. V. Tarannikov, P. Korolev and A. Botev. Autocorrelation coefficients and correlation immunity of Boolean functions. Accepted in *ASIACRYPT 2001*, to be published in Lecture Notes in Computer Science. Springer Verlag, 2001.

19. X. M. Zhang and Y. Zheng. GAC - the criterion for global avalanche characteristics of cryptographic functions. *Journal of Universal Computer Science*, 1(5):316–333, 1995.
20. Y. Zheng and X. M. Zhang. Plateaued functions. In *ICICS'99*, number 1726 in Lecture Notes in Computer Science, pages 284–300. Springer Verlag, 1999.
21. Y. Zheng and X. M. Zhang. Improving upper bound on nonlinearity of high order correlation immune functions. In *SAC 2000*, Lecture Notes in Computer Science. Springer Verlag, 2000.
22. Y. Zheng and X. M. Zhang. New results on correlation immunity. In *ICISC 2000*, Lecture Notes in Computer Science. Springer Verlag, 2000.
23. Y. Zheng and X. M. Zhang. On relationships among propagation degree, nonlinearity and correlation immunity. In *Advances in Cryptology - ASIACRYPT'00*, Lecture Notes in Computer Science. Springer Verlag, 2000.

On the Constructing of Highly Nonlinear Resilient Boolean Functions by Means of Special Matrices

Maria Fedorova and Yuriy Tarannikov

Mech. & Math. Department, Moscow State University,
119899 Moscow, Russia,
maria_fedorova@yahoo.com,
yutaran@mech.math.msu.su, taran@vertex.inria.msu.ru

Abstract. In this paper we consider matrices of special form introduced in [11] and used for the constructing of resilient functions with cryptographically optimal parameters. For such matrices we establish lower bound $\frac{1}{\log_2(\sqrt{5}+1)} = 0.5902\ldots$ for the important ratio $\frac{t}{t+k}$ of its parameters and point out that there exists a sequence of matrices for which the limit of ratio of these parameters is equal to lower bound. By means of these matrices we construct m-resilient n-variable functions with maximum possible nonlinearity $2^{n-1} - 2^{m+1}$ for $m = 0.5902\ldots n + O(\log_2 n)$. This result supersedes the previous record.

Keywords: stream cipher, Boolean function, nonlinear combining function, correlation-immunity, resiliency, nonlinearity, special matrices.

1 Introduction

Different types of ciphers use Boolean functions. So, LFSR based stream ciphers use Boolean functions as a nonlinear combiner or a nonlinear filter, block ciphers use Boolean functions in substitution boxes and so on. Boolean functions used in ciphers must satisfy some specific properties to resist different attacks. One of the most important desired properties of Boolean functions in LFSR based stream ciphers is *correlation immunity* introduced by Siegenthaler [9]. Another important properties are nonlinearity, algebraic degree and so on.

The most usual theoretic motivation for the investigation of highly nonlinear resilient Boolean functions is the using of such functions as nonlinear combiners in stream ciphers. But from the practical point of view the number of variables in such system can not be too big (in opposite case the key length will be too long). It is necessary to note that all important functions with small number of variables are found already by exhaustive search. At the same time another important practical type of stream ciphers uses Boolean functions as nonlinear filters. Here, in general, it is possible to use the functions with big number of variables. But the main problems here is that effective (from implementation point of view) constructions of such functions can not be found by exhaustive

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 254–266, 2001.

search, and also it was pointed out [4] that stream cipher of such type can be transformed into an equivalent (in some sence) with worse resiliency but the same nonlinearity. It emphasizes the importance of direct effective constructions of Boolean functions with big number of variables and optimal combination of resiliency and nonlinearity.

Correlation immunity (or resiliency) is the property important in cryptography not only in stream ciphers. This is an important property if we want that the knowledge of some specified number of input bits does not give a (statistical) information about the output bit. In this respect such functions are considered in [3], [2] and other works.

It was proved independently in [8], [10] and [12] that the nonlinearity of n-variable m-resilient function does not exceed $2^{n-1} - 2^{m+1}$ for $m \le n - 1$. It was proved that if this bound is achieved then $m > 0.5n - 2$. In [10] it was proved that if this bound is achieved then the algebraic degree of the function is maximum possible too (i. e. achieves *Siegenthaler's Inequality*) and equal to $n - m - 1$. In [10], [6] and [11] effective constructions of m-resilient n-variable functions with maximum possible nonlinearity $2^{n-1} - 2^{m+1}$ for $m \ge \frac{2n-7}{3}$, $m \ge \frac{2n-9}{3}$ and $m \ge 0.6n - 1$ correspondently were given. To obtain this result in [11] the concept of a *proper* (k_0, k, p, t)-matrix were introduced. In [11] it was pointed out that the mostly important to find a proper (k, k, p, t)-matrix where the ratio $\frac{t}{t+k}$ is as small as possible. In [11] it was given a proper $(4, 4, 6, 6)$-matrix for which this ratio is 0.6. At the same time the lowest possible value of the ratio $\frac{t}{t+k}$ for proper matrices was formulated in [11] as the open problem. In the present paper we investigate the problem of the lowest possible value of the ratio $\frac{t}{t+k}$ for proper matrices and establish that this ratio can not be less than $\frac{1}{\log_2(\sqrt{5}+1)} = 0.5902...$ At the same time we construct proper matrices that approach this lower bound with arbitrary precision. By means of these matrices we construct m-resilient n-variable functions with maximum possible nonlinearity $2^{n-1} - 2^{m+1}$ for $m = 0.5902\ldots n + O(\log_2 n)$. Note that our nonexistence results demonstrate that only proper matrices technique is not sufficient to construct m-resilient n-variable functions with maximum possible nonlinearity $2^{n-1} - 2^{m+1}$ for $m < 0.5902\ldots n + O(1)$. At the same time it is quite possible that such functions there exist for any m, n provide $0.5n - 2 < m \le n - 2$. At least an opposite result have not proved. Thus, the constructing of such functions demands new methods and new techniques.

The rest of this paper is organized as follows. In Section 2 we give preliminary concepts and notions. In Section 3 we formulate necessary concepts and results from the previous work [11] on proper matrices. In Section 4 we prove that there does not exist a proper (k_0, k, p, t)-matrix if $\frac{t}{k+t} < \frac{1}{\log_2(\sqrt{5}+1)} = 0.5902...$ In Section 5 we construct proper (k_0, k, p, t)-matrices with ratio $\frac{t}{k+t}$ close to $\frac{1}{\log_2(\sqrt{5}+1)}$ and $k > \alpha k_0$ where $\alpha < \sqrt{5}\log_2\left(\frac{\sqrt{5}+1}{2}\right) = 1.5523\ldots$. In Section 6 by means of proper matrices constructed in Section 5 we construct m-resilient n-variable functions with maximum possible nonlinearity $2^{n-1} - 2^{m+1}$ for $m = \frac{1}{\log_2(\sqrt{5}+1)}n + O(\log_2 n) = 0.5902\ldots n + O(\log_2 n)$. In Section 7 we discuss the method that probably gives the best possible in some sence concrete proper matrices.

2 Preliminary Concepts and Notions

We consider V^n, the vector space of n tuples of elements from $GF(2)$. A *Boolean function* is a function from V^n to $GF(2)$. The *weight* $wt(f)$ of a function f on V^n is the number of vectors x on V^n such that $f(x) = 1$. A function f is said to be *balanced* if $wt(f) = wt(f \quad 1)$. Obviously, if a function f on V^n is balanced then $wt(f) = 2^{n-1}$. A *subfunction* of the Boolean function f is a function f obtained by substitution some constants for some variables in f. If a variable x_i is not substituted by constant then x_i is called a *free* variable for f .

The *Hamming distance* $d(x\ , x\ )$ between two vectors x and x is the number of components where vectors x and x differ. For two Boolean functions f_1 and f_2 on V^n, we define the distance between f_1 and f_2 by $d(f_1, f_2) = \#\{x \quad V^n | f_1(x) = f_2(x)\}$. The minimum distance between f and the set of all affine functions (i. e. functions of the form $f(x) = c_0 \quad \overset{\boxed{n}}{\underset{i=1}{}} c_i x_i)$ is called the *nonlinearity* of f and denoted by $nl(f)$.

A Boolean function f on V^n is said to be *correlation-immune of order* m, with $1 \leq m \leq n$, if $wt(f\ ) = wt(f)/2^m$ for any its subfunction f of $n - m$ variables. This concept was introduced by Siegenthaler [9]. A balanced mth order correlation immune function is called an *m-resilient* function. From this point of view it is possible to consider formally any balanced Boolean function as 0-resilient (this convention is accepted in [1], [7], [5]) and an arbitrary Boolean function as (-1)-resilient (this convention is accepted in [10] and [11]). The concept of an m-resilient function was introduced in [3].

3 Results of Previous Work on Proper Matrices

In [11] for the constructing of new m-resilient n-variable Boolean functions with maximum possible nonlinearity $2^{n-1} - 2^{m+1}$ the concept of a *proper matrix* was introduced.

Definition 1. [11] *Let $B = (b_{ij})$ be $(2^k \times p)$ matrix of 2^k rows and p columns with entries from the set $\{1, 2, \quad \}$. Let k_0 and t be positive integers. We assume that*

(i) for every two rows i_1 and i_2 there exists a column j such that $b_{i_1 j} = 1$, $b_{i_2 j} = 2$ or $b_{i_1 j} = 2$, $b_{i_2 j} = 1$.

(ii) for every row i the inequality $\overset{\boxed{p}}{\underset{j=1}{}} b_{ij} \leq t$ holds (a sign does not give an influence to these sums).

(iii) in every row the number of ones does not exceed k_0.

If the matrix B satisfies all properties (i), (ii), (iii) we say that B is a proper *(k_0, k, p, t)-matrix.*

The proper (k_0, k, p, t)-matrix is denoted in [11] by $B_{k_0,k,p,t}$.

The next definitions were given in [11].

Definition 2. *A Boolean function $f = f(x_1, \dots, x_n)$ depends on a pair of its variables (x_i, x_j)* quasilinearly *if $f(x') = f(x'')$ for any two vectors x' and x'' of length n that differ only in ith and jth components. A pair (x_i, x_j) in this case is called a* pair of quasilinear variables *in f.*

Definition 3. *Let F be a set of Boolean functions such that for every s, $0 \le s \le k$, the set F contains an $(m+s)$-resilient function on V^{n+s} with nonlinearity at least $2^s(2^{n-1} - 2^{m+\lambda})$ (λ is not necessary integer). Moreover, we assume that each f_s contains s disjoint pairs of quasilinear variables. Then we say that F is a $S_{n,m,k,\lambda}$-*system *of Boolean functions.*

The next theorem was proved in [11].

Theorem 1. *[11] Suppose that there exists an $S_{n,m,k_0,\lambda}$-system of Boolean functions F and there exists a proper (k_0, k, p, t)-matrix B, $n \ge 2p - t$. Then there exists an $S_{n+k+t,m+t,k,\lambda}$-system of Boolean functions.*

An application of the construction given in Theorem 1 is denoted in [11] by $S_{n,m,k_0,\lambda}T_{k_0,k,p,t} = S_{n+k+t,m+t,k,\lambda}$.

Lemma 1. *[11] There exists an $S_{2,-1,2,1}$-system of Boolean functions.*

Indeed, the functions $f_0 = x_1x_2$, $f_1 = (x_1 \oplus x_2)x_3 \oplus x_1$, $f_2 = (x_1 \oplus x_2)(x_3 \oplus x_4) \oplus x_1 \oplus x_3$ forms the $S_{2,-1,2,1}$-system of Boolean functions, i. e. for $i = 0, 1, 2$ the system contains $(2+i)$-variable $(-1+i)$-resilient Boolean function of nonlinearity $2^{1+i} - 2^i$.

The results of [11] demonstrate that if there exists a proper (k, k, p, t)-matrix then there exists a constant C such that for any n and m provided $m \ge \frac{t}{k+t}n + C$ there exists an m-resilient n-variable Boolean function with the nonlinearity $2^{n-1} - 2^{m+1}$. Thus, the important problem is to construct a proper (k, k, p, t)-matrix with ratio $\frac{t}{k+t}$ as small as possible. In [11] it was given an example of a proper $(4, 4, 6, 6)$-matrix where the value $\frac{t}{k+t}$ is equal to 0.6.

In this work we study the problem of the existence of proper (k_0, k, p, t)-matrices.

4 Lower Bound for the Value $\frac{t}{k+t}$

In this Section we prove that there does not exist a proper (k_0, k, p, t)-matrix if $\frac{t}{k+t} < \frac{1}{\log_2(\sqrt{5}+1)} = 0.5902\dots$

Lemma 2. *If there exists a proper (k_0, k, p, t)-matrix B then for any $p' > p$ there exists a proper (k_0, k, p', t)-matrix.*

Proof. We obtain a proper (k_0, k, p', t)-matrix simply adding $p' - p$ new all-$*$ columns to B.

The next lemma is obvious.

Lemma 3. *If there does not exist a proper (k_0, k, p, t)-matrix B then for any $k_0' < k_0$ there does not exist a proper (k_0', k, p, t)-matrix.*

In this paper we consider *a Boolean cube* B^p as the set of all vectors $(x_1, \ldots, x_p)$ where $x_i \in \{1, 2\}$. The *lth level* of the Boolean cube B^p is the set of all vectors of B^p with exactly l ones. The cardinality of lth level of B^p is $\binom{p}{l}$. A proper (k_0, k, p, t)-matrix B can be interpreted [11] as a collection of 2^k disjoint subcubes in Boolean cube $\{1, 2\}^p$. Indeed, a row of B can be interpreted as a subcube where the components with $*$ are free whereas the components with 1 or 2 are substituted by correspondent constants. The property (i) of a proper matrix provides that subcubes are disjoint. The properties (ii) and (iii) characterize the location of subcubes in a cube and the size of subcubes.

Theorem 2. *There does not exist a proper (k_0, k, p, t)-matrix for*

$$\frac{t}{k+t} < \frac{1}{\log_2(\sqrt{5}+1)} = 0.5902\ldots$$

Proof. By Lemma 3 it is sufficient to prove this theorem for $k_0 = t$.

Let B be an arbitrary proper (t, k, p, t)-matrix. We can consider B as the set of disjoint subcubes of the Boolean cube B^p if we consider each row of B as a subcube. These subcubes are disjoint by item (i) in definition 1 of a proper matrix.

If t is even then we replace in rows with odd number of ones some asterisk by one (if there are not asterisks in a row then we add preliminary all-$*$ column to the matrix B, after this procedure the parameter p will increase but this is not important for us). If t is odd we do the same for all rows with even number of ones. Now for even t all rows contain even number of ones and for odd t all rows contain odd number of ones. If the matrix B contains rows where the sum of ones and twos is less than $t - 1$ then we replace asterisks in these rows by twos (adding if necessary new all-$*$ columns to B) until the sum of ones and twos will become greater than $t - 1$, i. e. t.

Thus, without loss of generality we can assume that the sum of ones and twos in any row of B is exactly t.

Consider a subcube defined by a row of B with exactly s twos and exactly r ones. Then lth level of Boolean cube B^p contains exactly $\binom{p-s-r}{l-r}$ vectors of this subcube if $l = r, \ldots, p - s$, and does not contain such vectors for another l.

Suppose that t is even (for odd t the reasoning is analogous). Then lth level of Boolean cube contains $\binom{p-t/2}{l}$ vectors from each subcube defined by the rows of B with exactly $t/2$ twos and exactly 0 ones, $\binom{p-t/2-1}{l-2}$ vectors from each subcube defined by the rows of B with exactly $t/2 - 1$ twos and exactly 2 ones and so on. Denote the number of rows of B with exactly i ones by c_i.

Then for any $l = 0, 1, \ldots, p$ the next inequality holds: $\sum_{i=0}^{t/2} c_{2i} \binom{p-t/2-i}{l-2i} \leq \binom{p}{l}$.
It follows $\sum_{i=0}^{t/2} c_{2i} \frac{(p-t/2-i)!}{(l-2i)!(p-t/2-l+i)!} \leq \frac{p!}{l!(p-l)!}$. Note that adding new all- columns to B we can obtain a proper (t, k, p, t)-matrix for any $p > p$. Thus, if there does not exist a proper (t, k, p, t)-matrix for any $p > p$ then there does not exist a proper (t, k, p, t)-matrix B. Therefore below we can suppose p as large as necessary. Then

$$\sum_{i=0}^{t/2} c_{2i} \frac{p^{t/2-i} + a_i^1 p^{t/2-i-1} + \ldots}{((\alpha p)^{t-2i} + a_i^2(\alpha p)^{t-2i-1} + \ldots)(((1-\alpha)p)^i + a_i^3((1-\alpha)p)^{i-1} + \ldots)} \leq$$

$$\frac{p^t + b^1 p^{t-1} + \ldots}{((\alpha p)^t + b^2(\alpha p)^{t-1} + \ldots)((p(1-\alpha))^{t/2} + b^3(p(1-\alpha))^{t/2-1} + \ldots)}$$

where $a_i^1, a_i^2, a_i^3, b^1, b^2, b^3$ — numbers that do not depend on p.

Next, we multiply both parts of this inequality by $p^{t/2}\alpha^t$ and transform the fractions. We have

$$\sum_{i=0}^{t/2} c_{2i} \left(\frac{\alpha^2}{1-\alpha}\right)^i (1 + a_i/p + O(1/p^2)) \leq$$

$$(1 + \max\{a_i\}/p + O(1/p^2)) \sum_{i=0}^{t/2} c_{2i} \left(\frac{\alpha^2}{1-\alpha}\right)^i \leq \frac{1}{(1-\alpha)^{t/2}}(1 + b/p + O(1/p^2))$$

where a_i, b do not depend on p. It follows

$$\sum_{i=0}^{t/2} c_{2i} \left(\frac{\alpha^2}{1-\alpha}\right)^i \leq \frac{1}{(1-\alpha)^{t/2}}(1 + b/p + O(1/p^2)).$$

Pointing in a view that we can take p as large as desired for fixed remained parameters, we have

$$\sum_{i=0}^{t/2} c_{2i} \left(\frac{\alpha^2}{1-\alpha}\right)^i \leq \frac{1}{(1-\alpha)^{t/2}}.$$

To find the sum of c_i we take $\alpha = \frac{\sqrt{5}-1}{2}$ (the root of the equation $\frac{\alpha^2}{1-\alpha} = 1$). This number is irrational but we can approach it by the sequence of rational numbers. As a result, we have:

$$\sum_{i=0}^{t/2} c_{2i} \leq \left(\frac{\sqrt{5}+1}{2}\right)^t.$$

Therefore, $k \leq \log_2 \sum_{i=0}^{t/2} c_{2i} \leq \log_2 \left(\frac{\sqrt{5}+1}{2}\right)^t$ and $\frac{t}{t+k} \geq \frac{1}{\log_2(\sqrt{5}+1)}$.

5 The Sequence of Proper Matrices with $\frac{t}{k+t} \to 0.5902\ldots$

In the previous Section we had demonstrated that for any proper (k_0, k, p, t)-matrix the inequality $\frac{t}{k+t} < \frac{1}{\log_2(\sqrt{5}+1)} = 0.5902...$ holds. Nevertheless, it appears that the ratio $\frac{t}{k+t}$ can approach the value $\frac{1}{\log_2(\sqrt{5}+1)} = 0.5902...$ with arbitrary precision. In this Section we construct proper (k_0, k, p, t)-matrices with ratio $\frac{t}{k+t}$ close to $\frac{1}{\log_2(\sqrt{5}+1)}$ and $k > \alpha k_0$ where $\alpha < \sqrt{5}\log_2\left(\frac{\sqrt{5}+1}{2}\right) = 1.5523\ldots$.

Lemma 4. *Suppose that* $\alpha < \sqrt{5}\log_2\left(\frac{\sqrt{5}+1}{2}\right)$. *Let*

$$k_0 = \left\lfloor \frac{t}{\alpha}\log_2\left(\frac{\sqrt{5}+1}{2}\right) + \frac{1}{\alpha}\log_2\left(\frac{\sqrt{5}+1}{2\sqrt{5}}\right) - 1 \right\rfloor .$$

Then $\binom{\lfloor\frac{t+k_0-1}{2}\rfloor}{k_0+1} \geq \binom{\lfloor\frac{t+k_0+1}{2}\rfloor}{k_0+3}(1+o(1))$ *and* $\binom{\lfloor\frac{t+k_0}{2}\rfloor}{k_0+2} \geq \binom{\lfloor\frac{t+k_0+2}{2}\rfloor}{k_0+4}(1+o(1))$.

Proof. We solve the inequality

$$\binom{\lfloor\frac{t+k_0-1}{2}\rfloor}{k_0+1} \geq \binom{\lfloor\frac{t+k_0+1}{2}\rfloor}{k_0+3} \tag{1}$$

(in the second case we have the same asymptotics). Using the factorial representation for binomial coefficients we solve the quadratic inequality for k_0 considering t as some parameter. As a result we obtain that the inequality (1) holds if

$$k_0 \geq \frac{1}{\sqrt{5}}t(1+o(1)). \tag{2}$$

But by the hypothesis of Lemma we have that k_0 is asymptotically $\frac{t}{\alpha}\log_2\left(\frac{\sqrt{5}+1}{2}\right)$ and $\alpha < \sqrt{5}\log_2\left(\frac{\sqrt{5}+1}{2}\right)$. It follows the same condition (2) on k_0 that completes the proof.

Theorem 3. *For any* α, $0 < \alpha < \sqrt{5}\log_2\left(\frac{\sqrt{5}+1}{2}\right) = 1.5523\ldots$, *and any* $\varepsilon > 0$ *there exists a proper* (k_0, k, p, t)*-matrix such that* $\frac{t}{t+k} < \frac{1}{\log_2(\sqrt{5}+1)} + \varepsilon$ *and* $k > \alpha k_0$.

Proof. If this Theorem holds for some α, $0 < \alpha < \sqrt{5}\log_2\left(\frac{\sqrt{5}+1}{2}\right)$, then, obviously, this Theorem holds for any α , $0 < \alpha < \alpha$. Therefore we can assume that $\alpha > \log_2\left(\frac{\sqrt{5}+1}{2}\right) = 0.6942\ldots$

At first, we construct recursively the sequence of matrices A_t, $t = 1, 2, \ldots$, that satisfy properties (i) and (ii) of proper matrices but the number of rows in these matrices is not necessary power of two. We denote by $s(t)$ the number of rows in the matrix A_t obtained after tth step.

At tth step we construct the matrix A_t such that the sum of ones and twos in any row of A_t does not exceed t and for any two different rows of A_t there exists a column such that one of these two rows has one in this column, and the second row has two in this column. We suppose that the matrices A_{t-1} and A_{t-2} were constructed at the previous steps. We suppose that the matrices A_{t-1} and A_{t-2} have the same number of columns (in opposite case we add to one of them the deficient number of all- columns). Next, we add to each of these matrices from the right side an additional column: the all-ones column to the matrix A_{t-1} and the all-twos column to the matrix A_{t-2}. Write the obtained matrices one over another. We say the resulting matrix is the matrix A_t, $A_t = \begin{pmatrix} A_{t-1} & \vec{1}^T \\ A_{t-2} & \vec{2}^T \end{pmatrix}$.

The matrix A_t is the matrix of desired form such that the sum of ones and twos in each row of A_t does not exceed t. The number of rows in A_t is equal to $s(t) = s(t-2) + s(t-1)$. Thus, $s(t)$ forms the Fibonacci sequence and $s(t)$ is asymptotically $\frac{1}{\sqrt{5}} \cdot \left(\frac{\sqrt{5}+1}{2}\right) \cdot \left(\frac{\sqrt{5}+1}{2}\right)^t$ if we take the matrices $A_1 = (1)$ and $A_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ as initial. In this construction the matrix A_t contains the rows with the number of ones greater than $k_0(t,\alpha) = \left(\frac{t}{\alpha}\log_2\left(\frac{\sqrt{5}+1}{2}\right) + \frac{1}{\alpha}\log_2\left(\frac{\sqrt{5}+1}{2\sqrt{5}}\right) - 1\right)$. Calculate the ratio of the number of rows that contain more than k_0 ones to the number of all rows in A_t (i. e. $s(t)$). Denote by $l_j(t)$ the number of rows with exactly j ones in the matrix A_t. By construction $l_0(t) = l_0(t-2)$, $l_j(t) = l_j(t-2) + l_{j-1}(t-1)$ for $j \geq 1$. These recursive relations follow the next direct formulas: $l_j(t) = \binom{\frac{t+j-2}{2}}{j} l_0(2) + \binom{\frac{t+j-4}{2}}{j-1} l_1(1) + a_2 l_2 + \ldots + a_j l_j$ if $(t+j)$ even and $l_j(t) = \binom{\frac{t+j-3}{2}}{j} l_0(1) + \binom{\frac{t+j-3}{2}}{j-1} l_1(2) + a_2 l_2 + \ldots + a_j l_j$ if $(t+j)$ odd where $a_2, \ldots, a_j$ — some numbers and arguments of $l_2, \ldots, l_j$ are 1 or 2 (it depends on the parity). For initial matrices A_1 and A_2 introduced above we have $l_0(1) = 0$, $l_0(2) = 1$, $l_1(1) = l_1(2) = 1$, $l_j(1) = l_j(2) = 0$ for $j \geq 2$. Therefore, $l_j(t) = \binom{\frac{t+j-2}{2}}{j} + \binom{\frac{t+j-4}{2}}{j-1}$ if $(t+j)$ even and $l_j(t) = \binom{\frac{t+j-3}{2}}{j-1}$ if $(t+j)$ odd.

It follows

$$\frac{\sum_{j=k_0(t,\alpha)+1}^{t} l_j(t)}{s(t)} \leq \frac{\sum_{j=k_0(t,\alpha)+1}^{t} \left(\binom{\frac{t+j-2}{2}}{j} \cdot 1 + \binom{\frac{t+j-3}{2}}{j-1} \cdot 1\right)}{\text{const} \cdot \left(\frac{\sqrt{5}+1}{2}\right)^t} \leq$$

(by Lemma 4 for $k_0(t,\alpha) = \left(\frac{t}{\alpha}\log_2\left(\frac{\sqrt{5}+1}{2}\right) + \frac{1}{\alpha}\log_2\left(\frac{\sqrt{5}+1}{2\sqrt{5}}\right) - 1\right)$)

$$\text{const}\,\frac{(t-k_0(t,\alpha))\left(\binom{\frac{t+k_0(t,\alpha)-1}{2}}{k_0(t,\alpha)+1}\right)}{\left(\frac{\sqrt{5}+1}{2}\right)^t} \leq$$

(denoting $v = \log_2\left(\frac{\sqrt{5}+1}{2}\right)$ and using the Stirling formula),

$$\leq \text{ const } \cdot t\,\frac{\sqrt{\frac{\frac{1}{2}t\left(1+\frac{v}{\alpha}\right)}{\frac{tv}{\alpha}\cdot\frac{1}{2}t\left(1-\frac{u}{\alpha}\right)}}\;\frac{\left(\frac{1}{2}t\left(1+\frac{v}{\alpha}\right)\right)^{\frac{1}{2}t\left(1+\frac{v}{\alpha}\right)}}{\left(\frac{tv}{\alpha}\right)^{\frac{tv}{\alpha}}\left(\frac{1}{2}t\left(1-\frac{v}{\alpha}\right)\right)^{\frac{1}{2}t\left(1-\frac{v}{\alpha}\right)}}}{\left(\frac{\sqrt{5}+1}{2}\right)^{t}} =$$

$$\text{const } \cdot \sqrt{t}\,\frac{\left(1+\frac{v}{\alpha}\right)^{\frac{1}{2}t\left(1+\frac{v}{\alpha}\right)}}{\left(\frac{2v}{\alpha}\right)^{\frac{tv}{\alpha}}\left(1-\frac{v}{\alpha}\right)^{\frac{1}{2}t\left(1-\frac{v}{\alpha}\right)}\left(\frac{\sqrt{5}+1}{2}\right)^{t}} =$$

$$\text{const } \cdot \sqrt{t}\left(\frac{\left(1+\frac{v}{\alpha}\right)^{\frac{1}{2}\left(1+\frac{v}{\alpha}\right)}}{\left(\frac{\sqrt{5}+1}{2}\right)^{1+\frac{1}{\alpha}}\left(\frac{v}{\alpha}\right)^{\frac{v}{\alpha}}\left(1-\frac{v}{\alpha}\right)^{\frac{1}{2}\left(1-\frac{v}{\alpha}\right)}}\right)^{t}.$$

It is easy to check that the expression in the parentheses increases monotonically on α for $\log_2\left(\frac{\sqrt{5}+1}{2}\right) = 0.6942\ldots < \alpha \leq \sqrt{5}\log_2\left(\frac{\sqrt{5}+1}{2}\right) = 1.5523\ldots$ and takes the value 1 for $\alpha = \sqrt{5}\log_2\left(\frac{\sqrt{5}+1}{2}\right)$. Therefore this expression takes values less than 1 for $\log_2\left(\frac{\sqrt{5}+1}{2}\right) < \alpha < \sqrt{5}\log_2\left(\frac{\sqrt{5}+1}{2}\right)$. It follows that $\frac{\sum_{j=k_0(t,\alpha)+1}^{t} l_j(t)}{s(t)} \xrightarrow{t\to\infty} 0$ for $\log_2\left(\frac{\sqrt{5}+1}{2}\right) < \alpha < \sqrt{5}\log_2\left(\frac{\sqrt{5}+1}{2}\right)$.

Thus, in the matrix A_t the number of rows that contain more than $k_0(t,\alpha)$ ones is asymptotically small with respect to the total number of rows. We eliminate from the matrix A_t all rows that contain more than $k_0(t,\alpha)$ ones. For sufficiently large t the number of such rows is smaller than $2^{k(t)}$ where $k(t) = \lfloor \log_2 s(t) \rfloor - 1$; therefore the obtained matrix will contain at least $2^{k(t)}$ rows. Now the matrix satisfies the property (iii) of a proper matrix (see Definition 1) for $k_0 = k_0(t,\alpha)$, $k = k(t)$. Next, we eliminate if necessary some rows more to obtain the matrix with exactly $2^{k(t)}$ rows. As a result, we have constructed the proper $(k_0(t,\alpha), k(t), p, t)$-matrix for some p. Thus, for the sequence of proper $(k_0(t,\alpha), k(t), p, t)$-matrices constructed above we have

$$\frac{t}{t+k(t)} = \frac{t}{t + \left\lfloor t\log_2\left(\frac{\sqrt{5}+1}{2}\right) + \log_2\left(\frac{\sqrt{5}+1}{2\sqrt{5}}\right)\right\rfloor - 1} \xrightarrow{t\to\infty} \frac{1}{\log_2(\sqrt{5}+1)}$$

and

$$\frac{k(t)}{k_0(t,\alpha)} = \frac{\left\lfloor t\log_2\left(\frac{\sqrt{5}+1}{2}\right) + \log_2\left(\frac{\sqrt{5}+1}{2\sqrt{5}}\right)\right\rfloor - 1}{\left\lfloor \frac{t}{\alpha}\log_2\left(\frac{\sqrt{5}+1}{2}\right) + \frac{1}{\alpha}\log_2\left(\frac{\sqrt{5}+1}{2\sqrt{5}}\right)\right\rfloor - 1} \xrightarrow{t\to\infty} \alpha,$$

moreover, if $\alpha > 1$ then $\frac{k(t)}{k_0(t,\alpha)} > \alpha$ for the infinite sequence of t.

The conclusion of the Theorem follows.

Remark. Note that in the construction in the proof of Theorem 3 in fact we have $p = 1$ for $t = 1$ and $p = t - 1$ for $t > 1$.

6 Constructions of New Record Highly Nonlinear Resilient Boolean Functions

In this Section by means of proper matrices constructed in the previous Section we construct m-resilient n-variable functions with maximum possible nonlinearity $2^{n-1} - 2^{m+1}$ for $m = \frac{1}{\log_2(\sqrt{5}+1)} n + O(\log_2 n) = 0.5902\ldots n + O(\log_2 n)$. Until now such functions with maximum possible nonlinearity $2^{n-1} - 2^{m+1}$ were known only for $m \geq 0.6n - 1$ [11] and some small set of concrete parameters n and m.

Lemma 5. *For any positive integer k there exists a proper $(1, k, 2^k + 1, 2^k + 1)$-matrix.*

Proof. We form the quadratic matrix B of order $2^k + 1$ writing in its rows all possible cyclic shifts of the row $(1\underbrace{22\ldots2}_{2^{k-1}}\underbrace{\ldots}_{2^{k-1}})$. It is easy to check that in this matrix for any two different rows there exists a column such that one of these two rows has one in this column, and the second row has two in this column. The sum of numbers in each row of B is exactly $2^k + 1$. Eliminating any row from B we obtain a proper $(1, k, 2^k + 1, 2^k + 1)$-matrix $B_{1,k,2^k+1,2^k+1}$.

Lemma 6. *For given positive integer k and infinite sequence of positive integer n there exist proper $S_{n,m,k,1}$-systems of Boolean functions for some m.*

Proof. By Lemma 1 there exists an $S_{2,-1,2,1}$-system of Boolean functions. Using Lemma 5 we apply

$$S_{2,-1,2,1}\,(T_{1,1,1,2})^h\,T_{1,k,2^k+1,2^k+1}.$$

By Theorem 1 this construction is valid if $2 + 3h \geq 2^k + 1$. Therefore for all h provided $h \geq \frac{2^k-1}{3}$ we construct $S_{2^k+k+3h+3,2^k+2h,k,1}$-system of Boolean functions.

Note that the constructions in Lemmas 5 and 6 are obviously nonoptimal from the practical point of view but more easy for the proof.

Theorem 4. *It is possible to construct m-resilient n-variable function with maximum possible nonlinearity $2^{n-1} - 2^{m+1}$ for $m = \frac{1}{\log_2(\sqrt{5}+1)} n + O(\log_2 n)$.*

Proof. We use proper $(k_0(t,\alpha), k(t), p, t)$-matrices constructed in the proof of Theorem 3. Note that by Remark after the proof of Theorem 3 we have $p = t-1$ for $t \geq 2$. We choose $1 < \alpha < \sqrt{5}\log_2\left(\frac{\sqrt{5}+1}{2}\right) = 1.5523\ldots$ and form the sequence t_0, t_1, $t_2, \ldots$ recursively. By Theorem 3 for given α beginning with sufficiently large t the matrices constructed in the proof of Theorem 3 are proper $(k_0(t,\alpha), k(t), p, t)$-matrices. We denote this sufficiently large t by t_0 (we can assume that $t_0 \geq 2$). Suppose that t_i and $k(t_i)$ are already defined positive integers. Then we define t_{i+1} as the maximal positive integer such that

$$k_0(t_{i+1},\alpha) = \left\lceil \frac{t_{i+1}}{\alpha}\log_2\left(\frac{\sqrt{5}+1}{2}\right) + \frac{1}{\alpha}\log_2\left(\frac{\sqrt{5}+1}{2\sqrt{5}}\right) - 1 \right\rceil = k(t_i). \qquad (3)$$

It is easy to see that $k_0(t,\alpha)$ is nondecreasing on t and $k_0(t+1,\alpha) - k_0(t,\alpha) \le 1$, therefore this definition of t_{i+1} is correct. Finally, we put

$$k(t_{i+1}) = \left\lceil t_{i+1}\log_2\left(\frac{\sqrt{5}+1}{2}\right) + \log_2\left(\frac{\sqrt{5}+1}{2\sqrt{5}}\right) - 1\right\rceil . \qquad (4)$$

The recursive definition is completed.

For defined t_0 by Lemma 6 we construct $S_{n_0,m_0,k(t_0),1}$-system of Boolean functions such that $n_0 \ge t_1 - 2$. After this we define recursively:

$$S_{n_i,m_i,k(t_i),1} T_{k(t_i),k(t_{i+1}),t_{i+1}-1,t_{i+1}} = S_{n_{i+1},m_{i+1},k(t_{i+1}),1}, \quad i = 0,1,2,\dots$$

Here $n_{i+1} = n_i + k(t_{i+1}) + t_{i+1}$, $m_{i+1} = m_i + t_{i+1}$.

By Theorem 1 this construction is valid if $n_i \ge 2p_{i+1} - t_{i+1} = t_{i+1} - 2$ for all i. We prove this statement by induction on i. We have $n_0 \ge t_1 - 2$ by construction. Next, suppose that $n_i \ge t_{i+1} - 2$. Then using (3) and (4) we have

$$n_{i+1} - t_{i+2} + 2 = n_i + k(t_{i+1}) + t_{i+1} - t_{i+2} + 2 \ge k(t_{i+1}) + 2t_{i+1} - t_{i+2} \ge$$
$$t_{i+1}\log_2\left(\frac{\sqrt{5}+1}{2}\right)(2-\alpha) + \sqrt{5}\left[\log_2\left(\frac{\sqrt{5}+1}{2}\right) - \log_2\left(\frac{\sqrt{5}+1}{2\sqrt{5}}\right) - 1\right] +$$
$$\frac{\log_2\left(\frac{\sqrt{5}+1}{2\sqrt{5}}\right)}{\log_2\left(\frac{\sqrt{5}+1}{2}\right)} \ge t_{i+1}\cdot 0.3107\ldots - 0.3123\ldots > 0$$

since $t_{i+1} \ge 2$. Thus, we use the Theorem 1 correctly. After q steps we have $n_q = n_0 + \sum_{i=1}^{q}(k(t_i)+t_i)$, $m_q = m_0 + \sum_{i=1}^{q} t_i$. From (4) we have $\frac{1}{\log_2\left(\frac{\sqrt{5}+1}{2}\right)}\left(k(t_i) - \log_2\left(\frac{\sqrt{5}+1}{2\sqrt{5}}\right)\right) < t_i \le \frac{1}{\log_2\left(\frac{\sqrt{5}+1}{2}\right)}\left(k(t_i) - \log_2\left(\frac{\sqrt{5}+1}{2\sqrt{5}}\right) + 1\right)$. It follows $\frac{m_q}{n_q} = \frac{m_0 + \sum_{i=1}^{q} t_i}{n_0 + \sum_{i=1}^{q}(k(t_i)+t_i)} = \frac{\frac{1}{\log_2\left(\frac{\sqrt{5}+1}{2}\right)}\sum_{i=1}^{q} k(t_i) + O(q)}{\left(1+\frac{1}{\log_2\left(\frac{\sqrt{5}+1}{2}\right)}\right)\sum_{i=1}^{q} k(t_i) + O(q)} = \frac{1}{\log_2(\sqrt{5}+1)} + O\left(\frac{q}{n_q}\right)$. It is easy to see that $q = O(\log_2 n_q)$. Therefore, $m_q = \frac{1}{\log_2(\sqrt{5}+1)} n_q + O(\log_2 n_q)$.

7 Constructions of Proper Matrices by Means of Cyclic Matrices

The construction of proper matrices in Section 5 gives the best limit value for the ratio $\frac{t}{t+k}$ but in general does not give the best possible matrices for concrete parameters. In this Section we discuss the method that probably gives the best possible in some sence concrete proper matrices.

We denote by $S(t)$ the maximum possible number of rows in matrices that satisfy properties (i) and (ii) of proper (t,k,p,t)-matrices but the number of rows in these matrices is not necessary power of two. By the proof of Theorem 2 we have $S(t) \le \left(\frac{\sqrt{5}+1}{2}\right)^t$. Below we show that $S(t) = \left\lfloor\left(\frac{\sqrt{5}+1}{2}\right)^t\right\rfloor$ at least for $1 \le t \le 10$. We search desired matrices for odd t in the class of matrices with $p = t$ that contain with each its row also all possible cyclic shifts of this row.

Theorem 5. $S(t) = \left\lfloor\left(\frac{\sqrt{5}+1}{2}\right)^t\right\rfloor$ *for* $1 \le t \le 10$.

Proof. For $t = 1, 3, 5, 7, 9$ we give the desired matrices M_t directly. Below we give in the matrices only one row from each class of cyclic shifts.

$$M_1 = \{1\},\ M_3 = \begin{pmatrix} 2 & 1 & \\ 1 & 1 & 1 \end{pmatrix},\ M_5 = \begin{pmatrix} 1 & 2 & 2 & & \\ & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

$$M_7 = \begin{pmatrix} 1 & 2 & 2 & 2 & & & \\ 1 & 2 & 1 & & 1 & 2 & \\ 1 & 1 & & & 1 & 2 & 2 \\ 2 & 1 & 1 & 1 & 1 & & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix},\ M_9 = \begin{pmatrix} 1 & 2 & 2 & 2 & 2 & & & & \\ 1 & 1 & & 2 & & 1 & 2 & 2 & \\ 1 & 2 & 1 & & & 1 & 2 & 2 & \\ 1 & 2 & 1 & 2 & & & 1 & & 2 \\ 1 & 2 & & 1 & 2 & & 1 & 2 & \\ 2 & & & 1 & 1 & 2 & 1 & 1 & 1 \\ 2 & 2 & & 1 & 1 & 1 & 1 & & 1 \\ & 2 & 1 & 2 & 1 & 1 & 1 & & 1 \\ 2 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

For $t = 2$ we put $M_2 = \begin{pmatrix} 2 & \\ 1 & 1 \end{pmatrix}$ (here we do not use cyclic shifts). Thus, $S(1) = 1$, $S(2) = 2$, $S(3) = 4$, $S(5) = 11$, $S(7) = 29$, $S(9) = 76$. If t is even, $t > 2$, then $\left\lfloor\left(\frac{\sqrt{5}+1}{2}\right)^t\right\rfloor = \left\lfloor\left(\frac{\sqrt{5}+1}{2}\right)^{t-1}\right\rfloor + \left\lfloor\left(\frac{\sqrt{5}+1}{2}\right)^{t-2}\right\rfloor$. Therefore if t is even, $t > 2$, and desired matrices M_{t-2} and M_{t-1} are constructed already then the matrix M_t can be constructed in the form

$$M_t = \begin{pmatrix} M_{t-1} & \vec{1}^T \\ M_{t-2}\vec{\ }^T & \vec{2}^T \end{pmatrix}.$$

Thus, $S(4) = 6$, $S(6) = 17$, $S(8) = 46$, $S(10) = 122$.

Hypothesis. $S(t) = \left\lfloor\left(\frac{\sqrt{5}+1}{2}\right)^t\right\rfloor$.

Note that if $k_0 < t$ then a proper (k_0, k, p, t)-matrix can be obtained from M_t by the cancelling all rows where the number of ones is greater than k_0 and some rows up to the nearest power of two.

Using the matrices M_9 and M_{10} as initial in the recursive construction of Theorem 3 we have constructed the 172-variable 102-resilient function with maximum possible nonlinearity as

$$S_{2,-1,2,1}T_{2,2,2,4}T_{2,4,7,8}T_{4,5,7,8}T_{5,6,9,9}T_{6,9,14,14}T_{9,10,15,15}$$
$$T_{10,11,16,16}T_{11,11,16,16}T_{11,9,13,13} = S_{172,102,9,1}.$$

These are the smallest parameters that we have found improving the bound in [11].

References

1. P. Camion, C. Carlet, P. Charpin, N. Sendrier, On correlation-immune functions, Advances in Cryptology: Crypto '91, Proceedings, Lecture Notes in Computer Science, V. 576, 1991, pp. 86–100.
2. R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, A. Sahai, Exposure-resilient functions and all-or-nothing transforms, In Advanced in Cryptology: Eurocrypt 2000, Proceedings, Lecture Notes in Computer Science, V. 1807, 2000, pp. 453–469.
3. B. Chor, O. Goldreich, J. Hastad, J. Friedman, S. Rudich, R. Smolensky, The bit extraction problem or t-resilient functions, IEEE Symposium on Foundations of Computer Science, V. 26, 1985, pp. 396–407.
4. C. Ding, G. Xiao, W. Shan, The stability theory of stream ciphers, Lecture Notes in Computer Science, V. 561, Springer-Verlag, 1991.
5. E. Pasalic, T. Johansson, Further results on the relation between nonlinearity and resiliency for Boolean functions, IMA Conference on Cryptography and Coding, Lecture Notes in Computer Science, Vol. 1746, 1999, pp. 35–44.
6. E. Pasalic, S. Maitra, T. Johansson, P. Sarkar, New constructions of resilient and correlation immune Boolean functions achieving upper bounds on nonlinearity, WCC2001 International Workshop on Coding and Cryptography, Paris, January 8–12, 2001, Electronic Notes in Discrete Mathematics, Volume 6, Elsevier Science, 2001.
7. P. Sarkar, S. Maitra, Construction of nonlinear Boolean functions with important cryptographic properties, In Advanced in Cryptology: Eurocrypt 2000, Lecture Notes in Computer Science, V. 1807, 2000, pp. 485–506.
8. P. Sarkar, S. Maitra, Nonlinearity bounds and constructions of resilient Boolean functions, In Advanced in Cryptology: Crypto 2000, Proceedings, Lecture Notes in Computer Science, V. 1880, 2000, pp. 515–532.
9. T. Siegenthaler, Correlation-immunity of nonlinear combining functions for cryptographic applications, IEEE Transactions on Information theory, V. IT-30, No 5, 1984, p. 776–780.
10. Yu. Tarannikov. On resilient Boolean functions with maximal possible nonlinearity, Proceedings of Indocrypt 2000, Lecture Notes in Computer Science, V. 1977, pp. 19-30, Springer-Verlag, 2000.
11. Yu. Tarannikov. New constructions of resilient Boolean functions with maximal nonlinearity, Preproceedings of 8th Fast Software Encryption Workshop, Yokohama, Japan, April 2-4, 2001, pp. 70-81, also available at Cryptology ePrint archive (http:// eprint.iacr.org/), Report 2000/069, December 2000, 11 pp.
12. Y. Zheng, X. M. Zhang, Improved upper bound on the nonlinearity of high order correlation immune functions, Selected Areas in Cryptography, 7th Annual International Workshop, SAC2000, Lecture Notes in Computer Science, V. 2012, pp. 264–274, Springer-Verlag, 2001.

A Twin Algorithm for Efficient Generation of Digital Signatures

D. Ramesh

New No 44/3, Old No 13/3 East Club Road,
Shenoy Nagar, Chennai-600 030, India,
Phone No +91 44 6450614
ramesh_panicker@yahoo.com

Abstract. The paper describes two algorithms for personnel identification, data authentication and digital signatures. Both are based on the intractability of finding square roots over finite fields and also can be an identity-based scheme. The outstanding feature of these two algorithms is its speed. Also a concept of interlocking equations is introduced which in effect acts like a one-way function.

Key words: Cryptography, digital signature, personnel identification, data authentication, interlocking equations.

1 Introduction

In an era where increasingly sensitive information is transmitted digitally and business transactions are done between people or firms located at far corners of the globe mostly using an insecure public medium, it has become imperative that enough security is guaranteed in such systems to boost the user confidence. The data authentication, digital signature and personnel identification schemes provide the much-needed security for these systems. These cryptographic features had its conception in a path breaking paper presented by Diffie and Hellman [1] in 1976. It is also to be noted that these added security measures should no way degrade the performance of such systems in terms of simplicity, response time, user friendliness, cost etc.

Most of the popular digital signature algorithms like RSA, DSA, Schnorr scheme etc use modular exponentiation with a modulus length of the order of 1000 bits. Using Brickell scheme [2] this will require on an average 250 modular multiplications. Also as time goes the scheme has to opt for longer keys for sustaining the security level, the required number of multiplications increase proportionately. This puts a constraint on the processing power of the system, which becomes a severe handicap in the case of smart cards, handheld secure devices, portable systems etc. Hence any cryptographic scheme which reduces the number of multiplications without doing any appreciable sacrifices on other fronts will always be welcome [3].

One more problem with the above schemes is that they are not identity based. This means an additional key certification phase as well as less flexibility for introducing user dependent features into the secure system.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 267–274, 2001.

The proposed algorithms address the problems plaguing the broad areas of personnel/entity identification, data/message authentication and digital signature generation. The main obstacle, which is seen in current schemes in these areas, is lack of speed. Also it is observed that if one scheme is superior in one specification it is left wanting in other specifications. The algorithms discussed below, apart from having speed, have a very few public keys; identity based and simpler to fabricate which is extremely handy in the case of smart card implementations for achieving cost effective systems.

2 Method 1

This method uses a set of unique interlocking equations as its backbone. The security derives from the interlocking feature of these equations as well as the intractability of finding the square roots of huge numbers over finite fields [4]. These equations taken independently resemble the scheme by Ong, Schnorr and Shamir [5].

2.1 Interlocking Equations

The two interlocking equations have three inputs I_1, I_2 and I_3 such that modulo square of one of them added/subtracted modulo to/from modulo square of each of the other two will give its outputs O_1 and O_2 respectively.

$$\begin{aligned} O_1 &= I_2^2 \pm I_1^2 \bmod n \\ O_2 &= I_3^2 \pm I_1^2 \bmod n \end{aligned}$$

It can be seen that given the inputs the outputs can be determined easily but the reverse process is not so. Even though each equation can be solved in polynomial time using Pollard method [6], when it comes to the second equation because of the interlocking nature, the problem boils down to finding square root over finite fields. In other words these set of equations exhibits a one-way property which can be used effectively for cryptographic purpose.

There are two modes of operation for issuing the private keys. These can be also considered as examples of identity based scheme [7].

2.2 Trusted Party Mode

In this mode the trusted party will issue the private keys. First it will generate a modulus n, which is a product of two big primes and then publishes n but keeps the prime factors secret. The modulus n will be common to all users in the group. Any user who wants to become a group member has to approach the trusted party with his/her/entity credentials like voter's identity card, passport, ration card, social security number, company registration details etc. If the trusted party is convinced about the user's identity, then it starts the process of issuing the private key. From the supplied credentials two identity strings $I_1(s)$ and $I_2(s)$ are generated based on a fixed and widely published format.

Now $I_1(s)$ and $I_2(s)$ are assigned to the outputs of a set of interlocking equations as shown below. The three private keys (r_1, r_2, r_3) are such that modulo square of one of them subtracted modulo from modulo square of the other two should get the identity strings $I_1(s)$ and $I_2(s)$ respectively.

$$I_1(s) = r_2^2 - r_1^2 \bmod n$$
$$I_2(s) = r_3^2 - r_1^2 \bmod n$$

For achieving this r_1 is replaced by a random number. Since in a finite field less than half of the numbers only will have square roots, first it is tested whether square root exists [8] for modulo sum of modulo square of r_1 with modulo squares of $I_1(s)$ and $I_2(s)$ respectively. Now r_1 is changed until square root exists for both conditions. The resultant square roots are the other two private keys r_2 and r_3.

It is to be noted that without the knowledge of the prime factors of n it is virtually impossible to find modulo square roots or in other words the private keys. The identity strings $I_1(s)$ and $I_2(s)$ are the two public keys of user A. The common public key for all the users is the common modulus n.

2.3 Independent Member Mode

In this mode each user will generate their private keys. So each user will create his own modulus n, which will be published as a public key apart from his identity strings as in the earlier mode. It may be noted that the trusted party is not fully dispensed with. Trusted party has a role to play in the form of authenticating user public keys where the user presents his credentials and public keys to the trusted party. If the trusted party is convinced about the credentials it issues its signatures for the user public keys. This is done to prevent somebody impersonating others. $I_1(s)$, $I_2(s)$ and n are the public keys of the user. There is no common modulus.

3 The Method of Operation for Different Applications

3.1 Personnel Identification

In this mode prover can prove his identity to the verifier or the verifier can verify prover's identity, either way. The protocol starts with the prover sending a prompt string to the verifier. For that the prover creates random numbers x, y and z using a cryptographically secure random generator [9]. Then they are assigned to the inputs of a set of interlocking equations. Its outputs p_1 and p_2 will be modulo difference of modulo square of x with modulo squares of y and z respectively.

$$p_1 = y^2 - x^2 \bmod n$$
$$p_2 = z^2 - x^2 \bmod n$$

Two more constituents of the prompt string p_3 and p_4 are obtained as follows.

$$\begin{aligned} p_3 &= 2(r_2 \;\; y - r_1 \;\; x) \bmod n \\ p_4 &= 2(r_3 \;\; z - r_1 \;\; x) \bmod n \end{aligned}$$

Prover sends p_1, p_2, p_3, p_4 along with his identity string to the verifier to initiate the process. In independent member mode the prover has to attach his public key signatures also so as to facilitate the verifier to authenticate the public keys.

After receiving the prompt string the verifier authenticates prover’s pubic keys using its signatures and the modulus of the trusted party which would have been widely published (only in the case of independent member mode) and then creates random numbers a and b and sends it to prover as a challenge string. When prover receives the challenge he creates his response or in other words his digital signature (p_5, p_6, p_7), which is unique in terms of his identity strings, verifier’s challenge string and the private key information, which he alone possesses.

$$\begin{aligned} p_5 &= r_1 \;\; a + x \;\; b \bmod n \\ p_6 &= r_2 \;\; a + y \;\; b \bmod n \\ p_7 &= r_3 \;\; a + z \;\; b \bmod n \end{aligned}$$

After getting the response from the prover the verifier imposes two conditions.

$$\begin{aligned} p_6^2 - p_5^2 &= I_1(s) \;\; a^2 + p_1 \;\; b^2 + p_3 \;\; a \;\; b \bmod n \\ p_7^2 - p_5^2 &= I_2(s) \;\; a^2 + p_2 \;\; b^2 + p_4 \;\; a \;\; b \bmod n \end{aligned}$$

This will verify whether prover possesses the private key information without leaking a single bit of information about it. This is an example of zero knowledge test. Proving of possession of private key information in turn proves the genuineness of the identity string he has presented. In other words the identity of the prover is proved beyond doubt.

It is important to mention that similar results can be obtained by changing the signs (i.e. + to −) in the equations p_1 to p_7 and then appropriately changing the signs in the test conditions such that the relations are valid.

Now all the hackers in the universe can lay their hands on p_1 to p_7 but will not be able to get any bit of information about the private key unless they know the prime factors of n. Since prime factors are not made public one has to factorize n, a huge number (of the order of 1000 bits) that may take millions of years with all the processing power one can get.

Before leaving this section a brief mention about the response time or processing time of the whole system. It can be seen that the prover module requires only six modular multiplications on line and only five modular multiplications off line. The verifier module requires only eight modular multiplications on line and only three modular multiplications off line. But the size of the signature (p_1 to p_7) is slightly large compared to other popular schemes.

3.2 Data Authentication

In this mode any message originating from prover can be authenticated as originated from prover himself and nobody else. Here the process is same as above till the prover receives the challenge. Then he hashes or in other words creates a message digest from the combination of challenge string, the actual message, the prompt string and the identity string of the prover and uses the resultant strings to create his signature (Inclusion of identity string for hashing is optional). This will be sent to the verifier. (In independent member mode the signatures of the public keys of the prover are also attached).

Verifier already in possession of all the strings creates the same message digest and then using the received signatures does the verification as mentioned earlier (In the case of independent member mode, signatures of prover public keys are verified first). If verified true then the verifier can be sure that the message has indeed originated from the prover as described by the identity string and further actions can be taken depending upon the content of the message.

3.3 Digital Signature Generation

In this mode the digital signatures is created as in the case of a data authenticator except that prompt string is not sent to the verifier and challenge string is absent. This means there is no communication between verifier and prover. Here the message digest is created from a combination of prompt string, prover's identity string and the message (Inclusion of identity string for hashing is optional). The resultant two strings are used by the prover to create the signatures. Now the signatures are attached to the message, prompt string and the identity string (In independent member mode the signatures of the public keys of the prover are also attached).

The verifier when supplied with the signed document/data retrieves the prompt string and message and creates the same message digest. (In the case of independent member mode the public key signatures supplied are also retrieved and then authenticated before proceeding further) The resultant strings are used to verify the signatures attached.

4 Method 2

This method is based on Pythagorean triplet [10]. The security is derived from the intractability of finding square root of huge numbers over finite fields. This can be also considered as identity based scheme.

There are two modes of issuing the private key as in the case of method I.

4.1 Trusted Party Mode

The procedure follows similar lines as in the case of method I till the generation of identity string. From the supplied credentials the identity string $I(s)$ is generated based on a fixed and widely published format. Since in a finite field less than half of the numbers only will have square roots a short random string of length a few bits long is concatenated with $I(s)$ which have to be varied until a square root is obtained.

Now $I(s)$ concatenated with the short random string is the public key (*Puk*) and reciprocal of the square root of *Puk* modulo n is the private key (*Prk*) of the user. It is to be noted that without the knowledge of the prime factors of n it is virtually impossible to find modulo square root or in other words the private key. The common public key for all the users is the common modulus n.

4.2 Independent Member Mode

In this mode each user will generate their private keys. So each user will have his modulus n, which will be published as a public key apart from his identity string as in the earlier mode. It may be noted that the trusted party is not fully dispensed with. Trusted party has a role to play in the form of authenticating user public keys where the user presents his credentials and public keys to the trusted party. If the trusted party is convinced about the credentials it issues its signatures for the user public keys. This is done to prevent somebody impersonating others. One noticeable difference from the earlier mode is the absence of the short random string. Here if $I(s)$ is not having a square root under the present modulus n, n is changed till the square root exists. Private key (*Prk*) is the reciprocal of square root of $I(s)$ modulo n. $I(s)$ (*Puk*) and n are the public keys of the user. There is no common modulus.

5 The Method of Operation for Different Applications

5.1 Personnel Identification

In this mode prover can prove his identity to the verifier or the verifier can verify prover's identity, either way. The noticeable difference is there is no need for a prompt string from the prover. In independent member mode the prover has to attach his public key signatures created by a trusted party also so as to facilitate the verifier to authenticate the public keys.

The verifier authenticates prover's pubic keys using its signatures and the modulus of the trusted party which would have been widely published (only in the case of independent member mode) and then creates a random number c and sends it to prover as a challenge. When prover receives the challenge he creates his response or in other words his digital signature, which is unique in terms of his identity string, verifier's challenge string c and the private key information he alone possesses. The two signatures, p_1 and p_2 are created as follows using diophantine solutions for the Pythagorean triplets [11].

$$\begin{aligned} v &= c \quad (2u)^{-1} && \bmod n \ (u \text{ is a random number}) \\ p_1 &= u + v && \bmod n \\ p_2 &= Prk(u^2 - v^2) && \bmod n \end{aligned}$$

After getting p_1 and p_2 from the prover the verifier imposes one condition, which will test whether prover possesses the private key information without any need to leak a single bit of information about it. This is an example of a zero knowledge test. The test is whether

$$Puk \quad p_2^2 \bmod n = p_1^2(p_1^2 - 2c) \bmod n$$

Proving of possession of private key information in turn proves the genuineness of the identity string he has presented. In other words the identity of the prover is proved beyond doubt.

The same result can be obtained by changing the signature equations.

$$p_1 = (u + v)Prk \bmod n$$
$$p_2 = (u^2 - v^2) \bmod n$$

The test equation becomes,

$$p_2^2 \bmod n = Puk \quad p_1^2(Puk \quad p_1^2 - 2c) \bmod n$$

The same results can be obtained by reversing the signs in the above equations as follows.

$$p_1 = (u - v) \qquad \bmod n$$
$$Puk \quad p_2^2 \bmod n = p_1^2(p_1^2 + 2c) \bmod n$$

Before leaving this section a brief mention about the response time or processing time of the whole system. It can be seen that the prover module requires only three modular multiplications on line and one modular inverse off line. The verifier module requires only four modular multiplications on line. The signature size (p_1 and p_2) is comparable to other popular schemes.

Here also the eavesdropper will not be able to get any bit of information about the private key unless he knows the prime factors of n. Since prime factors are not made public one has to factorize n, a huge number (of the order of 1000 bits) that may take millions of years with all the processing power one can get.

5.2 Data Authentication

In this mode any message originating from prover can be authenticated as originated from prover himself and nobody else. Here the process is same as above till the prover receives the challenge. Then he hashes or in other words creates a message digest from the combination of challenge string, the actual message and the identity string of the prover and uses the resultant string to create his signature (Inclusion of identity string for hashing is optional). This will be sent to the verifier (In independent member mode the signatures of the public keys of the prover is also attached).

Verifier already in possession of all the strings creates the same message digest and then using the received signatures does the verification as mentioned earlier (In the case of independent member mode signatures of prover public keys are verified first). If verified true then the verifier can be sure that the message has indeed originated from the prover as described by the identity string and further actions can be taken depending upon the content of the message.

5.3 Digital Signature Generation

In this mode the signatures for any message is created as in the case of a data authenticator except that challenge string is absent. This means there is no communication between verifier and prover. Here the message digest is created from a combination of the identity string and the message (Inclusion of identity string for hashing is optional). The resultant string is used by the prover to create the signatures. Now the signatures are attached to the message and the identity string (In independent member mode the signatures of the public keys of the prover are also attached).

The verifier when supplied with the signed document/data retrieves all the strings and message and creates the same message digest. (In the case of independent member mode the public key signatures supplied are also retrieved and then authenticated before proceeding further) The resultant string is used to verify the signatures attached.

6 Conclusion

Two algorithms for personnel identification, data authentication and digital signatures are discussed. It needs only a few multiplications for signing as well as verifying, as a result of doing away with modular exponentiation. Like majority of cryptographic algorithms, the security of these algorithms also relies on the absence of methods to find the factors of huge numbers in polynomial time.

References

[1] W.Diffie and M.Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. 22, No.6, Nov 1976.

[2] E.F.Brickell and K.S.McCurley, "Interactive identification and digital signatures," *AT&T Technical Journal*, Nov/Dec 1991.

[3] D.Ramesh, "Smart Cards – a smart way of achieving security," *Proceedings of Seminar on Embedded Systems for PC-Based Automation*, IEI, Chennai 2001.

[4] M.Rabin, "Digital signatures," in *Foundations of Secure Computation*. New York, NY: Academic Press, 1978.

[5] H.Ong, C.P.Schnorr and A.Shamir, "An efficient signature scheme based on quadratic equations," *Proceedings of 16^{th} ACM Symp. On Theory of Computing*, 1984.

[6] J.M.Pollard and C.P.Schnorr, "An efficient solution of the congruence $x^2 + ky^2 = m \pmod{n}$" IEEE Trans. Information Theory, vol. IT-33, no. 5 Sept 1987.

[7] A.Menezes, P. van Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Inc.

[8] L.M.Adleman, K.Manders, and G.L.Miller, "On taking roots in finite fields,"*Proc. 18^{th} Symp. on Foundation of Computer Science*, 1977.

[9] B.Schneier, *Applied Cryptography*, John Wiley & Sons.

[10] H.S.Zuckerman, *An Introduction to the Theory of Numbers*, 1960.

[11] W.J.LeVeque, *Topics in Number Theory*, 2 Vols., 1956.

Efficient "on the Fly" Signature Schemes Based on Integer Factoring

Takeshi Okamoto, Mitsuru Tada, and Atsuko Miyaji

School of Information Science,
Japan Advanced Institute of Science and Technology (JAIST),
Asahidai 1-1, Tatsunokuchi, Nomi, Ishikawa, 923-1292, Japan,
{kenchan,mt,miyaji}@jaist.ac.jp

Abstract. In 1999, Poupard and Stern proposed *on the fly signature scheme* (PS-scheme), which aims at minimizing the on-line computational work for a signer. In this paper, we propose more efficient on the fly signature schemes by improving the PS-scheme. In PS-scheme, the size of secret-key is fixed by modulus n, so that this feature leads to some drawbacks in terms of both the computational work and the communication load. The main idea of our schemes is to reduce the size of secret-key in PS-scheme by using a public element g which has a specific structure. Consequently, our schemes are improved with respect to the computational work (which means the computational cost for "pre-computation", "(on-line) signature generation" and "verification") and the data size such as a secret-key and a signature.

1 Introduction

As well-known, a *signature scheme* is an important tool for secure communication in an open network. Furthermore, a public-key infrastructure (PKI) actually requires compact signature schemes. *Compactness* on both computational work and data size, gives users' convenience, and is acceptable for various application to capacity limited devices such as a smart card.

Focus on the computational work in *a generic digital signature scheme*[1]. In such a signature scheme, there are two kinds of computation to generate a signature, that is, it consists of *pre-computation* and *(actual) signature generation*. To estimate the efficiency of a signature scheme, we should separately consider the computational cost for pre-computation and that for signature generation. The information generated at the pre-computation does not depend upon the message to be signed. Therefore the pre-computation can be executed in off-line, i.e. before a message to be signed is given. This means that such a computational cost does not influence the processing time after a message is given.

On the other hand, the computational cost in the signature generation step, does directly influence the processing time after being given a message. With

[1] As well as in [PS00], in this paper, *a generic (digital) signature scheme* means a signature scheme which can be derived from a three-pass identification scheme by using an appropriate hash function.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 275–286, 2001.

respect to a fast signature generation, Naccache et al. [NMVR94] proposed the efficient technique: a trusted authority computes the information in off-line, and treats those as *coupons*. In coupon based signature, the reduction of computation work in on line is the very target for fast signature. Consequently, we can say that it is a worthwhile work to make the computational cost small in signature scheme.

In 1992, Girault [Gir92] modified Schnorr's signature scheme [Sch91] in which an RSA-modulus[2] is used instead of a prime modulus. This modification leads to no modulo reduction in the signature generation. Therefore, in Girault's scheme, faster processing of the signature generation is possible than in Schnorr's one. In 1998, Poupard and Stern [PS98] investigated and gave provable security for Girault's scheme, and named that scheme GPS-scheme. In this paper, we call a generic signature scheme in which modulo reduction is not necessary at the (on-line) signature generation step, *on the fly signature scheme*.

In 1999, Poupard and Stern [PS99] proposed a generic signature scheme (PS-scheme), whose security relies on the difficulty of integer factoring. In this scheme, the size of the public-key is smaller than that in GPS-scheme. Consequently, compared with GPS-scheme, the computational cost and the data size can be decreased, and PS-scheme is seemed more secure under the *one-key attack* scenario [PS99]. However, PS-scheme has some drawbacks. For instance, the size of secret key is only dependent on modulus n, and considerably large (about $|n|/2$). This drawback leads to inefficient results in both communication work and data size. Moreover, computational cost in the verification is very high.

In this paper, we improve PS-scheme and propose new "on the fly" signature schemes (Scheme I and II) which is based on integer factoring. In our schemes, a public-key g has a specific structure. Consequently, in comparison with PS-scheme, the size of secret-key is small ($|n|/2$). In the following, our schemes realize a compactness of signature. Especially, the computation work in verification are much reduced by the changing n in $x = g^{y-ne} \bmod n$ (PS-scheme) into z in $x = g^{y-ze} \bmod n$ (our schemes).

As for Scheme I, a public-key n is RSA modulus, which is the same as that in PS-scheme. The performance in Scheme I is much superior to that in PS-scheme and the security is as secure as integer factoring problem for modulus n (in the random oracle model). To satisfy the security, Scheme I uses *asymmetric basis* g in $\mathbb{Z}_n$ which is a variant of [Po00],

As for Scheme II, a public-key n consists of three or more primes instead of RSA modulus in Scheme I (or PS-scheme). In [Sil99], we can see several trials to get faster computation for RSA cryptosystem [RSA78] by the technique of increasing the numbers of the factors of the modulus. Scheme II can make use of the very technique. The security is as secure as specially defined mathematical problem *finding order problem* (in the random oracle model), which is derived from integer factoring .

[2] In this paper, we call a modulus to be a product of two distinct primes *an RSA-modulus*.

Concrete to say, compared with PS-scheme, the size of a secret-key in Scheme I (resp. Scheme II) and a signature can be reduced by at least 69% and 47% (resp. 63% and 43%), respectively. Furthermore, Scheme I (resp. Scheme II) has an advantage that the computational cost can also be smaller. Compared with PS-scheme, the computational cost in Scheme I (resp. Scheme II) for pre-computation, signature generation and verification can be reduced by at least 38%, 69%, and 64% (resp. 54%, 63%, and 61%), respectively.

This paper is organized as follows. In Section ??, we will review PS-scheme and will discuss it. In Section 3, we will introduce our proposed signature scheme (Scheme I), will describe some features of ours, and will give provable security for ours. In Section 4, we will introduce an optimized scheme (Scheme II) and discuss in the same way as Section 3. In Section 5, we will discuss the security consideration with respect to (1)the size of n and (2)the number of prime factors with n in our schemes. In Section 6, we will evaluate the performance of our schemes by comparing with those of several existing schemes. The conclusion will be given in Section 7.

2 Previous Scheme

In this section, we review the signature scheme (PS-scheme) in [PS99]. This scheme is a generic signature scheme which is derived from the identification scheme. We first introduce some notation. The symbol $\phi(\cdot)$ denotes Euler totient function, that is, $\phi(n)$ is the number of the natural numbers less than n and coprime to n. The symbol $\lambda(\cdot)$ denotes so-called Carmichael function, that is, $\lambda(n)$ is the greatest number among the possible orders of elements in $\mathbb{Z}_n$. The order of an element g $\mathbb{Z}_n$ is represented as $\mathrm{Ord}_n(g)$.

2.1 Protocols

In PS-scheme, the following parameters exist: k and κ are *the security parameter* and *the information leak parameter*, respectively. The security parameter k is $|n|/2$, and the information leak parameter κ is assumed so that 2^κ-time computation is intractable. The parameters A and B satisfy $A < n$ and $|A| = \kappa + k + |B|$. Also B is assumed that B-time computation is intractable. We use an appropriate hash function $H : \{0, 1\} \quad \to \{0, 1\}^{|B|}$

Key generation step: The signer picks up two same-size primes p and q, and computes $n = pq$. After that, she picks up g $\mathbb{Z}_n$ satisfying $\mathrm{Ord}_n(g)$ $\{\lambda(n), \lambda(n)/2\}$ and computes $s = n - \phi(n)$ $(= p + q - 1)$. The secret-key is defined by s. The corresponding public-key is (n, g).

Signature generation step: Imagine that the signer generates a signature for a message m $\{0, 1\}$. The signer picks up a random number r $\mathbb{Z}_A$ to compute $x = g^r \mod n$, $e = H(x, m)$ and $y = r + se$. Note that y is the very value of $r + se$ on $\mathbb{Z}$. The signature for a message m is (e, y).

Verification step: Given the public-key of the signer (n, g), a message m and a signature (e, y), the verifier accepts the signature, if both $y < A$ and $e = H(g^{y-ne} \mod n, m)$ hold, and rejects it, otherwise.

2.2 Features and Drawbacks

A secret-key in PS-scheme is $s = n - \phi(n)$ which depends only upon (a part of) the public-key n. The two parameters n and s are congruent under the modulo $\phi(n)$, and the size of s is about a half of that of n.

Moreover, the computation of y is executed on $\mathbb{Z}$, and the information on a secret-key is protected by computing $r + se$ with condition $r \quad se$. Therefore, we can see that the size of r also depend upon that of se.

In the verification step, the size of y has to be explicitly verified whether the condition $y < A$ holds or not. This kind of verification cannot be seen in the existing signature schemes [ElG85,NIST91,RSA78,Sch91], hence we can say that such a verification indeed characterizes PS-scheme.

Unfortunately, PS-scheme has the following drawbacks.

High computational cost for verifier: In the verification step, $y \quad ne$ holds actually. And the order of $g \quad \mathbb{Z}_n$ is not open. Therefore, the computational cost for a verifier is considerably large as $|ne|$ increases. The verifier must compute full exponentiation ($|y - ne|$ bits) calculus such as $x = g^{y-ne} \bmod n$.

Inefficiency by the increase of a secret-key size: If the size of a secret-key s increase for the security reason, then this scheme shall get inefficient in view of (1) the computational cost for pre-computation, signature generation and verification, and (2) data size such as the size of signature.

Restriction for the structure of a public-key n: When we set up a public-key n to be the product of three or more primes, the size of a secret-key shall accordingly increase. For example, in case n is the product of three primes, that is, p, q and r, the secret-key s ($= n - \phi(n)$) turns out to be $n - (p-1)(q-1)(r-1)$ ($= pq + qr + rp - (p+q+r) + 1$), whose size is about 3/2 times of that in case n is the product of two primes.

3 Proposed Scheme

In this section, we introduce our signature scheme (Scheme I). The main idea of Scheme I is to reduce the size of secret-key by using element g which has a specific structure. Furthermore, we wish to construct that Scheme I has the same security of PS-scheme, so that the following basis is existed in Scheme I.

Definition 1 (Asymmetric basis) Let n be an RSA modulus such that $n = pq$. Then we say that g is an asymmetric basis in $\mathbb{Z}_n$ if the multiplicity of 2 in $\mathrm{Ord}_p(g)$ is not equal to the multiplicity of 2 in $\mathrm{Ord}_q(g)$. ■

We can say that this definition is more relaxed in comparison with that of [Po00].

3.1 Protocols

Scheme I has the parameters k, κ, a, b and c, where k is the security parameter, that is, the length of the secret-key, and κ is the information leak parameter,

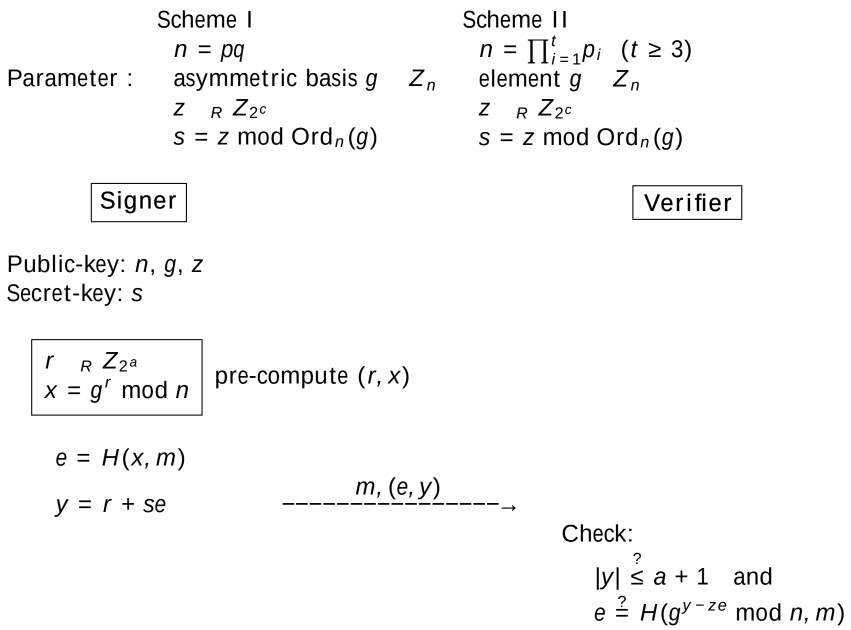

Scheme I
Scheme II
$n = pq$
$n = \prod_{i=1}^{t} p_i \quad (t \geq 3)$
Parameter :
asymmetric basis $g \quad Z_n$
element $g \quad Z_n$
$z \quad _R\ Z_{2^c}$
$z \quad _R\ Z_{2^c}$
$s = z \bmod \mathrm{Ord}_n(g)$
$s = z \bmod \mathrm{Ord}_n(g)$
Signer
Verifier
Public-key: n, g, z
Secret-key: s
$r \quad _R\ Z_{2^a}$
$x = g^r \bmod n$
pre-compute (r, x)
$e = H(x, m)$
$y = r + se$
$m, (e, y)$
Check:
$|y| \overset{?}{\leq} a + 1$ and
$e \overset{?}{=} H(g^{y-ze} \bmod n, m)$

Fig. 1. Proposed signature schemes

that is, 2^{κ}-time computation shall be intractable. Those parameters are assumed to satisfying $a \geq b + k + \kappa$ and $c \geq k + 2\kappa$. The detailed conditions on the parameters are mentioned in Section 3.2. We use an appropriate hash function $H : \{0, 1\} \quad \to \{0, 1\}^b$.

Key generation step: The signer picks up two same-size primes p and q, and computes $n = pq$. After that, she chooses an element $g \quad \mathbb{Z}_n$ which is an asymmetric basis in Z_n. She picks up a random number $z \quad \mathbb{Z}_{2^c}$ and computes $s = z \bmod q$, where $\mathrm{Ord}_n(g) = q$. The secret-key is s and the corresponding public-key is (n, g).

Signature generation step: Imagine that a signer having a public-key (n, g, z) and the corresponding secret-key s, generates a signature for a message $m \quad \{0, 1\}$. Then she picks up a random number $r \quad \mathbb{Z}_{2^a}$ to compute $x = g^r \bmod n$ and $e = H(x, m)$. She also computes $y = r + se$, where y is the very value of $r + se$ on $\mathbb{Z}$. The signature for a message m is (e, y).

Verification step: Given the public-key of the signer (n, g, z), a message m and a signature (e, y), the verifier accepts the signature, if both $|y| \leq a + 1$ and $e = H(g^{y-ze} \bmod n, m)$ hold, and rejects it, otherwise.

3.2 Parameter Generation

We describe remarks on the parameters for the security of Scheme I. In case of signature $y = r + se$, with $|r| = a$, $|s| = k$ and $|e| = b$, the values of a, b, k, κ shall satisfy $a \geq b + k + \kappa$ for its security.

If an adversary could figure out $r \quad \mathbb{Z}_q$ from x $(= g^r \bmod n)$ generated by the actual signer, then she could break the signature scheme. We can see the algorithms to extract r, such as *Pollard lambda method* in [Po78] and *the baby-step giant-step method* in [Knu98]. One may say that the former is better than the latter since it has same computational complexity (exponential-time: $O(\sqrt{q})$) but does not need memory. The size of q shall be set up not so that r can be figured out with such an algorithm.

The information leak parameter κ should be set up so that 2^κ-time computation should be intractable.

If $y > ze$ were allowed, then an adversary could impersonate the signer to easily compute y, along with the actual protocol, such that $x = g^{y-ze} \bmod n$ holds. To keep off such an attack, the condition of $c \geq k + 2\kappa$ shall be required from $c + b \geq a + \kappa \geq b + k + 2\kappa$. Furthermore, if $q > 2^c$ were satisfied, then $s = z$ would hold, that is, the secret-key would be disclosed. Hence also $q \leq 2^{c-\kappa}$ shall be required, and it is always held since $q < 2^k \leq 2^{c-2\kappa} \leq 2^{c-\kappa}$.

Next, we describe how to find p, q and an asymmetric basis g in $\mathbb{Z}_n$.

- Pick up two primes $p = 2p\,p \;\; + 1$ and $q = 2q\,q \;\; + 1$ such that p and q are also primes, and p and q are odd numbers.
- Choose $\alpha_p \quad \mathbb{Z}_p$ satisfying $g_p = \alpha_p^{(p-1)/p^{\square}} = 1 \bmod p$. In the same way, choose $\alpha_q \quad \mathbb{Z}_q$ satisfying $\alpha_q = q - 1 \bmod q$, $\alpha_q^{(q-1)/2} = 1 \bmod q$ and $g_q = \alpha_q^{(q-1)/2q^{\square}} = 1 \bmod q$.
- Compute $n = pq$ and $g = q(q^{-1} \bmod p)g_p + p(p^{-1} \bmod q)g_q \bmod n$.

In the last step, g is computed by using the technique of Chinese Reminder Theorem (CRT). Note that $\mathrm{Ord}_p(g) = p$ and $\mathrm{Ord}_q(g) = 2q$. Therefore $\mathrm{Ord}_n(g) = \mathrm{lcm}(p\ , 2q\ ) = 2p\ q$.

Finally, we discuss secure hash algorithm which we should adopt. If H were an *ideal* hash function, then the proposed signature scheme would be *secure* as described in Section 3.3. Since such a random function does not exist in the real world, in implementation, we are recommended SHA-1 by [NIST95] which is designed so that the algorithm can be a collision intractable hash function [Dam88].

3.3 Security Analysis

In this paper, we say that a signature scheme is *secure*, if no polynomial-time adversary A can existentially forge a signature under *the adaptive chosen message attack*. In this section, we show that Scheme I is secure, by using *the forking lemma* in [PS00], and showing protocol in signature generation step (see Section 3.1) can be simulated by a polynomial-time machine in *the random oracle model*

[BR93]. To discuss the provable security, we regard the signature for message m as (x, e, y).

As a strategy, we show that if there exists a polynomial-time adversary which can existentially forge a signature under the strongest attack, that is, an adaptive chosen-message attack, then we can construct a polynomial-time machine which can compute the integer factoring.

We say that a positive function $f(k) : \mathbb{N} \to \mathbb{R}$ is said to be *negligible*, if for any c, there exists a k_c such that $f(k) \le k^{-c}$ for any $k \ge k_c$. Otherwise f is said to be *non-negligible*.

Lemma 2 Let n be an RSA modulus and g be an asymmetric basis in $\mathbb{Z}_n$. Assume that we find $L > 0$ such that $g^L = 1 \bmod n$. Then we can construct a Turing machine M which on input n, g and L outputs a factor of n in time $O(|L||n|^2)$

Proof. (Sketch) This lemma is basically due to [Po00]. Hereafter, we describe how to construct M.

At first, M extract the odd part b of L, such that $L = 2^a b$. Since g is an asymmetric basis in $\mathbb{Z}_n$, it holds $g^{2b} = 1 \bmod p$ and $g^{2b} = 1 \bmod q$, and also holds $g^b = 1 \bmod p$ and $g^b = -1 \bmod q$. Then we have the following results: $p \mid g^b - 1$ and $n \nmid g^b - 1$. Consequently, M can find a factor of n by computing $\gcd(g^b - 1 \bmod n, n)$.

Note that modular exponentiation algorithm (resp. extended Euclidean algorithm) has a running time of $O(|L||n|^2)$ (resp. $O(|n|^2)$). Hence M can execute the above steps in time $O(|L||n|^2)$. □

Theorem 3 Let Q (resp. R) be the number of queries which a polynomial-time adversary A can ask to the random oracle (resp. the actual signer). Assume that $2^b q/2^a$ and $1/2^b$ are negligible. Also assume that, by executing adaptive chosen-message attack, A can forge a signature with non-negligible probability $\varepsilon \ge 10(R + 1)(R + q)/2^b$, and with the average running time T. Then we can construct a polynomial-time machine M which can factor n with non-negligible probability in expected time $O(QT/\varepsilon + |n|^{O(1)})$.

Proof. (Sketch) We firstly show that the signatures in the proposed scheme can be statistically simulated by a polynomial-time machine. This machine is simulated according to the protocol like in [PS00].

We denote, by $p(\alpha, \beta, \gamma)$ and $p\ (\alpha, \beta, \gamma)$, the probabilities that (α, β, γ) is output by the signature algorithm and the simulator, respectively. We set $\varphi = (2^b - 1)(2^k - 1)$, and let $R : \{0, 1\}\quad \to \{0, 1\}^b$ be an ideal hash function (random oracle) for a given message $m\quad \{0, 1\}\ $. For an integer A and a positive constant Δ, $N(R, A, \Delta)$ is defined to be the number of pairs $(e, y)\quad [0, 2^b) \times [A, A + \Delta)$ such that $R(g^{y-ze}, m) = e$. Tsavehen we have the following:

$$p(\alpha, \beta, \gamma) = \frac{\chi \begin{pmatrix} g^{\gamma - z\beta} \bmod n = \alpha, \\ R(\alpha, m) = \beta, \\ \gamma - s\beta \quad [0, 2^a) \end{pmatrix}}{2^a} \quad \text{and} \quad p\ (\alpha, \beta, \gamma) = \frac{\chi \begin{pmatrix} g^{\gamma - z\beta} \bmod n = \alpha, \\ R(\alpha, m) = \beta, \\ \gamma \quad [\varphi, 2^a) \end{pmatrix}}{N(R, \varphi, 2^a - \varphi)},$$

where for a predicate , $\chi(\ )$ is the characteristic function of , that is, $\chi(\ ) = 1$, if is true, and $\chi(\ ) = 0$, otherwise.

Therefore, the summation $\Sigma = \sum_{\alpha,\beta,\gamma} |p(\alpha,\beta,\gamma) - p\ (\alpha,\beta,\gamma)|$, has a upper bound of $8q(2^b - 1)/2^a$, because $\Sigma = 2(1 - N(R, \varphi, 2^a - \varphi)/2^a)$ holds similarly with [PS98], because $2^a - \Phi \leq N(R, \varphi, 2^a - \varphi)$ holds, and because $\varphi = (2^b - 1)(2^k - 1) \leq (2^b - 1)2q$ follows from $2^{k-1} \leq q < 2^k$. If $q/2^a$ is negligible, then so is $8q(2^b - 1)/2^a$, and consequently, the output by real signer and that by the simulator are statistically indistinguishable.

Next, by using the technique in [PS00], we can get a multiple of $\mathrm{Ord}_n(g)$ such that $g^L = 1 \bmod n$. Here g is an asymmetric basis in $\mathbb{Z}_n$, therefore by the result of Lemma 2 we can get a factor of n. □

4 Optimized Scheme

In this section, we give an optimized scheme (Scheme II) whish is superior to Scheme I in terms of computational work for a signer. The main feature in Scheme II is that the modulus n consists of three or more primes instead of using an RSA modulus in Scheme I. So a signer can make good use of the technique of CRT more efficiently. For example, in Scheme II with n having three prime factors, the computational cost for pre-computation $x\ (= g^r \bmod n)$ can be reduced to about 4/9 times of that in the Scheme I (or PS-scheme) with RSA modulus n. A preprint version of Scheme II can be seen in [OTM01]. In this paper, we consider further concrete security in Scheme II.

4.1 Protocols

Key generation step: The signer determines the number of factors, that is, $t \geq 3$, picks up same-size t primes p_i $(1 \leq i \leq t)$ and computes $n = \prod_{i=1}^{t} p_i$. After that, she chooses divisor q of $\lambda(n)$ and finds an order-q element $g\ \ \mathbb{Z}_n$. Also she picks up a random number $z\ \ \mathbb{Z}_{2^c}$ and compute $s = z \bmod q$. The secret-key is s and the corresponding public-key is (n, g).

The other steps are executed in the same way as Scheme I (see Section 3.1).

4.2 Description

The conditions of parameters such as k, κ, a, b and c are the same as those in Scheme I (see Section 3.2). Furthermore, primes p_i $(1 \leq i \leq t)$ and $g\ \ \mathbb{Z}_n$ will be generated under the line of work described in Section 3.2.

In [PS98,PS99] we can see the two types of attack: *one key attack*, an adversary try to forge valid signatures for fixed public key, and *possible key attack*, an adversary try to forge valid signatures for possible public keys, where possible public key means any public key satisfying the condition of the parameter. The security consideration under the one key attack scenario seems to be more strict analysis of security than that under the possible key attack scenario.

We have seen that the security in Scheme I is based on integer factoring. On the other hand, it is not unknown, under the one key attack scenario, whether Scheme II is as secure as the problem or not. To estimate more concrete security, we define the following problem.

Definition 4 (Finding order problem) This problem is as follows. Given $n \in \mathbb{N}_{>1}$ and $g \in \mathbb{Z}_n$, find L, where L is a multiple of $\mathrm{Ord}_n(g)$ and $|L|$ is bounded by a polynomial in $|n|$. ■

In Scheme II, if we assume the intractability of finding order problem, same result like Theorem 3 is obtained. Then the result (i.e. theorem) is proved, without loss of generality, using in the proof of the Theorem 3.

5 Integer Factoring Problem

In this section, we consider the secure size of n, and also discuss secure number of the prime factors for n in our schemes.

Of course, if the modulus n were factored, then the proposed signature schemes would be broken. In [LLMP90], we can see *the number field sieve method* for factorization, which is the most efficient algorithm ever proposed, and whose running time depends upon the size of n. On the other hand, in [Len87], we can see *the elliptic curve method*, which is also one of efficient algorithms for factorization, and whose running time depends upon the size of factors of n. Therefore, the faster one is determined according to the size of the input and upon the number of the factors of n.

As for Scheme II, referring to [Sil99] for computational cost of algorithms, in case that $|n| = 1024$ and that n has three prime factors, the number field sieve method is faster, whereas in case n has four prime factors, the other is faster. Hence supposing that $|n|$ is 1024 and t is 3 in the proposed scheme, and that $|n|$ is 1024 in PS-scheme, we can say that the number field sieve method is the faster (and fastest) algorithm to factor n in the respective schemes, and that the respective computational cost for factoring n can be almost the same.

6 Performance

In this section, we evaluate the efficiency of our schemes by comparing existing schemes. The parameters in the proposed Scheme I (resp. Scheme II) are set up to be $|n| = 1024$, $k = 160$ by taking $\kappa = 80$, $b = 80$ and $a = c = 320$ (resp. $|n| = 1024$, $t = 3$, $k = 192$ by taking $\kappa = 80$, $b = 80$ and $a = c = 352$).

Table 1 gives the performance of various signature schemes including ours.

Here, a primitive arithmetic of binary methods [Knu81] is used. For all schemes in the table, we set up the parameter under the line of the one key attack scenario. Hence the size of secret-key in GPS-scheme is 1024 bits. For more discussion on it, we refer to [Po00].

UMP means the underlying mathematical problem that the signature scheme relies on for its security. The terms CPC, CSG and CVF mean the computational

Table 1. Performance of signature schemes

Scheme	UMP	CPC (×M)	CSG	CVF (×M)	SPK (bits)	SSK (bits)	SSig (bits)
Scheme I $\|n\| = 1024$, $a = 320$, $\kappa = 80$	Integer factoring	240	80 × 160	600	2048	160	400
Scheme II $\|n\| = 1024$, $a = 352$ $t = 3$, $\kappa = 80$	Finding order	176	80 × 192	648	2048	192	432
PS-scheme [PS99] $\|n\| = 1024$, $\|A\| = 672$	Integer factoring	384	80 × 512	1656	1024	513	752
GPS-scheme [PS98] $\|n\| = 1024$	Discrete log. modulo n	384	80 × 1024	1796	3072	1024	1264

cost for pre-computation, signature generation and verification, respectively. The terms SPK, SSK and SSig means the size of a public-key, a secret-key and a signature, respectively.

In CPC, the signer uses the technique of CRT if it is possible. In SPK with our schemes, the size of public-key is optimized: we regard actual public-key as (n, g), and z is computed by $z = H(n, g)$, where H is a hash function $H : \{0, 1\} \rightarrow \{0, 1\}^c$.

For respective computational cost, the unit M represents the computational cost for one multiplication under a 1024-bit modulus, $\alpha \times \beta$ represents the computational cost for multiplication of an α-bit number and a β-bit number on $\mathbb{Z}$.

Since PS-scheme is intended to be used with a modulus product of two *strong* primes, $g = 2$ is a correct basis and do not have to be included in the public key. Consequently, we set SPC = 1024 for PS-scheme. Therefore, one may say that PS-scheme is more efficient than our schemes in terms of size of public key.

We can say that the proposed signature scheme is quite efficient one in view of both the computational cost and the data size. Concrete to say, Scheme I (resp. Scheme II) enables the computational cost to be reduced by 38% (resp. 54%) for pre-computation, by 69% (resp. 63%) for signature generation, and by 64% (resp. 61%) for verification, comparing with PS-scheme. For the data size, the secret-key size in ours is 69% (resp. 63%) of that in PS-scheme, and the signature size is 47% (resp. 43%) of that in PS-scheme.

By Table 1, we can say that the proposed signature scheme is efficient, and requires a relatively weak computational assumption for its security.

7 Conclusion

In this paper, we have proposed efficient signature schemes, which are derived from a three-pass identification scheme, and which are constructed by improving PS-scheme in terms of a compactness of signature. As well as PS-scheme (or GPS-scheme), the proposed schemes are so-called “on the fly” signature schemes, that is, it does not require modulo reduction in the signature generation step. We have shown that our schemes are existentially unforgeable against any polynomial-time adversaries that can execute adaptive chosen message attack in the random oracle model. Furthermore, the underlying computational problem in ours is the integer factoring problem in Scheme I and mathematically well defined problem (i.e. finding order problem) in Scheme II, respectively. We also have shown that ours are more efficient than PS-scheme in view of the computational cost and also in view of the size of a secret-key and a signature.

References

[BR93] M. Bellare and P. Rogaway: “*Random oracles are practical: a paradigm for designing efficient protocols*”, Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS), 1993.

[Dam88] I. Damgård: “*Collision free hash functions and public key signature schemes*”, Advances in cryptology - Eurocrypt’87, Lecture Notes in Computer Science 304, Springer-Verlag, pp.203-216, 1988.

[ElG85] T. ElGamal: “*A public-key cryptosystem and a signature scheme based on discrete logarithms*”, IEEE transactions of information theory, vol.IT-31, no.4, pp.469-472, 1985.

[FFS88] U. Feige, A. Fiat and A. Shamir: “*Zero-knowledge proofs of identity*”, Journal of cryptology, vol.1, pp.77-95, 1988.

[FMO92] A. Fujisaki, S. Miyaguchi and T. Okamoto: “*ESIGN: an efficient digital signature implementation for smart cards*”, Advances in cryptology - Eurocrypt’91, Lecture Notes in Computer Science 547, Springer-Verlag, pp.446-457, 1992.

[Gir91] M. Girault: “*An identity-based identification scheme based on discrete logarithms modulo a composite number*”, Advances in cryptology - Eurocrypt’90, Lecture Notes in Computer Science 473, Springer-Verlag, pp.481-486, 1991.

[Gir92] M. Girault: “*Self-certified public keys*”, Advances in cryptology - Eurocrypt’91, Lecture Notes in Computer Science 547, Springer-Verlag, pp.490-497, 1992.

[Knu81] D. E. Knuth: “*Seminumerical Algorithms*”, The art of computer programming, vol.2, Second edition, Addison-Wesley, 1981.

[Knu98] D. E. Knuth: “*Sorting and Searching*”, The art of computer programming, vol.3, Second edition, Addison-Wesley, 1998.

[LLMP90] A. K. Lenstra, H. W. Lenstra Jr., M. S. Manasse and J. M. Pollard: “*The number field sieve*”, Proceedings of ACM Annual Symposium on Theory of Computing, pp.564-572, 1990.

[Len87] H. W. Lenstra Jr.: “*Factoring integers with elliptic curves*”, Annuals of Mathematics, vol.126, pp.649-673, 1987.

[NMVR94] D. Naccache, D. M’raihi, S. Vaudenay and D. Raphaeli : “*Can DSA be improved ?*”, Advances in cryptology - Eurocrypt’94, Lecture Notes in Computer Science 950, 1995.

[NIST91] National Institute of Standards and Technology (NIST): "*Digital signature standards (DSS)*", Federal Information Processing Standards, 1991.

[NIST95] National Institute of Standards and Technology (NIST): "*Secure hash standards (SHS)*", Federal Information Processing Standards, 1995.

[OTM01] T. Okamoto, M. Tada and A. Miyaji: "*Proposal of Efficient Signature Schemes Based on Factoring*" (in Japanese), Transactions of Information Processing Society of Japan, vol. 42, no. 8, pp.2123-2133, 2001.

[Po00] D. Pointcheval: "*The Composite Discrete Logarithm and Secure Authentication*", Advances in cryptology - PKC'00, Lecture Notes in Computer Science 1751, 2000.

[Po78] J. Pollard: "*Monte Carlo methods for index computation mod p*", Mathematics of Computation, vol 32, pp.918-924, 1978.

[PS96] D. Pointcheval and J. Stern: "*Security proofs for signature schemes*", Advances in cryptology - Eurocrypt'96, Lecture Notes in Computer Science 1070, 1996.

[PS00] D. Pointcheval and J. Stern: "*Security arguments for digital signatures and blind signatures*", Journal of cryptology, vol.13, no.3, Springer-Verlag, pp.361-396, 2000.

[PS98] G. Poupard and J. Stern: "*Security analysis of a practical 'on the fly' authentication and signature generation*", Advances in cryptology - Eurocrypt'98, Lecture Notes in Computer Science 1403, Springer-Verlag, pp.422-436, 1998.

[PS99] G. Poupard and J. Stern: "*On the fly signatures based on factoring*", Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS), pp.48-57, 1999.

[Riv92] R. L. Rivest: "*The MD5 message-digest algorithm*", Internet Request for Comments, RFC 1321, 1992.

[RSA78] R. L. Rivest, A. Shamir and L. M. Adleman: "*A method for obtaining digital signatures and public-key cryptosystems*", Communications of the ACM, vol.21, no.2, pp.120-126, 1978.

[Sch91] C. P. Schnorr: "*Efficient signature generation by smart cards*", Journal of cryptology, vol.4, Springer-Verlag, pp.161-174, 1991.

[Sil99] R. D. Silverman: "*A cost-based security analysis of symmetric and asymmetric key length*", RSA Laboratories, CryptoBytes, Bulletins, no.13, 1999. Available from: `http://www.rsasecurity.com/rsalabs/bulletins/bulletin13.html`.

[SZ00] R. Steinfeid and Y. Zheng: "*A Signencryption Scheme Based on Integer Factorization*", Advances in cryptology - ISW'00, Lecture Notes in Computer Science 1975, 2000.

Clock-Controlled Shift Registers and Generalized Geffe Key-Stream Generator

Alexander Kholosha

Department of Mathematics and Computer Science,
Technische Universiteit Eindhoven,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands,
A.Kholosha@tue.nl

Abstract. In this paper we devise a generalization of the Geffe generator that combines more than three periodic inputs over GF(q). In particular, clock-controlled shift registers are suggested as inputs. The period and the linear complexity of the generated key-stream are estimated. We also prove some new results about the period of the sequence generated by a clock-controlled shift register.

Keywords: cryptography, key-stream generator, clock-controlled shift register, Geffe generator.

1 Introduction

The basic building block that we want to use for constructing a key-stream generator, consists of a control register CR and a clock-controlled generating register GR. A control register generates a sequence of nonnegative integers $a = \{a_i\}_{i\geq 0}$ and cycles periodically with period π. Hereafter in this paper by period we mean least period of a sequence, as opposed to multiple period. A generating register is an LFSR over $P = \mathrm{GF}(q)$ with *irreducible* feedback polynomial $f(x)$ of degree $m > 1$ and order M. Let $b = \{b(i)\}_{i\geq 0}$ denote the output sequence from the GR when clocked regularly and let α be a root of $f(x)$ in the splitting field of $f(x)$. In some cases, further in this paper, primitiveness of $f(x)$ will be required. Then $\lambda = q^m - 1$ will denote the maximal possible order of $f(x)$. Let also S denote $\sum_{k=0}^{\pi-1} a_k$.

In the clock-controlled mode, the output sequence $u = \{u(t)\}_{t\geq 0}$ is generated in the following way (see Fig. 1). The initial output is $u(0) = b(a_0)$. Further, after output $u(t-1)$ has been generated, the CR specifies the nonnegative integer a_t, the GR is shifted a_t times and then produces the next output $u(t)$. After that, the CR is shifted once to be ready for the next iteration. Thus, the general form of an output sequence element is

$$u(t) = b\left(\sum_{i=0}^{t} a_i\right) \quad \text{for} \quad t \geq 0 \ . \tag{1}$$

In the sequel, by irregular clocking will we mean the above type of clock control applied to the GR.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 287–296, 2001.

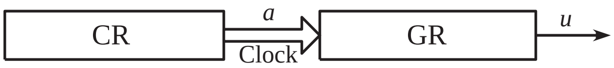


Fig. 1. Clock-controlled arrangement

Section 2 of the paper starts with some results about uniform decimation of linear recurring sequences in the field $P = \mathrm{GF}(q)$. These results are used to estimate the period of a sequence generated by a clock-controlled LFSR. We derive also some new conditions for sequences, obtained by uniform decimation, to reach their maximum linear complexity. Further, we estimate the period of the output sequence generated by an arbitrary clock-controlled LFSR with an irreducible feedback polynomial and an arbitrary structure of the control sequence. A sufficient condition for this period to reach its maximal value is formulated. Some specific configurations of clock-controlled arrangements with a maximal period of the output sequence are defined. Relevant recommendations for estimating the linear complexity are also presented.

In Sect. 3 we construct a key-stream generator based on the one suggested by Geffe in [1]. Unlike the Geffe generator that has three binary input m-sequences, our generator runs over the field $P = \mathrm{GF}(q)$ and combines multiple inputs having arbitrary periods. In particular, this implies that clock-controlled shift registers can be used as inputs. The original Geffe generator can not be used for key-stream generation since its combining function is zero-order correlation immune and correlation attacks are applied easily. Using clock-controlled registers and multiple inputs makes this generator immune against fast correlation attacks and less susceptible to basic attacks. We analyze some relevant algebraic properties of the suggested generator.

2 Period and Linear Complexity of Clock-Controlled LFSR's

First, we need some results about sequences obtained by uniform decimation of linear recurring sequences with irreducible characteristic polynomial. These results will be used further to estimate the period of a sequence generated by a clock-controlled LFSR.

Definition 1. *Let l and k be arbitrary nonnegative integers and $k > 0$. Then sequence $v = \{v(i)\}_{i\geq 0}$ defined by $v(i) = u(l + ki)$ for $i \geq 0$ is called the uniform (l, k)-decimation of sequence $u = \{u(i)\}_{i\geq 0}$. Also we will say that v is obtained by uniform (l, k)-decimation of u.*

Let $f(x)$ be an irreducible polynomial of degree $m > 0$ and order M over $P = \mathrm{GF}(q)$. Further, taking into account the fact that $Q = \mathrm{GF}(q^m)$ is the splitting field of $f(x)$, let α be a root of $f(x)$ in an extension field $Q = \mathrm{GF}(q^m)$ of P. Let $m(k)$ denote the degree of $R_k = P(\alpha^k)$ over P. Let also $f_k(x)$ denote the minimal polynomial of α^k over P. Note that $f_k(x)$ is irreducible in $P[x]$. Then directly from the definition of extension degree it follows that $\deg f_k(x) = m(k)$ and evidently $m(k) \mid m = m(1)$.

We denote the set of all homogeneous linear recurring sequences in P with characteristic polynomial $f(x)$ by $L_P(f)$. If degree of $f(x)$ is m then $L_P(f)$ is an m-dimensional vector space over P. Item (a) of the following theorem is a particular case of [2, Proposition]. Item (b) is an easy generalization of [3, Lemma 17].

Theorem 1. *Under the conditions imposed above, let l and k be arbitrary nonnegative integers and $k > 0$, then:*

(a) The uniform (l, k)-decimation defines a homomorphism of the vector space $L_P(f)$ onto $L_P(f_k)$. This homomorphism is an isomorphism if and only if $m(k) = m$.

(b) If $f(x)$ is a primitive polynomial and if u is a nonzero sequence belonging to $L_P(f)$ then every nonzero sequence w $L_P(f_k)$ can be obtained as a uniform (l, k)-decimation of u using exactly $q^{m-m(k)}$ different values of l $\{0, \ldots, \lambda - 1\}$, and the zero sequence can be obtained similarly using exactly $q^{m-m(k)} - 1$ different values of l $\{0, \ldots, \lambda - 1\}$.

Note 1. Polynomial $f_k(x)$ is the minimal polynomial of α^k, so it is irreducible. Since the order of α^k (that is equal to the order of $f_k(x)$) is given by $\frac{\operatorname{ord}\alpha}{\gcd(k, \operatorname{ord}\alpha)} = \frac{M}{\gcd(k,M)}$, we conclude that $f_k(x)$ has order M if and only if k is relatively prime to M. Further, if $\gcd(k, M) = 1$ then $f_k(x)$ has degree m. Indeed, the degree of $f_k(x)$ is equal to the least value of t, $t > 0$, for which $(\alpha^k)^{q^t} = \alpha^k$ or equivalently $\alpha^{k(q^t-1)} = 1$. But $\operatorname{ord}\alpha = M$ and $\gcd(k, M) = 1$. It follows that $M \mid q^t - 1$ and thus that $t = m$.

Corollary 1. *Let $\gcd(k, M) = 1$. Then every uniform (l, k)-decimation sequence of any nonzero sequence u $L_P(f)$ is equal to a nonzero sequence belonging to $L_P(f_k)$ and none nonzero sequence w $L_P(f_k)$ can be obtained as a uniform (l, k)-decimation of u using more than one value of l $\{0, \ldots, M - 1\}$.*

Proof. When applying the uniform decimation with parameters $l \geq 0$ and $k > 0$ to sequences in $L_P(f)$ we can assume that $l < M$ since all these sequences have the multiple period M. Moreover, if we fix some arbitrary value of $0 \leq \tilde{l} < M$ then for any $l > 0$, the uniform (l, k)-decimation of any nonzero sequence from $L_P(f)$ is equal to the uniform $(\tilde{l}, k)$-decimation of some other nonzero sequence from $L_P(f)$. Thus, for any fixed value of $\tilde{l}$, $0 \leq \tilde{l} < M$, the set containing uniform (l, k)-decimation sequences of any nonzero sequence u $L_P(f)$, when $k > 0$ is fixed and l takes all possible nonnegative values, is equal to the set containing uniform $(\tilde{l}, k)$-decimation sequences of some M-cardinal subset of nonzero sequences in $L_P(f)$. Now since $m = m(k)$, the statement easily follows from Item (a) of Theorem 1.

Corollary 2. *If the degree m of polynomial $f(x)$ is a prime number then $m(k) = m$ if and only if k is not a multiple of $\frac{M}{\gcd(M,q-1)}$. Moreover, if $\frac{M}{\gcd(M,q-1)} \nmid k$ then* *every uniform (l, k)-decimation sequence of any nonzero sequence $u \in L_P(f)$ is equal to a nonzero sequence belonging to $L_P(f_k)$ and none nonzero sequence $w \in L_P(f_k)$ can be obtained as a uniform (l, k)-decimation of u using more than one value of $l \in \{0, \ldots, M-1\}$.*

Proof. Since $m(k) \mid m$ and m is prime, only two alternatives are possible: either $m(k) = m$ or $m(k) = 1$, in which case $(\alpha^k)^q = \alpha^k$. So, $m(k) = 1$ if and only if M divides $k(q-1)$, i.e. $\frac{M}{\gcd(M, q-1)} \mid k$. The rest of the proof goes the same way as in Corollary 1.

Corollary 3. *If $f(x)$ is a primitive polynomial and $k \leq q^{m/2}$ then* $\deg f_k(x) = m$*. Moreover, under these conditions, every uniform (l, k)-decimation sequence of any nonzero sequence $u \in L_P(f)$ is equal to a nonzero sequence belonging to $L_P(f_k)$ and every nonzero sequence $w \in L_P(f_k)$ can be obtained as a uniform (l, k)-decimation of u using a unique value of $l \in \{0, \ldots, \lambda-1\}$.*

Proof. By virtue of Theorem 1, Item (a), all uniform (l, k)-decimation sequences of u belong to $L_P(f_k)$ and we have to prove that $m(k) = m$.

By definition, $\operatorname{ord} \alpha^k = \frac{\lambda}{\gcd(k, \lambda)} \mid (q^{m(k)} - 1)$ and $m(k) \mid m$, as was noted before. Hence, if $m(k) < m$ then $m(k) \leq \frac{m}{2}$ and therefore $\frac{\lambda}{\gcd(k, \lambda)} \leq q^{m/2} - 1$, i.e. $\gcd(k, \lambda) \geq q^{m/2} + 1$. In particular, $k \geq q^{m/2} + 1$ that contradicts the condition imposed.

Therefore, $m(k) = m$ and by Theorem 1, Item (b), the zero sequence can be obtained as a uniform (l, k)-decimation of u using exactly $q^{m-m(k)} - 1 = 0$ different values of $l \in \{0, \ldots, \lambda-1\}$. So, all uniform (l, k)-decimation sequences of u are nonzero. Every nonzero linear recurring sequence $w \in L_P(f_k)$ can be obtained as a uniform (l, k)-decimation of u using exactly $q^{m-m(k)} = 1$ value of $l \in \{0, \ldots, \lambda-1\}$.

Further in this section, we continue to use the terminology and notations introduced in Sect. 1. As a generalization of Definition 1 of a uniform decimation, we can consider the output sequence u, obtained from (1) as a *nonuniform* decimation of b according to the control sequence a as follows:

$$u(i + j\pi) = b(\sigma(i) + jS) \quad \text{for} \quad 0 \leq i < \pi,\ j \geq 0\ , \tag{2}$$

where $S = \sum_{k=0}^{\pi-1} a_k$ and $\sigma(i) = \sum_{k=0}^{i} a_k$. Hence, any uniform (i, π)-decimation of u is a uniform $(\sigma(i), S)$-decimation of b. By Theorem 1, Item (a), the latter decimation belongs to $L_P(f_S(x))$. The output sequence u consists of π such sequences interleaved and belongs to $L_P(f_S(x^\pi))$.

Since the period of the sequence b divides the order M of $f(x)$, we conclude that all elements of a can be reduced modulo M without any effect on the output sequence u. So, from now on we assume without loss of generality that all elements of a are nonnegative integers less than M.

It is obvious that the minimum of the degrees of irreducible factors of $f_S(x^\pi)$ provides a lower bound for the linear complexity of the output sequence u and the lowest possible order of any irreducible factor of $f_S(x^\pi)$ gives a lower bound for the period of u.

Since $\operatorname{ord} f_S(x) = \operatorname{ord} \alpha^S = \frac{M}{\gcd(S,M)}$ and u consists of π interleaved sequences belonging to $L_P(f_S(x))$, then from (2) it easily follows that the period of u divides $\frac{\pi M}{\gcd(S,M)}$. From [4, Lemma 1] it follows that if u is nonzero then its period is a multiple of $\frac{\pi' M}{\gcd(S,M)}$ where π' is the product of all prime factors of π, not necessarily distinct, which are also factors of $\frac{M}{\gcd(S,M)}$. This provides the lower bound for the period. In particular, if every prime factor of π also divides $\frac{M}{\gcd(S,M)}$ then the period of u reaches the maximal value $\frac{\pi M}{\gcd(S,M)}$. We also note that zero output sequences can be generated even if the initial state of the GR is nonzero and $f(x)$ is primitive.

By Note 1, if S is relatively prime to M then $f_S(x)$ is irreducible of degree m and order M. For $P = \mathrm{GF}(2)$ and such an $f_S(x)$, Theorem 2 in [5] provides an exact lower bound for the degree of any irreducible factor of $f_S(x^\pi)$. From this theorem it easily follows that if $f(x)$ is primitive, if $\gcd(S,\lambda) = 1$, and if every prime factor of π also divides λ then $f_S(x^\pi)$ is irreducible. In this case the linear complexity of u reaches its maximal possible value πm (this is equal to the degree of $f_S(x^\pi)$).

In many cases the period of sequence u can be determined more precisely. The following theorem extends [6, Theorem 4]. Recently, in [4, Theorem 2] Golić generalized this result for an arbitrary GR having an LFSR structure.

Theorem 2. *The output sequence u is periodic. If for $l \in \{0,\ldots,M-1\}$ the uniform (l,S)-decimation sequences of b are all distinct then the period of u is equal to*

$$\tau(\pi, M, S) = \frac{\pi M}{\gcd(S, M)} .$$

Let assume that b is a nonzero sequence. Then, by Theorem 1, Item (a), all the uniform (l,S)-decimation sequences of b for $l \in \{0,\ldots,M-1\}$ are distinct if $m(k) = m$ (see [4, Proposition 2], where a similar fact was proved for an arbitrary GR having LFSR structure).

Proposition 1. *Let $f(x)$ be a primitive polynomial of degree m, so it has the maximal possible order $\lambda = q^m - 1$. Then all uniform (l,S)-decimation sequences of b are distinct for $l \in \{0,\ldots,\lambda-1\}$ if and only if for any $l \in \{0,\ldots,\lambda-1\}$ the uniform $(l,\gcd(S,\lambda))$-decimation of b is nonzero.*

Proof. Let us first consider the congruence $xS \equiv y\gcd(S,\lambda) \pmod{\lambda}$ where $x \geq 0$ and $y \geq 0$. It is evident that for any fixed value of $x = 0,1,2,\ldots$ this congruence is solvable with respect to y and for any fixed value of $y = 0,1,2,\ldots$ it is solvable with respect to x. Thus, for any $l \geq 0$ a uniform (l,S)-decimation of b contains exactly the same elements as a uniform $(l,\gcd(S,\lambda))$-decimation.

Suppose now that for some $k,t \in \{0,\ldots,\lambda-1\}$ with $k = t$, the uniform (k,S) and (t,S)-decimation sequences of b are equal. By Theorem 1, Item (b), they can be equal if and only if $q^{m-m(S)} \geq 2$ and this is so if and only if for some l $\{0,\dots,\lambda-1\}$ the uniform (l,S)-decimation of b is zero. But then the uniform $(l,\gcd(S,\lambda))$-decimation is zero too.

The following corollary easily follows from corollaries 2 and 3, Proposition 1 and Theorem 2.

Corollary 4. *Let b be a nonzero sequence and suppose that one of the following two conditions holds*

(a) degree m of $f(x)$ is prime and S is not a multiple of $\frac{M}{\gcd(M,q-1)}$,
(b) $f(x)$ is a primitive polynomial (so, of order $\lambda = q^m - 1$) and $\gcd(S,\lambda) \le q^{m/2}$.

Then the period of u is equal to $\tau(\pi, M, S) = \frac{\pi M}{\gcd(S,M)}$.

Note that if $f(x)$ is primitive then one has $M = \lambda = q^m - 1$. If conditions of Theorem 2, Proposition 1 and Corollary 4 do not hold then the period of the decimated sequence may be equal to or smaller than $\frac{\pi M}{\gcd(S,M)}$. If S is relatively prime to M, it follows from Corollary 1 and Theorem 2 that the period of u reaches the maximal value πM (this is Theorem 4 in [6]).

3 Generalized Geffe Generator

Combining linear feedback shift registers with a memoryless nonlinear function F is a well-known way to increase the period and the linear complexity of the key-stream, as well as to reduce the correlation between the key-stream sequence and the LFSR sequences that are used as input of F, see [7]. The key-stream generator discussed in this section is a memoryless combiner based on a specific combining function that implements a nonuniform decimation of input sequences. The key-stream sequence is obtained by irregularly interleaving the decimated sequences. Both decimation and interleaving operations are controlled by the same sequence being one of combining function inputs. This construction can be seen as a generalization of the Geffe generator from [1].

First, we need to define and fix an ordering in the finite field $P = \mathrm{GF}(q)$ by numbering the elements from 0 to $q-1$, thus $P = \{p_0,\dots,p_{q-1}\}$. Let the combining function F from P^{q+1} to P be defined by $F(p_j, x_0,\dots,x_{q-1}) = x_j$ for $j = 0,\dots,q-1$. Thus, the first argument of F selects which of the remaining q arguments is taken as an output of the function. Let assume that a periodic sequence $a = \{a_i\}_{i\ge 0}$ in P (we will also call it the control sequence of F) with the period π and linear complexity $\hat{L}$ is fed to the first argument of F and that q periodic sequences $b^j = \{b^j_i\}_{i\ge 0}$ $(j = 0,\dots,q-1)$ in P with periods λ_j and linear complexity L_j respectively are fed to the remaining q arguments. Let $u = \{u_i\}_{i\ge 0}$ denote the output sequence generated by the function F (see Fig. 2). The period and linear complexity of u are estimated further in this section.

Before we can continue, we need some preliminary lemmas. The first one is a special case of a fundamental result on the period of nonuniformly decimated sequences, as established in [8, Theorem 3].

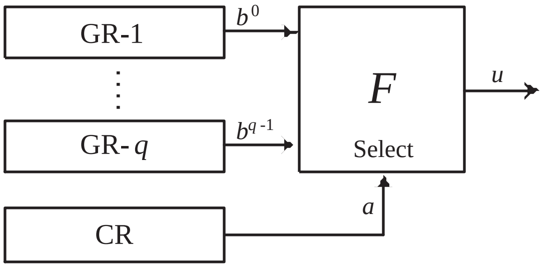


Fig. 2. Generalized Geffe generator

Lemma 1. *Let $c = \{c_i\}_{i\geq 0}$ be a periodic sequence with the period T and let sequence $c = \{c_i\}_{i\geq 0}$ be a uniform d-decimation of c for some integer $d > 0$. Then c is periodic and if T denotes its period then*

(a) $T \mid \frac{T}{\gcd(T,d)}$;

(b) If $\gcd(T,d) = 1$ *then* $T = T$.

Let K denote the least common multiple of the periods of the sequences b^j $(j = 0,\dots,q-1)$, so $K = \mathrm{lcm}(\lambda_0,\dots,\lambda_{q-1})$ and let d denote $\gcd(\pi, K)$. It is obvious that K is equal to the period of the sequence of q-grams $B = \{(b_i^0,\dots,b_i^{q-1})\}_{i\geq 0}$.

Lemma 2. *Suppose that sequence a contains all elements of P and that the q-gram sequence B with the period K contains a q-tuple that is equal to P in the sense of set equality. Suppose moreover that* $\gcd(\pi, K) = 1$. *Then* $\tau = \pi K$.

Proof. Under the hypothesis of the lemma, we can list a set of integers $t_j \geq 0$ $(j = 0,\dots,q-1)$ such that $a_{t_j} = p_j$. Let us consider q uniform (t_j,π)-decimation sequences of the output u by taking $j = 0,\dots,q-1$. Since π is the period of the control sequence a, the (t_j,π)-decimation of u is equal to the (t_j,π)-decimation of b^j. But hypothesis of the lemma claims that $\gcd(\pi, K) = 1$ whence it follows that $\gcd(\pi,\lambda_j) = 1$ for $j = 0,\dots,q-1$. Hence by Lemma 1, Item (b), the period of the (t_j,π)-decimation of b^j is λ_j for $j = 0,\dots,q-1$. But since these decimation sequences are decimation sequences of u as well, by Lemma 1, Item (a), $\lambda_j \mid \tau$ for $j = 0,\dots,q-1$ and thus $K \mid \tau$.

Under the hypothesis of the lemma, there exists an integer $t \geq 0$ such that the q-tuple $(b_t^0,\dots,b_t^{q-1})$ can be obtained by permutating the elements in $(p_0,\dots, p_{q-1})$. Let us now consider the uniform (t, K)-decimation of the output sequence u. Since K is the period of the q-gram sequence B, this decimation is equal to the (t, K)-decimation of a which elements are substituted afterwards according to the rule defined by the permutation transforming $(p_0,\dots,p_{q-1})$ into $(b_t^0,\dots,b_t^{q-1})$. A one-to-one mapping applied to the elements of a sequence does not affect its period. Since $\gcd(\pi, K) = 1$, by Lemma 1, Item (b), the period of the (t, K)-decimation of a is π. But since this decimation is a decimation of u as well, by Lemma 1, Item (a), $\pi \mid \tau$.

Now since $K \mid \tau$, $\pi \mid \tau$ and $\gcd(\pi, K) = 1$ we can conclude that $\pi K \mid \tau$. On the other hand, it is obvious that $\tau \mid \pi K$ and thus $\tau = \pi K$.

Theorem 3. *The sequence u is periodic. Let τ denote the period of u. Then $\tau \mid \mathrm{lcm}(\pi, K)$. Moreover, if sequence a is such that each of its uniform d-decimation sequences contains all the elements of P and the q-gram sequence B is such that all its uniform d-decimation sequences contain a q-tuple that is equal to P in the sense of set equality then*

$$\frac{\pi K}{\gcd(\pi, K)^2} \,\Bigg|\, \tau \ .$$

Proof. It is obvious that in every $\mathrm{lcm}(\pi, K) = \mathrm{lcm}(\pi, \lambda_0, \ldots, \lambda_{q-1})$ steps all input sequences complete their full cycle. Since function F is memoryless, the output sequence u completes a full cycle as well in $\mathrm{lcm}(\pi, K)$ steps. Thus u is periodic and $\tau \mid \mathrm{lcm}(\pi, K)$.

Let us consider the q-gram sequence B. Since all sequences b^j $(j = 0, \ldots, q-1)$ are periodic with the period equal to λ_j respectively, it is obvious that the q-gram sequence B is periodic as well with the period equal to $\mathrm{lcm}(\lambda_0, \ldots, \lambda_{q-1}) = K$.

Now we fix an arbitrary $t \in \{0, \ldots, d-1\}$ and consider uniform (t, d)-decimation sequences of a, u and B. Let π_t, τ_t and K_t denote the respective periods of these decimation sequences. Then, by Lemma 1, Item (a),

$$\pi_t \,\Bigg|\, \frac{\pi}{\gcd(\pi, d)} = \frac{\pi}{d}\,, \quad \tau_t \mid \tau \quad \text{and} \quad K_t \,\Bigg|\, \frac{K}{\gcd(K, d)} = \frac{K}{d} \ . \tag{3}$$

Since $\gcd(\frac{\pi}{d}, \frac{K}{d}) = 1$, it follows that $\gcd(\pi_t, K_t) = 1$.

Let us now consider the memoryless combiner described above when uniform (t, d)-decimation sequences of the respective original sequences are fed into the arguments of F. Thus, the control sequence of F has period π_t and the q-gram sequence, feeding the rest of the arguments of F, has period K_t satisfying $\gcd(\pi_t, K_t) = 1$. We note that the output sequence of F has period τ_t since it is a uniform (t, d)-decimation of sequence u. So, the conditions of Lemma 2 are met and thus it follows that

$$\tau_t = \pi_t K_t \ , \tag{4}$$

for all $t \in \{0, \ldots, d-1\}$.

By (3), π_t divides $\frac{\pi}{d}$ for $t = 0, \ldots, d-1$ and therefore $\mathrm{lcm}(\pi_0, \ldots, \pi_{d-1}) \,\Big|\, \frac{\pi}{d}$. Sequence a can be reconstructed by interleaving d sequences obtained by (t, d)-decimating of a for $t = 0, \ldots, d-1$ and thus $d \cdot \mathrm{lcm}(\pi_0, \ldots, \pi_{d-1})$ is a multiple period of a, that is $\pi \mid d\, \mathrm{lcm}(\pi_0, \ldots, \pi_{d-1})$. Hence $\mathrm{lcm}(\pi_0, \ldots, \pi_{d-1}) = \frac{\pi}{d}$. In the same way it is easy to show that $\mathrm{lcm}(K_0, \ldots, K_{d-1}) = \frac{K}{d}$.

From (3) it also follows that $\gcd(\pi_i, K_j) = 1$ $(i, j = 0, \ldots, d-1)$. Thus

$$\begin{aligned}
\mathrm{lcm}(\tau_0, \ldots, \tau_{d-1}) &\overset{(4)}{=} \mathrm{lcm}(\pi_0 K_0, \ldots, \pi_{d-1} K_{d-1}) = \\
&= \mathrm{lcm}(\mathrm{lcm}(\pi_0, K_0), \ldots, \mathrm{lcm}(\pi_{d-1}, K_{d-1})) = \\
&= \mathrm{lcm}(\pi_0, \ldots, \pi_{d-1}, K_0, \ldots, K_{d-1}) =
\end{aligned}$$

$$= \mathrm{lcm}(\mathrm{lcm}(\pi_0, \dots, \pi_{d-1}), \mathrm{lcm}(K_0, \dots, K_{d-1})) =$$
$$= \mathrm{lcm}(\pi_0, \dots, \pi_{d-1}) \cdot \mathrm{lcm}(K_0, \dots, K_{d-1}) = \frac{\pi K}{d^2} \ .$$

Also by (3), τ_t divides τ for $t = 0, \dots, d-1$ and therefore $\mathrm{lcm}(\tau_0, \dots, \tau_{d-1}) = \frac{\pi K}{d^2} \mid \tau$.

The following lemma, that easily follows from [2, Proposition], will be needed to estimate the linear complexity of u.

Lemma 3. *Let $c = \{c_i\}_{i \geq 0}$ be a periodic sequence having linear complexity L and let $c \ = \{c_i\}_{i \geq 0}$ be a uniform d-decimation of c for some integer $d > 0$. Then there exists a polynomial $f_{(d)}(\cdot)$ annihilating c as well as all d-decimation sequences of c, where the degree of $f_{(d)}(\cdot)$ is not greater than L.*

Proposition 2. *Let L denote the linear complexity of an output sequence u. Then $L \leq \pi(L_0 + \dots + L_{q-1})$. If $q = 2$, the sequences b^0 and b^1 are nonzero, and the respective periods π, λ_0, and λ_1 are pairwise coprime then $L \geq (\hat{L} - 1)(L_0 + L_1 - 2)$.*

Proof. To prove the claimed upper bound on the linear complexity of the sequence u it is sufficient to present a polynomial $P(\cdot)$ of degree not greater than $\pi(L_0 + \dots + L_{q-1})$, for which $P(u) = 0$ (i.e. P is an annihilating polynomial of u). Let us consider an arbitrary uniform π-decimation of u. Since π is the period of the control sequence a, this decimation is equal to the (t_j, π)-decimation of b^j for some $j \quad \{0, \dots, q-1\}$ and $t_j \quad \{0, \dots, \lambda_j - 1\}$. Then, by Lemma 3, there exists a polynomial $Q_j(\cdot)$ of degree not greater than L_j annihilating this decimation as well as all the other π-decimation sequences of b^j. The polynomial $Q_j(\cdot)$ also annihilates the uniform π-decimation of u that we consider.

Now let $Q(\cdot)$ be the least common multiple of polynomials $Q_0(\cdot), \dots, Q_{q-1}(\cdot)$ where $Q_j(\cdot)$ is the polynomial annihilating any π-decimation of b^j. Then $Q(\cdot)$ annihilates any π-decimation of u and thus polynomial $P(\cdot) = Q(x^\pi)$ of degree not greater than $\pi(L_0 + \dots + L_{q-1})$ annihilates u. Thus the linear complexity of u is at most $\pi(L_0 + \dots + L_{q-1})$.

The second part of the proposition follows from [9, Theorem 6] since the algebraic normal form of the combining function for $q = 2$ is $F(a, x_0, x_1) = a(x_0 \quad x_1) \quad x_0$. Condition $q = 2$ is required since only then the algebraic normal form of F is free from powers.

It remains an open problem how to estimate a nontrivial lower bound for the linear complexity of the output sequence u when $q > 2$.

If we assume that input sequences of the combining function F are sequences of uniform, independent and identically distributed random variables (i.e. purely random sequences) then its output sequence is purely random as well since the combining function of the generator is balanced. Thus the balance quality of the combining function ensures good statistical properties of the key-stream.

Sequences produced by linear feedback shift registers (clocked regularly or irregularly) could be used as inputs for function F in practical implementations of the key-stream generator described above. Let us note that the combining function F of the generator is memoryless, balanced and zero-order correlation immune (its output is correlated to inputs $x_0, \dots, x_{q-1}$ and this correlation decreases if q is increased). Thus when all shift registers are clocked regularly, it is possible to apply the basic or fast correlation attack in order to reconstruct the initial state of shift registers that produce sequences b^j ($j = 0, \dots, q-1$). Therefore it is reasonable to use large q and/or clock-controlled LFSR's to generate sequences b^j ($j = 0, \dots, q-1$). We note that knowing the periods of the control and the generating registers, one can easily verify the condition of coprimality in Proposition 2. Memoryless combiners of clock-controlled LFSR's can also be susceptible to certain types of correlation attacks. But the essential benefit of these combiners consists in their immunity against fast correlation attacks.

For practical implementation of the suggested generator it may be reasonable to select q as a power of 2, and to generate binary sequences a and b^j ($j = 0, \dots, q-1$), to feed them as input to the $(q+1)$-input combining function F. The control sequence is split into $\log_2 q$-long tuples that are used to index sequences b^j ($j = 0, \dots, q-1$). Following the first half of the proof of Lemma 2, it can be readily shown that if the control sequence splits into $\log_2 q$-tuples consisting of all q possible values and if $\gcd(\pi, K) = 1$ then $K \mid \tau$.

Acknowledgment

The author is grateful to Henk van Tilborg and to anonymous referees for their comments which helped to improve the paper.

References

1. Geffe, P.R.: How to protect data with ciphers that are really hard to break. Electronics **46** (1973) 99–101
2. Golić, J.D.: On decimation of linear recurring sequences. Fibonacci Quarterly **33** (1995) 407–411
3. Zierler, N.: Linear recurring sequences. Journal of the Society for Industrial and Applied Mathematics **7** (1959) 31–48
4. Golić, J.D.: Periods of interleaved and nonuniformly decimated sequences. IEEE Transactions on Information Theory **44** (1998) 1257–1260
5. Chambers, W.G.: Clock-controlled shift registers in binary sequence generators. IEE Proceedings - Computers and Digital Techniques **135** (1988) 17–24
6. Gollmann, D., Chambers, W.G.: Clock-controlled shift registers: a review. IEEE Journal on Selected Areas in Communications **7** (1989) 525–533
7. Rueppel, R.A.: Analysis and Design of Stream Ciphers. Communications and Control Engineering Series. Springer-Verlag, Berlin (1986)
8. Blakley, G., Purdy, G.: A necessary and sufficient condition for fundamental periods of cascade machines to be product of the fundamental periods of their constituent finite state machines. Information Sciences: An International Journal **24** (1981) 71–91
9. Golić, J.D.: On the linear complexity of functions of periodic GF(q) sequences. IEEE Transactions on Information Theory **35** (1989) 69–75

Efficient Software Implementation of Linear Feedback Shift Registers

Sandeepan Chowdhury[1] and Subhamoy Maitra[2]

[1] Applied Statistics Unit, Indian Statistical Institute,
203, B T Road, Calcutta 700 035, India,
sandeepan@consultant.com

[2] Computer and Statistical Service Center, Indian Statistical Institute,
203, B T Road, Calcutta 700 035, India,
subho@isical.ac.in

Abstract. Linear Feedback Shift Registers (LFSRs) are used as pseudorandom keystream generators in cryptographic schemes. Hardware implementation of LFSRs are simple and fast but their software implementation is not quite efficient. Here we present a fast software implementation strategy for the LFSRs. The output will be available as a block of bits after each operation. We discuss theoretical issues for such block oriented implementation and present necessary algorithms. We clearly identify the constraints in the choice of connection polynomials for block oriented implementation. Actual implementation results have been presented in support of our claims. The results emphasise the usability of LFSRs in software based stream cipher systems.

Keywords: Block Oriented LFSR, Connection Polynomials, Stream Cipher.

1 Introduction

In this paper we deal with the issues related to fast software implementation of Linear Feedback Shift Registers. We here introduce the concept of block oriented LFSR. The LFSR of length n is divided into y blocks of b bits each. We present an equivalent linear recurrence relation between these y blocks for the software implementation. After each operation, the output of the LFSR is one block of b bits. In the next section we discuss some preliminary concepts of LFSRs. Section 3 deals with the issues of software implementation techniques for block LFSRs and provides a concrete design strategy. The performance analysis of this implementation is discussed in Section 4. Note that there are some constraints regarding the choice of connection polynomials in the strategy proposed in Section 3. We partially solve this problem in Section 5.

2 Preliminaries

An LFSR consists of a set of registers each of which can take the value 0 or 1. The connection pattern of an LFSR can be indicated by a polynomial over GF(2)

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 297–307, 2001.

and this polynomial is called the *connection polynomial*. We consider a degree n polynomial over GF(2) as $x^n \oplus \bigoplus_{i=0}^{n-1} a_i x^i$, where $a_i \in \{0, 1\}$. Note that $\oplus$ is addition modulo 2 and + indicates ordinary integer addition. By *weight* of this polynomial we mean the number of nonzero terms, i.e., $1 + \#\{a_i = 1\}$. Next we consider an LFSR of length n (i.e., n number of registers) corresponding to this n degree polynomial. We denote the bit positions of the LFSR by $z_0, z_1, \ldots, z_{n-1}$. By ν_i we mean the value at the bit position z_i, $i = 0, \ldots, n-1$. For the LFSR in Figure 1, $n = 6$, and bit positions are $z_0, z_1, \ldots, z_5$. The *Least Significant Bit* (LSB) starts from the extreme right side (in Figure 1, z_0). The leftmost bit is the Most Significant Bit (MSB). We denote the locations of the t taps (i.e., where $a_i = 1$) in the LFSR by $p_0, p_1, \ldots, p_{t-1}$, where $p_0 < p_1 < \ldots < p_{t-1}$, i.e., p_0 is closest to the LSB while p_{t-1} is closest to the MSB. In Figure 1, $t = 4$ and $p_0 = 0, p_1 = 2, p_2 = 3, p_3 = 5$.

An output bit is obtained from the LFSR by the following mechanism. A new bit ν_n is obtained by XORing the bit values of the positions corresponding to the taps p_j i.e., $\nu_n = \bigoplus_{j=0}^{t-1} \nu_{p_j}$. The LSB comes out as the output bit and each of the remaining bits undergoes one right shift. Thereafter, the vacant MSB is filled up by the new bit ν_n, already generated. In hardware, this entire operation is completed by one *clock*.

For an LFSR with t number of taps, the connection polynomial actually contains $(t + 1)$ number of terms, t places where $a_i = 1$ and the term x^n. The polynomial $x^n \oplus \bigoplus_{i=0}^{n-1} a_i x^i$, yields a recurrence relation of form $\nu_{k+n} = \bigoplus_{j=0}^{t-1} \nu_{k+p_j}$, for $k \geq 0$. By ν_{k+i} we denote the value of the bit z_i after k clocks. In Figure 1, recurrence relation is $\nu_{k+6} = \nu_{k+5} \oplus \nu_{k+3} \oplus \nu_{k+2} \oplus \nu_k$. For more details about LFSRs see [2,1,5] and the references in these documents.

The LFSR outputs a single bit in each clock. Hardware implementation of such a structure is simple and fast. The main problem is an efficient software implementation of such a system. In [3] a method for simulation of an LFSR has been described using binary matrix. It is clear that we can represent the state of an LFSR (of length n) by an $n \times 1$ binary column vector x. If x_0 is the initial state and x_i is the state after i clocks, x_i can be derived from x_0 by the operation $\mathrm{x}_i = A^i \mathrm{x}_0$, where A is an $n \times n$ binary matrix, called the state

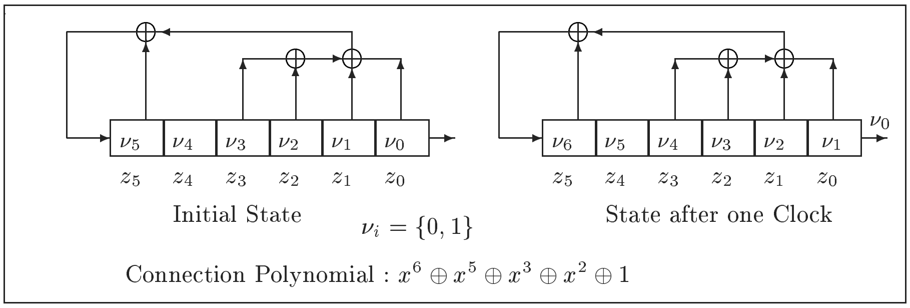

ν_5 ν_4 ν_3 ν_2 ν_1 ν_0
z_5 z_4 z_3 z_2 z_1 z_0
Initial State
ν_0
ν_6 ν_5 ν_4 ν_3 ν_2 ν_1
z_5 z_4 z_3 z_2 z_1 z_0
State after one Clock
$\nu_i = \{0, 1\}$
Connection Polynomial : $x^6 \oplus x^5 \oplus x^3 \oplus x^2 \oplus 1$

Fig. 1. Bit operation of an LFSR.

transition matrix (see [3] for details). Thus, we can write $\mathrm{x}_{nk} = A^n \mathrm{x}_{n(k-1)}$, for $k \geq 1$. But the process involves matrix multiplication over GF(2), and required number of steps is of $O(n^2)$ for n output bits, i.e., $O(n)$ for a single bit.

An algorithm for fast software implementation has been described in [4] by packing the LFSR into integers of size 32 bits. However, a closer look at the algorithm in [4] reveals that the number of logical operations needed to generate a single bit is at least 9, which makes it inefficient.

3 Block Oriented Implementation of LFSRs

Here we consider an LFSR as an aggregate of *blocks* of fixed size. By the term *block* we mean a number of contiguous bits. As the constituent elements of the LFSR are now blocks, the output will be one block instead of a single bit. Our main objective will be to find out an efficient algorithm which outputs one block after each operation. First we fix a few parameters to explain the algorithm for block operation.

1. The LFSR of size n corresponds to the connection polynomial of degree n.
2. We consider that the LFSR consists of y blocks, each of size b. So, $n = yb$. As example, we can consider a 32 bit LFSR with block size 8, i.e., $n = 32$ and $b = 8$, so, $y = 4$ (see Figure 2).
3. These y number of blocks are denoted by $Y_0, Y_1, \ldots, Y_{y-1}$. Here, Y_0 is the right most (least significant) block. Bits in any block Y_i are denoted by $Y_{i,0}, Y_{i,1}, \ldots, Y_{i,b-1}$ (from LSB to MSB). The bit position z_l, $l = 0, \ldots, n-1$, for an LFSR can be related with corresponding bit position in blocks by $Y_{i,j}$, where block position $i = l/b$ (the quotient) and relative position in block Y_i

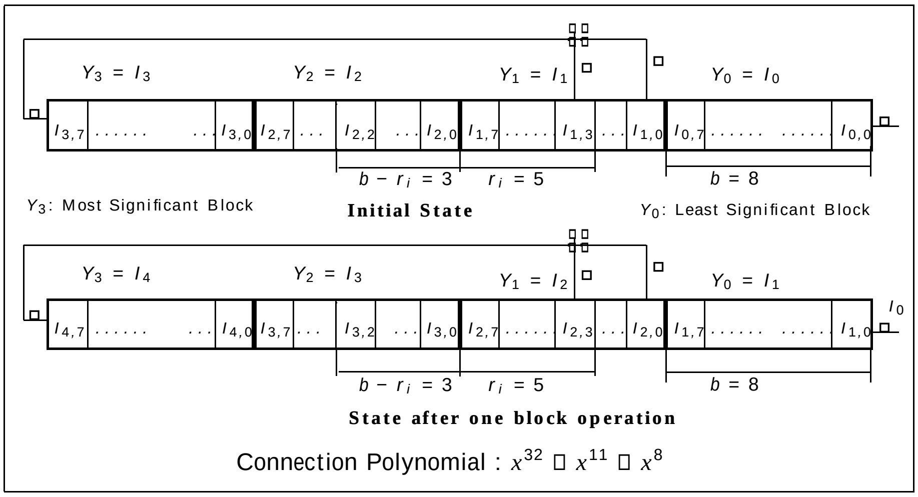


Fig. 2. Block-oriented LFSR.

is $j = l\%b$ (the remainder). In any block Y_i, the rigthmost bit position $Y_{i,0}$ is termed as the *boundary position*.

4. Initial state of a block Y_i (i.e., set of values for all the b bits in this block) is denoted by I_i. In terms of bits, $Y_{i,0} = I_{i,0}$, $Y_{i,1} = I_{i,1}$, $\ldots$, $Y_{i,b-1} = I_{i,b-1}$, $I_{i,j} \quad \{0, 1\}$. The state of a block changes after completion of one block operation. After obtaining k number of output blocks (i.e., at the end of k block operations), we denote value of Y_i by I_{k+i}. In Figure 2, initial state of the LFSR is $Y_0 = I_0, Y_1 = I_1, Y_2 = I_2, Y_3 = I_3$ and after one block operation, $Y_0 = I_1, Y_1 = I_2, Y_2 = I_3, Y_3 = I_4$.
5. The position of any tap p_j $(j = 0, \ldots, t-1)$ is mapped to bit position Y_{q_j,r_j}, where $q_j = p_j/b$ and $r_j = p_j\%b$. In Figure 2, there are two taps $p_0 = 8$ and $p_1 = 11$, correspondingly $q_0 = 1, q_1 = 1$ and $r_0 = 0, r_1 = 3$.

The motivation behind defining the block operation is to find out a fast method of computation to get I_y out of $I_0, I_1, \ldots, I_{y-1}$ and thereby to arrive at a recurrence relation for I_{k+y} in terms of $I_{k+0}, I_{k+1}, I_{k+2}, \ldots, I_{k+y-1}$.

Example 1. We consider $n = 32, b = 8, y = 4$, i.e., each block is of one byte. Let the connection polynomial be $x^{32} \quad x^{11}$. Here, the tap $p_0 = 11$ is in bit position $Y_{1,3}$ as $q_0 = 1$ and $r_0 = 3$. After one block operation, the state of the LFSR is $Y_0 = I_1, \ldots, Y_3 = I_4$. The new block I_4 is to be expressed as a combination of the block states I_0, I_1, I_2, I_3. The 8 successive new bits generated from the LFSR constitutes I_4. These 8 new bits $I_{1,3}, I_{1,4}, I_{1,5}, I_{1,6}, I_{1,7}$ and $I_{2,0}, I_{2,1}, I_{2,2}$ are the 8 successive values of tap position $Y_{1,3}$. We can rewrite the block I_4 as $I_4 = (I_{2,2}, I_{2,1}, I_{2,0}, 0, 0, 0, 0, 0) \quad (0, 0, 0, I_{1,7}, I_{1,6}, I_{1,5}, I_{1,4}, I_{1,3})$. Simplifying, $I_4 = (I_2 << 5) \quad (I_1 >> 3)$. Thus, the byte oriented recurrence relation will be $I_{k+4} = (I_{k+2} << 5) \quad (I_{k+1} >> 3)$, for $k \geq 0$.

Now we generalise the expression. For a single tap p_0, the new block generated from one block operation actually consists of b successive new bits generated from the LFSR, i.e., b successive values of bit position Y_{q_0,r_0}. Thus the new block consists of b contiguous bits, starting from position Y_{q_0,r_0}, towards left. This is the *basic principle* behind generation of a new block for a tap by one complete block operation. Evidently these b number of bits are the left most $(b - r_0)$ bits from the block Y_{q_0} and the right most r_0 bits of the adjacent left block Y_{q_0+1}. So, $I_y = (I_{q_0+1,b-r_0-1}, \ldots, I_{q_0+1,0}, 0, \ldots, 0) \quad (0, \ldots, 0, I_{q_0,b-1}, \ldots, I_{q_0,r_0})$. Thus the recurrence relation for a block operation is

$$I_{k+y} = (I_{k+q_0+1} << (b - r_0)) \quad (I_{k+q_0} >> r_0), \text{ for } k \geq 0. \qquad (3a)$$

In total we need three logical operations for getting a new block corresponding to a single tap. It is important to note that r_0 bits of its adjacent left block are also required for construction of the resultant block. When $q_0 = y - 1$ and $r_0 > 0$, to generate the new block I_y we require the r_0 bits of block I_y which is actually the new block to be obtained from the LFSR. Hence it will not hold when the position of the tap $p_0 > n - b$, i.e., beyond the boundary position $Y_{(y-1),0}$. *Considering this restriction, for now, our discussion for defining block operation is kept confined to tap positions* $\leq n - b$. We will tackle this partially in Section 5.

Next we investigate the case for $t > 1$ number of taps. Considering the bit operation (see Section 2) of block LFSRs, the first new bit generated is $\bigoplus_{i=0}^{t-1} I_{q_i,r_i}$. Thus for the initial state $I_0,\ldots,I_{y-1}$ of the block oriented LFSR, the resultant block I_y generated for t taps can be obtained by XORing all the new blocks generated from each tap considering them individually. So $I_y = \bigoplus_{i=0}^{t-1} I_y^{p_i}$, *where $I_y^{p_i}$ denotes the new block generated by the tap p_i had it been the only tap of the LFSR for the given state.*

Example 2. Consider the connection polynomial $x^{32} \oplus x^{15} \oplus x^{11}$. Here both the taps are in block Y_1 and $p_0 = 11$, $p_1 = 15$. Hence, $q_0 = q_1 = 1$ and $r_0 = 3$ and $r_1 = 7$. So, new block $I_4 = I_4^{p_0} \oplus I_4^{p_1}$, where $I_4^{p_0} = (I_{2,2}, I_{2,1}, I_{2,0}, I_{1,7},\ldots,I_{1,3})$ and $I_4^{p_1} = (I_{2,5},\ldots,I_{2,0}, I_{1,7}, I_{1,6})$. Thus, $I_4^{p_0}$ and $I_4^{p_1}$ are the two new blocks generated by the block operation, considering each of the taps p_0 and p_1 separately for LFSR state $I_0,\ldots,I_3$.

Without loss of generality, we can extend the relation for any given state of the LFSR, $I_{k+y-1},\ldots,I_k$, where $k \geq 0$. Thus the recurrence relation for the resultant new block I_{k+y} for t taps is $I_{k+y} = \bigoplus_{i=0}^{t-1} I_{k+y}^{p_i}$ for $k \geq 0$. (3*b*)

Using Equation (3*a*), $I_{k+y}^{p_i} = (I_{k+q_i+1} << (b - r_i)) \oplus (I_{k+q_i} >> r_i)$. Combining Equations (3*a*) and (3*b*), with the restriction for tap positions $0 \leq p_0 < p_1,\ldots < p_{t-1} \leq n - b$, the recurrence relation for block operation is presented by the following lemma.

Lemma 1. *Consider a polynomial $x^n \oplus \bigoplus_{i=0}^{n-1} a_i x^i$ over GF(2), where $n = yb$, b is the block size and y is the number of blocks. Let there be $t = \#\{a_i = 1\}$ taps at the positions $p_0, p_1,\ldots,p_{t-1}$, such that $0 \leq p_0 < p_1 < \ldots < p_{t-1} \leq n - b$. Consider $p_i = bq_i + r_i$, where $0 \leq r_i \leq b - 1$. Then the block oriented recurrence relation for this LFSR is*
$I_{k+y} = \bigoplus_{i=0}^{t-1}((I_{k+q_i+1} << b - r_i) \oplus (I_{k+q_i} >> r_i))$, for $k \geq 0$.

Now we consider a special case of the above lemma with each tap p_i at the boundary position, i.e., $r_i = 0$. This, using Lemma 1, gives $I_{k+y} = \bigoplus_{i=0}^{t-1}((I_{k+q_i+1} << b) \oplus (I_{k+q_i} >> 0))$, i.e., $I_{k+y} = \bigoplus_{i=0}^{t-1} I_{k+q_i}$ for $k \geq 0$. (3*c*)

Thus, we can generate the new block for a boundary tap without any bit shifting operation. Equation (3*c*) indicates that for the taps at boundary location, the generation of new block requires a single logical operation. This can be successfully exploited for more efficient software implementation. Let us now present the corollary as follows.

Corollary 1. *Consider a polynomial $x^n \oplus \bigoplus_{i=0}^{n-1} a_i x^i$ over GF(2), where $n = yb$, b is the block size and y is the number of blocks. Let there be $t = \#\{a_i = 1\}$ taps at the positions $p_0, p_1,\ldots,p_{t-1}$, such that $0 \leq p_0 < p_1 < \ldots < p_{t-1} \leq n - b$. Consider $p_i = bq_i$, for all i. Then the block oriented recurrence relation for this LFSR is $I_{k+y} = \bigoplus_{i=0}^{t-1} I_{k+q_i}$ for $k \geq 0$.*

Combining Lemma 1 and Corollary 1, we get the following theorem defining the block oriented recurrence relation with certain restrictions.

Theorem 1. *Consider a polynomial* $x^n \quad \square_{i=0}^{n-1} a_i x^i$ *over GF(2), where* $n = yb$, b *is the block size and* y *is the number of blocks. Let there be* $t = \#\{a_i = 1\}$ *taps* ($t = t_1 + t_2$), *such that the taps* $0 \le p_0 < p_1 < \ldots < p_{t_1-1} \le n - b$ *are at boundary locations with* $r_i = 0$ *for* $0 \le i \le t_1 - 1$ *and the remaining* t_2 *taps,* $0 < p_{t_1} < p_{t_1+1} < \ldots < p_{t-1} < n - b$ *are not in boundary locations, i.e.,* $p_i = bq_i + r_i$, *where* $0 < r_i \le b - 1$, $t_1 \le i \le t - 1$. *Then the block oriented recurrence relation for this LFSR is*

$$I_{k+y} = \square_{i=0}^{t_1-1} I_{k+q_i} \quad \square_{i=t_1}^{t-1} ((I_{k+q_i+1} << (b - r_i)) \quad (I_{k+q_i} >> r_i)),\ \textit{for}\ k \ge 0.$$

Next we present a simple C like algorithm for the implementation of this theorem.

Implementation blockLFSROutput

```
for ($i = 0; i < t; i++$) { $q[i] = p[i]/b$;  $r[i] = p[i]\%b$; }
$k = 0$;
while output is required {
    $I[k + y] = 0$;
    for ($i = 0; i < t_1; i++$) $I[k + y] = I[k + y] \quad I[k + q[i]]$;
    for ($i = t_1; i < t; i++$)
        $I[k + y] = I[k + y] \quad ((I[k + q[i] + 1] << b - r[i]) \quad (I[k + q[i]] >> r[i]))$;
    Output $I[k]$; $k = k + 1$;
}
```

The space overhead for storing the array I can be easily avoided by allocating this array dynamically for certain number of output blocks at a time. After generation of a good number of blocks the array can be released, retaining only the last y blocks which may be required to generate further blocks. It should be noted that it is always preferable to implement an LFSR using array, since the use of linked list or other standard data structure requires higher number of operations for accessing each element. *According to the above implementation, we require one logical operation for each boundary tap and four logical operations for each non boundary taps.* Thus we have the following result.

Theorem 2. *Consider a block oriented LFSR as in Theorem 1 with* t_1 *boundary and* t_2 *nonboundary taps. Then the software implementation* **blockLFSROutput** *will require* $\frac{t_1+4t_2}{b}$ *logical operation on average to generate each output bit.*

4 Implementation Results

To get the maximum linear complexity, the connection polynomials of LFSRs are generally taken as primitive over GF(2). We execute the following steps to generate the connection polynomials.

1. Choose a polynomial $x^n \quad \square_{i=0}^{y-1} a_{ib} x^{ib}$ over GF(2), where $a_{ib} \quad \{0, 1\}$ and $\#\{a_{ib} = 1\} = t_1$. Let us denote these tap positions as $p_1, \ldots, p_{t_1-1}$.
2. Apart from these positions $p_1, \ldots, p_{t_1-1}$, randomly choose t_2 other positions from 1 to $n - b - 1$ such that $t = t_1 + t_2 + 1$ is odd (weight of primitive polynomials are always odd). Let us denote these tap positions as $p_{t_1}, \ldots, p_{t-1}$.
3. Check whether the polynomial $x^n \quad \square_{i=0}^{t-1} x^{p_i}$ is primitive. If it is primitive, then report it and terminate. Else go to step 2.

Table 1. Results for 32 degree primitive polynomials.

$t = t_1 + t_2$	Connection polynomial	σ
6	$p_1(x)$ □ x^{14} □ x	1.50
8	$p_1(x)$ □ x^7 □ x^5 □ x^2 □ x	2.50
10	$p_1(x)$ □ x^9 □ x^5 □ x^4 □ x^3 □ x^2 □ x	3.50
12	$p_1(x)$ □ x^{23} □ x^7 □ x^6 □ x^5 □ x^4 □ x^3 □ x^2 □ x	4.50
14	$p_1(x)$ □ x^{15} □ x^{11} □ x^9 □ x^7 □ x^6 □ x^5 □ x^4 □ x^3 □ x^2 □ x	5.50

Next we provide some concrete examples with respect to 32 degree polynomials. We consider byte oriented LFSRs, i.e., $b = 8$. Let $p_1(x) = x^{32} \quad x^{24} \quad x^{16} \quad x^8 \quad 1$, which means we choose all the 4 boundary taps. Then we execute the above three steps to get primitive polynomials. In Table 1 below, we provide the average logical operations per output bit $\sigma(= \frac{t_1+4t_2}{b}$, see Theorem 2) for different number of non boundary taps. Note that we will get better results (low values of σ) when the block size is larger. To demonstrate this, we consider 128 bit LFSRs with different block sizes $b = 8, 16$ and 32 and LFSRs with total number of taps $t = 24, 32, 48, 64$. Theorem 2 clearly shows that for same number of taps, average number of operations are much less for greater block size, which is also reflected in Table 3.

Table 2 and 3 indicate that it is not encouraging to use small block size (e.g., 8, 16) for a large LFSR (e.g., 128). Consider a primitive polynomial of degree 128 having weight 25 ($t = 24$). Table 2 shows that for block size 32 such a system needs less than 3 logical operations to generate 1 bit, which is encouraging. Moreover, for $n = 128$ and $b = 32$, we can consider a primitive polynomial with weight 11 ($t = 10, t_1 = 2, t_2 = 8$), which will take only $\frac{2+4\times8}{32} = 1.063 \approx 1$ logical operation on an average to produce 1 output bit. This is clearly competitive with the hardware implementation of LFSRs where one bit is generated in each clock. Actual bit generation rates obtained by software implementation of the above algorithm has been furnished in Table 3. We get these results using a personal computer having Pentium III 500 MHz microprocessor with 128 MB RAM on Windows NT (version 4.0)/ Windows 95 platform and Microsoft Visual C++ (version 6) compiler. The speed is measured in Mega bits per second (Mbps). Much better speed is expected if assembly language programming is used on a

Table 2. σ for different cases ($n = 128$).

Taps (t)	$b = 8$			$b = 16$			$b = 32$		
	t_1	t_2	σ	t_1	t_2	σ	t_1	t_2	σ
24	16	8	6	8	16	4.5	4	20	2.625
32	16	16	10	8	24	6.5	4	28	3.625
48	16	32	18	8	40	10.5	4	44	5.625
64	16	48	26	8	56	14.5	4	60	7.625

Table 3. Bit generation speed.

Length of LFSR	Block Size	Mbps	Taps		
			t	t_1	t_2
128	32	19.40	24	4	20
128	8	5.90	24	16	8
128	32	42.40	10	2	8
128	32	116.20	4	2	2
32	8	9.59	12	4	8
32	na	1.10	12	4	8

dedicated machine where operating system overhead is low. Note that the last two rows provide a direct comparison between the speed of our scheme and the speed of the algorithm from [4] (the last row). Even for the worst case of our algorithm (small block size of 8 bits) the speed is around 9 times faster than the existing one [4].

The idea of *word oriented* LFSR for software stream cipher was indicated in [6]. The connection polynomial considered by the authors [6] $p(x) = x \cdot (x^{127} + x^{63} + 1)$ is not a primitive one, instead it has a primitive polynomial $x^{127} + x^{63} + 1$ as its factor. This 128 degree polynomial $(x^{128} + x^{64} + x)$ [6] has been implemented using 4 blocks of 32 bits each. The recurrence relation has been presented in [6] as $I_{k+4} = I_{k+2} \quad ((I_k >> 1) \quad (I_{k+1} << 31))$, for $k \geq 0$ (average logical operation per bit ≈ 0.13). There is no primitive trinomial of degree 128. But considering a primitive five-nomial (with $t_1 = 2, t_2 = 2$), we get $\sigma \approx 0.31$ for block size 32 (speed $\approx$ 116 Mbps, see Table 3). The bit generation speed is also very competitive for higher number of taps. So instead of going for *word-oriented non primitive connection polynomial* as in [6], one can easily find out primitive polynomials with specific weight and use that for efficient block operation.

5 Tap at Most Significant Block

Here we try to remove the constraint regarding taps in most significant block. In this direction, we consider a single non boundary tap in the most significant block Y_{y-1} i.e., $p_{t-1} > n - b$. The remaining $t-1$ taps consists of t_1 boundary taps $p_0 < p_1 < \ldots < p_{t_1-1}$ and t_2 non boundary taps $p_{t_1} < p_1 < \ldots < p_{t-2} \leq n - b$. As usual, the initial state of the LFSR is $I_0, \ldots, I_{y-1}$.

First consider the case without taking into account the tap p_{t-1}. In this case we get the recurrence relation $I^{s}_{k+y} = \; {}^{t-2}_{j=0} I^{p_j}_{k+y}$ (see Theorem 1). Once again we like to mention that $I^{p_j}_{k+y}$ is the output block contributed by the tap p_j after k block operations without considering any other taps. Considering all the taps (including p_{t-1}), we will present a recurrence relation of the form

$$I_{k+y} = I^{s}_{k+y} \quad J_{k+y}. \tag{5a}$$

Here J_{k+y} is the contribution of the tap p_{t-1}, which is not the individual contribution of the tap p_{t-1} itself. Rather for generation of J_{k+y} we need to consider the contribution from the other taps also.

We now present an example considering the connection polynomial x^{32} x^{29} x^{11}. Here $n = 32, b = 8, y = 4$. Now $p_1 = 29, q_1 = 3, r_1 = 5, p_0 = 11, q_0 = 1, r_0 = 3$ and $t = 2$. Tap p_1 is in the most significant block Y_3. Initial state of the LFSR is I_3, I_2, I_1, I_0. Now, $I^s_4 = I^{11}_4$. From Equation (5*a*), after one block operation we obtain $I_4 = I^{11}_4 \quad J_4$. We find out the composition of J_4 considering the values of bit positions $Y_{3,5}$ for 8 successive clocks. This is explained in Table 4. The bits corresponding to tap position $Y_{3,5}$ is highlighted in the table. Note that the bit compositions presented below are those *before* the corresponding clock number.

Now, the block operation for obtaining J_4 is to be defined in terms of $I_3, \ldots, I_0$. Note that $J_4 = (I_{4,4}, I_{4,3}, I_{4,2}, I_{4,1}, I_{4,0}, I_{3,7}, I_{3,6}, I_{3,5})$ which are the 8 successive values of $Y_{3,5}$. Using the above table and rearranging the terms

Table 4. Bit composition of most significant block.

Clock	Bit composition of Y_3	New bit
1	$I_{3,7}, I_{3,6}, \underline{I_{3,5}}, I_{3,4}, I_{3,3}, I_{3,2}, I_{3,1}, I_{3,0}$	$I_{4,0} = I_{3,5} \oplus I^{11}_{4,0}$
2	$I_{4,0}, I_{3,7}, \underline{I_{3,6}}, I_{3,5}, I_{3,4}, I_{3,3}, I_{3,2}, I_{3,1}$	$I_{4,1} = I_{3,6} \oplus I^{11}_{4,1}$
3	$I_{4,1}, I_{4,0}, \underline{I_{3,7}}, I_{3,6}, I_{3,5}, I_{3,4}, I_{3,3}, I_{3,2}$	$I_{4,2} = I_{3,7} \oplus I^{11}_{4,2}$
4	$I_{4,2}, I_{4,1}, \underline{I_{4,0}}, I_{3,7}, I_{3,6}, I_{3,5}, I_{3,4}, I_{3,3}$	$I_{4,3} = I_{4,0} \oplus I^{11}_{4,3} = I_{3,5} \oplus I^{11}_{4,0} \oplus I^{11}_{4,3}$
5	$I_{4,3}, I_{4,2}, \underline{I_{4,1}}, I_{4,0}, I_{3,7}, I_{3,6}, I_{3,5}, I_{3,4}$	$I_{4,4} = I_{4,1} \oplus I^{11}_{4,4} = I_{3,6} \oplus I^{11}_{4,1} \oplus I^{11}_{4,4}$
6	$I_{4,4}, I_{4,3}, \underline{I_{4,2}}, I_{4,1}, I_{4,0}, I_{3,7}, I_{3,6}, I_{3,5}$	$I_{4,5} = I_{4,2} \oplus I^{11}_{4,5} = I_{3,7} \oplus I^{11}_{4,2} \oplus I^{11}_{4,5}$
7	$I_{4,5}, I_{4,4}, \underline{I_{4,3}}, I_{4,2}, I_{4,1}, I_{4,0}, I_{3,7}, I_{3,6}$	$I_{4,6} = I_{4,3} \oplus I^{11}_{4,6} = I_{3,5} \oplus I^{11}_{4,0} \oplus I^{11}_{4,3} \oplus I^{11}_{4,6}$
8	$I_{4,6}, I_{4,5}, \underline{I_{4,4}}, I_{4,3}, I_{4,2}, I_{4,1}, I_{4,0}, I_{3,7}$	$I_{4,7} = I_{4,4} \oplus I^{11}_{4,7} = I_{3,6} \oplus I^{11}_{4,1} \oplus I^{11}_{4,4} \oplus I^{11}_{4,7}$

$J_4 = (I^{11}_{4,4}, I^{11}_{4,3}, 0, \ldots, 0) \oplus (I^{11}_{4,1}, I^{11}_{4,0}, I^{11}_{4,2}, I^{11}_{4,1}, I^{11}_{4,0}, 0, 0, 0) \oplus (I_{3,6}, I_{3,5}, I_{3,7}, I_{3,6}, I_{3,5}, I_{3,7}, I_{3,6}, I_{3,5})$. The last term can be rearranged again as $(I_{3,6}, I_{3,5}, 0, \ldots, 0) \oplus (0, 0, I_{3,7}, I_{3,6}, I_{3,5}, 0, 0, 0) \oplus (0, \ldots, 0, I_{3,7}, I_{3,6}, I_{3,5})$ $= (I_{tmp} << 6) \oplus (I_{tmp} << 3) \oplus I_{tmp}$ where I_{tmp} is $(I_3 >> 5)$. Thus we get, $J_4 = (I^{11}_4 << 6) \oplus (I^{11}_4 << 3) \oplus ((I_{tmp} << 6) \oplus (I_{tmp} << 3) \oplus I_{tmp})$. We can rewrite it as, $J_4 = (I^{11}_4 << 2 \cdot 3) \oplus (I^{11}_4 << 1 \cdot 3) \oplus ((I_{tmp} << 2 \cdot 3) \oplus (I_{tmp} << 1 \cdot 3) \oplus I_{tmp})$. In this case $I^s_4 = I^{11}_4$ and we get $J_4 = \bigoplus_{l=1}^{m} (I^s_4 << l \cdot (b - r_1)) \oplus \bigoplus_{l=0}^{m} (I_{tmp} << l \cdot (b - r_1))$, where $m = \lfloor \frac{b}{b - r_1} \rfloor - 1$ and $b = 8, r_1 = 5$. Hence, $I_4 = I^s_4 \oplus J_4 = \bigoplus_{l=0}^{m} (I^s_4 << l \cdot (b - r_1)) \oplus \bigoplus_{l=0}^{m} (I_{tmp} << l \cdot (b - r_1))$.

Now we present the generalized result in the following theorem.

Theorem 3. *Consider a polynomial* $x^n \oplus \bigoplus_{i=0}^{n-1} a_i x^i$ *over GF(2), where* $n = yb$, b *is the block size and* y *is the number of blocks. Let there be* $t = \#\{a_i = 1\}$ *taps such that* $t = t_1 + t_2 + 1$. *Let* $p_0 < p_1 < \ldots < p_{t_1 - 1}$ *are at boundary locations, and* $0 < p_{t_1} < p_{t_1+1} \ldots < p_{t-2} < n - b$ *are at non boundary locations. Also* $n - b + 1 \leq p_{t-1} \leq n - 1$. *Then the block oriented recurrence relation for this LFSR is* $I_{k+y} = \bigoplus_{l=0}^{m} ((I_{tmp} << l \cdot (b - r_{t-1})) \oplus (I^s_{k+y} << l \cdot (b - r_{t-1})))$, *where* $I_{tmp} = (I_{k+y-1} >> r_{t-1})$, $I^s_{k+y} = \bigoplus_{i=0}^{t-2} I^{p_i}_{k+y}$, *and* $m = \lfloor \frac{b}{b - r_{t-1}} \rfloor - 1$.

We now provide a C like algorithm for implementation of this theorem.

Implementation lastBlockTap

```
$m = \lfloor b/(b - r[t-1]) \rfloor - 1$; $k = 0$;
while output is required {
    $var1 = blockLFSROutput()$; /* This is for $I^s_{k+y}$ */
    $var2 = I[k + y - 1] >> r[t - 1]$; /* This is for $I_{tmp}$ */
    $I[k + y] = var1 \oplus var2$; /* This is for $m = 0$ */
    for ($j = 1; j \leq m; j++$) {
        $var1 = var1 << (b - r[t - 1])$;
        $var2 = var2 << (b - r[t - 1])$;
        $I[k + y] = I[k + y] \oplus var1 \oplus var2$;
    }
    Output $I[k]$; $k = k + 1$;
}
```

It is evident that for the single tap in the most significant block $4m + 4$ number of logical operations are required. We consider assignment to different variables as 1 operation. The value of m will vary from a minimum of 1 (when $r_{t-1} \leq b/2$) to a maximum of $b - 1$ (when $r_{t-1} = b - 1$). In case of non boundary tap in the last block, the best case is when $m = 1$. Even for this case, the number of logical operations required is $4 \times 1 + 4 = 8$ which is higher than the logical operations required for a non boundary tap in any other block. Moreover, it is clear that for more than one non boundary taps in the most significan block, the number of required logical operations will be much higher in this technique. For a block-oriented LFSR with taps $t = t_1 + t_2 + 1$, as in Theorem 3, average number of logical operations required for each output bit is $\sigma = \frac{t_1+4t_2+(4m+4)}{b}$ (see Theorem 2). Table 5 shows variation of σ values for different block sizes (b) and different values of m (similar to Table 2). Note that $m = 0$ indicates no non boundary tap in most significant block and in that case $\sigma = \frac{t_1+4t_2}{b}$.

Table 5. σ for different cases ($n = 128$).

Taps (t)	m	$b = 8$			$b = 16$			$b = 32$		
		t_1	t_2	σ	t_1	t_2	σ	t_1	t_2	σ
24	0	16	8	6.000	8	16	4.500	4	20	2.625
24	1	16	7	6.500	8	15	4.750	4	19	2.750
24	2	16	7	7.000	8	15	5.000	4	19	2.875
24	3	16	7	7.500	8	15	5.250	4	19	3.000

Table 6. Bit generation speed.

Length of LFSR	Block Size	Mbps	Taps			m
			t	t_1	t_2	
128	32	19.40	24	4	20	0
128	32	18.30	24	4	19	1
128	32	42.40	10	2	8	0
128	32	40.30	10	2	7	1
128	32	38.15	10	2	7	2
32	8	9.59	12	4	8	0
32	8	9.10	12	4	7	1

It is clear from Table 5 that for smaller block sizes, presence of a single tap in the most significant block increases the average number of logical operations σ considerably. For $b = 8, 16$, it is better to confine r_{t-1} within $b/2$ (i.e., $m = 1$). For block size 32, the value can be increased upto 20 (i.e., $m = 3$). We provide Table 6 similar to Table 3 to show the variation in bit generation speed for different positions of non boundary tap (i.e., m) in the most significant block.

A further work in this direction could be the design of efficient software implementation for LFSRs of any length.

References

1. C. Ding, G. Xiao, and W. Shan. *The Stability Theory of Stream Ciphers*. Number 561 in Lecture Notes in Computer Science. Springer-Verlag, 1991.
2. S. W. Golomb. *Shift Register Sequences*. San Fransisco, CA, Holden-Day, 1967.
3. R. Lidl and H. Niederreiter. *Finite Fields*. Addison Wesley, 1983.

4. S. Maitra and P. Sarkar. Efficient implementation of ciphertext only attack on LFSR based encryption schemes. In *National Seminar on Cryptology*, pages 1–12, July 9-10 1998.
5. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
6. M. Zhang, C. Carrol, and A. Chan. The software-oriented stream cipher SSC2. In *Fast Software Encryption - FSE 2000*, in Lecture Notes in Computer Science. Volume 1978, Springer Verlag, 2001.

Comments on a Signature Scheme Based on the Third Order LFSR Proposed at ACISP2001

Seongan Lim[1], Seungjoo Kim[1], Ikkwon Yie[2,], and Jaemoon Kim[2,]

[1] KISA (Korea Information Security Agency),
5th FL., Dong-A Tower, 1321-6, Seocho-Dong, Seocho-Gu, Seoul 137-070, Korea,
{seongan, skim}@kisa.or.kr
[2] Department of Mathematics, Inha University,
YongHyun-Dong, Nam-Gu, Incheon, Korea,
{ikyie, jmkim}@math.inha.ac.kr

Abstract. In this paper we will compare two signature schemes proposed by different sets of authors. One is the XTR-Nyberg-Rueppel signature proposed by A.K. Lenstra and E.R. Verheul in [3] and the other is the signature scheme proposed by C.H. Tan, X. Yi and C.K. Siew (We will call it TYS signature.) in [9]. XTR-NR signature uses the third degree trace projection $\mathrm{Tr} : \mathrm{GF}(p^6) \to \mathrm{GF}(p^2)$ and has been generalized in [8] by Lim et. al. as a scheme in $\mathrm{GF}(p^{6m})$ using $\mathrm{Tr} : \mathrm{GF}(p^{6m}) \to \mathrm{GF}(p^{2m})$. On the other hand, TYS signature is based on a third order LFSR. Tan et. al. claimed that TYS signature is as secure as Schnorr signature scheme. We will explain why these two schemes are essentially the same. In addition, we will point out that TYS signature as it is has some flaws in their arguments. We will show that in order to cure the flaws of TYS signature, one should bring in exactly the same security and efficiency consideration of XTR scheme as in [8].

Key words: Trace Projection, XTR, LFSR, digital signature scheme

1 Introduction

In [3] A.K. Lenstra and E.R. verheul proposed the public key scheme XTR (which stands for Efficient and Compact Subgroup Trace Representation). As a computational tool, XTR uses the third degree trace projection $\mathrm{Tr} : \mathrm{GF}(p^6) \to \mathrm{GF}(p^2)$. They also showed that XTR can be used to materialize Nyberg-Rueppel message recovery signature scheme. In [8] it was shown by Lim et. al. that XTR can be naturally generalized as a scheme in $\mathrm{GF}(p^{6m})$ using $\mathrm{Tr} : \mathrm{GF}(p^{6m}) \to \mathrm{GF}(p^{2m})$. XTR-NR signature can also be naturally generalized in the same way.

In [9] C.H. Tan, X. Yi and C.K. Siew proposed a signature scheme using third order linear feed back shift registers. These LFSR's are generated by irreducible cubic polynomials of the form $f(x) = x^3 - ax^2 + bx - 1$ over the field $\mathrm{GF}(q)$. These cubic polynomials are assumed to have order $Q = q^2 + q + 1$, i.e., the

Yie and Kim's work was supported by Basic Science Research Institute Program, Korea Research Foundation 1998-015-D00017.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 308–315, 2001.

multiplicative order of a root g GF(q^3) of $f(x)$ is Q. Tan et. al. claimed that the security of their signature scheme is equivalent to the security of Schnorr signature scheme. We will call their signature scheme the TYS signature.

In this paper we will compare XTR-NR signature and TYS signature. Although Tan et. al. didn't refer to the trace projection, we will show that TYS signature can be described in terms of the third degree trace projection $\mathrm{Tr}: \mathrm{GF}(q^3) \to \mathrm{GF}(q)$. Thus XTR and TYS signature will be shown to share the same computational tools. By comparing with XTR, we will show that TYS signature, as it is, is not as secure as Tan et. al. claimed. In order to maintain the security of TYS signature as desired, one must bring in precisely the same conditions for parameters as in XTR. Then it will be apparent that TYS signature is essentially the same as XTR-NR signature scheme. Our conclusions are:

1. Unless the parameters Q and $f(x)$ are chosen very carefully, TYS signature scheme is not as secure as it was claimed to be.
2. XTR scheme can be generalized so that the computation is done over an odd degree extension field GF(q) of GF(p). But in this case we lose much of the computational advantage of XTR we had in an even degree extension.

2 Computational Tools

In this section we briefly describe the computational aspects of XTR and TYS signature. Let p be a prime and $q = p^t$ be a power of p for some positive integer t.

2.1 Description of XTR

XTR starts with an irreducible polynomial of the form $F(c, x) = x^3 - cx^2 + c^{p^m}x - 1$ over GF(q)[x], where $t = 2m$ is assumed to be even and c does not belong to any proper subfield of GF(q). It follows then that the multiplicative order of a root g of $F(c, x)$ is a factor of $p^{2m} - p^m + 1$ and $c = \mathrm{Tr}(g)$ and $c^{p^m} = \mathrm{Tr}(g^{-1})$, where Tr is the trace projection of GF(q^3) onto GF(q).

In fact, one could have started describing XTR by considering irreducible polynomials of the form $f(x) = x^3 - ax^2 + bx - 1$. But once we assume, for security reason, that the order of a root g of $f(x)$ divides $p^{2m} - p^m + 1$, it immediately follows that $a = \mathrm{Tr}(g)$ and $b = \mathrm{Tr}(g^{-1}) = a^{p^m}$.

Now XTR defines basically an Diffie-Hellman type key agreement scheme on the cyclic subgroup G generated by g in the multiplicative group $\mathrm{GF}(q^3)^\times$. But XTR uses Tr(h) to represent the element h G to enhance the computational and communicational efficiency. Also for security concern and practical reason, the order of g is taken as a prime of at least 160 bits.

2.2 Description of TYS

TYS signature starts with a third order LFSR $c_n = ac_{n-1} - bc_{n-2} + c_{n-3}$, $n \geq 3$ generated by a cubic irreducible polynomial of the form $f(x) = x^3 - ax^2 + bx - 1$

over GF(q). The order of $f(x)$, that is, the multiplicative order of a root g of $f(x)$ is assumed to be $Q = q^2 + q + 1$. It follows then that $a = \mathrm{Tr}(g)$ and $b = \mathrm{Tr}(g^{-1})$, where $g \in \mathrm{GF}(q^3)$ is a root of $f(x)$ and $\mathrm{Tr}: \mathrm{GF}(q^3) \to \mathrm{GF}(q)$ is the trace projection. Note that by Newton's formula for elementary symmetric polynomials, we see that $c_n = \mathrm{Tr}(g^n)$ for $n \geq 1$ (See [7]). Thus the set of c_n's is nothing but the set of $\mathrm{Tr}(h)$'s for $h \in G$ as in XTR.

2.3 Computations

Thus in both schemes, $c_n = \mathrm{Tr}(g^n)$ for large n will frequently be computed. To perform these computations efficiently, the following lemmas (Lemma 2.3 of [8] and Algorithm 1 of [9]) were employed:

Lemma 2.1. *Let $q = p^t = p^{2m}$. Let $F(c, x) = x^3 - cx^2 + c^{p^m}x - 1$ be an irreducible polynomial over GF(q) and g be a root of $F(c, x)$ in GF(q^3). If we let $c_n = \mathrm{Tr}(g^n)$, where* Tr *is the obvious trace projection, then we have following formulas.*

1. $c_{n+2} = c_{n+1}c - c_n c^{p^m} + c_{n-1}$, $c_{n-1} = c_{n+2} - c_{n+1}c + c_n c^{p^m}$;
2. $c_{2n} = c_n^2 - 2c_n^{p^m}$;
3. $c_{2n+1} = c_n c_{n+1} - c c_n^{p^m} + c_{n-1}^{p^m}$;
4. $c_{2n-1} = c_n c_{n-1} - c^{p^m} c_n^{p^m} + c_{n+1}^{p^m}$.

Lemma 2.2. *Let $f(x) = x^3 - ax^2 + bx - 1$ be an irreducible polynomial over GF(q) and g be a root of $f(x)$ in GF(q^3). If we let $c_n = \mathrm{Tr}(g^n)$, where* Tr *is the obvious trace projection, then we have following formulas.*

1. $c_{2n} = c_n^2 - 2c_{-n}$;
2. $c_{2n+1} = c_n c_{n+1} - ac_{-n} + c_{-n+1}$;
3. $c_{2n-1} = c_n c_{n-1} - bc_{-n} + c_{-n-1}$.

Note that, in TYS signature, c^{p^m} is meaningless since q may not be a square. However, the formulas in Lemma 2.2 are easily obtained from the formulas in Lemma 2.1 simply by replacing c by a, c^{p^m} by b and $c_n^{p^m}$ by c_{-n}.

Following [3], we denote $S_n(c) = (c_{n-1}, c_n, c_{n+1})$ for any integer n. (Caution: the notation in [9] is slightly different as $S_k = (c_k, c_{k+1}, c_{k+2})$.) Now if we are given $S_0(c), S_1(c), S_2(c)$, by repeatedly applying Lemmas 2.1 or 2.2 as a slight variation of the 'square and multiply' algorithm, we can quickly compute $S_{n-1}(c), S_n(c), S_{n+1}(c)$.

As we have seen so far, XTR and TYS signature have common computational feature. We give Table 1 as a summary of this section.

Remark: In TYS signature, one has to compute both c_n and c_{-n} every time, whereas in XTR, $c_{-n} = c_n^{p^m}$ is for free once c_n is obtained.

If we want to generalize the XTR scheme over the odd degree extension GF(p^t), where t is an odd positive integer, the first problem we face is that we cannot write a third degree polynomial of the form $F(c, x) = x^3 - cx^2 + c^{\sqrt{q}}x - 1$ because q is not a square. Also, $q^2 - q + 1$ is no longer a factor of $q^3 - 1$, the order of

Table 1. Common computational feature

Item	Property	Remark
base field	GF(q)	$q = p^t$. In XTR, $t = 2m$.
polynomial	$f(x) = x^3 - ax^2 + bx - 1 \quad \mathrm{GF}(q)[x]$	In XTR, $a = c$, $b = c^{p^m}$.
splitting field	GF(q^3)	
trace	Tr : GF(q^3) → GF(q)	
ambient group	GF(q^3)$^\times$	
subgroup	the cyclic group G generated by g	g is a root of $f(x)$.
representation	Tr(h)	$h \quad G$
computation	$c_{2n} = c_n^2 - 2c_{-n}$ $c_{2n+1} = c_n c_{n+1} - ac_{-n} + c_{-n+1}$ $c_{2n-1} = c_n c_{n-1} - bc_{-n} + c_{-n-1}$	In XTR, c_{-n} can be easily computed as $c_n^{p^m}$.

the multiplicative group GF(q^3)$^\times$. Instead, we have $q^3 - 1 = (q-1)(q^2+q+1)$. Thus we need to start, as in [9], with an irreducible polynomial of the form $f(x) = x^3 - ax^2 + bx - 1$. If g is a root of $f(x)$, then the norm $N(g)$ of g is $g^{q^2+q+1} = 1$. Therefore the order of $f(x)$ automatically becomes a factor of $q^2 + q + 1$. Now the rest of computational setup will follow exactly in the same way as above.

3 Signaure Schemes

XTR-NR signature is a message recovery signature scheme and TYS signature is a signature scheme with appendix. But as was remarked in [3], signature scheme with appendix using XTR can be defined in the same way. We compare the XTR-NR signature and TYS signature and display the comparison as Table 2. Note that t is assumed to be an even number $t = 2m$ for XTR.

4 Security of TYS Signature

Both XTR and TYS signature have their security based on DLP. Since every computation can be performed inside the splitting field GF(q^3), the security level of TYS signature is at best equivalent to that of DLP in GF(q^3). Also, since the only elements we deal with are elements of the cyclic subgroup G generated by a root g of $f(x) = x^3 - ax + bx - 1$, the security level of TYS signature is at best equivalent to that of DLP in G. Based on this observation, we discuss below the security of TYS signature scheme.

4.1 If the Polynomial $f(x) = x^3 - ax^2 + bx - 1$ is Defined over a Proper Subfield of GF(q)

If the subgroup G can be caught inside a proper subfield K of GF(q^3), then the security level will go down to the security level of DLP of K. This case occurs

Table 2. XTR-NR and TYS signature schemes

Item	XTR	TYS signature
Trace	$\mathrm{Tr}: \mathrm{GF}(p^{6m}) \to GF(p^{2m})$	$\mathrm{Tr}: \mathrm{GF}(p^{3t}) \to \mathrm{GF}(p^{t})$
irr. poly.	$f(x) = x^3 - ax^2 + a^{p^m}x - 1$, $a \quad \mathrm{GF}(p^{2m})$ but not in any proper subfield	$f(x) = x^3 - ax^2 + bx - 1$, $a, b \quad \mathrm{GF}(p^{t})$
a	$a = Tr(g)$ for some $g \quad \mathrm{GF}(p^{6m})$	$a = Tr(g)$ for some $g \quad \mathrm{GF}(p^{3t})$
Order of g	Prime factor Q of $p^{2m} - p^m + 1$	$Q = p^{2t} + p^t + 1$
Secret key k	$1 < k < Q - 2$	$\gcd(k, Q) = 1$
Nonce z	$1 < z < Q - 2$	$\gcd(z, Q) = 1$
Message Auth.	$h = Hash(E_K(m))$, E: agreed symmetric cipher, $K = Tr(g^z)$	$h_1 = Hash(m, s_k, s_{k+1}, s_{k+2})$
s	$s = hk + z \pmod{Q}$	$s = h_1k - z \pmod{Q}$
Sig. for m	$(s, E_K(m))$	$(s, m, s_k, s_{k+1}, s_{k+2})$

exactly when the polynomial $f(x) = x^3 - ax^2 + bx - 1$ is defined over a proper subfield of $\mathrm{GF}(q)$. Tan et. al. in [9] gave this condition implicitly by requesting the order of $f(x)$ to be $Q = q^2 + q + 1$. But requesting the order to be exactly $q^2 + q + 1$ makes it difficult to find the polynomial $f(x)$. Also the vague role of Lemma 1 of [9] and the example below it may give wrong impression that one may even choose $f(x)$ over the prime subfield $\mathrm{GF}(p)$.

4.2 If the Order Q of the Subgroup Factors into Small Primes

One of the known attacks on DLP is Pohlig-Hellman algorithm which is designed to work when the order of the group used is a product of small primes. Hence we should be careful in choosing the parameter Q so that it has a large prime factor. With respect to the current computing ability, it is usually required that Q must have a prime factor of at least 160 bits. When one construct a cryptographic scheme based on the subgroup DLP, it is a common practice to make the order Q of the subgroup to be a prime of at least 160 bits because of efficiency and security consideration.

Thus, in order to make TYS signature secure as claimed, it is required to impose the condition that Q have a prime factor B of at least 160 bits. Unfortunately, if the order of the base field $\mathrm{GF}(q)$ is a square $q = p^{2m}$, this condition is not enough. In that case, we have $q^2 + q + 1 = (p^{2m} - p^m + 1)(p^{2m} + p^m + 1)$. And if B is a factor of $(p^{2m} + p^m + 1)$, then the subgroup of $\mathrm{GF}(q^3)^\times$ of order B is in fact contained in the proper subfield $\mathrm{GF}(p^{3m})$ of $\mathrm{GF}(q^3)$. Therefore, if $q = p^{2m}$ is a square, the condition should be stronger so that $(p^{2m} - p^m + 1)$ have a prime factor of at least 160 bits.

5 Parameter Generation for TYS Signature

Since Tan et. al. do not provide any method of generating parameters and since we need to add some restrictive conditions on parameters, we discuss how to generate parameters for TYS signature. The discussions about computation in Section 2 and about security in Section 4 show us that the parameter selection should be different according as whether $q = p^t$ is a square or not. So we deal these two cases separately. Note also that the order Q of the polynomial $f(x)$ doesn't have to be $q^2 + q + 1$ but is enough to be a prime factor of $q^2 + q + 1$ or $p^{2m} - p^m + 1$ of at least 160 bits according as t is odd or even.

5.1 Parameter Generation when t Is Even

In case $q = p^{2m}$ is a square, Q is a prime factor of $p^{2m} - p^m + 1$ and the computational detail of TYS signature becomes exactly the same as XTR. Therefore we can generate parameters the same way as we would in [8] for XTR.

5.2 Parameter Generation when t is Odd

In case $q = p^t$ is not a square, the description of XTR should be modified as we noted at the end of Section 2. In fact, if we fix the flaws reported so far, the computational feature of TYS signature can readily serve as the generalization of XTR over the odd degree extention field GF(q).

Selection of $q = p^t$. Since the security level of TYS signature (or XTR) is bounded by the security level of DLP in the splitting field GF(q^3) and of DLP in the subgroup G, q^3 must be selected so that these DLP's are secure. Currently, q must be at least 1024 bits and the order Q of G, which is a prime factor of $q^2 + q + 1$, must be at least 160 bits.

If $t > 1$, since every computation will be done over GF(p^t), it is desirable to have efficient arithmetic in GF(p^t). Therefore, we need to select $q = p^t$ so that GF(p^t) has good bases. Note that if $t > 1$ is odd, GF(p^t) never has an optimal normal basis of type I. Not much is known about optimal normal bases of type II of extension fields with characteristic $p > 2$. Study on a good optimal normal basis of type II that is well suited for XTR-TYS scheme would be an interesting subject.

Let us let $t = 2m + 1$. We will consider only the case when GF(p^t) has an optimal normal basis of type II. In this case, $2t + 1 = 4m + 3$ is a prime number. We further assume that $t = 2m + 1$ is a prime so that Z_{4m+3} has as many primitive elements as possible. Thus we made the situation similar to the case when t is even.

We also need to construct the subgroup G of GF(p^{3t}) = GF(p^{6m+3}) of order Q so that G is not contained in any proper subfield of GF(p^{3t}) = GF(p^{6m+3}). The following Lemma, which follows directly from Lemma 2.4 of [2], gives a sufficient condition for a subgroup of GF(p^{6m+3}) not to be contained in any proper subfield of GF(p^{6m+3}).

Lemma 5.1. *Let Q be a prime factor of $\Phi_{6m+3}(p)$, where $\Phi_n(X)$ denotes the n-th cyclotomic polynomial. Then the subgroup of* GF(p^{6m+3}) *of order Q is not contained in any proper subfield of* GF(p^{6m+3}).

The 3(2m + 1)-th cyclotomic polynomial $\Phi_{6m+3}(x)$ in GF(p) are as follows:

- $\Phi_9(x) = x^6 + x^3 + 1$
- $\Phi_{6m+3}(x) = \frac{x^{4m+2}+x^{2m+1}+1}{x^2+x+1} = \frac{x^{2t}+x^t+1}{x^2+x+1}$, $m > 1$.

Thus we have that if Q is a prime factor of $p^{2t} + p^t + 1$ of at least 160 bits (assuming that p^{3t} is of 1024 bits) then the subgroup g of GF(p^{3t}) with g GF(p^{3t}) of order Q is not contained in any proper subfield of GF(p^{3t}).

Unfortunately, when $t > 1$, there is no known easy way of constructing such p, Q pair satisfying above conditions. One has to keep constructing prime numbers p until $p^{2t} + p^t + 1$ has a prime factor Q of at least 160 bits.

Selection of the Cubic Polynomial $f(x) = x^3 - ax^2 + bx - 1$. As was noted at the end of Section 2, the irreducibility of $f(x) = x^3 - ax^2 + bx - 1$ over GF(q) implies that the order of $f(x)$ is a factor of $q^2 + q + 1$. However, it is very difficult to deterministically compute b from a given a, or vice versa, so that $f(x)$ is irreducible. We give two ways of finding such (a, b) pairs, one of which is much preferable.

One conceptually easier way is to start with an element g GF(q^3) whose norm in GF(q) is 1. Then a = Tr(g) and b = Tr(g^{-1}). But when GF(q) is not a prime field (i.e., $t > 1$), it is not easy to compute the trace.

Another way of generating $f(x)$ is to randomly choose (a, b) pair and test whether $x^3 - ax^2 + bx - 1$ is irreducible. The probability for $x^3 - ax^2 + bx - 1$ to be irreducible for randomly chosen (a, b) is about 1/3.

6 Conclusion

In this paper we compared XTR-NR signature scheme and TYS signature scheme. We explained why these two schemes are essentially the same. In addition, we pointed out that TYS signature as it is has some flaws in their arguments and shown that in order to cure the flaws of TYS signature, one should bring in exactly the same security and efficiency consideration of XTR scheme as in [8].

As a summary we list the following:

- To make TYS signature scheme secure, the order Q of the irreducible polynomial $x^3 - ax^2 + bx - 1$ must have a prime factor B of at least 160 bits. The prime B should be a factor of $p^{2t} + p^t + 1$ when t is odd, and should be a factor of $p^{2m} - p^m + 1$ when $t = 2m$ is even.
- When t is odd, the cost of computation as well as communication is almost twice of that of t even case. Also if t is odd, we lose the advantage of XTR of easy generation of the cubic polynomial $F(c, x)$.

- Thus for efficiency reason, it is desirable to take t to be even. Also for security reason, it is desirable to take the order Q of the cubic polynomial to be a prime factor of $p^{2m} - p^m + 1$ of at least 160 bits, where $t = 2m$. Then TYS signature scheme is practically a variation of XTR.

References

1. A. E. Brouwer, R. Pellikaan, Eric R. Verheul, *Doing More with Fewer Bits*, Advances in Cryptology – Asiacrypt'99, **LNCS 1716** (1999), pp. 321–332.
2. Arjen K. Lenstra, *Using Cyclotomic Polynomials to Construct Efficient Discrete Logarithm Cryptosystems over Finite Fields*, ACISP'97 (1997), **LNCS 1270**, pp. 127–138.
3. Arjen K. Lenstra, Eric R. Verheul, *The XTR public key system*, Advances in Cryptology – CRYPTO'00 **LNCS 1880** (2000), pp. 1–19
4. Arjen K. Lenstra, Eric R. Verheul, *Key improvements to XTR*, Advances in Cryptology – Asiacrypt'00 **LNCS 1976** (2000), pp. 220–233
5. Arjen K. Lenstra, Eric R. Verheul, *Selecting Cryptographic Key Sizes*, http://www.cryptosavvy.com (1999).
6. Arjen K. Lenstra, Eric R. Verheul, *Fast irreduciblility and subgroup membership testing in XTR*, Proceedings of the PKC'01 **LNCS 1992** (2001), pp. 73–86
7. Rudolf Lidl, Harald Niederreiter, *Introduction to finite fields and their applications*, Cambridge, 1994.
8. Seongan Lim, Seungjoo Kim, Ikkwon Yie, Jaemoon Kim and Hongsub Lee, *XTR Extended to GF*(p^{6m}), to appear at SAC'01, **LNCS** (2001).
9. Chik How Tan, Xun Yi and Chee Kheong Siew, *Signature Schemes Based on 3rd Order Shift Registers*, ACISP'01, **LNCS 2119** (2001), pp.445 – 459.

Pseudo-random Bit Generator Based on Couple Chaotic Systems and Its Applications in Stream-Cipher Cryptography

Li Shujun, Mou Xuanqin, and Cai Yuanlong

Institute of Image Processing, School of Electronics and Information Engineering,
Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P. R. China,
hooklee@mail.com, {xqmou,ylcai}@xjtu.edu.cn

Abstract. Chaotic cryptology is widely investigated recently. This paper reviews the progress in this area and points out some existent problems in digital chaotic ciphers. As a comprehensive solution to these problems, a novel pseudo-random bit generator based on a couple of chaotic systems called CCS-PRBG is presented. Detailed theoretical analyses show that it has perfect cryptographic properties, and can be used to construct stream ciphers with higher security than other chaotic ciphers. Some experiments are made for confirmation. Finally, several examples of stream ciphers based on digital CCS-PRBG are given, and their security is discussed.

1 Introduction

Chaotic cryptography has received much attention in recent years, both digital and analog chaotic encryption methods have been proposed and analyzed [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]. Most analog chaotic ciphers are designed to realize secure communications through noisy channel using chaotic synchronization technique [1]. This paper chiefly focuses on the digital chaotic ciphers.

The tight relationship between chaos theory and cryptography has been pointed out by some researchers [2, 1, 16, 31]. Many fundamental characteristics of chaos, such as mixing and sensitivity to initial conditions, can be connected with those of good ciphers, such as confusion and diffusion. Since chaos theory has developed well in recent decades, and numerous chaotic systems can be employed in ciphers, chaos should be a new rich source of cryptography.

Generally speaking, there are two chief ways to design digital chaotic ciphers: 1) using chaotic systems to generate pseudo-random keystream to encrypt plaintext [3, 5, 6, 7, 8, 10, 11, 12]; 2) using plaintext and/or secret key as the initial conditions and/or control parameters, iterating/counter-iterating chaotic systems n times to obtain ciphertext [2, 9, 13, 14, 15, 16]. The first way corresponds to the stream ciphers and the second does to the block ciphers. Some other ways also have been proposed [17, 18, 19]. Meanwhile, some efficient attacks have been presented [20, 21, 22, 23, 24, 25]. In the following of this section, we will give a brief survey of the proposed digital chaotic ciphers, and discuss some problems existing in them.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 316–329, 2001.

1.1 Overview

Digital chaotic stream ciphers: Many different chaotic systems have been employed to generate pseudo-random keystream, 2-D Hénon attractor in [3], logistic map in [10], generalized logistic map in [6], quasi-chaotic nonlinear filter in [7], piecewise linear chaotic map in [4,5,8,19], and first-order nonuniformly sampling digital phase-locked loop (DPLL) circuits in [11]. In [12] multiple different chaotic maps are suggested, Bernoulli shift and logistic map are used for demonstration. The algorithms generating chaotic pseudo-random keystreams can be divided into three classes: A1) – extracting from some bits of the chaotic orbits [4,5,6,12]; A2) – determining by which interval the chaotic orbits reach [3,8,10,11]; A3) – just equaling the chaotic orbits themselves [7]. It should be noticed that some algorithms in A2) [8,10,11] can be considered as the corresponding ones in A1), and A3) can be deemed as a special case of A1). Several chaotic stream ciphers [3,6,7] have been known not secure enough [20,21,22,23].

Digital chaotic block ciphers: Inverse tent map is used by T. Habutsu et al. in a chaotic cryptosystem [13], in which the plaintext represents the initial condition of the inverse tent map and the ciphertext is obtained by iterating this map N times. Because of the weakness of piecewise linearity of tent map and the use of 75 random bits, E. Biham presented a known-plaintext attack and a chosen-plaintext attack to break it [24]. Zbigniew Kotulski and Janusz Szczepanski generalized the method presented in [13] using other chaotic systems [15,14]. In Jiri Fridrich's chaotic cipher [16], 2-D digital Barker map is introduced to realize secure pseudo-random permutation of 2-D plaintext such as digital images. A discrete version of chaotic inverse system encryption approach is presented by Zhou Hong et al. in [9].

Other digital chaotic ciphers: M. S. Baptista suggested a new encryption method in [17]: a chaotic attractor is divided into S units representing different plaintexts, the ciphertext is the number of iteration from an initial value to the unit representing the plaintext, logistic map is used for demonstration. In [18], such an idea is introduced: run a chaotic system, and use a threshold to generate a pseudo-random sequence from its orbit, find the position that plaintext occurs in the sequence and take the corresponding information about the position as the ciphertext, tent map is used as an example. G. Alvarez et al. pointed out that it is not secure at all if the tent map is used [25]. Li Shujun et al. improve the original chaotic cryptosystem to resist the proposed attacks [19].

1.2 Problems

Although many digital chaotic ciphers have been proposed and some of them have not been confronted with effective attacks, there are still many problems existing in them. To design a really good digital chaotic cipher, they must be carefully considered. The following is brief discussions on these problems:

1) Discrete Dynamics: When chaotic systems are realized discretely in finite computing precision, their discrete dynamics will be far different from continuous ones. Some severe degradation will arise, such as short cycle-length, non-ideal distribution and correlation, etc. This problem has been firstly noticed by J. Palmore, C. Herring [32] and D. Wheeler [21, 22], and then Ghobad Heidari-Bateni [33]. Up till now, there is not an established theory to measure the discrete dynamics of chaos exactly, and to indicate how to improve such degradation (we have proved some limited theoretical results in [34] recently). Only several engineering methods are suggested: using higher finite precision [21, 22], perturbation-based algorithm [4, 5, 35], and cascading multiple chaotic systems [33]. Actually, this problem is neglected in most digital chaotic ciphers [3, 9, 8, 10, 11, 12, 13, 14, 15, 17, 18], so their security cannot be adequately ensured.

2) Employed Chaotic Systems: Because logistic map has been widely investigated in chaos theory and is very simple to be realized, it has been used by some digital chaotic ciphers [6, 10, 12, 17]. However, only when control parameters r is 4.0, logistic map is a surjective function and has perfect chaotic properties. So r must be selected near 4.0 in these ciphers, which makes the key space much smaller. Other good candidates for simple realization are piecewise linear chaotic maps, such as tent map [13, 18] and the ones used in [4, 5, 9, 8, 19]. But we must be very careful to use them since there exist some weaknesses for their piecewise linearity [24, 25, 34]. In fact, it is desired that a digital chaotic cipher can work well with a large number of chaotic systems; such a property is called *chaotic-system-free* in this paper. Several chaotic ciphers are chaotic-system-free to some extent [12, 15, 17, 18]. Some others can be chaotic-system-free since different chaotic systems are not essentially excluded by their design [10, 11].

3) Encryption Speed: Some digital chaotic ciphers work so slowly that they are infeasible for real-time encryption [13, 14, 15, 17, 18, 19]. While the chaotic systems are running in finite precision, the floating-point or fixed-point arithmetic must be employed. Since the floating-point arithmetic is much slower than the fixed-point one, we suggest using fixed-point arithmetic as possible. But several chaotic systems defined by some complicated functions [6, 15] must run under floating-point arithmetic, they should be avoided in chaotic ciphers. The piecewise linear chaotic maps are the fastest chaotic systems, since only one division and several additions are needed in one iteration. Another problem about the encryption speed is: in order to enhance security, many ciphers need multiple chaotic iterations to generate one ciphertext [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19], which will lower the encryption speed. In addition, some ciphers [17, 18, 19] have time-variant speed, so they cannot encrypt plaintext with constant bit-rate, such as MPEG video stream.

4) Practical Security: Most digital chaotic ciphers are claimed to be secure by the authors, but many of them are actually not. Because chaotic systems are deterministic systems, there are some tools in chaos theory to discern chaos. Once an intruder finds some information about the chaotic systems from their orbits, he might use such information to lessen the complexity of finding the

secure key. For almost all digital chaotic ciphers [2,1,3,4,5,6,7,9,8,10,11,13,14, 15,16,17,18,19], the ciphertext directly depends on the chaotic orbit of a single chaotic system, so the extraction of such information may be possible. In fact, based on such a fact, many cryptanalysis methods [26,27,28,29,30] have been developed to break the analog secure communication approaches. If multiple chaotic systems are used [12,33], the cryptanalysis of chaotic ciphers will be more difficult since the output is determined by many different mixed chaotic orbits.

5) Realization: Simple realization by hardware and software at low cost is a very important requirement for a good digital cipher. In consideration of the above fact, the fixed-point arithmetic is better than the floating-point one since the latter needs more cost. Another desired requirement is the extensible security with considerably more cost and complexity. In fact, problems of realization are the crucial factors influencing the use of a cipher in many final applications, since there are so many kinds of ciphers that can provide enough security.

Although many problems have not been settled in most digital chaotic ciphers, we still believe that the chaotic and conventional cryptology will benefit each other from the mutual relationship between them; some other researchers hold the same opinion [2,1,16,31]. In this paper, we suggest a comprehensive solution to the existent problems. A novel pseudo-random bit generator (PRBG) based on a couple of chaotic systems, called CCS-PRBG, is presented, which has perfect cryptographic properties and can be used to construct stream ciphers with high security. In these ciphers, most above-mentioned problems can be overcome satisfactorily.

The outline of this paper is as follows. In Sect. 2, CCS-PRBG and its digital realization with finite precision are introduced. Analyses on cryptographic properties of CCS-PRBG, including some experimental results, are given in Sect. 3. In Sect. 4, several examples of chaotic stream ciphers based on CCS-PRBG are established; discussion on the security is also given. The conclusion is given and some open research topics are pointed out in the last section.

2 Couple Chaotic Systems Based PRBG (CCS-PRBG)

As mentioned in Sect. 1, using chaos to generate pseudo-random numbers (PRN) is a general way to design digital chaotic stream ciphers. Besides in chaotic cryptography area, chaotic pseudo-random number generators (PRNG) have also attracted much attention in other research areas, such as communications [36,33, 37] and physics [38]. Most chaotic PRNG-s are based on single chaotic system and generate PRN directly from its orbit. In Sect. 1.2, we have discussed that such chaotic PRNG-s are potentially insecure, since the output PRN may expose some information about chaotic systems. In this paper, we present a novel pseudo-random bit generator (PRBG) based on a couple of chaotic systems, which can provide higher security than other ciphers because two chaotic systems are employed to generate PRN. Here, we call it CCS-PRBG as abbreviation. Since the PRN is generated by comparing two different chaotic orbits, it is difficult for an eavesdropper to extract information about both chaotic systems. More detailed discussions on security will be given in Sect. 4, after some chaotic stream ciphers based on CCS-PRBG are described.

2.1 Definition

Assume there are two different one-dimensional chaotic maps $F_1(x_1, p_1)$ and $F_2(x_2, p_2)$: $x_1(i+1) = F_1(x_1(i), p_1)$, $x_2(i+1) = F_2(x_2(i), p_2)$, where p_1, p_2 are control parameters, $x_1(0), x_2(0)$ are initial conditions, and $\{x_1(i)\}$, $\{x_2(i)\}$ denote the two chaotic orbits.

Define a pseudo-random bit sequence $k(i) = g(x_1(i), x_2(i))$, where

$$g(x_1, x_2) = \begin{cases} 1, & x_1 > x_2 \\ \text{no outut}, & x_1 = x_2 \\ 0, & x_1 < x_2 \end{cases} . \qquad (1)$$

When some requirements are satisfied, the chaotic PRBG will have perfect cryptographic properties and be called "a Couple of Chaotic Systems based Pseudo-Random Bit Generator" (CCS-PRBG). These requirements are: *R1*) – $F_1(x_1, p_1)$ and $F_2(x_2, p_2)$ are surjective maps defined on a same interval $I = [a, b]$; *R2*) – $F_1(x_1, p_1)$ and $F_2(x_2, p_2)$ are ergodic on I, with unique invariant density functions $f_1(x)$ and $f_2(x)$; *R3*) – One of the following conditions holds: $f_1(x) = f_2(x) = f(x)$, or $f_1(x)$, $f_2(x)$ are both even symmetrical to $x = (a+b)/2$; *R4*) – $\{x_1(i)\}, \{x_2(i)\}$ are asymptotically independent as $i \to \infty$.

If one of chaotic map is replaced by a constant $c \in I$, $k(i)$ will be simplified to the pseudo-random sequence in [11] and the chaotic threshold sequence in [36]. From such a viewpoint, CCS-PRBG can be regarded as the generalized version of them with "pseudo-random and time-variant threshold parameter" [1].

2.2 Digital Realization with Perturbation

It is obvious that CCS-PRBG can be applied to both analog and digital chaotic ciphers. We will only consider digital CCS-PRBG in this paper. The perturbation-based algorithm in [4] is suggested improving statistical properties of digital CCS-PRBG. The algorithm can be described as follows.

Use two PRNG-s to generate two pseudo-random distributed signals [2], which are used to perturb l lowest bits of $\{x_1(i)\}, \{x_2(i)\}$, with intervals Δ_1, Δ_2 [4]. The maximal length linear feedback shift registers (m-LFSR) are the best perturbing PRNG-s for hardware realization, and the linear congruential generators for software realization [39]. Different from [4], this paper suggests determining l as follows: $l \geq \lceil \lambda \cdot \log_2 e \rceil = \lceil 1.44\lambda \rceil$, where λ is Lyapunov exponent of the perturbed chaotic map and $\lceil x \rceil$ denotes the least integer not less than x. It is based on such a

[1] $g(x_1, x_2)$ can be considered as follows: one chaotic orbit is binarized by anther chaotic orbit, the second chaotic orbit behaves like the threshold constant in [36, 11].

[2] Please see [4] for more details on how to generate the perturbing signals. Of course, we can use some other generation algorithms, the only requirement is that the generated signals should be pseudo-randomly distributed.

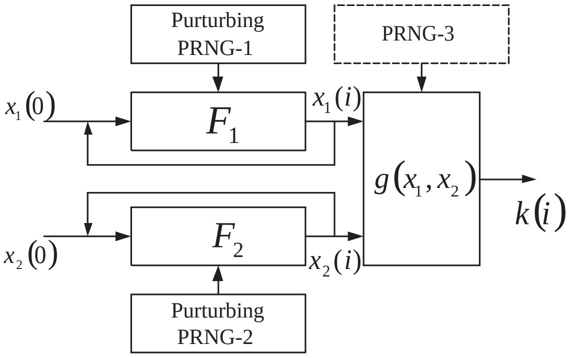


Fig. 1. The digital CCS-PRBG with perturbation

fact: when the finite computing precision is n (bits), the least difference between two signals 2^{-n} will become $e^{\lambda} \cdot 2^{-n}$ after one iteration averagely (under fixed-point arithmetic). To keep the characteristics of the chaotic systems, $l \quad n$ should also be satisfied. Although the perturbing signal is much smaller than chaotic signal, it can still drive $\{x_1(i)\}$, $\{x_2(i)\}$ to a very complex way since chaos is sensitive to initial conditions. The combination of digital chaos and pseudo-randomness of PRNG-s will make both chaos-theory-based and conventional cryptanalysis difficult.

Another trivial problem existing in digital CCS-PRBG is: when $x_1 = x_2$, $g(x_1, x_2)$ will not output pseudo-random bit. An extra simple PRNG-3 can be introduced to determine $k(i)$. The digital CCS-PRBG with perturbation is shown in Fig. 1. We can see that it can be easily realized by both hardware and software.

3 Cryptographic Properties of Digital CCS-PRBG

For $\{k(i)\}$ generated by digital CCS-PRBG, the following cryptographic properties are satisfied: 1) balance on $\{0, 1\}$; 2) long cycle-length; 3) high linear complexity approximating to half of the cycle-length; 4) δ-like auto-correlation; 5) cross-correlation near to zero; 6) chaotic-system-free (see Sect. 1.2). Detailed discussions are given as follows, with some experimental results.

3.1 Balance

Theorem 1. *If two chaotic maps satisfy the above requirement R1–R4, we can get $P\{k(i) = 0\} = P\{k(i) = 1\}$, i.e., $k(i)$ is balanced on $\{0, 1\}$.*

Proof. Because $F_1(x_1, p_1)$ and $F_2(x_2, p_2)$ are ergodic on $I = [a, b]$ (requirement *R2*), the orbits generated from almost all initial conditions will lead to the same distribution functions $f_1(x)$, $f_2(x)$ [40]. From requirement *R4*, the orbits $\{x_1(i)\}$, $\{x_2(i)\}$ are asymptotically independent, so the probabilities of $x_1 > x_2$ and $x_1 < x_2$ as $i \to \infty$ will be:

$$P\{x_1 > x_2\} = \int_a^b \int_a^x f_1(x) f_2(y)\,\mathrm{d}y\,\mathrm{d}x \tag{2}$$

$$P\{x_1 < x_2\} = \int_a^b \int_a^x f_2(x) f_1(y)\,\mathrm{d}y\,\mathrm{d}x \tag{3}$$

When requirement $R3$ holds, we can prove $P\{x_1 > x_2\} = P\{x_1 < x_2\}$:
$R3$–1) $f_1(x) = f_2(x) = f(x)$:

$$P\{x_1 > x_2\} = P\{x_1 < x_2\} = \int_a^b \int_a^b f(x) f(y)\,\mathrm{d}y\,\mathrm{d}x. \tag{4}$$

$R3$–2) $f_1(x), f_2(x)$ are both even symmetrical to $x = (a + b)/2$:
Define the mirror orbits of x_1, x_2 as $x_1 = b - x_1, x_2 = b - x_2$. From the symmetry of $f_1(x), f_2(x)$, x_1, x_2 will have the same distribution $f_1(x), f_2(x)$, then we have:

$$P\{x_1 > x_2\} = P\{x_1 < x_2\} = \int_a^b \int_a^{x'} f_2(x\,) f_1(y\,)\,\mathrm{d}y\,\mathrm{d}x = P\{x_1 < x_2\}. \tag{5}$$

Consider $x_1 > x_2 \rightarrow k(i) = 1$ and $x_1 < x_2 \rightarrow k(i) = 0$, $P\{x_1 > x_2\} = P\{x_1 < x_2\} \quad P\{k(i) = 0\} = P\{k(i) = 1\}$. The proof is complete.

Apparently, the above deduction is still based on the continuous conditions. When chaotic systems are discretely realized with perturbation, every chaotic orbit will be perturbed timely to a certain neighbor orbit by the small perturbing signal. Consequently, almost all orbits reach to the discrete versions of $f_1(x), f_2(x)$ with a little smoothing. For the discrete versions of $f_1(x), f_2(x)$, the above deduction also holds if $\int$ is replaced by $\sum$[3]. Therefore, the balance will be approximately preserved in the digital CCS-PRBG with perturbation.

3.2 Long Cycle-Length

When the ergodic chaotic systems are realized continuously, the cycle-length will be infinite for the orbit beginning at almost every initial condition [40]. However, as we have pointed out in Sect. 1, when they are discretely realized with finite precision, the short cycle-length problem will arise. Employing perturbation can solve this problem. Without loss of generality, assume two m-LFSR-s are used as the perturbing PRNG-s, whose degrees are L_1, L_2, and perturbing intervals are Δ_1, Δ_2. Then the cycle-length of $x_1(i)\}, \{x_2(i)\}$ are $\sigma_1 \Delta_1 (2^{L_1} - 1), \sigma_2 \Delta_2 (2^{L_2} - 1)$, where σ_1, σ_2 are two positive integers [4]. So the cycle-length of $\{k(i)\}$ will be:

$$\mathrm{lcm}(\sigma_1 \Delta_1 (2^{L_1} - 1), \sigma_2 \Delta_2 (2^{L_2} - 1)). \tag{6}$$

[3] Equation (2) and (3) are replaced by $P\{x_1 > x_2\} = \sum_{x=a}^{b} \sum_{y=a}^{x} P_1\{x_1 = x\} \cdot P_2\{x_2 = y\}$ and $P\{x_2 > x_1\} = \sum_{x=a}^{b} \sum_{y=a}^{x} P_2\{x_1 = x\} \cdot P_1\{x_2 = y\}$. From the approximate symmetry to $x = 1/2$ of x_1, x_2 when a digital CCS-PRBG is realized with perturbation, we can obtain the following result $P\{x_1 > x_2\} \approx P\{x_1 < x_2\}$.

When Δ_1, Δ_2 and L_1, L_2 are selected to satisfy $\gcd(\Delta_1, \Delta_2)=1$ and $\gcd(2^{L_1}-1, 2^{L_2}-1)=1$, the cycle-length of $\{k(i)\}$ will be:

$$\mathrm{lcm}(\sigma_1, \sigma_2) \cdot \Delta_1\Delta_2(2^{L_1} - 1)(2^{L_2} - 1) \approx \mathrm{lcm}(\sigma_1, \sigma_2) \cdot \Delta_1\Delta_2 2^{L_1+L_2}. \qquad (7)$$

Such a cycle length is long enough for most secure applications. Furthermore, there are still some methods that can be used to further prolong the cycle length, such as the one in [5].

3.3 High Linear Complexity and Good Correlation Properties

Actually, the requirement *R4* and the balance of $\{k(i)\}$ imply that $\{k(i)\}$ is an independent and identically distributed (i.i.d.) bit sequence as $i \to \infty$. Therefore, it will have δ-like auto-correlation and near-to-zero cross-correlation. What's more, it has been proved (see [41]) that i.i.d. binary sequence has half-length linear complexity, so $\{k(i)\}_{i=1}^{n}$ will also have high linear complexity approximating to $n/2$ [4]. So let us discuss under what condition requirement *R4* will be satisfied for digital CCS-PRBG.

For any chaotic maps, even if the initial conditions or the control parameters have a very small difference, their orbits will become entirely different after limited iterations. If there is some initial information about the orbits, the information will decrease to zero as $i \to \infty$. The relation between two chaotic orbits can be considered as such information. In chaos theory, Kolmogorov entropy is defined to measure the decreasing rate of the information. For one-dimensional chaotic maps, Kolmogorov entropy is equal to Lyapunov exponent [42]. If the initially known information is H, it will lose completely after $\eta \approx H/\lambda$ iterations [11], where λ is Lyapunov exponent. When chaotic systems are realized discretely, the information will decrease even faster since the quantization errors and small perturbing signals makes two orbits depart faster. So we can see, as long as there is initial difference between two chaotic orbits, they will become asymptotically independent as $i \to \infty$. Therefore, the equivalent requirement of *R4* is $\{x_1(i)\} = \{x_2(i)\}$, that is to say, $F_1 = F_2$, or $x_1(0) = x_2(0)$, or $p_1 = p_2$.

Because the independence of $\{x_1(i)\}$, $\{x_2(i)\}$ holds after η iterations, we suggest discarding the first m bits of $\{k(i)\}$, where $m > \eta$. It means m pre-iterations for the two chaotic maps should be done before $\{k(i)\}$ is output. Since m is not very large, such pre-iterations need only a little extra computation.

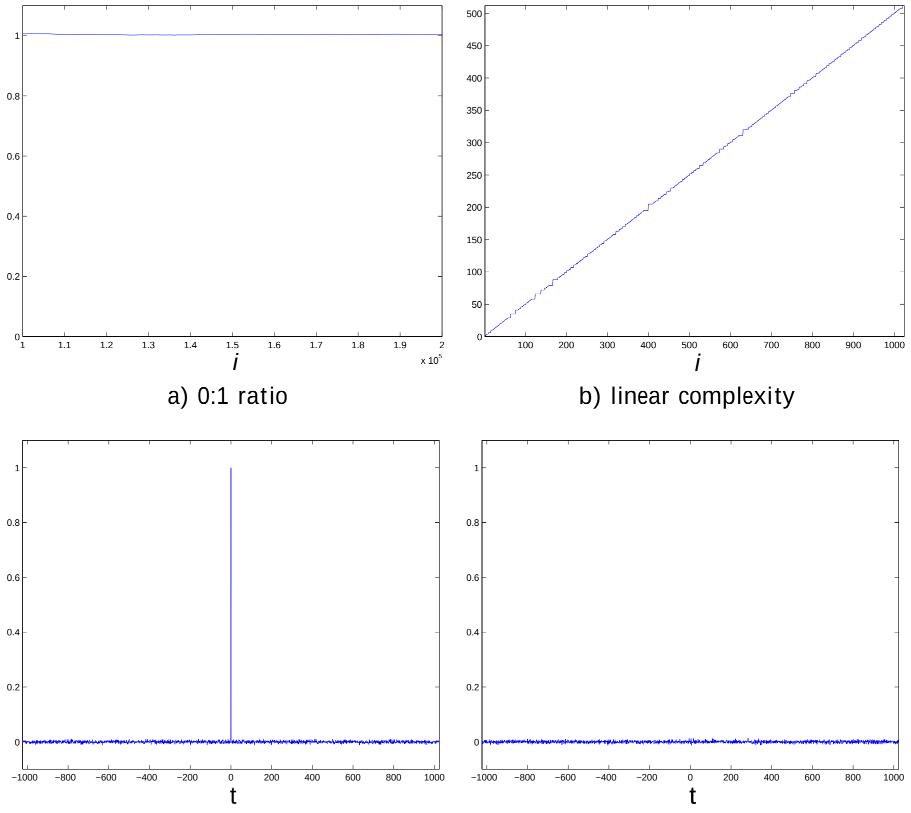
Although analyses given here are entirely theoretic, the experiments strongly support the theoretical results (see the following Fig. 2. and Sect. 3.5 for more details). In the future research, we will try to find the strict proof of $\{k(i)\}$ generated by CCS-PRBG is real i.i.d. binary sequence.

3.4 Chaotic-System-Free Property

Consider there are many chaotic maps satisfy the requirements *R1* and *R2*, and the requirement *R3* and *R4* just restrict the relation between the two

[4] The cycle-length of $\{k(i)\}$ is $L = \mathrm{lcm}(\sigma_1\Delta_1(2^{L_1} - 1), \sigma_2\Delta_2(2^{L_2} - 1))$, not infinity. Hence, the linear complexity of $\{k(i)\}_{i=1}^{\infty}$ should be about $L/2$, not infinity either.

chaotic systems, CCS-PRBG is chaotic-system-free obviously. Since piecewise linear chaotic maps satisfy the requirements *R1–R4*, they are strongly suggested being used, from the viewpoint of the encryption speed and realization (recall section 1.2).



a) 0:1 ratio

b) linear complexity

c) auto-correlation

d) cross-correlation

Fig. 2. Cryptographic properties of digital CCS-PRBG

3.5 Experimental Results

In order to verify the theoretical results on cryptographic properties of digital CCS-PRBG with perturbation, some experiments are made. The two chaotic maps are both selected as the following piecewise linear maps define on $I = [0, 1]$, which are used in [9] and detailed analyzed in [34]:

$$F_1(x,p) = F_2(x,p) = F(x,p) = \begin{cases} x/p, & x \in [0,p) \\ (x-p)/(\frac{1}{2}-p), & x \in [p, \frac{1}{2}] \\ F(1-x,p), & x \in [\frac{1}{2}, 1] \end{cases}, \qquad (8)$$

The finite computing precision is $n = 32$ (bits). The perturbing PRNG-s are selected as two m-LFSR-s, whose degrees are $L_1 = 16$, $L_2 = 17$ and whose perturbing intervals are $\Delta_1 = 99$, $\Delta_2 = 101$. The number of pre-iteration m is 16. Both initial conditions and control parameters are generated randomly, and a large number of sub-sequences of $k(i)$ are extracted from random positions to test the cryptographic properties. The 0:1 ratio, linear complexity and auto-correlation of one sub-sequence are shown in Fig. 2a–c respectively. In Fig. 2d, the cross-correlation of two sub-sequences with identical initial conditions but slightly different (2^{-n}) control parameters is given. We can see the experimental results coincide well with the theoretical analyses.

4 Construct Stream Ciphers Using Digital CCS-PRBG

Based on digital CCS-PRBG, many different practical stream ciphers can be constructed. We will see these stream ciphers can provide feasible solutions to the problems existing in other digital chaotic ciphers. Using different configurations of CCS-PRBG, many stream ciphers can be obtained conveniently with considerably low cost and simple realization. Here, digital CCS-PRBG replaces the kernel role of LFSR in conventional stream-cipher cryptography.

4.1 Some Examples of Stream Ciphers

• **Cipher 1**: Give a digital CCS-PRBG with perturbation, initial conditions $x_1(0), x_2(0)$ and control parameters p_1, p_2 are the secure key. $\{k(i)\}$ is directly used to encrypt (generally XOR) plaintext and decrypt ciphertext.

The above Cipher 1 is the simplest stream cipher based on digital CCS-PRBG. If finite computing precision is n (bits), the key entropy will be $4n$. Moreover, it is easy to be realized by hardware or software with rather low cost. On a Pentium III 800MHz PC, a software version based on piecewise linear chaotic map (8) is developed with Turbo C 2.0 for test. The actual encryption speed reaches 9 Mbps under fixed-point arithmetic. Such a speed is faster than many other chaotic ciphers and can be acceptable in many secure applications. Under hardware realization, the speed will be promoted much.

If some simple modifications are made on cipher 1, some enhanced stream ciphers with larger key entropy (higher security), faster speed can be obtained with a little extra complexity and cost. Two examples are given as follows.

• **Cipher 2**: Give four one-dimensional chaotic systems CS_0 CS_3, and five m-LFSR-s $m\text{-}LFSR_0$ $m\text{-}LFSR_4$, in which $m\text{-}LFSR_0$ $m\text{-}LFSR_3$ are used to perturb CS_0 CS_3. Before each iteration of CS_0 CS_3, firstly use $m\text{-}LFSR_4$ to generate two 2-bits pseudo-random numbers $pn1(i)$ and $pn2(i)$. If $pn2(i) = pn1(i)$, do $pn2(i) = pn1(i)$ 1. Then select $CS_{pn1(i)}$ and $CS_{pn2(i)}$ to compose the digital CCS-PRBG to generate $k(i)$. The secure key contains the initial conditions and control parameters of the four chaotic systems.

The key entropy will be $8n$ under n (bits) computing precision. $m\text{-}LFSR_4$ adds more complexity to the cryptanalysis so such a cipher is securer, with only double cost of realization and approximate encryption speed to cipher 1.

• **Cipher 3**: For piecewise linear chaotic maps defined on $I = [0, 1]$, such as the map (8), the invariant density functions are $f(x) = 1$. When they are realized discretely, every bit of the orbits will be balanced on $\{0, 1\}$. Based on such a fact, we can define a generalized version of digital CCS-PRBG. Here assume finite computing precision is n (bits). For one iteration of $F_1(x_1, p_1)$ and $F_2(x_2, p_2)$, generate n bits $K(i) = k_0(i) \ldots k_{n-1}(i)$ as follows:

```
for $j = 0$ to $n - 1$ do
    $x_1(i, j) = x_1(i) \quad j$
    $x_2(i, j) = x_2(i) \quad j$
    $k_j(i) = g(x_1(i, j), x_2(i, j))$
end
```

Where ( ) denotes circular right (left) shift operation. Apparently, a stream cipher based on generalized CCS-PRBG will run nearly n times faster than the one based on common CCS-PRBG, without loss of high security. When cipher 3 is realized by hardware with parallel arithmetic technique, the encryption speed of cipher 3 will close to s Mbps when the clock frequency is s MHz [5]. Such a speed approximately equals to the speed of many conventional stream ciphers based on LFSR-s, such as Geffe generator and clock-controlled generator, and faster than some complicated stream ciphers [39]. If we combine cipher 2 and cipher 3, both the security and the encryption speed can be improved much. Actually, in order to further enhance the security of Cipher 3, we can introduce another $m\text{-}LFSR_5$ to pseudo-randomly control the direction of the circular shift operation of x_1 and x_2.

4.2 Security

Generally speaking, the security of the above ciphers can be ensured by the perfect cryptographic properties of digital CCS-PRBG. But we have known that many chaotic ciphers are not secure although they have some “good” statistical properties. So we should still investigate whether or not the ciphers based on digital CCS-PRBG is secure enough to known cryptanalysis methods.

Many methods have been proposed to break analog chaotic encryption schemes, such as chaotic masking, switching and modulating approaches [26, 27, 28, 29, 30]. They work well because chaotic synchronization makes it possible to extract dynamical information of the chaotic systems. Since the transmitted signal must be used to realize synchronization of the transmitter and receiver, such information may be useful to restore the chaotic orbit and then extract the hidden message. For digital CCS-PRBG, because chaotic synchronization is not used and two different chaotic orbits are employed to make pseudo-random keystream $k(i)$, the dynamics of the two chaotic systems cannot be obtained

[5] Apparently, the speed is chiefly determined by the fixed-point divisions needed in chaotic iterations. Since a n-bit digital divider consumes about n clock cycles for one n-bit division, the encryption speed of cipher 3 will be close to $\frac{s}{n} \cdot n = s$ Mbps.

from the ciphertext. In addition, the pseudo-random perturbation also makes the cryptanalysis more difficult. Even if the plaintext is known, it is impossible to extract the two chaotic orbits just from $k(i)$. Hence, those methods, which are available to break secure communication approaches based on chaotic synchronization, cannot be used to break the ciphers based on digital CCS-PRBG.

Other known cryptanalysis methods aim at different weaknesses of concerned chaotic ciphers. The one in [21,22] is available because of the degraded statistical properties of discrete chaotic systems, which has been considered carefully and been avoided by perturbation-based algorithm in digital CCS-PRBG. The one in [20] is based on a specific weakness of 2-D Hénon map and cannot be generalized to other chaotic systems. The ones in [23, 24, 25] can work well for the special weaknesses in the corresponding ciphers and also cannot be extended to break CCS-PRBG based ciphers with entirely different encryption structure.

We can see the ciphers based on digital CCS-PRBG are secure to all known cryptanalysis methods of chaotic ciphers. Of course, before we can finally say "digital CCS-PRBG based ciphers are secure enough", further research on cryptanalysis of digital CCS-PRBG should be done. But the above discussion implies that digital CCS-PRBG may be a new promising candidate to construct stream ciphers with high security and low cost.

There is one notable defect in digital CCS-PRBG that should be mentioned here. Assume $x_1(0) = x_2(0)$, when the control parameters are p_1, p_2, the generated pseudo-random bit sequence is $k(i)$; exchange the control parameters of the two chaotic maps, the generated pseudo-random bit sequence is $k\ (i)$. If the two chaotic maps are perturbed with identical perturbing PRNG-s and identical perturbing intervals ($\Delta_1 = \Delta_2$), it is obvious that $k\ (i) = \overline{k(i)}$, which is the natural result of $g(x_2, x_1) = \overline{g(x_1, x_2)}$. Such an effect will cause the key space size of the ciphers decrease 1/2. To avoid this defect, different perturbing PRNG-s or perturbing intervals should be used, and $m > \max(\Delta_1, \Delta_2)$ is suggested.

5 Conclusion

Nowaday digital chaotic ciphers are surveyed, and some existent problems in them are discussed in this paper. A novel chaotic PRBG called CCS-PRBG is proposed to solve these problems. Theoretical analyses and experiments show that digital CCS-PRBG has perfect cryptographic properties. The digital CCS-PRBG can be a kernel part in the design of new stream ciphers. In the future, some details on hardware realization of CCS-PRBG based stream ciphers will be concerned. As we have mentioned in Sect. 3.3, the strict proof of $\{k(i)\}$ is i.i.d. sequence will be further studied, too. Possible cryptanalysis methods of the digital CCS-PRBG will be another open topic.

Acknowledgement

The authors would like thank the anonymous reviewers for their helpful comments, Miss Han Lu at Xi'an Foreign Language University and Miss Cao Yanghui at Xi'an Jiaotong University for their helps in the preparation of this paper.

References

1. G. Alvarez, G. Pastor F. Monotoya, and M. Romera. Chaotic cryptosystems. In *Proc. IEEE Int. Carnahan Conf. Security Technology*, pages 332–338. IEEE, 1998.
2. Ljupčo Kocarev, Goce Jakimoski, Toni Stojanovski, and Ulrich Parlitz. From chaotic maps to encryption schemes. In *Proc. IEEE Int. Symposium Circuits and Systems*, volume 4, pages 514–517. IEEE, 1998.
3. R. Forré. The Hénon attractor as a keystream generator. In *Advances in Cryptology – EuroCrypt'91*, Lecture Notes in Computer Science 0547, pages 76–81, Berlin, 1991. Spinger-Verlag.
4. Sang Tao, Wang Ruili, and Yan Yixun. Perturbance-based algorithm to expand cycle length of chaotic key stream. *Electronics Letters*, 34(9):873–874, 1998.
5. Sang Tao, Wang Ruili, and Yan Yixun. Clock-controlled chaotic keystream generators. *Electronics Letters*, 34(20):1932–1934, 1998.
6. R. Matthews. On the derivation of a 'chaotic' encryption algorithm. *Cryptologia*, XIII(1):29–42, 1989.
7. D. R. Frey. Chaotic digital encoding: An approach to secure communication. *IEEE Trans. Circuits and Systems II*, 40(10):660–666, 1993.
8. Zhou Hong and Ling Xieting. Generating chaotic secure sequences with desired statistical properties and high security. *Int. J. Bifurcation and Chaos*, 7(1):205–213, 1997.
9. Hong Zhou and Xie-Ting Ling. Problems with the chaotic inverse system encryption approach. *IEEE Trans. Circuits and Systems I*, 44(3):268–271, 1997.
10. M. E. Bianco and D. A. Reed. Encryption system based on chaos theory. US Patent No. 5048086, 1991.
11. G. M. Bernstein and M. A. Lieberman. Secure random number generation using chaotic circuits. *IEEE Trans. Circuits and Systems*, 37(9):1157–1164, 1990.
12. V. A. Protopopescu, R. T. Santoro, and J. S. Tollover. Fast and secure encryption – decryption method based on chaotic dynamics. US Patent No. 5479513, 1995.
13. T. Habutsu, Y. Nishio, I. Sasase, and S. Mori. A secret key cryptosystem by iterating a chaotic map. In *Advances in Cryptology - EuroCrypt'91*, Lecture Notes in Computer Science 0547, pages 127–140, Berlin, 1991. Spinger-Verlag.
14. Zbigniew Kotulski and Janusz Szczepanski. Application of discrete chaotic dynamical systems in cryptography – dcc method. *Int. J. Bifurcation and Chaos*, 9(6):1121–1135, 1999.
15. Zbigniew Kotulski and Janusz Szczepanski. Discrete chaotic cryptography. *Annalen der Physik*, 6(5):381–394, 1997.
16. Jiri Fridrich. Symmetric ciphers based on two-dimensional chaotic maps. *Int. J. Bifurcation and Chaos*, 8(6):1259–1284, 1998.
17. M. S. Baptista. Cryptography with chaos. *Physics Letters A*, 240:50–54, 1998.
18. E. Alvarez, A. Fernández, P. García, J. Jiménez, and A. Marcano. New approach to chaotic encryption. *Physics Letters A*, 263:373–375, 1999.
19. Li Shujun, Mou Xuanqin, and Cai Yuanlong. Improving security of a chaotic encryption approach. *Physics Letters A (to be published)*.
20. D. Erdmann and S. Murphy. Hénon stream cipher. *Electronics Letters*, 28(9):893–895, 1992.
21. D. D. Wheeler. Problems with chaotic cryptosystems. *Cryptologia*, XIII(3):243–250, 1989.
22. D. D. Wheeler and R. Matthews. Supercomputer investigations of a chaotic encryption algorithm. *Cryptologia*, XV(2):140–151, 1991.

23. W. G. Chambers. Comments on ‘chaotic digital encoding: An approach to secure communication’. *IEEE Trans. Circuits and Systems II*, 46(11):1445–1447, 1993.
24. E. Biham. Cryptoanalysis of the chaotic-map cryptosystem suggested at EuroCrypt’91. In *Advances in Cryptology - EuroCrypt’91*, Lecture Notes in Computer Science 0547, pages 532–534, Berlin, 1991. Spinger-Verlag.
25. G. Alvarez, F. Montoya, M. Romera, and G. Pastor. Cryptanalysis of a chaotic encryption system. *Physics Letters A*, 276:191–196, 2000.
26. Kevin M. Short. Signal extraction from chaotic communications. *Int. J. Bifurcation and Chaos*, 7(7):1579–1597, 1997.
27. Tao Yang, Lin-Bao Yang, and Chun-Mei Yang. Cryptanalyzing chaotic secure communications using return maps. *Physics Letters A*, 245:495–510, 1998.
28. Maciej J. Ogorzatek and Hervé Dedieu. Some tools for attacking secure communication systems employing chaotic carriers. In *Proc. IEEE Int. Symposium Circuits and Systems 1998*, volume 4, pages 522–525. IEEE, 1998.
29. Chang-Song Zhou and Tian-Lun Chen. Extracting information masked by chaos and contaminated with noise: Some considerations on the security of communication approaches using chaos. *Physics Letters A*, 234:429–435, 1997.
30. Th. Beth, D. E. Lazic, and A. Mathias. Cryptanalysis of cryptosystems based on remote chaos replication. In *Advances in Cryptology - EuroCrypt’94*, Lecture Notes in Computer Science 0950, pages 318–331, Berlin, 1994. Spinger-Verlag.
31. R. Brown and L. O. Chua. Clarifying chaos: Examples and counterexamples. *Int. J. Bifurcation and Chaos*, 6(2):219–249, 1996.
32. Julian Palmore and Charles Herring. Computer arithmetic, chaos and fractals. *Physica D*, 42:99–110, 1990.
33. Ghobad Heidari-Bateni and Clare D. McGillem. A chaotic direct-sequence spread-spectrum communication system. *IEEE Trans. Communications*, 42(2/3/4):1524–1527, 1994.
34. Li Shujun, Li Qi, Li Wenmin, Mou Xuanqin, and Cai Yuanlong. Statistical properties of digital piecewise linear chaotic maps and their roles in cryptography and pseudo-random coding. In *Cryptography and Coding - 8th IMA Int. Conf. Proc. (to be published)*, Lecture Notes in Computer Science, Berlin, 2001. Springer-Verlag.
35. Zhou Hong and Ling Xieting. Realizing finite precision chaotic systems via perturbation of m-sequences. *Acta Eletronica Sinica* (In Chinese), 25(7):95–97, 1997.
36. Tohru Kohda and Akio Tsuneda. Statistics of chaotic binary sequences. *IEEE Trans. Information Theory*, 43(1):104–112, 1997.
37. Shin’ichi Oishi and Hajime Inoue. Pseudo-random number generators and chaos. *Trans. IECE Japan*, E 65(9):534–541, 1982.
38. Jorge A. González and Ramiro Pino. A random number generator based on unpredictable chaotic functions. *Computer Physics Communications*, 120:109–114, 1999.
39. Bruce Schneier. *Applied Cryptography – Protocols, algorithms, and souce code in C*. John Wiley & Sons, Inc., New York, second edition, 1996.
40. Andrzej Lasota and Michael C. Mackey. *Chaos, Fractals, and Noise - Stochastic Aspects of Dynamics*. Springer-Verlag, New York, second edition, 1997.
41. Yang Yixian and Lin Xuduan. *Coding Theory and Cryptology* (In Chinese). People’s Post and Telecommunications Press, Beijing, China, 1992.
42. Hao Bai-Lin. *Starting with Parabolas: An Introduction to Chaotic Dynamics* (In Chinese). Shanghai Scientific and Technological Education Publishing House, Shanghai, China, 1993.

Re-dividing Complexity between Algorithms and Keys (Key Scripts)

Gideon Samid

Technion – Israel Institute of Technology, Haifa, Israel,
samidg@tx.technion.ac.il

Abstract. For decades cryptography strived for its goals by packing complexity into the exposed program, all the while pressing down the size of the secret key. Alas, modern technology (1) makes small keys a secondary requirement, (2) allows for layering of program logic, and (3) offers privacy and security offenders clever eavesdropping tools; altogether warranting a re-examination of the relative roles of the "passive" key and the "active" algorithm. We describe a working system where the nominal key is merged with some JavaScript code to become the "effective key," thereby conferring upon the JavaScript interpreter (standard part in modern browsers), the role of the exposed cryptographic algorithm. We show that such Key-Script offers equivocation, (deniability), and we provide a secure key-distribution scheme that is not based on one-way functions, rather on the attribute of equivocation. We examine this new setting, and argue that it formally defeats cryptanalysis, where in practice such robustness is somewhat qualified.

1 Introduction

To "cook" a plaintext into its corresponding ciphertext, two standard ingredients are customarily needed: an encryption algorithm, and an encryption key. Over the years, the key was envisioned as a secret bit sequence, and the algorithm was a process that operated on the plaintext and the key as two sources of passive data. The result of the operation was the ciphertext.

Early in the life of the profession, the acclaimed Dutch cryptographer, A. Kerckhoff formulated his famous principle which clarified the functional distinction between key an process. The process, Kerckhoff argued, should be open for broad examination, so that any mathematical weakness therein would be readily exposed. In turn, absence of such discovered weakness builds users' confidence in its merit. The key, said Kerckhoff, is the sole element of the system which should remain secret (apart from the plaintext, of course).

Based on this much regarded principle, we may switch around our definitions. The key will be defined as the element that, if changed, restores full security to a compromised system. In other words, by equipping a fully penetrated system with a new key, it is as if it was never penetrated.

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 330–338, 2001.

This, or the former definition, does not restrict the key to a specific format. The definition is functional. Same for the process. It is only by tradition that the key and the process have settled into die-cast formats. The key is invariably a rather small random bit sequence, and the process is almost invariably fixed in terms of its operation. Thus the following setup will be considered non-traditional:

Case I: Process: RSA Encryption, (R); DES Encryption, (D); choice-algorithm, (C).

Key: random-sequence of bits. (K).

Based on the parity of the key the choice-algorithm will decide whether to activate DES, or RSA.

This is a case of compounding well known encryption packages. One might argue that there is no practical reason for such a setup because the two algorithms are so different from each other, and each is best for particular circumstances. Accordingly it would be foolish to pack one grand-encryption box with $R + D + C$ above. Such argument may be valid on its merit, but it is premature. We first want to stress the option of compounding and how it fits into the standard definitions of key and process.

Case II: Opposite to compounding we find the case of disassembly: In any particular version of DES the plaintext block and the key undergo a fixed sequence of processing steps. Now suppose that the "P-boxes" and the "S-boxes" are implemented as individual processing units, and the key contains bits which specify which of the boxes to use, in which order. In other words, the processing configuration is not fixed, as in the DES standard, but rather dependent upon the contents of the key. Even the block size may be key dependent.

Case III: A process comprised of DES, (D), and Null, (N) where Null is a "do nothing" operator: input = output. The key is a stream of bits such that one of its attributes determines whether D or N is activated.

We now examine these three cases according to Kerckhoff's principle. On its face all three cases comply with Kerckhoff's dictum. Alas, Kerckhoff calls for examination of robustness of the open algorithm. In Case-I, assuming that both DES and RSA are robust, then the combined set up is also robust. In case II it is an open question. The key may dictate such a combination of P-boxes and S-boxes that the result will be easy prey for cryptanalysts. Accordingly Case-II is suspect because the process (the open part) does not contain sufficient information to allow an examiner to determine robustness. Case-III is weak on its face. If the key determines the Null processor, then the message is in the clear. A further examination will qualify that conclusion. The Case-III setup could be an intermediate step in an encryption series. In that case the input will be 'garbled' (looking like ciphertext), and on examination of the output it would be impossible to determine prima facie that the Null option was used, (rather than DES).

So far this analysis seems very academic. To avert this reaction, we now consider the major proposition of this article. But before that we review the modern trend for distributed computing.

Before browsers came to be, one could display graphics on a screen by communicating to the local computer a bit-by-bit pixel representation of that graphics. The data volume was prohibitive. With browsers, the communicated file contains instructions for the local computer to construct the graphics on its own. The construction instructions are much smaller in volume than the actual picture. Similarly browser-associated languages, like VisualBasic Script and JavaScript contain computing parameters which are processed by the browser itself. The browsers grow bigger, more efficient and more powerful, and the programs become smaller and by reference also more powerful. Say then that smaller and smaller data volume of a browser-associated language will generate more and more action.

Also, as a modern trend: the cryptographic key is no longer a letter sequence one spy remembers by heart, it is rather a lengthy, random looking, bit sequence.

Taking these trends together, one may wonder: is it possible and advisable to merge the nominal key with some script language code to create a functional "key" which will refer to the browser (or the script interpreter therein), as the Kerckhoff open process?

2 Script Key

We consider a binary file (Ks) which when processed by a browser program is interpreted as an HTML code with a script language embedded therein. The Ks file will order the browser to open two text windows on the screen. One marked: plaintext, (the p-window, or the message window), and one marked ciphertext (the c-window, or the encryption window). The user types in a message in the p-window (or pastes one from the clipboard), then clicks a clearly marked "encrypt" button, which causes the system to generate a ciphertext, C, which is placed at the c-window. The ciphertext, comprised of only printable characters may be pasted into the clipboard and from there attached to emails, or files, as the user sees fit. The same ciphertext can be pasted back to the c-window, at any time, and by pressing a clearly marked "decrypt" button, the system reverses the former process and regenerates the original plaintext message in the p-window. (From where it can be taken to any other application). This setup is a complete encryption/decryption system We can write: $C = Br(Ks, P)$.[1]

Where C and P are the ciphertext and plaintext respectively, Br is the browser program, and Ks the HTML/Script file. The browser, Br, is fully specified. Ks serves as the formal, or functional key for this system.

[1] While a script allows for a limited size message to be placed in the window, a file-based equivalent can be easily extended from this simple script configuration.

There are several glaring distinctions between Ks and the traditional key (Kt). Kt has a known fixed length, but its bit sequence is unrestricted. Ks may be of any desired length, but its contents must comply with HTML/Script specification. This is the seminal difference. Because if Ks where of fixed, known size, it would have been patently inferior to Kt, since statistically most of the random combinations for Ks would not constitute a valid sequence of HTML/Script statements. However, a browser will process an HTML/Script file of any length; statement by statement as long as it lasts. A cryptanalyst ignorant about the key and its length, will have to suspect any combination of legal HTML/Script statements.

Moreover, browsers grow. Early versions of Internet Explorer, and Navigator provided the JavaScript method: Math.random() without a seed. In later versions Math.random(seed) appeared. In future versions one might expect a choice among specific generators, say:

```
Math,random,LFSR[32, 11, 5](seed)
```

specifying the LFSR method with 32 bits long register where the 32nd, the 11th, and the 5th bits are XOR-ed to generate the new leftmost bit. As of today, such special case generator must be implemented in the JavaScript file (Ks). The more elaborate the browser, the shorter Ks, for the same results. Now, browsers are mainstay software, and there is a great deal of economic pressure to develop them, refine them and add more and more capabilities, much beyond the cause of encryption. But encryption can take a ride on this trend.

Shift registers are notoriously inefficient in software compared to their hardware implementation. Accordingly, if the volume will justify it, an LFSR may be firmware or hardware supplied to support the browser.

For an HTML/Script file to serve as a valid key there is a need to change it quite often, and properly. This requirement is analogous to the nominal key generation challenge. Borrowing from modern computer lingo, we refer to Ks generation as mutant-generation: generating mutants to a given Ks file, so that a cryptanalyst will be unable to discover it.

3 Mutant Generation

Functionally this is a parsing capability combined with data manipulation and symbolic manipulation. The mutant generator program will operate on a given Ks and generate a different K's, or many distinct ones.

It is a rather simple task to recognize data statements and change the data around. For example, the following JavaScript function substitutes numerals with strings of four symbols comprised of: X,Y,Z, and W:

```
function numeric(number)
{
line = "0YZZY1YZZZ2ZXXX3ZXXY4ZXXZ5ZXYX6ZXYY7ZXYZ8ZXZX9ZXZY";
place = line.indexOf(number);
if (place == "-1") return place;
else numericxyz=line.substring((place+1),(place+5));
return numericxyz;
}
```

The mutant generator will find the "line =" line and randomize the string to the right, complying with the rule that all digits 0-9 will appear in some order and between them there will be a non-repeat sequence of the letters X,Y,Z,W.

We have seen lately a fascinating advance in symbol manipulation: the ability to generate mutants based on modifying the symbols that dictate the logic followed by the browser. It's an irony that the most brilliant examples thereof are exhibited by malicious viruses which mutate to evade tracking software – and quite successfully.

The mutant generator itself may, or may not be exposed. What matters is that the key space (mutant variability) will be large enough. In principle it is the unspecified length of Ks which guarantees unbounded variability, in theory at least, if not in practice. One must note that as mentioned before, per given key size Ks is much more restricted than Kt.

The open ended key space, also offers a shot at equivocation.

4 Equivocation

The prevailing cryptographies suffer from a serious weakness, cryptographers don't like to talk about: zero equivocation. A DES cipher is highly unlikely to find a second key (not the one actually used) that will decrypt it to a plausible message. Same for RSA, elliptic curves, etc. This means one must rely on the assumption of intractability, that it is sufficiently difficult to locate the one and only key. There is no dispute that once found, it can not be repudiated because there is no other key to repudiate it with. Equivocation is the probability that more than a single plausible message will fit a given ciphertext.

With HTML/Script keys one could pose the following question: Consider a cipher C, generated from plaintext P by K_s. Now picking a message of choice P , is it possible to construct an HTML/Script file, K_s, such that:

$$C = Br(K_s, P) = Br(K_s, P\ ) \tag{1}$$

We describe a procedure to accomplish this task. We write: (omitting the subscript, s, for clarity):

$$K\ (C, P\ ) = K_a(C_a, P_a) + K_b(C_b, P_b) + K_{ab} \tag{2}$$

Where: $C\ = C_a(+)C_b; P\ \ = P_a(+)P_b$. The symbol (+) represents string concatenation. K_a, K_b, and K_{ab} are all script keys.

The idea is that one can separate C_a bits from C, and some P_a bits from P and associate them through a script key K_a, while K_b associates the balance of bits. K_{ab} is a script key that will contain the information needed to direct the browser to the separation of bits for the plaintext and the ciphertext.

In other words we replace the task of finding K with the task of finding smaller key scripts K_a and K_b. This process can be repeated, and each iteration will reduce the bit counts of P and C which must be matched with a key. If necessary such repeated breakdown will reduce the size of plaintext and ciphertext bits to a single or few bits where the basic Boolean logic (AND, NOT, XOR) will guarantee a match (finding of a corresponding key script).

When the various keys and the respective K_{ab} keys are summarized, one might end up with a rather large K s but that would still be a legal key, that satisfies (1). One might note that the above is only a worst case scenario. With some insight and ingenuity much smaller K s may be found. And if one adds to this the reality that in order to establish equivocation (or deniability), it is only necessary for some plausible messages to fit the ciphertext (not one message in particular), then the room for deniability maneuvers is that much higher.

Equivocation serves as a basis for deniability. The latter refers to the credible denial by an encryption user of a claim that a particular plaintext is hidden in a captured ciphertext. We may consider the case of formal deniability where the user denies that plaintext P was hidden in ciphertext C, and insists that it was rather plaintext P that was encrypted into C. As long as the user can point to a key script K s such that C = Br(K s, P), the claim has formal credibility. Alas, if K s   Ks (where Ks encrypts P into C), the practical credibility of the K s claim is low. As shown above, a contrived key is likely to be a large one. Yet, a smart user might encrypt his true message with a rather large key script, so that a cryptanalyst will find some shorter keys, and be genuinely baffled by the equivocation.

5 Usage

The prospect of equivocation suggests a variety of very serious applications for the key script idea. Most of them will have to evolve as the concept takes a hold. In this preliminary stage we wish to outline a more casual usage, and discuss its merit.

The average savvy Internet surfer is not using cryptography today, although, most of us have, at least occasionally some reasons to be discreet, and insure that a sensitive message is read only by whom we intend to read it. The reasons for this lack of personal use of cryptography are apparently: (1) complexity of usage, and (2) the black-box syndrome.

Central to any encryption system is the need to manage the cryptographic keys: make them readily available when needed, and generally secure them from prying eyes. Often one requirement is served on account of the other. Also, most encryption environments are monolithic, that is, when activated they apply encryption to anything that goes through them. In reality even two intimates will have most of their communication in the ‘ordinary’ category, and only a minority thereof in the ‘sensitive’ or ‘supersensitive’ category. Therefore a monolithic environment is an imposition.

The key script idea as it has already been implemented and in use today (albeit in beta testing mode), removes the notion of traditional key. The key and some logic appear as a simple WEB page with two windows: one for the plaintext, and one for ciphertext. Each window is associated with a click button that sends the data from one window to the other. So a writer who jots down a long email to a close friend, might appraise one paragraph to be of some sensitive nature. She will then cut that paragraph off the email, and paste its contents into the plaintext window in the key script page. Then simply click the “encrypt” button, and observe the encrypted version appear in the encryption window. Copied from there and pasted in the original letter, this paragraph will now be secure.

When the full email arrives to the receiver, he quickly recognizes the encrypted paragraph (a random looking sequence of lower case letters, upper case letters, digits and some symbols), copies it to the same WEB page but into the encryption window where a click on the ‘decrypt’ key will generate the original message. Task complete.

In other words, the encryption ready WEB page which is the key script facilitates occasional encryption activity without upsetting the normal course of email flow, where encryption is in reality a rarity.

To so use the key script, one needs an effective way to share the key. Using the mutant generator a single user can create a new key at will. The key can be saved on a floppy diskette, which can be mailed or hand delivered to a communication party. The users might invoke the WEB page (the key script) directly from the floppy, so that it is never copied into the hard drive. They use it to encrypt or decrypt and copy the results to other files or emails. Subsequently they kill that WEB page on the screen and remove the floppy.

The “black-box” syndrome of most current cryptographies is disturbing to quite a few sophisticated users. If the encryption and decryption are carried out by mysterious executables one can not be sure that an additional trap door activity is not incorporated into the encrypted file. Using key script, the logic is viewable and readable by invoking the file through a text editor. There is no mystery. The data and the logic are open and can be analyzed step by step through tracking and debugging software, if so desired. The only black-box attributes are with the browser itself, or more precisely the script interpreter therein. Alas, there are numerous independent browsers on the market, and unless someone theorizes wide conspiracy, it is safe to assume that the various browsers all comply with the language definition, and nothing more.

6 Key Distribution

Key script equivocation may serve as a basis for a key distribution scheme, which will further the desirability of this paradigm. Today, key distribution is based on the notion of one-way function, or intractability. It has been argued above that intractability is a tentative assumption that taints this method intrinsically. The following scheme will offer an alternative: Alice wishes to change the encryption key which she and Bob use in their secret communications. To that end, Alice builds a history file that chronicles the detailed chronology of her communication with Bob. It also includes information she knows about Bob before they began their secret communication. We designate this history file as H_0. Now, Alice prepares fake history files: $H_1, H_2, H_3, ...H_n$, which are all plausible, but all false. Then banking on the key script equivocation, for each of the H files, Alice produces a key script that relates them to a given ciphertext, C:

$$C = E(K_i, H_i) \qquad \text{for } i = 0, 1, 2, ...n \tag{3}$$

Then Alice communicates, C, and $K_0, K_1, K_2, ...K_n$ to Bob. Bob knows the history of his communication with Alice, and thereby identifies K_0, as the proper key. Eve, the eavesdropper will be confused by the multiplicity of the keys since she can not distinguish between the n history files which she is generating. Of course, this key distribution mechanism is stronger in proportion to the value of n. Also, it can be used in conjunction with one-way function, serving as an extra layer thereof.

7 Case Study

A full implementation of key script cryptography is given at
http://www.agsencryptions.com/dnlnotes.htm.

8 Outlook

For the past three decades computing has shown consistent preference in favor of the open system. The secretive does not survive. Details must be reviewable, and black boxes don't stand. One may extend this trend to cryptography. Despite the large number of users, there are very few professional cryptographers who have the skills to judge an encryption setup. The rest, rely on these judgments. The small size of the experts circle creates a festering of suspicion and a sense of discomfort. Such sense can be alleviated if more of what happens between the plaintext and the ciphertext is up, in the open, and readable – not just by the few, but by the many. Psychologically and practically the key script serves this trend.

The other attribute of key script: equivocation, may prove even more powerful, whether it would be quantum decoders, or some other nifty computing devices, it is hard to imagine that with all the innovation that the human race accomplished, one feat will remain beyond human reach: efficient factorization of large numbers. The day will come when RSA, DES and elliptic curve intractability will be so eroded that these mainstay cryptographies will no longer cut it. As we approach this day, we also invite more attention to any equivocation based cryptography, be it quantum cryptography, or be it key scripts.

References

1. Rolf Oppliger, Security Technologies for the World Wide Web, ISBN 1-58053-045-1, 2000, 444 pp.
2. Vesna Hassler, Security Fundamentals for E-Commerce, ISBN 1-58053-108-3, 2001, 416 pp.
3. Rolf Oppliger, Secure Messaging with PGP and S/MIME, ISBN 1-58053-161-X, 2001, 332 pp.
4. United States General Accounting Office Accounting and Information Management Division Information: Security Risk Assessment GAO Practices of Leading Organizations; November 1999
5. Sheila Frankel, Demystifying the IPsec Puzzle, ISBN 1-58053-079-6, 2001, 296 pp.
6. James P. Anderson, Computer Security Technology Planning Study, ESD-TR-73-51, ESD/AFSC, Hanscom AFB, Bedford, MA 01731 (Oct. 1972) [NTIS AD-758 206]
7. Department of Defense, Trusted Computer System Evaluation Criteria (Orange Book), DoD 5200.28-STD (1983, 1985)
8. Philip Myers, Subversion: The Neglected Aspect of Computer Security, Master Thesis. Naval Postgraduate School, Monterey, CA 93940 (June 1980)
9. Peter G. Neumann, L. Robinson, Karl N. Levitt, R. S. Boyer, and A. R. Saxena, A Provably Secure Operating System, M79-225, Stanford Research Institute, Menlo Park, CA 94025 (June 1975)
10. Grace H. Nibaldi, Proposed Technical Evaluation Criteria for Trusted Computer Systems, M79-225, The Mitre Corporation, Bedford, MA 01730 (Oct. 1979)
11. G. Vernam, "The Vernam Cipher" US Patent No 1,310,719
12. C. E. Shannon "A Mathematical Theory of Cryptography" Technical Report 45-110-92, Bell Laboratories, 1945.
13. C. E. Shannon "Communication Theory of Secrecy Systems", Bell Systems Tech. Jr. Vol 28, pages 656-715, 1949
14. R. Canetti, U. Feige, O. Goldreich and M. Naor "Adaptively Secure Computation", 28th STOC, 1996
15. R. Canetti, R. Gennaro, "Incoercible Multiparty Computation", FOCS'96
16. D. Beaver: "Plausible Deniability (extended abstract)"; Pragocrypt '96 Proceedings, 1996
17. G. Davida "Ciphertext Transformations and Deniability" 1997 Information Security Workshop Ishikawa High-Tech Conference Center Japan Advanced Institute of Science and Technology
18. M Roe "Cryptography and Evidence" Doctoral Dissertation, Univ of Cambridge, UK, 1997
19. Ran Canetti, Cynthia Dwork, Moni Naor, Rafail Ostrovsky "Deniable Encryption" Crypto'97
20. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone "Handbook of Applied Cryptography" CRC Press 1997
21. J. Kelsey, B. Schneier, D. Wagner, and C. Hall, "Cryptanalytic attacks on pseudo-random number generators", Fast Software Encryption, Fifth International Proceedings, pp. 168-188, Springer-Verlag, 1988.
22. D. Hoover, B. Kausik "Software Smart Cards via Cryptographic Camouflage" Proceedings of the 1999 IEEE Symposium on Security and Privacy

A Tool Box of Cryptographic Functions Related to the Diffie-Hellman Function

Eike Kiltz

Lehrstuhl Mathematik & Informatik,
Fakultät für Mathematik, Ruhr-Universität Bochum,
44780 Bochum, Germany,
kiltz@lmi.ruhr-uni-bochum.de,
http://www.ruhr-uni-bochum.de/lmi/kiltz/

Abstract. Given a cyclic group G and a generator g, the *Diffie-Hellman* function (DH) maps two group elements (g^a, g^b) to g^{ab}. For many groups G this function is assumed to be hard to compute. We generalize this function to the *P-Diffie-Hellman* function (P-DH) that maps two group elements (g^a, g^b) to $g^{P(a,b)}$ for a (non-linear) polynomial P in a and b. In this paper we show that computing DH is *computationally equivalent* to computing P-DH. In addition we study the corresponding decision problem. In sharp contrast to the computational case the decision problems for DH and P-DH can be shown to be *not generically equivalent* for most polynomials P. Furthermore we show that there is no *generic algorithm* that computes or decides the P-DH function in polynomial time.

1 Introduction

Let G be a cyclic finite group and let g be a generator of G. The Diffie-Hellman function, $\mathrm{DH} : G \times G \to G$ is given by $\mathrm{DH}(g^a, g^b) = g^{ab}$. This function is used, for instance, in the Diffie-Hellman cryptosystem [3]. Here two parties, say Alice and Bob, agree on a common pair (G, g), a is the private key of Alice, b is the private key of Bob, g^a is sent from Alice to Bob, g^b is sent vice-versa, and finally both of them are able to compute g^{ab}. The Computational Diffie-Hellman assumption claims that the function DH is hard to evaluate.

In this work we are generalizing the Diffie-Hellman function in the following way. Let $P(a, b)$ be a function in a and b. We define the P-Diffie-Hellman function, P-DH: $G \times G \to G$ as

$$P\text{-DH}(g^a, g^b) := g^{P(a,b)}.$$

Clearly, the Diffie-Hellman function is achieved by setting $P(a, b) = ab$. We will restrict our studies to the case where P is a *non-linear polynomial* in a and b.

The function that computes $g^{(a^2)}$ from g^a is called the *Square Exponent* function. A motivation for the analysis of this variant of the Diffie-Hellman function is that certain cryptographic systems exist whose security relies on the hardness of this function. An example is a scheme for key escrow with limited time span [1].

C. Pandu Rangan, C. Ding (Eds.): INDOCRYPT 2001, LNCS 2247, pp. 339–349, 2001.

Maurer and Wolf [5] prove the equivalence of computing the Diffie-Hellman function and computing the Square Exponent function. Further theoretical research about the Square Exponent function was done [2,8].

Clearly computing the P-DH function cannot be harder than computing the DH function for a polynomial $P(a,b)$. In Section 3 we also show the converse direction, i.e. that computing the Diffie-Hellman function is *computational equivalent* to computing the P-DH function for non-linear polynomials $P(a,b)$. As we will see, the strength of our result will depend on the smallest prime factor of the group order. In Section 4 we study the corresponding decision problem: For random group elements g^a, g^b and (in random order) g^c and $g^{P(a,b)}$ decide between g^c and $g^{P(a,b)}$. In sharp contrast to the results in Section 3 we show that the decision problem for the Diffie-Hellman function and the P-Diffie-Hellman function are provably *not generically equivalent* for most polynomials $P(a,b)$. On the other hand we show that no efficient generic algorithm can decide the P-Diffie-Hellman function. Finally, in Section 5 we mention some open problems.

2 Definitions

We say that an algorithm is efficient if it runs in probabilistic polynomial time. We call a function α negligible in n if $\alpha(n) < 1/P(n)$ holds for every polynomial P and for sufficiently large n.

P-Diffie-Hellman function. Let G be a finite cyclic group whose order $|G|$ is an n-bit integer. Let $\mathbb{Z}_{|G|}$ denote the ring of integer residue classes modulo $|G|$. Let $k = k(n)$ and $l = l(n)$ be two functions mapping integers to integers. Let $\mathcal{P}_l^k = \mathcal{P}_l^k(n)$ be the family of sets of all non-linear polynomials $P(a,b)$ over $\mathbb{Z}_{|G|}$ of the form $P(a,b) = \sum_{i,j \in \{0...l\}} c_{ij} a^i b^j$ with coefficients $c_{ij} \in \mathbb{Z}_{|G|}$ and absolute values $|c_{ij}|$ bounded by k. We restrict the polynomials $P(a,b)$ to non-linear polynomials, i.e. at least for one (i,j) with $i+j \geq 2$, $c_{ij} = 0$ must hold. To simplify our notation we introduce $\mathcal{P}_l := \mathcal{P}_l^{|G|/2}$ (no restrictions to coefficients) and $\mathcal{P} := \mathcal{P}_{|G|-1}$.

For a cyclic, finite group G, a fixed generator g of G and a polynomial $P \in \mathcal{P}$ we define the *P-Diffie-Hellman function*, P-DH: $G \times G \to G$ as

$$P\text{-DH}(g^a, g^b) := g^{P(a,b)},$$

where P is called the *defining polynomial* of the P-Diffie-Hellman function.

Examples of the P-DH function are:

Name	Defining polynomial	P-Diffie-Hellman function
Diffie-Hellman function [3]	$P(a,b) = ab$	$\mathrm{DH}(g^a, g^b) = g^{ab}$
Square Exponent function [5]	$P(a,b) = a^2$	$\mathrm{SE}(g^a) = g^{(a^2)}$
To-the-s Diffie-Hellman function	$P(a,b) = a^s$	$\mathrm{DH}^s(g^a) = g^{(a^s)}$
-	$P(a,b) = a^2b + ab^2$	$P\text{-DH}(g^a, g^b) = g^{a^2b+ab^2}$

Considered Group Families. Let $G := (G_n, g_n)_{n \quad \mathrm{N}}$ be a family of finite cyclic groups and generators. We define G as the set of all families G, where the bitlength of the (efficiently computable) group order $|G_n|$ is of the order n. We define G(nsprime) := $\{G : \quad$ polynomials $R \quad n_0 \quad n \geq n_0 : \mathrm{minpf}(|G_n|) > R(n)\}$ as the set of all families G such that the minimal prime factor of the group order $|G_n|$ is larger than any polynomial (nsprime stands for “no small prime factor”).

Computational Assumptions. Let $(G, g) = (G_n, g_n)_{n \quad \mathrm{N}} = G$ be a family of groups and generators and let (n) be a function in n taking values in the interval $[0, 1]$. For $P \quad$ P the *(n)-P Computational Diffie-Hellman assumption for G* ((n)-P-CDH(G)) is: There is no efficient algorithm that, given random group elements g^a and g^b, outputs $g^{P(a,b)}$ with probability at least (n) (taken over the uniformly distributed input and coin tosses of the algorithm).

We define (n)-CDH(G) as the assumption (n)-P-CDH(G) for $P(a, b) := ab$ and (n)-CSE(G) as the assumption (n)-Q-CDH(G) for $Q(a, b) := a^2$.

The assumption that for all polynomials R there is no efficient algorithm that, given g^a and g^b, outputs $g^{P(a,b)}$ with (asymptotical) probability at least $1/R(n)$ is denoted as $\frac{1}{\mathrm{poly}(n)}$-$P$-CDH($G$). Vice-versa, the assumption, that there is no efficient algorithm that, given g^a and g^b, outputs $g^{P(a,b)}$ with probability $1 - \alpha(n)$, where $\alpha(n)$ is a negligible function in n, is denoted as P-CDH(G).

We say that assumption (n)-P-CDH holds, if (n)-P-CDH(G) holds for every family $G \quad$ G. We say that (n)-P-CDH$_{\mathrm{nsprime}}$ holds, if (n)-P-CDH(G) holds for every family $G \quad$ G(nsprime).

Relations. To express relations among assumptions we will use the following notation: $A \quad B$ means that if assumption A holds, so does assumption B. Vice-versa, it also means that if there is a efficient algorithm A_B breaking assumption B then we can build another efficient algorithm $A_A^{A_B}$ with (oracle) access to A_B which breaks assumption A.

Generic Algorithms (Notation of Shoup [7]). An encoding function on the additive group $(\mathrm{Z}_m, +)$ is an unknown injective map $\sigma : \mathrm{Z}_m \to \{0, 1\}^n$ for some integer n. For a generic algorithm nothing is known about the structure (representation) of the underlying algebraic group. More precisely a generic algorithm A for Z_m is a probabilistic algorithm that takes as input an encoding list $(\sigma(x_1), \ldots, \sigma(x_k))$ where σ is an encoding function. Operations can only be performed via addition and subtraction oracles which given two indices i, j, return the encoding of $\sigma(x_i + x_j)$ and $\sigma(x_i - x_j)$ respectively. The new encoding is then added to the encoding list. The output of the algorithm is denoted by $A(\sigma; x_1, \ldots, x_k)$. An example of a generic algorithm is the Pohlig-Hellman algorithm that computes the discrete logarithm.

Relations between assumptions that make only use of generic reduction algorithms are marked by the appearance of σ. For instance, $A \overset{\sigma}{\not\Rightarrow} B$ means that no efficient reduction is possible when computation is restricted to generic algorithms. And *true* $^{\sigma}$ B means that there is no efficient generic algorithm can break assumption B. Note that such "impossibility statements" for generic algorithms are very weak, because problems might get substantially easier when adding an encoding to the group G.

3 The Computational Case

3.1 Previous Work

Theorem 1. *1. true* $^{\sigma}$ $\frac{1}{\mathrm{poly}(n)}$-CDH$_{\mathrm{nsprime}}$ *(Shoup [7]).*
2. true $^{\sigma}$ $\frac{1}{\mathrm{poly}(n)}$-CSE$_{\mathrm{nsprime}}$ *(Wolf [9]).*
3. $\frac{1}{\mathrm{poly}(n)}$-CDH $\Leftrightarrow$ CDH *(Shoup's Diffie-Hellman self-corrector [7]).*
4. $\frac{1}{\mathrm{poly}(n)}$-CDH $\Leftrightarrow$ $\frac{1}{\mathrm{poly}(n)}$-CSE *(Maurer and Wolf [5]) .*

3.2 This Work

The following two main theorems of this section state the equivalence of the two assumptions P-CDH and Q-CDH for two defining polynomials P and Q. Note that the size of the smallest prime factor of the group order turns the balance of the strength of the two theorems.

Theorem 2. *For every constant l and for every P, Q P_l we have:*

$$\frac{1}{\mathrm{poly}(n)}\text{-}P\text{-CDH}_{\mathrm{nsprime}} \Leftrightarrow \frac{1}{\mathrm{poly}(n)}\text{-}Q\text{-CDH}_{\mathrm{nsprime}}.$$

Theorem 3. *For l $O(\sqrt{\log n})$ and for every P, Q $\mathrm{P}_l^{\mathrm{poly}(n)}$ we have:*

$$P\text{-CDH} \Leftrightarrow Q\text{-CDH}.$$

No *generic* algorithm can efficiently break the assumption $\frac{1}{\mathrm{poly}(n)}$-$P$-CDH$_{\mathrm{nsprime}}$:

Theorem 4. *For every P $\mathrm{P}_{\mathrm{poly}(n)}$ we have: true* $^{\sigma}$ $\frac{1}{\mathrm{poly}(n)}$-$P$-CDH$_{\mathrm{nsprime}}$.

The proof of Theorem 4 uses techniques due to Shoup [7] and can be found in the full version of this paper [4].

Theorem 5 (P-DH self-corrector). *For every constant l and every P P_l we have:* $\frac{1}{\mathrm{poly}(n)}$-$P$-CDH $\Leftrightarrow$ P-CDH.

3.3 Proofs

Computing Roots in G will be an important building stone for the proofs of our theorems. We will shortly summarize some known theoretical results from [9]. For a finite cyclic group G of known order $|G|$ let d $\mathbb{Z}_{|G|}$ and x, a G. Then the equation

$$x^d = a$$

has exactly $s := \gcd(|G|, d)$ different solutions $x_1, \ldots, x_s$ (there must be at least one, x). They are called d-th roots of a and can be computed by a probabilistic algorithm in expected $O(sn^3)$ bit operations. In fact, for this algorithm to work one has to know which prime factors are shared by d and $|G|$. But in our application d is always small enough to compute this relation. Therefore a complete factorization of $|G|$ is not needed, only $|G|$ must be known. The proof of the following simple lemma can be found in the full version [4]:

Lemma 1. *For* $(G, g) = (G_n, g_n)_{n\ \mathrm{N}}$ G(nsprime) *let* d $\mathrm{Z}_{|G|}$ *and* x, a G *be random elements. Then the equation* $x^d = a$ *has with overwhelming probability a unique solution* x*.*

The following central lemma says that once we are given an algorithm that computes DH with *non-negligible* probability of success, then we can compute P-DH with *overwhelming* probability of success for any polynomial $P(a, b)$. Recall that, for instance, P-CDH is the assumption that there is *no efficient algorithm* that computes the P-Diffie-Hellman function.

Lemma 2. *For every* P $\mathrm{P}_{\mathrm{poly}(n)}$ *we have:* P-CDH $\frac{1}{\mathrm{poly}(n)}$-CDH*.*

Proof. Fix the family $(G, g) = (G_n, g_n)_{n\ \mathrm{N}}$ G. Assume $\frac{1}{\mathrm{poly}(n)}$-CDH is wrong, i.e. there is an oracle that computes DH with non-negligible probobility of success. Use the Diffie-Hellman self-corrector of Theorem 1 (3) to get an algorithm that computes DH with overwhelming probability of success. With this reliable algorithm for DH at hand, given g^a and g^b, any monomial $g^{c_{ij} a^i b^j}$ can be computed by repeated multiplication or squaring in the exponent. Hence, P-DH can be constructed "monomial-by-monomial" (there are at most polynomial many) by addition in the exponent. This brakes assumption P-CDH.

With this observation at hand the proof of Theorem 5 (P-DH self-corrector) is easy. Clearly " " holds. To prove " " we "detour" over DH. This will be a very frequently used strategy in our proofs. Let P P_l and let an oracle $O_{P\text{-DH}}$ be given that computes P-DH with non-negligible probability of success. Due to Theorem 2 we can construct an algorithm $A^{O_{P\text{-DH}}}$ that computes DH with non-negligible probability of success. Now apply Lemma 2.

Proof Outline of Theorem 2 and Theorem 3: Due to Lemma 2 in both cases it is sufficient to show that given an algorithm that computes P-DH for a P P_l then there is an algorithm that computes DH. Lemma 3 deals with the special case P P_2. It can be viewed as the induction base. In Lemma 4 computing P-DH for a P P_l is reduced through an efficient algorithm to computing Q-DH for a Q P_{l-1}. This lemma can be viewed as the induction step which is then applied recursively $l - 2$ times. As we will see we have to take care of a blow-up of the coefficients of the polynomial Q in the induction step.

Lemma 3. *1. For* P $\mathrm{P}_2^{\mathrm{poly}(n)}$ *we have:* P-CDH CDH*.*
2. For P P_2 *we have:* $\frac{1}{\mathrm{poly}(n)}$-$P$-$\mathrm{CDH}_{\mathrm{nsprime}}$ $\frac{1}{\mathrm{poly}(n)}$-$\mathrm{CDH}_{\mathrm{nsprime}}$*.*

Proof. We first prove part 1 of the lemma. Let $P \in \mathcal{P}_2^{\text{poly}(n)}$. Because of Lemma 2 it is sufficient to show P-CDH $\Leftarrow \frac{1}{\text{poly}(n)}$-CDH. Let $(G, g) = (G_n, g_n)_{n \in \mathbb{N}} \in \mathcal{G}$. Let $O_{P\text{-DH}}$ be an oracle that computes P-DH, i.e. given g^a, g^b, $O_{P\text{-DH}}$ outputs

$$g^{P(a,b)} = g^{c_{20}a^2+c_{21}a^2b+c_{22}a^2b^2+c_{10}a+c_{11}ab+c_{12}ab^2+c_{00}+c_{01}b+c_{02}b^2}.$$

We want to design an algorithm $A^{O_{P\text{-DH}}}$ that computes DH with non-negligible probability of success. The main idea of the proof is to "eliminate" any appearance of a^ib^2 and a^2b^j in the exponent for every $0 \leq i, j \leq 2$ by the multiplicative combination of calls to $O_{P\text{-DH}}$. For this, $A^{O_{P\text{-DH}}}$ queries the oracle for $Y_+ = O_{P\text{-DH}}(g^{a+b}, g) = P\text{-DH}(g^{a+b}, g)$ and $Y_- = O_{P\text{-DH}}(g^{a-b}, g) = P\text{-DH}(g^{a-b}, g)$. Division of the two outputs yields $C = Y_+ \cdot (Y_-)^{-1} = g^{4c_2 \cdot ab + 2c_1 \cdot b}$ where all $c_i := \sum_{j=0}^{2} c_{ij}$ are known. First assume $c_2 = 0$. Now $g^{4c_2 \cdot ab} = C \cdot (g^b)^{-2c_1}$ can be computed. Assume $4c_2$ is positive, otherwise invert. Now compute all $4c_2$-th roots of $g^{4ab \cdot c_2}$ (there are $s := \gcd(4c_2, |G|)$), i.e. all solutions of the equation

$$x^{4c_2} = g^{4ab \cdot c_2}, \tag{1}$$

with $x = g^{ab}$. This can be done in time $O(sn^3) = \text{poly}(n)$, because for all coefficients, $c_{ij} = \text{poly}(n)$ holds. Now output one of the roots of equation (1) at random, one of them is the correct one, g^{ab}. Hence, for the case $c_2 = 0$ the success probability of the $A^{O_{P\text{-DH}}}$ is $\epsilon(n) \geq 1/s \geq 1/(4c_2) = 1/\text{poly}(n)$.

In the case $c_2 = 0$ we query the oracle for $P\text{-DH}(g^{a \pm b}, g^2)$ or $P\text{-DH}(g^{a \pm b}, g^3)$. As shown in the full paper [4] at least one of those queries leads to a successful computation of g^{ab}. This completes the proof of part 1.

Now let $(G, g) = (G_n, g_n)_{n \in \mathbb{N}} \in \mathcal{G}(\text{nsprime})$ and let $P \in \mathcal{P}_2$. We show part 2 of the lemma. Let $O_{P\text{-DH}}$ be an oracle that outputs P-DH with success probability at least $\epsilon(n) = 1/\text{poly}(n)$. First the algorithm queries the oracle for $Y_+ = O_{P\text{-DH}}(g^{a+b+s}, g^u)$ and $Y_- = O_{P\text{-DH}}(g^{a-b+t}, g^v)$ for random and known values s, t, u, v. Note that the queries are random and independent. Therefore the probability that both calls give the correct answer is at least $\epsilon^2(n)$. Assume this is the case, thus $Y_+ = P\text{-DH}(g^{a+b+s}, g^u)$ and $Y_- = P\text{-DH}(g^{a-b+t}, g^v)$. Now the key observation is that because the minimal prime factor of G is not too small, every coefficient in the exponent has an unique inverse with overwhelming probability (Lemma 1). In this case the inverse is efficiently computable. From

$$Y_+ = g^{c_2(a+b+s)^2+c_1(a+b+s)+c_0} = g^{c_2(a+b)^2+2c_2(a+b)s+c_2s^2+c_1(a+b+s)+c_0}$$

for $c_i = \sum_{j=0}^{2} c_{ij}u^j$ a simple computation gives us $g^{(a+b)^2}$. For the same reason we get $g^{(a-b)^2}$ from Y_-. Again by division g^{2ab} can be computed and hence g^{ab}. We have constructed an algorithm $A^{O_{P\text{-DH}}}$ that computes g^{ab} with success probability of at least $\epsilon^2(n) = 1/\text{poly}(n)$.

The next lemma is the "induction step" to proof Theorems 3 and 2.

Lemma 4. *1. For a $P \in \mathcal{P}_l^k$ let $O_{P\text{-DH}}$ be an oracle that breaks P-CDH. Then for $k' := 2kl2^l$ there is a $Q \in \mathcal{P}_{l-1}^{k'}$ and an efficient algorithm $A^{O_{P\text{-DH}}}$ that breaks Q-CDH making at most 3 queries to $O_{P\text{-DH}}$.*

2. For a function > 0 *and for a* P P_l *let* $O_{P\text{-DH}}$ *be an oracle that breaks* (n)-P-CDH$_{\mathrm{nsprime}}$. *Then there is a* Q P_{l-1} *and an efficient algorithm* $A^{O_{P\text{-DH}}}$ *that breaks* $^3(n)$-Q-CDH$_{\mathrm{nsprime}}$ *making at most* 3 *queries to* $O_{P\text{-DH}}$.

Proof. We start proving the first part of this lemma. Let P P_l^k and let $(G, g) = (G_n, g_n)_{n\ \mathrm{N}}$ G. The main idea of this proof again is to eliminate any appearance of $g^{a^i b^j}$ or $g^{a^i b^l}$ for any i, j by computing P-DH$(g^{a+b}, g)\cdot(P$-DH$(g^{a-b}, g))^{-1}$.

Case 1: l even. Making *two* queries to the oracle $O_{P\text{-DH}}$, algorithm $A^{O_{P\text{-DH}}}$ gets

$$Y_+ = P\text{-DH}(g^{a+b}, g) = g^{c_l(a+b)^l + c_{l-1}(a+b)^{l-1} + \cdots + c_0(a+b)}$$
$$\text{and}\quad Y_- = P\text{-DH}(g^{a-b}, g) = g^{c_l(a-b)^l + c_{l-1}(a-b)^{l-1} + \cdots + c_0(a-b)},$$

where $c_i := □_{j=1}^{l} c_{ij}$. Now assume $c_l = 0$. c_l might be 0, but in this case continue with the same trick as in the proof of Lemma 3 (1). Because l is even division leads to $g^{Q(a,b)} = Y_+ \cdot (Y_-)^{-1}$ where

$$g^{Q(a,b)} = g^{c_l((a+b)^l-(a-b)^l)+c_{l-1}((a+b)^{l-1}-(a-b)^{l-1})+\sum_{i=0}^{l-2} c_i((a+b)^i-(a-b)^i))}$$
$$= g^{2c_l\left(\binom{l}{l-1}a^{l-1}b+\binom{l}{l-3}a^{l-3}b^3+\cdots+\binom{l}{1}ab^{l-1}\right)+2c_{l-1}\left(\binom{l-1}{l-2}a^{l-2}b+\cdots+b^{l-1}\right)+\cdots}.$$

Each coefficient of $a^i b^j$ of this polynomial $Q(a, b)$ is either 0 (if j is even) or $2c_{i+j}\binom{i+j}{i} \leq 2kl\binom{l}{l/2} \leq 2kl2^l =: k$ (if j is odd). Note that the coefficients of the monomials a^l and b^l are always 0. Thus $Q(a, b)$ P_{l-1}^{k}. Algorithm $A^{O_{P\text{-DH}}}$ outputs $g^{Q(a,b)} = \text{Q-DH}(g^a, g^b)$.

Case 2: l odd. With *three* queries to the oracle $O_{P\text{-DH}}$, algorithm $A^{O_{P\text{-DH}}}$ gets

$$g^{Q(a,b)} = P\text{-DH}(g^{a+b}, g) \cdot P\text{-DH}(g^{a-b}, g)^{-1} \cdot P\text{-DH}(g^b, g)^{-1}$$

where $Q(a, b) = 2c_l l a^{l-1} b - 2c_{l-1} b^{l-1} + \cdots$. A similar computation as in the even case shows that $Q(a, b)$ P_{l-1}^{k} with k defined as above. Algorithm $A^{O_{P\text{-DH}}}$ outputs $g^{Q(a,b)} = Q\text{-DH}(g^a, g^b)$. This completes the proof of the first part.

The proof of the second part of the lemma can be found in [4].

Now we are ready to give the proof of Theorem 3.

Proof (of Theorem 3). Let $(G, g) = (G_n, g_n)_{n\ \mathrm{N}}$ G. For l $O(\sqrt{\log n})$ and k poly(n) let P P_l^k. Due to Lemma 2 it is sufficient to show P-CDH CDH. Let $O_{P\text{-DH}}$ be an oracle that computes P-DH. Now apply $(l - 2)$-times Lemma 4 (1) recursively to get a polynomial Q and an efficient algorithm $A^{O_{P\text{-DH}}}$ that computes Q-DH. Q $\mathrm{P}_2^{f(k,l)}$, where $f(k, l) = k \cdot □_{i=2\ldots l} i2^i \leq k \cdot l!2^{\frac{(l+1)l}{2}} =$ poly$(n) \cdot$ poly$(n) =$ poly(n). The number of queries to $O_{P\text{-DH}}$ is at most $3^{l-2} =$ poly(n). Now use Lemma 3 (1) to construct an efficient algorithm $B^{O_{P\text{-DH}}}$ that computes DH(g^a, g^b) with overwhelming probability of success.

The proof of Theorem 2 is similar and can be found in the full paper [4].

4 The Decisional Case

Let $G = (G_n, g_n)_{n \quad \mathrm{N}} = (G, g)$ be a family of groups and generators and let (n) be a function in n. Then the *(n)-P-Decision Diffie-Hellman assumption for G* ((n)-P-DDH(G) is: There is no efficient algorithm A that, given random group elements g^a, g^b and (in random order) $g^{P(a,b)}$ and another random group element g^c, identifies $g^{P(a,b)}$ with probability $1/2 +$ (n) (taken over the input and coin tosses of A). Let (n)-P-DDH$_{\mathrm{nsprime}}$, (n)-P-DDH$_{\mathrm{nsprime}}$, (n)-P-DDH as well as the Decision Square Exponent assumption (n)-DSE and the Decision Diffie-Hellman assumption (n)-DDH be defined as in the computational case.

4.1 Previous Work

Theorem 6. *1. true* $^{\sigma}$ $\frac{1}{\mathrm{poly}(n)}$-DDH$_{\mathrm{nsprime}}$ *(Shoup [7]).*

2. true $^{\sigma}$ $\frac{1}{\mathrm{poly}(n)}$-DSE$_{\mathrm{nsprime}}$ *(Wolf [9]).*

3. $\frac{1}{\mathrm{poly}(n)}$-DDH $\overset{\sigma}{\not\Rightarrow}$ $\frac{1}{\mathrm{poly}(n)}$-DSE$_{\mathrm{nsprime}}$ *(Wolf [9]).*

4. $\frac{1}{\mathrm{poly}(n)}$-DDH $\frac{1}{\mathrm{poly}(n)}$-DSE *(Wolf [9]).*

4.2 This Work

We define the set of polynomials P$_2$ for which the reduction from P-DDH to DDH is possible. Let P$_2$ P$_2$ be the set of polynomials $P(a, b)$ given by

$$P(a, b) = (d_{00}a + d_{01}b)(d_{10}a + d_{11}b) + c_{10}a + c_{01}b + c_{00}, \quad c_{ij}, d_{ij} \quad \mathrm{Z}_{|G|}.$$

Example polynomials of P$_2$ include $P(a, b) = a^2 - b^2$ and $P(a, b) = (a + b)^2$.

We characterize the relation between the assumptions $\frac{1}{\mathrm{poly}(n)}$-DDH and $\frac{1}{\mathrm{poly}(n)}$-$P$-DDH in the following two theorems. Remember that P$_1$ is the set of polynomials $P(a, b) = c_{11}ab + c_{01}a + c_{10}b + c_{00}$ satisfying $c_{11} = 0$.

Theorem 7. *For every P* P$_{\mathrm{poly}(n)}$ \ P$_2$ *and Q* P$_2$ *we have:*

1. $\frac{1}{\mathrm{poly}(n)}$-DDH$_{\mathrm{nsprime}}$ $\overset{\sigma}{\not\Leftarrow}$ $\frac{1}{\mathrm{poly}(n)}$-$P$-DDH$_{\mathrm{nsprime}}$.
2. $\frac{1}{\mathrm{poly}(n)}$-DDH $\frac{1}{\mathrm{poly}(n)}$-$Q$-DDH.

Theorem 8. *For every P* P$_{\mathrm{poly}(n)}$ \ P$_1$ *we have:*

$$\frac{1}{\mathrm{poly}(n)}\text{-DDH}_{\mathrm{nsprime}} \overset{\sigma}{\not\Rightarrow} \frac{1}{\mathrm{poly}(n)}\text{-}P\text{-DDH}_{\mathrm{nsprime}}.$$

The proof of Theorem 8 uses techniques due to Shoup [7] and can be found in the full paper [4]. The next theorem is a direct corollary of Theorem 7 (1).

Theorem 9. *For every P* P$_{\mathrm{poly}(n)}$ *we have: true* $^{\sigma}$ $\frac{1}{\mathrm{poly}(n)}$-$P$-DDH$_{\mathrm{nsprime}}$.

4.3 Proofs

The following lemma gives an alternative characterization of $\mathcal{P}_2$.

Lemma 5. *1. If $P(a,b)$ $\mathcal{P}_2$ then there are non-trivial linear combinations R, S and T of* 1, *a, b and $P(a,b)$ over $\mathbb{Z}_{|G|}$ that satisfy the relation*

$$a,b: \quad R(1,a,b,P(a,b))\cdot S(1,a,b,P(a,b)) = T(1,a,b,P(a,b)). \qquad (2)$$

2. Let (G,g) $\mathcal{G}$(nsprime) and let P $\mathcal{P}_{\mathrm{poly}(n)}$. If relation (2) is satisfied, then P $\mathcal{P}_2$ holds with overwhelming probability (over the choices of P).

Proof. Let $R(a,b) = r_0 + r_1a + r_2b + r_3P(a,b)$, $S(a,b) = s_0 + s_1a + s_2b + s_3P(a,b)$ and $T(a,b) = t_0 + t_1a + t_2b + t_3P(a,b)$. Relation (2) can only hold for all a, b if $r_3 = s_3 = 0$. Thus, viewed as polynomials over a and b, relation (2) is satisfied iff $(r_1a + r_2b)\cdot(s_1a + s_2b) = u_0 + u_1a + u_2b + t_3P(a,b)$, where $u_0 := t_0 - r_0s_0$, $u_1 := t_1 - r_0s_1 - r_1s_0$ and $u_2 := t_2 - r_0s_2 - r_2s_0$. Consequently, for any $P(a,b)$ $\mathcal{P}_2$, relation (2) can be satisfied (setting $t_3 = 1$). Now let (G,g) $\mathcal{G}$(nsprime) and relation (2) be satisfied. Due to Lemma 1, t_3 is invertible in $\mathbb{Z}_{|G|}$ with overwhelming probability. In this case obviously P $\mathcal{P}_2$ holds.

The next lemma proves Theorem 7 (2).

Lemma 6. *Let P $\mathcal{P}_2$ and let O_{DDH} be an oracle that breaks (n)-DDH. Then there is an efficient algorithm $A^{O_{\mathrm{DDH}}}$ that breaks (n)-P-DDH.*

Proof. We construct $A^{O_{\mathrm{DDH}}}$ as follows: Let g^a, g^b and in random order $g^{P(a,b)}$ and g^c be given. Since P $\mathcal{P}_2$ we can compute $g^{R(a,b)}, g^{S(a,b)}$ and $g^{T(a,b)}$ with R, S, T satisfying relation (2) of Lemma 5 (1). $g^{T(a,b)}$ is computed twice, first with $g^{P(a,b)}$, second with g^c in the role of $g^{P(a,b)}$. Lets denote them as g^{T_1} and g^{T_2}. Now feed O_{DDH} with the input $(g^R, g^S, g^{T_1}, g^{T_2})$ which immediately identifies which one, g^{T_1} or g^{T_2}, has been computed from $g^{P(a,b)}$. Note that we called O_{DDH} only once, thus the success probability of algorithm $A^{O_{\mathrm{DDH}}}$ is (n).

The next lemma says that for P $\mathcal{P}_{\mathrm{poly}} \setminus \mathcal{P}_2$ every generic algorithm that breaks (n)-P-DDH$_{\mathrm{nsprime}}$ for a non-negligible (n) needs at least super-polynomial time. It proves Theorem 7 (1). The proof uses techniques due to Shoup [7].

Lemma 7. *Let $m = m(n)$ be a family of integers whose smallest prime factors $p = p(n)$ are (asymptotically) lower bounded by a polynomial $R(n)$. Let S $\{0,1\}$ be a set of at least m binary strings. Let P $\mathcal{P}_l \setminus \mathcal{P}_2$. Let $A = A(n)$ be generic algorithms that work for groups of order m, run in time at most $T = T(n)$ and make calls to a (perfect)* DDH*-oracle. Let a, b, c $\mathbb{Z}_m$ be chosen at random, let $\sigma : \mathbb{Z}_m \to S$ be a random encoding function, and let t be a random bit. Set $w_0 = P(a,b)$ and $w_1 = c$. Then* $\Pr[A(\sigma; 1, a, b, w_t, w_{1-t}) = t] \leq 1/2 + O(lT^2/p)$.

Proof (Sketch). The proof follows two ideas. First, if the algorithm has a slight chance to decide P-DH only by making computations in the group, then this happens by an "accident" that is not very likely to happen. And second, the algorithm has no chance to get a single bit of information from the DDH-oracle, i.e. the probability that it gets a non-trivial answer from it is very small (here Lemma 5 (2) comes to application). Hence, the oracle is useless. See the full version [4] for a formal treatment of the proof.

5 Conclusions and Open Problems

We presented a theoretical approach of a generalization of the Diffie-Hellman function. This P-Diffie-Hellman function is provably computationally equivalent to the Diffie-Hellman function for a certain class of groups. As the title of this paper suggests this set of functions should be viewed as a tool box. The same way the Square Exponent function was introduced as a theoretical concept first and later exploited in a cryptographic setting, we hope that one will find a useful application in some cryptographic protocols or maybe one can use it to simplify some proofs in the context of the Diffie-Hellman function.

Note that the P-DH function can replace the DH function in some applications. For instance the to-the-s Diffie-Hellman function introduced in Section 1 can be used in protocols like the scheme for key escrow with limited time span [1].

Open Problems: As mentioned above it would be nice to have some more "real-world" applications of the P-Diffie-Hellman function.

The results in the computational case leave a lot of room for improvement. It would be interesting to see if one can improve our results to show that $\frac{1}{\mathrm{poly}(n)}$-CDH $\Leftrightarrow$ $\frac{1}{\mathrm{poly}(n)}$-$P$-CDH holds for $P \in P_{\mathrm{poly}(n)}$. In [6] the *Inverse Exponent* function $\mathrm{IE}(g^a) = g^{(a^{-1})}$ is proven to be computationally equivalent to the DH function. Consequently one might ask the question what kind of functions f (others than polynomials) lead to f-Diffie-Hellman functions $f\text{-DH}(g^a, g^b) = g^{f(a,b)}$ that are computationally equivalent to the DH function. Also, a generalization of the defining polynomial $P(a, b)$ to $P(a_1, \cdots, a_k)$ is possible.

Acknowledgment

Thanks to Jürgen Forster for great help on this paper as well to Andreas Bomke, Dario Catalano, Ahmad-Reza Sadeghi, Hans-Ulrich Simon and Michael Steiner.

References

1. M. Burmester, Y. Desmedt, and J. Seberry. Equitable key escrow with limited time span (or, how to enforce time expiration cryptographically). In *Advances in Cryptology – ASIACRYPT ' 98*, pages 380–391, 1998.
2. D. Coppersmith and I. Shparlinski. On polynomial approximation of the Discrete Logarithm and the Diffie-Hellman mapping. *Journal of Cryptology*, 13(3):339–360, March 2000.

3. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on information Theory*, 22(6):644–654, 1976.
4. E. Kiltz. A tool box of cryptographic functions related to the Diffie-Hellman Function (full version). *Manuscript*, 2001.
5. U. Maurer and S. Wolf. Diffie-Hellman oracles. *Proc. of CRYPTO'96. Lecture Notes in Computer Science*, 1109:268–282, 1996.
6. A.-R. Sadeghi and M. Steiner. Assumptions related to discrete logarithms: Why subtleties make a real difference. In *Advances in Cryptology – EUROCRYPT '01*, pages 243–260, 2001.
7. V. Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology – EUROCRYPT '97*, pages 256–266, 1997.
8. I. Shparlinski. Security of most significant bits of g^{x^2}. *Inf. Proc. Letters*, (to appear).
9. Stefan Wolf. *Information-theoretically and Computionally Secure Key Agreement in Cryptography*. PhD thesis, ETH Zürich, 1999.

Author Index