Lecture Notes in Computer Science 1746

Edited by G. Goos, J. Hartmanis and J. van Leeuwen

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

Michael Walker (Ed.)

Cryptography and Coding

7th IMA International Conference
Cirencester, UK, December 20-22, 1999
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editor

Michael Walker
Vodafone Limited
The Courtyard, 2-4 London Road
Newbury, Berkshire RG14 1JX, UK
E-mail: mike.walker@vf.vodafone.co.uk

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Cryptography and coding : . . . IMA international conference . . . ; proceedings. - 5[?]-. - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 1995[?]-
(Lecture notes in computer science ; . . .)

7. Cirencester, UK, December 20 - 22, 1999. - 1999
(Lecture notes in computer science ; 1746)
ISBN 3-540-66887-X

CR Subject Classification (1998): E.3-4, G.2.1, C.2, J.1

ISSN 0302-9743
ISBN 3-540-66887-X Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN: 10750021 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

Preface

The IMA conferences on Cryptography and Coding are not only a blend of these two aspects of information theory, but a blend of mathematics and engineering and of theoretical results and applications. The papers in this book show that the 1999 conference was no exception. Indeed, we again saw the mathematics underlying cryptography and error correcting coding being applied to other aspects of communications, and we also saw classical mathematical concepts nding new applications in communications theory.

As usual the conference was held at the Royal Agricultural College, Cirencester, shortly before Christmas - this time 20-22 December 1999. The papers appear in this book in the order in which they were presented, grouped into sessions, each session beginning with an invited paper. These invited papers were intended to reflect the invitees' views on the future of their subject - or more accurately where they intended to take it. Indeed the focus of the conference was the *future of cryptography and coding* as seen through the eyes of young researchers.

The rst group of papers is concerned with mathematical bounds, concepts, and constructions that form a common thread running through error correcting coding theory, cryptography, and codes for multiple access schemes. This is followed by a group of papers from a conference session concerned with applications. The papers range over various topics from arithmetic coding for data compression and encryption, through image coding, biometrics for authentication, and access to broadcast channels, to photographic signatures for secure identi cation. The third set of papers deals with theoretical aspects of error correcting coding, including graph and trellis decoding, turbo codes, convolution codes and low complexity soft decision decoding of Reed Solomon codes. This is followed by a collection of papers concerned with some mathematical techniques in cryptography - elliptic curves, the theory of correlations of binary sequences, primality testing, and the complexity of nite eld arithmetic. The nal collection of papers is concerned primarily with protocols and schemes. There is a diversity of papers covering lattice based cryptosystems, protocols for sharing public key parameters and for delegating decryption, and arithmetic coding schemes.

It is my pleasure to record my appreciation to the members of the conference organising committee for their help in refereeing the papers that make up this volume. They were Michael Darnell, Paddy Farrell, Mick Ganley, John Gordon, Bahram Honary, Chris Mitchell, and Fred Piper. Sincere thanks also to Pamela Bye, Hilary Hill, Adrian Lepper, and Deborah Sullivan of the IMA for all their help with the organisation of the conference and with the publication of this collection of papers.

Finally, I hope that those of you who attended the conference found it rewarding and stimulating. For those of you who did not, I hope this book of papers will encourage you to participate in the next one.

December 1999 Mike Walker

Contents

Applications of Exponential Sums in Communications Theory
[Invited Paper]

Kenneth G. Paterson

Mathematics, Cryptography and Security Group,
Hewlett-Packard Laboratories,
Filton Road, Stoke-Gi ord,
Bristol BS34 8QZ, U.K.
kp@hplb.hpl.hp.com

Abstract. We provide an introductory overview of how exponential sums, and bounds for them, have been exploited by coding theorists and communications engineers.

1 Introduction

An exponential sum is a sum of complex numbers of absolute value one in which each term is obtained by evaluating a function of additive and/or multiplicative characters of a nite eld $\mathbb{F}_q$, and where the sum is taken over the whole of $\mathbb{F}_q$. Exponential sums date back to early work of Lagrange and Gauss, the latter explicitly evaluating certain basic exponential sums now called Gauss sums in his honour. Since then, much more general exponential sums have been considered, but generally, it is impossible to nd explicit expressions evaluating these more complicated sums. However their evaluation is intimately connected to the problem of counting the numbers of points on related curves (more generally, algebraic varieties) de ned over nite extensions of $\mathbb{F}_q$ and deep methods in algebraic geometry have been developed to nd good bounds on such numbers. Two major achievements of these methods are Weil's 1940 announcement of the proof of the Riemann hypothesis for curves over nite elds [66] and Deligne's Fields medal winning proof of the Weil conjectures for algebraic varieties [8]. These results are justly regarded as being high-points of twentieth century mathematics, and from them, good bounds for many classes of exponential sums can easily be deduced.

In contrast to the depth and sophistication of the techniques used by Weil and Deligne, the bounds they proved are rather easy to state and to use. Coding theorists and communications engineers have been extraordinarily fecund in exploiting this ease of use. In this paper, we quote some bounds for exponential sums, briefly sketch the connection to curves over nite elds and examine some applications of exponential sums in communications theory. We make no attempt to be exhaustive in our coverage. Rather our aim is to provide an introductory tour, focusing on salient points, basic techniques and a few applications.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 1{24, 1999.

For this reason, all of our applications will involve, in various guises, a class of codes called dual BCH codes. We provide pointers to the vast literature for more advanced topics, and immediately recommend the survey [21] for a snapshot of the whole area.

We show how the minimum distances of dual BCH codes and other cyclic codes can be evaluated in terms of exponential sums. We then consider the problem, important in multiple-access spread-spectrum communications, of designing sequence sets whose periodic cross-correlations and auto-correlations are all small. Then we look at how exponential sums can be used to study binary sequences with small partial and aperiodic correlations. These are also important in spread-spectrum applications. We also consider the application of exponential sums in a relatively new communications application, the power control problem in Orthogonal Frequency Division Multiplexing (OFDM). Finally, we briefly consider some more advanced applications of exponential sums.

2 Finite Fields, Their Characters, and the Dual BCH Codes

We set out some facts concerning the trace map on a nite eld, assuming the reader to be familiar with the basic properties of nite elds (existence, uniqueness, primitive elements and so on). Basic references for nite elds are [23, 31, 32]. We will almost exclusively be concerned with elds of characteristic two in this paper, though almost everything we say can be generalised to characteristic p with appropriate modi cations.

Throughout, $m; n$ will denote positive integers with mjn. Also, $\mathbb{F}_{2^n}$ denotes the nite eld with 2^n elements and $\mathbb{F}_{2^n}$ the set of non-zero elements of $\mathbb{F}_{2^n}$. The relative trace function from $\mathbb{F}_{2^n}$ to $\mathbb{F}_{2^m}$ is de ned by

$$\mathrm{tr}_m^n(x) = \sum_{i=0}^{n=m-1} x^{2^{mi}}:$$

The trace map $\mathrm{tr}_m^n(x)$ has the following properties:

It is an $\mathbb{F}_{2^m}$-linear mapping onto $\mathbb{F}_{2^m}$.
For each $b \in \mathbb{F}_{2^m}$, the equation

$$\mathrm{tr}_m^n(x) = b$$

has exactly 2^{n-m} solutions $x \in \mathbb{F}_{2^n}$. In other words, the trace map is 'equidistributed' on sub- elds.
$\mathrm{tr}_1^m(\mathrm{tr}_m^n(x)) = \mathrm{tr}_1^n(x)$ for $x \in \mathbb{F}_{2^n}$.

Next we introduce the characters of $\mathbb{F}_{2^n}$. Of course these can be de ned more generally for any nite eld $\mathbb{F}_q$. Even more generally, the characters of an abelian group are just the homomorphisms from that group onto the set U of complex numbers of absolute value 1. The eld $\mathbb{F}_{2^n}$ contains two abelian subgroups of

particular interest, namely the additive and multiplicative groups of the nite eld, and so we have two corresponding sets of characters.

For each $b \in \mathbb{F}_{2^n}$, de ne a map χ_b from $\mathbb{F}_{2^n}$ to the set $\{1, -1\}$ by writing

$$\chi_b(x) = (-1)^{\mathrm{tr}_1^n(bx)}; \quad x \in \mathbb{F}_{2^n}.$$

The maps χ_b are called the *additive characters* of $\mathbb{F}_{2^n}$: by linearity of trace, it can be seen that these maps are homomorphisms from the group $(\mathbb{F}_{2^n}, +)$ to U. The map χ_0 is called the *trivial* additive character because $\chi_0(x) = 1$ for all $x \in \mathbb{F}_{2^n}$. Notice that if $b \neq 0$, then

$$\sum_{x \in \mathbb{F}_{2^n}} \chi_b(x) = 0 \tag{1}$$

because of the equi-distribution properties of the trace map.

Now let $N = 2^n - 1$ and let $\omega = \exp(2\pi i/N)$ be a complex N-th root of unity. Let α be a primitive element in $\mathbb{F}_{2^n}$. For each integer j with $0 \leq j < 2^n - 1$, we de ne a map ψ_j from $\mathbb{F}_{2^n}$ to the set U of powers of ω by writing

$$\psi_j(\alpha^i) = \omega^{ji}; \quad 0 \leq i < 2^n - 1.$$

The maps ψ_j are called the *multiplicative characters* of $\mathbb{F}_{2^n}$: they are homomorphisms from $(\mathbb{F}_{2^n}, \cdot)$ to U. The map ψ_0 is called the *trivial* multiplicative character.

For much more information about characters of nite elds, see [22, 23, 32]

Next we de ne the main class of codes that we'll work with in this paper, the dual BCH codes. In fact, we work with a sub-class of these codes, more properly called binary, primitive, dual BCH codes.

Let α be primitive in $\mathbb{F}_{2^n}$ and let t be a positive integer with $1 \leq 2t - 1 \leq 2^{dn/2e} + 1$. Let G_t denote the set of polynomials

$$G_t = \{g_1 x + g_3 x^3 + \cdots + g_{2t-1} x^{2t-1} : g_i \in \mathbb{F}_{2^n}\}.$$

For each $g \in G$, de ne a length $2^n - 1$, binary word

$$c_g = (\mathrm{tr}_1^n(g(1)), \mathrm{tr}_1^n(g(\alpha)), \ldots, \mathrm{tr}_1^n(g(\alpha^{2^n-2}))).$$

and de ne a code C_t by:

$$C_t = \{c_g : g \in G_t\}.$$

So the words of C_t are obtained by evaluating certain degree $2t - 1$ polynomials on the non-zero elements $1, \alpha, \ldots, \alpha^{2^n-2}$ of $\mathbb{F}_{2^n}$, and then applying the trace map.

It follows from the linearity of the trace map that the code C_t is linear. It can be shown that the dimension of the code is equal to nt over $\mathbb{F}_2$, the set of polynomials $\{cx^{2i-1} : 1 \leq i \leq t; c \in \mathbb{F}_{2^n}\}$ leading to a basis for the code. By examining these 'basis polynomials', it's now easy to show that the code is cyclic.

It is a consequence of a theorem of Delsarte that the code C_t is the dual of the primitive, binary BCH code with designed distance $2t + 1$ whose zeros include $; \ ^3; : : : ; \ ^{2t-1}$. See [34, Chapters 8 and 9] for more background on BCH codes and their duals.

In Section 4 we will obtain bounds on the minimum Hamming distances of the codes C_t by using Weil's bound on the size of exponential sums with polynomial argument.

3 Exponential Sums

As we stated in the introduction, exponential sums are sums in which each term is obtained by evaluating a function of additive and/or multiplicative characters of a nite eld $\mathbb{F}_q$, and where the sum is taken over the whole of $\mathbb{F}_q$. Here we consider some classes of sums over nite elds of characteristic 2, stating bounds for such sums. We also sketch the connection between exponential sums and the problem of counting the numbers of points on certain curves over nite elds. For a much more detailed exposition of the theory of exponential sums, we recommend [32, Chapter 5].

Let be a non-trivial additive character of $\mathbb{F}_{2^n}$ and let g be a polynomial of odd degree $r < 2^n$ over $\mathbb{F}_{2^n}$. We are interested in sums of the form

$$\sum_{x \in \mathbb{F}_{2^n}} (g(x))$$

which are called *exponential sums with polynomial argument* or *Weil sums*. For special choices of g, the sums can be evaluated explicitly (for example, when $g(x) = x$ we know from (1) that the sum is identically zero). Usually though, we have to settle for bounds on the size of the sums. The following result, known as Weil's theorem or the Carlitz-Uchiyama/Weil bound, is the fundamental estimate on the size of Weil sums:

Result 1 *[66, 4] With notation as above,*

$$\sum_{x \in \mathbb{F}_{2^n}} (g(x)) \qquad (r - 1)2^{n/2}:$$

Notice the Weil sums, being sums of 2^n complex numbers of absolute magnitude 1, are potentially of size $O(2^n)$. The above bound shows that (at least when r is not too large), the Weil sums are much smaller than this. Notice also that the case $r = 1$ of Weil's bound recovers (1). The condition that g have odd degree r can be replaced by much weaker criteria, for example that the polynomial $y^2 + y + g(x)$ in two variables be absolutely irreducible, or that the polynomial g *not* be of the form $h(x)^2 + h(x) + d$ for any polynomial h over $\mathbb{F}_{2^n}$ and any $d \in \mathbb{F}_{2^n}$.

3.1 Exponential Sums and Curves over Finite Fields

We sketch the connection between Weil exponential sums and the problem of counting points on curves over finite fields and outline how Weil's theorem is proved using algebraic-geometric methods. For modern and accessible approaches to the proof of Weil's theorem and related results, see the books [36, 61]. For an elementary approach avoiding algebraic geometry, see [54]. For introductory explanations, see [22, Chapters 10 and 11] and [32, Notes to Chapter 6].

To make the connection, we need the following simple result:

Lemma 1 *[32, Theorem 2.25] For $b \in \mathbb{F}_{2^n}$, we have $tr_1^n(b) = 0$ if and only if $y^2 + y = b$ for some $y \in \mathbb{F}_{2^n}$.*

Now consider the exponential sum

$$\sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{tr}_1^n(g(x))} = |\{x \in \mathbb{F}_{2^n} : \mathrm{tr}_1^n(g(x)) = 0\}| - |\{x \in \mathbb{F}_{2^n} : \mathrm{tr}_1^n(g(x)) = 1\}|$$
$$= 2|\{x \in \mathbb{F}_{2^n} : \mathrm{tr}_1^n(g(x)) = 0\}| - 2^n.$$

But we know that $\mathrm{tr}_1^n(g(x)) = 0$ if and only if there exists a solution $y \in \mathbb{F}_{2^n}$ to the equation $y^2 + y = g(x)$, in other words, if and only if there is a y such that (x, y) is a point on the *affine curve* C whose equation is $h(x, y) = 0$ where $h(x, y) = y^2 + y + g(x)$. Notice though that if y is a solution to $h(x, y) = 0$, then so too is $y + 1$. So the points on C come in pairs and are in 2-1 correspondence with the x satisfying $\mathrm{tr}_1^n(g(x)) = 0$. We deduce that

$$\sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{tr}_1^n(g(x))} = |C| - 2^n$$

where $|C|$ denotes the number of points on the affine curve C.

Next we introduce a projective version of C. We consider a homogeneous version of the equation defining C:

$$H(x, y, z) = y^2 z^{r-2} + yz^{r-1} + z^r g(x/z)$$

(where r is the degree of g) and count the projective points $[x, y, z]$ satisfying $H(x, y, z) = 0$. Notice that $H(x, y, 1) = h(x, y)$ for all x, y, so the set of projective points $[x, y, z]$ satisfying $H(x, y, z) = 0$ accounts for all the points on the affine curve, once each. But the projective curve has one additional point $[0, 1, 0]$, called a point at infinity. So if N denotes the number of projective points on C, then we have $N = |C| + 1$ and

$$\sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{tr}_1^n(g(x))} = N - 1 - 2^n. \tag{2}$$

In his paper [66], Weil considered the numbers of points on general absolutely irreducible projective curves. Let C be such a curve defined over a finite field $\mathbb{F}_q$.

For $s \geq 1$, let N_s denote the number of projective points on C whose coordinates all lie in the extension $\mathbb{F}_{q^s}$, called $\mathbb{F}_{q^s}$-rational points. Then the function

$$Z(u) = \exp\left(\sum_{s=1}^{\infty} \frac{N_s u^s}{s}\right)$$

is called the zeta function of C. This function contains all the information about the numbers of projective points on C over extensions of $\mathbb{F}_q$. Weil was able to show that $Z(u)$ is actually a rational function of u, in fact, he showed:

$$Z(u) = \frac{P(u)}{(1-u)(1-qu)}$$

where $P(u)$ is a degree $2g$ polynomial with integer coefficients and constant term 1. Here g, the genus of C, is a topological number associated with the curve. Writing

$$P(u) = \prod_{i=1}^{2g} (1 - \omega_i u)$$

Weil also showed that the $2g$ complex numbers $\omega_1, \ldots, \omega_{2g}$ all satisfy $|\omega_i| = q^{1/2}$. This last fact, conjectured by Artin and proved by Weil, is *the Riemann hypothesis for curves over finite fields*, so-called by analogy with the Riemann hypothesis for the classical zeta function.

Now a straightforward calculation shows that

$$u\frac{d\log Z(u)}{du} = \sum_{s=1}^{\infty} N_s u^s.$$

On the other hand,

$$u\frac{d\log Z(u)}{du} = u\frac{Z'(u)}{Z(u)} = u\left(\sum_{j=1}^{2g} \frac{-\omega_i}{1-\omega_i u} + \frac{1}{1-u} + \frac{q}{1-qu}\right)$$
$$= \sum_{s=1}^{\infty} \left(\sum_{i=1}^{2g} (\omega_i)^s + 1 + q^s\right) u^s.$$

By comparing the two power series, we get

$$N_s = q^s + 1 - \sum_{i=1}^{2g} (\omega_i)^s$$

and so

$$|N_s - q^s - 1| \leq 2gq^{1/2}. \tag{3}$$

We can now specialise to the projective curve C arising from our exponential sum. It turns out that the curve is always absolutely irreducible when r is odd and has genus $g = (r-1)/2$. Taking $q = 2^n$ and $s = 1$, the bound (3) tells us that $N = N_1$, the number of projective points on our curve, satisfies $|N - 2^n - 1| \leq (r-1)q^{1/2}$. Comparing with the identity (2), we now obtain the bound of Result 1.

These results have been generalised considerably to the situation where C is replaced by any non-singular algebraic variety V. Dwork [12] showed that the analogous zeta function is rational while Deligne [8] finally proved Weil's conjectures concerning the analogue of the Riemann hypothesis for such varieties. These deep results have also been exploited by coding theorists. We will summarise this work briefly in the final section.

3.2 Hybrid Exponential Sums

We loosely define hybrid exponential sums to be exponential sums in which the summand is a product of a multiplicative and an additive character. Perhaps the simplest hybrid sums are the Gaussian sums:

Definition 1 *Let ψ be an additive character and χ a multiplicative character of $\mathbb{F}_{2^n}$. Then the Gaussian sum $G(\chi, \psi)$ is defined by*

$$G(\chi, \psi) = \sum_{x \in \mathbb{F}_{2^n}} \psi(x)\,\chi(x).$$

The following result about Gaussian sums is basic; elementary proofs can be found in [32, Theorem 5.11] and [22, Proposition 8.2.2].

Result 2 *Let ψ be a non-trivial additive character and χ a non-trivial multiplicative character of $\mathbb{F}_{2^n}$. Then*

$$|G(\chi, \psi)| = 2^{n/2}.$$

Why should this result be surprising? The sum is of size $2^{n/2}$, only slightly bigger than the square root of the size of the domain over which the sum is taken. Moreover, the sum has exactly this absolute value for *every* pair of non-trivial characters.

Hybrid exponential sums with polynomial arguments have also been considered; the following is a useful general purpose bound on such sums, again due to Weil [66].

Result 3 *Let χ be a non-trivial multiplicative character of $\mathbb{F}_{2^n}$ of order d with $d|(2^n - 1)$. Let ψ be a non-trivial additive character of $\mathbb{F}_{2^n}$. Let $f(x) \in \mathbb{F}_{2^n}[x]$ have m distinct roots and $g(x) \in \mathbb{F}_{2^n}[x]$ have degree r. Suppose that* $\gcd(d, \deg f) = 1$ *and that r is odd. Then*

$$\left| \sum_{x \in \mathbb{F}_{2^n}} \psi(g(x))\ \chi(f(x)) \right| \leq (m + r - 1)2^{n/2}.$$

Here, the technical conditions on the polynomials f and g are needed to rule out various degenerate cases. They can be replaced by weaker conditions | see [54, Theorem 2G, p.45]. We emphasise again that the bound shows that the hybrid sums are much smaller than the size of the eld over which the sum is taken.

4 Application: Minimum Distance of Dual BCH Codes

When $t = 1$, the code C_t is called the simplex code. The minimum Hamming distance of this code is exceedingly simple to calculate. Recall that the code is linear, so we need to nd the minimum Hamming weight of a non-zero codeword of C_1. Now a non-zero codeword c has components of the form $\mathrm{tr}_1^n(b^{\ i})$ where $b \in \mathbb{F}_{2^n}$ and $0 \quad i < 2^n - 1$. As i runs through the range $0; 1; \ldots; 2^n - 2$, so ^{i} runs over the whole of $\mathbb{F}_{2^n}$, the non-zero elements of $\mathbb{F}_{2^n}$.

Consider the exponential sum (1):

$$\begin{aligned} 0 &= \sum_{x \in \mathbb{F}_{2^n}} {}_b(x) \\ &= 1 + \sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{tr}_1^n(bx)} \\ &= 1 + \#\{x \in \mathbb{F}_{2^n} : \mathrm{tr}_1^n(bx) = 0\} - \#\{x \in \mathbb{F}_{2^n} : \mathrm{tr}_1^n(bx) = 1\} \\ &= 1 + (2^n - 1 - \mathrm{wt}_H(c)) - \mathrm{wt}_H(c) \\ &= 2^n - 2\mathrm{wt}_H(c) \end{aligned}$$

Here we have used the fact that the number of components in which c equals 0 is just the code length less the Hamming weight of c. It follows from our last equality that $\mathrm{wt}_H(c) = 2^{m-1}$. So every non-zero codeword of C_1 has Hamming weight equal to 2^{m-1}, and the minimum distance of the code is also 2^{m-1}.

We can apply the same technique, and the Weil bound, to bound the minimum distance of the code C_t. Recall that a non-zero codeword c_g of C_t comes from a non-zero polynomial $g(x)$ with zero constant term and of odd degree at most $2t - 1$. Reversing the steps in the previous calculation, we get:

$$2^n - 2\mathrm{wt}_H(c_g) = \sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{tr}_1^n(g(x))} = \sum_{x \in \mathbb{F}_{2^n}} {}_1(g(x)) \qquad (4)$$

But this last sum is bounded in absolute value by $(2t - 2)2^{n=2}$ according to Result 1. We deduce that

$$2^{n-1} - (t-1)2^{n=2} \quad \mathrm{wt}_H(c_g) \quad 2^{n-1} + (t-1)2^{n=2};$$

and the following theorem is now obvious:

Theorem 4. *Suppose* $1 \quad 2t - 1 \quad 2^{dn=2e} + 1$. *Then the minimum Hamming distance of* C_t *is at least* $2^{n-1} - (t-1)2^{n=2}$.

This bound can be improved in certain cases [38].

5 Application: Sequence Sets with Low Periodic Correlations

The correlation properties of sets of binary sequences are important in Code-Division Multiple-Access (CDMA) spread-spectrum communications as well as in ranging and synchronisation applications.

We begin in this section by de ning the periodic correlation functions for sequences and then stating a basic sequence design problem. This is motivated by a simpli ed description of how sequences with favourable correlation properties are used in CDMA communications. Then we de ne a class of sequences, the m-sequences, and look at their correlation properties. Finally, we show how exponential sums can be used to bound the correlations of some sets of sequences obtained from m-sequences and the dual BCH codes.

5.1 Periodic Correlation Functions

Let $u = u_0; u_1; u_2; \ldots$ and $v = v_0; v_1; v_2; \ldots$ be two complex-valued sequences of period N (by which we mean $u_{i+N} = u_i$ and $v_{i+N} = v_i$ for all $i \quad 0$). We de ne the *periodic cross-correlation* of u and v at a relative shift $\quad , 0 \quad < N$, to be:

$$\mathrm{CC}(u; v)(\ ) = \sum_{i=0}^{N-1} x_i \ \overline{y_{i+}}:$$

and call $CC(u; v)(\ )$ the *periodic cross-correlation function* of u and v. This function is a measure of the similarity of the sequences u and v at various shifts. We also de ne the *periodic auto-correlation* of u at a shift $\quad , 0 \quad < N$, to be:

$$\mathrm{AC}(u)(\ ) = \mathrm{CC}(u; u)(\ ):$$

The periodic auto-correlation is a measure of the self-similarity of the sequence u when compared to shifts of itself. The auto-correlation of u at shift 0, $\mathrm{AC}(u)(0) = \sum_{i=0}^{N-1} |u_i|^2$, is in many applications a measure of the energy in the transmitted signal corresponding to sequence u. The auto-correlations of u at non-zero shifts are usually called *non-trivial auto-correlations*.

5.2 A Simpli ed Model for CDMA Communications

We next discuss a simpli ed model for CDMA communications. In our model, we have K users, all transmitting data simultaneously and without coordination or synchronisation on the same channel. The transmitted signal is the sum of users' individual signals, and is corrupted by noise. The users are transmitting to a single receiver, whose job it is to take the received signal and process it to obtain individual user's data.

Each user is assigned a *spreading code*, which in our model is just a complex-valued sequence of period N. User j is assigned the sequence

$$u^j = u_0^j; u_1^j; u_2^j; \ldots$$

To send a data bit $a_j \in \{0,1\}$, user j actually transmits the sequence $(-1)^{a_j}u^j$, i.e. the sequence of bits:

$$(-1)^{a_j}u^j_0, (-1)^{a_j}u^j_1, (-1)^{a_j}u^j_2, \ldots$$

In other words, he transmits a $\{+1,-1\}$-version of his data bit *spread* by his sequence u^j.

The received signal can be modelled by a sequence $s = s_0, s_1, s_2, \ldots$ where

$$s_i = \sum_{j=0}^{K-1} (-1)^{a_j}u^j_{i+\tau_j}.$$

Here τ_j is the *delay* of user j relative to the receiver. Because the users are transmitting in an uncoordinated fashion, these delays are unknown to the receiver. We have also assumed an ideal situation where there the transmission channel is noiseless.

Now suppose the receiver wishes to estimate the data bit a_ℓ for user ℓ. The receiver calculates, for each τ with $0 \le \tau < N$, the function $CC(s, u^\ell)(\tau)$. Notice that:

$$\begin{aligned} CC(s,u^\ell)(\tau) &= \sum_{i=0}^{N-1}\left(\sum_{j=0}^{K-1}(-1)^{a_j}u^j_{i+\tau_j}\right)\overline{u^\ell_{i+\tau}} \\ &= \sum_{j=0}^{K-1}\left(\sum_{i=0}^{N-1}((-1)^{a_j}u^j_{i+\tau_j}\,\overline{u^\ell_{i+\tau}}\right) \\ &= (-1)^{a_\ell}AC(u^\ell)(\tau-\tau_\ell) + \sum_{j\neq\ell}(-1)^{a_j}CC(u^j,u^\ell)(\tau-\tau_j). \end{aligned}$$

Now suppose that all the non-trivial autocorrelations and all the cross-correlations of the sequences u^j are small. In other words, we assume that for every ℓ and $\tau \neq 0$, $AC(u^\ell)(\tau)$ is small and that for every j, ℓ and every τ, $CC(u^j, u^\ell)(\tau)$ is small.

Then when $\tau = \tau_\ell$, the expression above for $CC(s, u^\ell)(\tau)$ has a first term $(-1)^{a_\ell}AC(u^\ell)(0)$ whose sign reveals a_ℓ, and whose relatively large magnitude dominates the remaining correlation terms. When $\tau \neq \tau_\ell$ then all the terms are small. Thus the receiver, after calculating $CC(s, u^\ell)(\tau)$ for each τ should focus on the largest resulting correlation value to estimate the delay τ_ℓ and use the sign of this value to estimate the data bit a_ℓ.

Clearly, the success of this approach to transmitting information crucially depends on the term $(-1)^{a_\ell}AC(u^\ell)(0)$ not being swamped by the other correlations. In other communications applications, for example, in synchronisation, single sequences with small non-trivial auto-correlations are called for. Thus we are motivated to consider the following basic sequence design problem:

For a set U containing K complex-valued sequences of period N, define

$$AC_{max}(U) = \max_{u\in U;\ \tau\neq 0} |AC(u)(\tau)|,$$

$$CC_{max}(U) = \max_{u \neq v \in U;\ 0 \le \tau < N} |CC(u, v)(\tau)|.$$

and

$$C_{max}(U) = \max\{AC_{max}, CC_{max}\}.$$

Find sequence sets U which minimise $AC_{max}(U)$ (when $K = 1$) or $C_{max}(U)$ (when $K > 1$).

There are a number of lower bounds on $C_{max}(U)$ for sequence sets consisting of K sequences of period N [30, 51, 59, 67] which can be used to judge how good a particular design is.

For further details of how sequence sets with favourable correlation properties can be exploited in communications applications, see [9, 13, 14, 53, 57, 60].

5.3 The m-Sequences and Their Periodic Correlations

We introduce a class of sequences, called the m-sequences, which have good auto-correlation properties.

Let α be a primitive element of $\mathbb{F}_{2^n}$. The sequence $s = s_0, s_1, \ldots$ with

$$s_i = tr_1^n(\alpha^i)$$

is called a *binary m-sequence*. Because α is an element of period $2^n - 1$ in $\mathbb{F}_{2^n}$, the sequence s has period $2^n - 1$. Notice that taking one period of an m-sequence gives us a length $2^n - 1$ vector that is a codeword of the simplex code. From the equi-distribution property of the trace map, we see that s contains 2^{n-1} ones and $2^{n-1} - 1$ zeros in a period. We de ne a related $\{+1, -1\}$-valued sequence u of period $2^n - 1$ by $u_i = (-1)^{s_i}$.

Lemma 2 *Let s be a binary m-sequence of period $2^n - 1$ and let u be the corresponding complex-valued sequence. Then for $\tau \not\equiv 0 \bmod 2^n - 1$, we have $AC(u)(\tau) = -1$.*

Proof. We have:

$$\begin{aligned}
AC(u)(\tau) &= \sum_{i=0}^{2^n-2} (-1)^{s_i}\ \overline{(-1)^{s_{i+\tau}}} \\
&= \sum_{i=0}^{2^n-2} (-1)^{tr_1^n(\alpha^i)}\ (-1)^{tr_1^n(\alpha^{i+\tau})} \\
&= \sum_{i=0}^{2^n-2} (-1)^{tr_1^n[(1+\alpha^\tau)\alpha^i]} \\
&= \sum_{x \in \mathbb{F}_{2^n}} \chi_\gamma(x) \quad \text{where } \gamma = 1 + \alpha^\tau \\
&= -1 + \sum_{x \in \mathbb{F}_{2^n}} \chi_\gamma(x) \\
&= -1
\end{aligned}$$

where we have again used (1) and the fact that $\gamma = 1 + \alpha^{\delta} \neq 0$ provided $\alpha^{\delta} \neq 0 \bmod 2^n - 1$. □

Of course, for m-sequences to be useful in applications, we need to have a convenient method for generating them. It turns out that an m-sequence of period $2^n - 1$ satisfies a linear recurrence relation of degree n and can be generated using a simple electronic device called a Linear Feedback Shift Register. For more details, see [31, 32].

5.4 Sequence Sets from m-Sequences

Since the auto-correlations of m-sequences are so neatly described, might we not expect the cross-correlations of two different m-sequences to be calculable? In fact, we can always express such cross-correlations as a Weil exponential sum, as we now show.

Let β be a second primitive element of $\mathbb{F}_{2^n}$ and define a second m-sequence $t = t_0, t_1, \ldots$ by $t_i = \mathrm{tr}_1^n(\beta^i)$. Since α and β are both primitive, we can write $\beta = \alpha^d$ for some d with $\gcd(d, 2^n - 1) = 1$. We also define a $\{+1, -1\}$-valued sequence v corresponding to t.

Now consider the cross-correlation:

$$\begin{aligned} CC(v,u)(\tau) &= \sum_{i=0}^{2^n-2} (-1)^{\mathrm{tr}_1^n(\alpha^{di})} (-1)^{\mathrm{tr}_1^n(\alpha^{i+\tau})} \\ &= \sum_{i=0}^{2^n-2} (-1)^{\mathrm{tr}_1^n(a\alpha^i + \alpha^{di})} \quad \text{where } a = \alpha^{\tau} \\ &= -1 + \sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{tr}_1^n(ax + x^d)}. \end{aligned}$$

More generally, if we consider the sequence set G consisting of u, v and the term-by-term product of u with all the cyclic shifts of v, then any cross- or auto-correlation of sequences in this set of size $2^n + 1$ can be expressed in a similar way as a Weil exponential sum involving functions of the form $g(x) = ax + x^d$.

In certain special cases, not only can the sums (and therefore $C_{\max}(G)$) be bounded, but the spectrum of values taken on by the correlations as the pairs of sequences range over G can be calculated explicitly.

For example, when $d = 2^k + 1$, n is odd and $\gcd(n, k) = 1$, Gold [15, 16] showed that $C_{\max}(G) = t(n)$ where $t(n) = 1 + 2^{\lfloor (n+2)/2 \rfloor}$, and that the values taken on by non-trivial correlations of sequences in G, called a Gold code, lie in the set $\{-1, -t(n), t(n) - 2\}$. It can also be shown that the set G has optimal correlation properties, meeting a lower bound of [59] on the correlations of *binary* sequence sets.

More general results of this type are summarised in [53, Theorem 1]. The analysis used to prove these correlation results is interesting in itself, though not a direct application of exponential sums. One shows that, by choosing a basis for $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$, the functions $\mathrm{tr}_1^n(ax + x^d)$ can be described by quadratic forms in n variables over $\mathbb{F}_2$ for these special d. Essentially, this is because the exponent

d has a binary expansion of weight 2. According to the theory of Dickson [34, Theorems 4 and 5, Chapter 15], the distribution of values taken on by such a form is determined by an invariant called the *rank* of the form. This rank, and therefore the spectrum of correlation values, depends on the parameters k and n and can be explicitly calculated. We refer the reader to [35] for a nice treatment of this topic.

Also of special note are very recent results [10, 11, 20] which together resolve the long-standing Welch and Niho conjectures concerning the correlation spectra in the cases $d = 2^m + 3$, $n = 2m + 1$ and $d = 2^{2r} + 2^r - 1$, $4r + 1 = 0 \bmod n$, respectively.

Among the many papers considering other families of sequence sets with good correlation properties are [3, 17, 24, 40, 41, 42, 43, 50, 56]. See also the survey by Helleseth and Kumar in [47, Vol. II].

5.5 Sequence Sets from Dual BCH Codes

We will examine in more detail the correlations a family of sequences obtained from the dual BCH codes. Let G_t denote the set of polynomials

$$G_t = \{x + g_3x^3 + \quad + g_{2t-1}x^{2t-1} : g_i \in \mathbb{F}_{2^n}\}.$$

For each $g \in G$, define a period $2^n - 1$, binary sequence s_g with terms $(s_g)_i$ where

$$(s_g)_i = \mathrm{tr}_1^n(g(\ ^i)); \quad i \quad 0$$

and let u_g be the $\{+1; -1\}$-valued sequence corresponding to s_g. We define two sequence sets S_t; U_t by:

$$S_t = \{s_g : g \in G_t\}; \quad U_t = \{u_g : g \in G_t\}.$$

We see that the sequences in U_t are $\{+1; -1\}$ versions of the sequences in S_t and that $|S_t| = |U_t| = 2^{n(t-1)}$. Single periods of the sequences of S_t are just codewords of the dual BCH code C_t that are all distinct under cyclic shifting (this explains the restriction to polynomials with coefficient of x equal to 1 in the definition of G_t). For example, when $t = 1$, the single sequence in S_t is just an m-sequence coming from the simplex code.

Consider a pair of sequences u_g and u_h from U_t, where

$$g(x) = x + \sum_{i=0}^{t-1} g_{2i+1}x^{2i+1}; \quad h(x) = x + \sum_{i=0}^{t-1} h_{2i+1}x^{2i+1}.$$

Then a straightforward calculation shows that

$$\mathrm{CC}(u_g; u_h)(\ ) = -1 + \sum_{x \in \mathbb{F}_{2^n}} {}_1(e(x))$$

where $e(x)$ is the polynomial

$$(1 + \quad)x + \sum_{i=0}^{t-1} (g_{2i+1} + \quad ^{(2i+1)}\ h_{2i+1})x^{2i+1}.$$

So the correlations of sequences in our set can be expressed as Weil exponential sums. Notice that if $\not\equiv 0 \bmod 2^n - 1$, or if $g(x) \not\equiv h(x)$ and $= 0 \bmod 2^n - 1$, then $e(x)$ is a non-zero polynomial of odd degree and lies in the set G_t. A direct application of Result 1 yields:

Theorem 5. *The sequence set U_t contains $2^{n(t-1)}$ sequences of period $2^n - 1$ and satisfies $C_{\max}(U_t) \quad 1 + (t-1)2^{(n+2)=2}$.*

There is a simple relationship between the correlations $\mathrm{CC}(u_g; u_h)(\ )$ and the Hamming weights of words of C_t, as we now show. The analysis above shows that c_e is in the code C_t. We already know from (4) that

$$2^n - 2\mathrm{wt}_H(c_e) = -1 + \sum_{x \in \mathbb{F}_{2^n}} {}_1(e(x))$$

so we have

$$\mathrm{CC}(u_g; u_h)(\ ) = 2^n - 2\mathrm{wt}_H(c_e);$$

an identity linking the correlations of pairs of sequences in U_t with the Hamming weight of a related codeword in C_t.

When $t = 1$, Theorem 5 gives us the correct bound on the non-trivial periodic auto-correlations of m-sequences. When $t = 2$, the polynomials in G_2 are of the form $x + g_3x^3$ and when n is odd, the sequence set U_2 is a subset of the Gold code with $d = 3$ (omitting just the sequence v from the Gold code). The theorem gives a bound on $\mathrm{C}_{\max}$ which is slightly weaker than Gold's bound. When n is even, we get a new sequence set satisfying $\mathrm{C}_{\max}(U_2) \quad t(n)$. This set is a special case of what are called in [53] *Gold-like codes*.

6 Application: Aperiodic and Partial Correlations

Traditionally, it is the periodic auto- and cross-correlations of sequence sets discussed above that have received most attention in the literature. But *aperiodic* and *partial* correlations of sequences emerge as being at least if not more important parameters to study when more realistic models of communications systems are considered.

As usual, u and v will denote complex-valued sequences of period N. Aperiodic correlations are correlations taken over only nite sequences: suppose $0 \quad < N$; Then the *aperiodic cross-correlation* between u and v at a relative shift , $0 \quad < N$ is de ned to be

$$\mathrm{ACC}(u; v)(\ ) = \sum_{i=0}^{N-1} u_i \overline{v_{i+}};$$

We can also de ne the *aperiodic auto-correlation function* of sequence u via:

$$\mathrm{AAC}(u)(\ ) = \mathrm{ACC}(u; u)(\ ):$$

Aperiodic correlations are important in, for example, CDMA systems where consecutive periods of a spreading sequence are used to spread di erent data bits [53, Section V].

Partial correlations are correlations taken over only subsequences of sequences: suppose $0 \quad j; \; ; ' < N$. Then the *partial cross-correlation* between u and v of period N over the subsequence of length $'$ beginning at position j and with relative shift , denoted $\mathrm{PCC}(u;v)(j;k;')$, is de ned by

$$\mathrm{PCC}(u;v)(j; \; ;') = \sum_{i=0}^{'-1} u_{j+i} \; \overline{v_{j+i+}}:$$

Similarly, we de ne the *partial auto-correlation* by:

$$\mathrm{PAC}(u)(j; \; ;') = \mathrm{PCC}(u;u)(j; \; ;'):$$

Notice that when $' = N$, the partial correlations revert to the usual periodic correlations. Partial correlations arise as natural parameters of study in CDMA systems where many (possibly several hundred) data bits are spread by each copy of a user's spreading sequence [48, 49, 55] and in systems [9, 14], where a long sequence (typically an m-sequence) is used for synchronisation, but where correlations are computed over only a short subsequence of that sequence for faster acquisition.

So we are motivated to study the problem of constructing sequence sets for which the maximum value of non-trivial aperiodic or partial correlation is as small as possible. But these correlations are much less well understood than periodic correlations. One reason for this is that in the periodic case, we can use the algebraic structure of, for example, a nite eld to de ne sequences and their periodic correlations are then calculated from certain sums taken over the whole nite eld. We have seen an example of this in our calculation of the periodic auto-correlations of m-sequences. In contrast, the corresponding aperiodic and partial correlations for such sequences lead to sums over only part of the nite eld and the exponential sum results are no longer directly applicable. Nevertheless, as we show next, hybrid exponential sums can be employed to obtain bounds for these new correlations. We concentrate on partial correlations, though very similar methods can be used to handle aperiodic cases too. We make use of a technique called the Polya-Vinogradov method. For a survey of applications of this technique in number theory and communications see [63] and for related results [29, 33, 52].

For $i \quad 0$ we de ne $(j;')_i$ by:

$$(j;')_i = \begin{cases} 1 \text{ if } j + kN \quad i < j + kN + ', \text{ for some } k \; 2 \; \mathbb{Z}, \\ 0 \text{ otherwise.} \end{cases}$$

Then the sequence $(j;')$ has period N and we can write:

$$\mathrm{PCC}(u;v)(j; \; ;') = \sum_{i=0}^{N-1} (j;')_i \; u_i \overline{v_{i+}}: \tag{5}$$

Next we bring the Discrete Fourier Transform (DFT) of the sequence $(j; \ell)$ into play. Generally, if $u = u_0, u_1, \ldots$ is a complex-valued sequence of period N and $\omega = \exp(2\pi i/N)$ then the sequence

$$\hat{u} = \hat{u}_0, \hat{u}_1, \ldots$$

with terms

$$\hat{u}_k = \sum_{i=0}^{N-1} u_i \omega^{ik}, \qquad k = 0$$

is called the *Discrete Fourier Transform (DFT) of* u. It is a simple exercise in manipulation of geometric series to show that u can be recovered from $\hat{u}$ via the *Inverse DFT*:

$$u_i = \frac{1}{N} \sum_{k=0}^{N-1} \hat{u}_k \omega^{-ik}, \qquad i = 0.$$

The sequence $(j; \ell)$ has a particularly nice DFT:

Lemma 3 *Let* $(j; \ell)$ *be defined as above. Then*

$$\widehat{(j;\ell)}_k = \begin{cases} \omega^{jk} \dfrac{\omega^{\ell k} - 1}{\omega^k - 1} & \text{if } k \neq 0 \bmod N \\ \ell & \text{if } k = 0 \bmod N \end{cases}$$

Replacing terms of $(j; \ell)$ in (5) by expressions involving the inverse DFT, we get:

$$\mathrm{PCC}(u,v)(j;\ell;\tau) = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} \widehat{(j;\ell)}_k \omega^{-ik} u_i \overline{v_{i+\tau}}$$

and then reversing the order of summation:

$$\mathrm{PCC}(u,v)(j;\ell;\tau) = \frac{1}{N} \sum_{k=0}^{N-1} \widehat{(j;\ell)}_k \left(\sum_{i=0}^{N-1} u_i \overline{v_{i+\tau}} \omega^{-ik} \right).$$

Now consider a non-trivial partial auto-correlation of the $\{+1, -1\}$-valued version of a period $N = 2^n - 1$ m-sequence. We take $u = v$ and $u_i = (-1)^{\mathrm{tr}_1^n(\alpha^i)}$ for some primitive element $\alpha \in \mathbb{F}_{2^n}$ in the above expression. We also take $\ell \neq 0 \bmod 2^n - 1$ and define $\gamma = 1 + \alpha^\tau$ so that $\gamma \neq 0$. Then we have, for $k \neq 0$,

$$\begin{aligned} \sum_{i=0}^{N-1} u_i \overline{u_{i+\tau}} \omega^{-ik} &= \sum_{i=0}^{N-1} (-1)^{\mathrm{tr}_1^n(\gamma \alpha^i)} \omega^{-ik} \\ &= \sum_{x \in \mathbb{F}_{2^n}} \chi_\gamma(x) \psi_{-k}(x) \qquad \text{(substituting } x = \alpha^i\text{)} \\ &= G(\chi_\gamma, \psi_{-k}); \end{aligned}$$

a Gaussian sum. Separating the contributions due to $k = 0$ and $k \neq 0$ and using $\widehat{(j; \cdot)}_0 = \cdot$ we get:

$$\mathrm{PAC}(u)(j; \ ; \cdot) = \frac{\cdot}{N} \ \mathrm{AC}(u)(\ ) + \frac{1}{N} \sum_{k=1}^{N-1} \widehat{(j; \cdot)}_k \ G(\ _\gamma; \ _{-k})$$

showing that the partial auto-correlation of an m-sequence can be expressed as a sum involving Gaussian sums and a periodic auto-correlation term. To get a bound on $|\mathrm{PAC}(u)(j; \ ; \cdot)|$, we use the facts that $\mathrm{AC}(u)(\ ) = -1$ and $|G(\ _\gamma; \ _{-k})| = 2^{n=2}$ with the triangle inequality. We obtain:

$$|\mathrm{PAC}(u)(j; \ ; \cdot)| \quad \frac{\cdot}{N} + \frac{2^{n=2}}{N} \sum_{k=1}^{N-1} \widehat{(j; \cdot)}_k \tag{6}$$

and we are now left with the problem of estimating a sum involving terms of the DFT of $(j; \cdot)$. A bound of $N \log N$ for this sum in the case $j = 0$ is reported by Vinogradov [65]; improvements in the constant have been obtained since by Sarwate [52]. Since $|\widehat{(j; \cdot)}_k| = |\widehat{(0; \cdot)}_k|$, Vinogardov's bound applies to the more general case of $j \neq 0$ too. Combining Vinogradov's bound with inequality (6) we obtain:

Theorem 6. *Let u be the $\{+1; -1\}$-valued version of an m-sequence of period $N = 2^n - 1$. Then for any j and any $\neq 0$, we have*

$$|PAC(u)(j; \ ; \cdot)| \quad 1 + (N + 1)^{1=2} \log N:$$

An almost identical method can be used to bound the partial and aperiodic correlations of the sequence set U_t. The only di erences are that the Gauss sum is replaced by a hybrid exponential sum with polynomial argument, we use Result 3 rather than Result 2 to bound this sum, and we use Theorem 5 to bound the periodic cross-correlations of sequences from U_t. See also [45] for applications of the Polya-Vinogradov method to other sequence families and for a survey of other approaches to working with the partial correlations of sequence sets.

In this Section, we started with sequences with favourable periodic correlation properties and then looked at their partial correlations. But computational evidence [45] suggests that the bounds we can obtain using the Polya-Vinogradov method are rarely tight. Is it possible to design sequence families with better partial correlations from scratch?

7 Application: The Power Control Problem in OFDM

Orthogonal Frequency Division Multiplexing (OFDM) is a communications technique that has recently seen rising popularity in wireless and wire-line communications [1, 2, 5, 6]. OFDM-based solutions have important advantages over more traditional data transmission approaches: OFDM has greater inherent resistance to a certain kind of noise called multi-path interference that plagues

wireless communications, while implementations of OFDM systems can be realised using standard digital signal processing techniques and can avoid the use of an expensive channel equalisation process.

In an OFDM system, the transmitted signal is a sum of phase-shifted sinusoidal carriers, the phase shifts carrying the data. The data itself is coded because of channel noise. Given a length N binary code C and a codeword $c = (c_0, c_1, \ldots, c_{N-1}) \in C$, the transmitted signal at time t for the codeword c can be modelled as the real part of the sum

$$\sum_{j=0}^{N-1} (-1)^{c_j} e^{(2\pi i)jt}.$$

If we define a degree $N-1$ polynomial $c(z)$ by

$$c(z) = (-1)^{c_0} + (-1)^{c_1} z + \cdots + (-1)^{c_{N-1}} z^{N-1},$$

then the OFDM signal at time t is then just the real part of $c(e^{2\pi it})$ so that the transmitted signal is related to the values of polynomials on $|z| = 1$, the unit circle in the complex plane.

The *envelope power* of the OFDM signal at time t is defined to be $|c(e^{2\pi it})|^2$. The mean value of this function as t ranges over $[0, 1]$ is equal to N (this can be shown simply by computing the integral of $|c(z)|^2 = c(z)\,\overline{c(1/z)}$ around the unit circle). So we define the *peak-to-mean envelope power ratio* or PMEPR of the OFDM signal to be:

$$\mathrm{PMEPR}(c) = \frac{1}{N} \max_{|z|=1} |c(z)|^2$$

and the PMEPR of the code C to be:

$$\mathrm{PMEPR}(C) = \frac{1}{N} \max_{c \in C} \max_{|z|=1} |c(z)|^2.$$

The number $\mathrm{PMEPR}(C)$ is a measure of the dynamic range of the power in the OFDM signals that are obtained from the code C. It is desirable to work with codes C which have 'small' values of PMEPR, acutely so for low-cost wireless applications. This is because a low value of PMEPR leads to signals that can be amplified by cheap electronic components without too much distortion being introduced, and which make efficient use of regulatory limits that are commonly imposed on the power of wireless signals. Notice that if $c = (0, 0, \ldots, 0)$, then $\mathrm{PMEPR}(c) = N$. In fact this is the largest value of PMEPR that can occur, so by 'small' in this context we mean substantially less than N. We can summarise the *OFDM power control problem* as:

> Find binary codes C which simultaneously are good error correcting codes and have $\mathrm{PMEPR}(C)$ small.

For a summary of previous work on this problem and references to the engineering literature, see [46]. Here we will show how hybrid exponential sums (and a little analysis) can be used to obtain bounds on the PMEPRs of the non-zero words of dual BCH codes. A similar analysis for many other code families can be found in [46]. We also recommend [7, 44] for a completely di erent approach to the power control problem.

Consider then a general non-zero codeword c_g of the length $N = 2^n - 1$ dual BCH code C_t. We have

$$(c_g)_j = \mathrm{tr}_1^n(g(\ ^j)); \quad 0 \quad j < N;$$

where g is a polynomial of degree $2t - 1$ over $\mathbb{F}_{2^n}$ and is primitive in $\mathbb{F}_{2^n}$. So the polynomial $c_g(z)$ corresponding to the codeword c_g is

$$c_g(z) = \sum_{j=0}^{N-1} (-1)^{\mathrm{tr}_1^n(g(\ ^j))} z^j$$

and we want to obtain a bound for $\max_{|z|=1} |c_g(z)|^2$. Notice that at $z = e^{(2\ i)\ '=N}$, an N-th root of unity, we have

$$c_g(e^{(2\ i)\ '=N}) = \sum_{j=0}^{N-1} (-1)^{\mathrm{tr}_1^n(g(\ ^j))} e^{(2\ i)j\ '=N} = \sum_{x \in \mathbb{F}_{2^n}} {}_1(g(x))\ \ '(x)$$

| a hybrid exponential sum with polynomial argument. When $' \neq 0$, the sum satis es the conditions of Result 3 with $f(x) = x$ and $m = 1$. So we can immediately say:

$$|c_g(e^{(2\ i)\ '=N})| \quad (2t - 1)2^{n=2} \quad \text{for } ' \neq 0.$$

On the other hand, for $' = 0$, we get:

$$|c_g(1)| = \sum_{j=0}^{N-1} (-1)^{\mathrm{tr}_1^n(g(\ ^j))} = -1 + \sum_{x \in \mathbb{F}_{2^n}} {}_1(g(x)) \quad 1 + (2t - 2)2^{m=2}$$

according to Result 1. Thus we see that, at the N-th roots of unity, the polynomial $c_g(z)$ has absolute value no greater than $(2t - 1)2^{n=2}$.

Now we convert this bound holding at the N-th roots of unity to a bound that is valid on the entire unit circle. We make use of the following lemma, obtained by bounding the coe cients that occur in a Lagrange interpolation of a degree $N - 1$ polynomial from the N-th roots of unity to $|z| = 1$:

Lemma 4 *[46] Let $c(z)$ be a degree $N - 1$ polynomial and write $\gamma = e^{(2\ i)=N}$. Then*

$$\max_{|z|=1} |c(z)| \quad \frac{2}{} \log(2N) + 2 \quad \max_{0 \ j < N} |c(\gamma^j)|:$$

The lemma shows that the interpolation can be achieved at the expense of a factor of $O(\log N)$. Combining the lemma with our exponential sum estimate, we obtain:

Theorem 7. *Let C_t denote the code obtained by removing the all-zero word from the length $N = 2^n - 1$ dual BCH code C_t. Then*

$$PMEPR(C_t) \quad \frac{N+1}{N} \quad (2t-1)^2 \quad \frac{2}{} \log(2N) + 2 \quad ^2 :$$

Thus the theorem shows that the PMEPR of the dual BCH codes is $O((\log N)^2)$ for xed t and large N, clearly much better than the worst case PMEPR value of N. Notice however that for xed t, we have shown the dual BCH codes to have normalised envelope power at most $(2t-1)^2$ at $t = `=N$, $0 \quad ` < N$. The factor of $(\log N)^2$ in the theorem comes from our use of Lagrange interpolation. This indicates that there is room for improvement in our bound on PMEPR(C_t). Indeed a result of [46] shows that there do exist codes which are asymptotically good (i.e. their normalised minimum distances and rates are both bounded away from zero as $N \; ! \; 1$) and which have PMEPR growing only as $O(\log N)$. Unfortunately, the proof is non-constructive.

For a collection of open problems related to the power control problem, see the closing comments of [46].

8 Further Applications and Literature

In this section, we provide brief notes and pointers to some of the literature on exponential sums and applications that we have not touched upon in earlier sections.

Kloosterman sums [32, Chapter 5] in characteristic 2 are exponential sums of the form

$$\sum_{x \in \mathbb{F}_{2^n}} (ax + \frac{b}{x}); \qquad a;b \; 2 \; \mathbb{F}_{2^n} :$$

Their evaluation is intimately connected with counting points on *elliptic curves*. If $a;b$ are not both zero, the sum can be bounded by $2 \; 2^{n=2}$. In much the same way as for the dual BCH codes, we can de ne a Kloosterman code to be the set of words $c_{a;b}$ where

$$(c_{a;b})_i = \mathrm{tr}_1^n(a^{\;i} + \frac{b}{i}); \qquad 0 \quad i < 2^n - 1; \; a;b \; 2 \; \mathbb{F}_{2^n} :$$

It is easy to mimic our previous arguments to show that the Kloosterman code has minimum distance at least $2^{n-1} - 2^{n=2}$. In fact the complete distribution of weights occuring in this code has been calculated [28]. We can also de ne sequence sets with favourable periodic correlations using the Kloosterman code.

The sequences are term-by-term sums of an m-sequence corresponding to a primitive element and the shifts of its *reciprocal* m-sequence which comes from the element $^{-1}$.

Kumar and Moreno [27] have used Deligne's results bounding the numbers of points on algebraic varieties to construct sequence families with good correlations. Their sequences have terms that are powers of a p-th root of unity (rather than the $f+1; -1g$-valued sequences we have studied here). They have also used Deligne's results to bound the minimum distances of certain binary codes [37]. These two papers contain a wealth of other references to previous work on codes and sequence designs. The Carlitz-Uchiyama/Weil bound has also been extended using Deligne's theorem and applied to coding theory [39].

Exponential sums have been applied to the study of the covering radii of BCH codes [19, 62] and Goppa codes [21, Section 12 and 13].

Exponential sums over Galois *rings* (rather than Galois elds) have recently received a lot of attention, beginning with the influential paper [18]. There are analogues of the Carlitz-Uchiyama/Weil bound [25] and of Result 3 for hybrid exponential sums [58]. These bounds have been used to construct quaternary codes with large minimum distances and to design quaternary (more generally $p^{'}$-ary) sequence families with low periodic and aperiodic correlations. See also [26, 64] and the references therein for related sequence designs. The hybrid sum results of [58] were used in [46] to bound the PMEPR properties of some quaternary codes.

As we have already noted, a complete survey of the whole area of exponential sums and their applications can be found in [21]. We hope this paper serves as a useful introduction to what is a fascinating and thriving area.

References

[1] M. Alard and R. Lasalle. Principles of modulation and channel coding for digital broadcasting for mobile receivers. *EBU Review*, 224: 47{69, Aug. 1987.

[2] J.A.C. Bingham. Multicarrier modulation for data transmission: an idea whose time has come. *IEEE Commun. Magazine*, 28(1): 5{14, May 1990.

[3] S. Boztas and P.V. Kumar. Binary sequences with Gold-like correlation but larger linear span. *IEEE Trans. Inform. Theory*, IT-40(2): 532{537, March 1994.

[4] L. Carlitz and S. Uchiyama. Bounds for exponential sums. *Duke Math. J.*, 24:37{41, 1957.

[5] P.S. Chow, J.M. Cio , and J.A.C. Bingham. DMT-based ADSL: concept, architecture, and performance. In *IEE Colloquium on 'High Speed Access Technology and Services, Including Video-on-Demand'*, pages 3/1{6, Oct. 1994.

[6] L.J. Cimini, Jr. Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing. *IEEE Trans. Commun.*, 33:665{675, July 1985.

[7] J.A. Davis and J. Jedwab. Peak-to-mean power control in OFDM, Golay complementary sequences and Reed-Muller codes. *IEEE Trans. Inform. Theory*, to appear Nov. 1999.

[8] P. Deligne. La conjecture du W. *Publ. Math. IHES*, 43: 273{307, 1974.

[9] R. C. Dixon. *Spread Spectrum Systems with Commercial Applications (3rd edition)*. Wiley{Interscience, New York, 1994.
[10] H. Dobbertin. Almost perfect nonlinear power functions on $GF(2^n)$: the Welch case. *IEEE Trans. Inform. Theory*, IT-45: 1271{1275, 1999.
[11] H. Dobbertin. Almost perfect nonlinear power functions on $GF(2^n)$: the Niho case. *Information and Computation*, to appear.
[12] B. Dwork. On the rationality of the zeta function. *Amer. J. Math.*, 82:631{648, 1959.
[13] P. Fan and M. Darnell. *Sequence design for communications applications*. John Wiley and Sons, New York, 1996.
[14] H. Fukumasa, R. Kohno, and H. Imai. Pseudo{noise sequences for tracking and data relay satellite and related systems. *Trans. of IEICE*, E(5): 1137{1144, May 1991.
[15] R. Gold. Optimal binary sequences for spread spectrum multiplexing. *IEEE Trans. Inform. Theory*, IT-13: 619{621, 1967.
[16] R. Gold. Maximal recursive sequences with 3-valued cross-correlation functions. *IEEE Trans. Inform. Theory*, IT-14: 154{156, 1968.
[17] G. Gong. Theory and applications of q-ary interleaved sequences. *IEEE Trans. Inform. Theory*, IT-41(2): 400{411, March 1995.
[18] A.R. Hammons, Jr., P.V. Kumar, A.R. Calderbank, N.J.A. Sloane, and P. Sole. The $\mathbb{Z}_4$-linearity of Kerdock, Preparata, Goethals and related codes. *IEEE Trans. Inform. Theory*, IT-40: 301{319, 1994.
[19] T. Helleseth. On the covering radius of cyclic linear codes and arithmetic codes. *Discrete. Appl. Math.*, 11: 157{173, 1985.
[20] H. Hollmann and Q. Xiang. A proof of the Welch and Niho conjectures on crosscorrelations of binary m-sequences. *preprint*, 1999.
[21] N.E. Hurt. Exponential sums and coding theory: A review. *Acta Applicandae Mathematicae*, 46: 49{91, 1997.
[22] K. Ireland and M. Rosen. *A Classical Introduction to Modern Number Theory (2nd edition), Graduate Texts in Mathematics Vol. 84*. Springer, Berlin, 1990.
[23] D. Jungnickel. *Finite Fields | Structure and Arithmetics*. B.I. Wissenschaftsverlag, Mannheim, 1993.
[24] A.M. Klapper. d-form sequences: Families of sequences with low correlation values and large linear spans. *IEEE Trans. Inform. Theory*, IT-41(2): 423{431, March 1995.
[25] P.V. Kumar, T. Helleseth, and A.R. Calderbank. An upper bound for Weil exponential sums over Galois rings and applications. *IEEE Trans. Inform. Theory*, IT-41(2): 456{468, March 1995.
[26] P.V. Kumar, T. Helleseth, A.R. Calderbank, and A.R. Hammons Jr. Large families of quaternary sequences with low correlation. *IEEE Trans. Inform. Theory*, IT-42(2): 579{592, March 1996.
[27] P.V. Kumar and O. Moreno. Prime-phase sequences with periodic correlation properties better than binary sequences. *IEEE Trans. Inform. Theory*, IT-37(3): 603{616, May 1991.
[28] G. Lachaud and J. Wolfmann. Sommes de kloosterman, coubes elliptiques et codes cycliques en caracteristique 2. *Comptes Rendu Academie Science Paris*, 305: 881{883, 1987.
[29] J. Lahtonen. On the odd and aperiodic correlation properties of the Kasami sequences. *IEEE Trans. Inform. Theory*, IT-41(5): 1506{1508, Sept. 1995.
[30] V.I. Levenshtein. Bounds on the maximal cardinality of a code with bounded modulus of the innner product. *Soviet Math. Dokl.*, 25(2): 526{531, 1982.

[31] R. Lidl and H. Niederreiter. *Introduction to finite fields and their applications (2nd Edition)*. Cambridge University Press, Cambridge, 1994.

[32] R. Lidl and H. Niederreiter. *Finite Fields*. Encyclopedia of Mathematics and Its Applications, Vol. 20 (2nd Edition), Cambridge University Press, Cambridge, 1997.

[33] S. Litsyn and A. Tietäväinen. Character sum constructions of constrained error-correcting codes. *AAECC*, 5: 45{51, 1994.

[34] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North Holland, Amsterdam, 1977.

[35] R.J. McEliece. *Finite fields for computer scientists and engineers*. Kluwer, Boston, 1987.

[36] C.J. Moreno. *Algebraic Curves over Finite Fields*. Cambridge University Press, Cambridge, 1991.

[37] O. Moreno and P.V. Kumar. Minimum distance bounds for cyclic codes and Deligne's theorem. *IEEE Trans. Inform. Theory*, IT-39(5): 1524{1534, Sept. 1993.

[38] O. Moreno and C.J. Moreno. The MacWilliams-Sloane conjecture on the tightness of the Carlitz-Uchiyama bound and the weights of duals of BCH codes. *IEEE Trans. Inform. Theory*, IT-40(6): 1894{1907, Nov. 1994.

[39] O. Moreno, V.A. Zinoviev, and P.V. Kumar. An extension of the Weil-Carlitz-Uchiyama bound. *Finite Fields and their Applications*, 1: 360{371, 1995.

[40] J.-S. No. Generalization of GMW sequences and No sequences. *IEEE Trans. Inform. Theory*, IT-42(1): 260{262, Jan. 1996.

[41] J.-S. No and P.V. Kumar. A new family of binary pseudorandom sequences having optimal periodic correlation properties and large linear span. *IEEE Trans. Inform. Theory*, IT-35(2):371{379, March 1989.

[42] J.D. Olsen, R.A. Scholtz, and L.R. Welch. Bent{function sequences. *IEEE Trans. Inform. Theory*, IT-28(6): 858{864, Nov. 1982.

[43] K.G. Paterson. Binary sequence sets with favourable correlation properties from di erence sets and MDS codes. *IEEE Transactions on Information Theory*, 44:172{180, 1998.

[44] K.G. Paterson. Generalised Reed-Muller codes and power control in OFDM. *IEEE Transactions on Information Theory*, to appear.

[45] K.G. Paterson and P.J.G. Lothian. Bounds on partial correlations of sequences. *IEEE Transactions on Information Theory*, 44:1164{1175, 1998.

[46] K.G. Paterson and V. Tarokh. On the existence and construction of good codes with low peak-to-average power ratios. *Hewlett-Packard Laboratories Technical Report HPL-1999-51, submitted*, 1999. `http://www.hpl.hp.com/techreports/1999/HPL-1999-51.html`.

[47] V.S. Pless and W. Hu man, eds. *Handbook of Coding Theory Vols. I & II*. Elsevier, 1998.

[48] M.B. Pursley. On the mean-square partial correlation of periodic sequences. In *Proc. of Conf. on Information Sciences and Systems*, pages 377{379, John Hopkins Univ., Baltimore MD, March 28{30 1979.

[49] M.B. Pursley, D.V. Sarwate, and T.U. Basar. Partial correlation e ects in direct{ sequence spread{spectrum multiple{access communications systems. *IEEE Trans. Commun.*, COM-32(5): 567{573, May 1984.

[50] L.C. Quynh and S. Prasad. New class of sequences sets with good auto- and crosscorrelation functions. *IEE Proc. (F)*, 133(3): 281{287, June 1986.

[51] D.V. Sarwate. Bounds on crosscorrelation and autocorrelation of sequences. *IEEE Trans. Inform. Theory*, IT-25(6): 720{724, Nov. 1979.

[52] D.V. Sarwate. An upper bound on the aperiodic autocorrelation function for a maximal-length sequence. *IEEE Trans. Inform. Theory*, IT-30(4): 685{687, July 1984.

[53] D.V. Sarwate and M.B. Pursley. Cross-correlation properties of pseudorandom and related sequences. *Proc. IEEE*, 68: 593{618, May 1980.

[54] W. Schmidt. *Equations Over Finite Fields | An Elementary Approach. Lecture Notes in Mathematics, Vol. 536*. Springer, Berlin, 1976.

[55] R.A. Scholtz. Criteria for sequence set design in CDMA communications. In G. Cohen, T. Mora, and O. Moreno, editors, *Applied Algebra, Algebraic Algorithms and Error{Correcting Codes (AAECC{10)* , pages 57{65, Puerto Rico, May 10{14 1993. Springer-Verlag, Berlin.

[56] R.A. Scholtz and L.R. Welch. GMW sequences. *IEEE Trans. Inform. Theory*, IT-30(3): 548{553, Nov. 1984.

[57] M.R. Schroeder. *Number Theory in Science and Communication (3rd edition)*. Springer, Berlin, 1997.

[58] A.G. Shanbag, P.V. Kumar, and T. Helleseth. Upper bound for a hybrid sum over Galois rings with applications to aperiodic correlation for some q-ary sequences. *IEEE Trans. Inform. Theory*, IT-42: 250{254, 1996.

[59] V.M. Sidelnikov. On mutual correlation of sequences. *Soviet Math. Dokl.*, 12(1): 197{201, 1971.

[60] M.K. Simon, J.K. Omura, R.A. Scholtz, and B.K. Levitt. *Spread Spectrum Communications, Vol. 1*. Computer Science Press, Rockville, MD, 1985.

[61] H. Stichtenoth. *Algebraic Function Fields and Codes*. Springer-Verlag, New York, 1993.

[62] A. Tietäväinen. An asymptotic bound on the covering radii of binary bch codes. *IEEE Trans. Inform. Theory.*, IT-36: 211{213, 1990.

[63] A. Tietäväinen. Vinogradov's method and some applications. Technical Report TUCS No. 28, Turku Centre for Computer Science, Turku, Finland, 1996.

[64] P. Udaya and M.U. Siddiqi. Optimal biphase sequences with large linear complexiy derived from sequences over $\mathbb{Z}_4$. *IEEE Trans. Inform. Theory*, IT-42(1): 206{216, Jan. 1996.

[65] I.M. Vinogradov. *Elements of Number Theory*. Dover, New York, 1954.

[66] A. Weil. *Sur les courbes algebriques et les varietes qui s'en deduisent, Actualites Sci. et Ind. no. 1041*. Hermann, Paris, 1948.

[67] L.R. Welch. Lower bounds on the maximum correlation of signals. *IEEE Trans. Inform. Theory*, IT-20(3): 397{399, May 1974.

Some Applications of Bounds for Designs to the Cryptography

Svetla Nikova* and Ventsislav Nikov

Department of Mathematics and Informatics
Veliko Tarnovo University
5000 Veliko Tarnovo, Bulgaria
svetla_venci@hotmail.com

Abstract. Recent years have seen numerous examples where designs play an important role in the study of such topics in cryptography as secrecy and authentication codes, secret sharing schemes, correlation-immune and resilient functions. In this paper we give applications of some methods and results from the design theory, especially bounding the optimal size of the designs and codes, to cryptography. We give a new bound for the parameter t, when $(n;T;t)$-resilient functions and correlation-immune functions of order t exist. In the last section we present analogous bound for the parameter N of T-wise independent t-resilient function.

1 Introduction

Let M be a metric space with distance $d(x;y)$ and a normalized measure , $(M) = 1$. Any nite subset (*code* or *design*) C, $C \quad M$ is characterized by its *minimal distance* $d(C) = \min_{x;y2C;x6=y} d(x;y)$. For any $C \quad M$, let (C) denote the set of values of $d(x;y)$ when $x;y \ 2 \ C$ i.e. the distance distribution of C. For any code C the parameter $s(C) = j \ (C) \ n \ f0gj$ characterizes the number of distinct distances between distinct points of C. The diameter of the whole space M can be de ned as $D(M) = \max(\ (M) \ n \ f0g)$. We will de ne a -design by means of the strictly decreasing real function (substitution) (d) considered on the interval $[0; D(M)]$.

De nition 1.1 *A set C will be referred to as a -design in M with respect to the substitution (d) if for any polynomial $f(t)$ in a real t of degree at most ,*

$$\int_M \int_M f(\ (d(x;y)))d \ (x)d \ (y) = \frac{1}{jCj^2} \sum_{x;y2C} f(\ (d(x;y))): \qquad (1)$$

* Supported by a junior research fellowship of the Katholieke Universiteit Leuven, Belgium.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 25{34, 1999.

The maximum integer ($s(M)$) such that a set C is a -design is called the strength of C and denoted by (C). Suppose M is nite and $(C) = fd_0; d_1; :::; d_ng$, is the *distance distribution* of C. The *dual distance distribution* (so called MacWilliams transform [15, p.137]) of C is de ned to be $^0(C) = fd_0^0; d_1^0; :::; d_n^0g$. The *dual distance* of the code C is the smallest i, $i = 1; :::; n$, such that $d_i^0 \neq 0$; the *external distance* $s^0(C)$ of C is the number of i, $i = 1; :::; n$; such that $d_i^0 \neq 0$. It is proved in [3] that $(C) + 1 = d^0(C)$.

In this paper we are interested in applications of some methods and results of design theory, especially bounding the optimal size of the designs and codes, to cryptography. A function $f(x_1; :::; x_n)$, where $x_i \in Z_v = f0; 1; :::; v - 1g$ $v \quad 2$ and $f(x_1; :::; x_n) \in Z_w (w \quad 2)$ can be considered as a random variable provided that the input variables x_i, $i = 1; :::; n$, are independent and uniformly distributed random variables. Then f is characterized by probabilities $p(b)$ to take a value $b \in Z_w$. The function f is called correlation-immune of order t ($t = 0; 1; :::; n$) if any function in $n - t$ variables, obtained from f by a substitution of any constants from Z_v for any t input variables, has the same probability $p(b)$, $b \in Z_w$. If $w = v^T$ ($T = 1; 2; :::$) and all probabilities $p(b)$, $b \in Z_w$, are equal, then any f can be considered as a system of T functions $f_1(x_1; :::; x_n); :::; f_T(x_1; :::; x_n)$ of the form $Z_v^n \to Z_v$, which are independent and uniformly distributed random variables (we call such systems of T functions in n variables *balanced*). If f is correlation-immune function of order t, then the system of T function preserves the property to be balanced system under a substitution of any constants of Z_v for any t input variables. Such balanced system of T functions is called $(n; T; t)$-*resilient*, (or simply resilient of order t).

Another interesting application is the designs in product association schemes. Let $(Y; A)$ [1] be an association scheme with primitive idempotents $E_0; E_1; :::; E_d$. For $T \quad f1; :::; dg$ a Delsarte T-design in $(Y; A)$ is a subset D of Y whose characteristic vector is annihilated by the idempotents E_j ($j \in T$). The most studied case is that in which $(Y; A)$ is Q-polynomial and $T = f1; :::; tg$. For $1 \quad i \quad m$, let $(Y_i; A_i)$ be Q-polynomial association schemes. Assume that Delsarte t-designs in each $(Y_i; A_i)$ are characterized as poset t-designs in a Q-poset P_i attached to that scheme. With these assumption, we consider the product association scheme ($Y_i; A$) and the corresponding linear programming bound and the Delsarte bound on size of degree of such a T-design analogous to Delsarte's bound for t-designs in Q-polynomial association schemes.

The paper is organized as follows. In the rst part we present some preliminary results about -designs in Hamming and Johnson space. We give briefly our previous investigations, which give necessary and su cient conditions for improving the Delsarte bound for designs. We present also the analytical form of the extremal polynomials and the analytical form of the new bound for non-antipodal spaces. The results in this section are valid for all PMS, in particular for Hamming and Johnson space. The proofs of the theorems can be found in [10,11,12]. In Section 3 we are interested in correlation-immune and resilient

[1] Here and in Section 4 we will follow the notations used in [9]

functions and the connection with the designs theory. As a direct application we give a new bound for the parameter t, when $(n;T;t)$-resilient function and correlation-immune function of order t exist. In the last section we present analogous bound for the parameter N of T-wise independent t-resilient function.

2 Improvement of the Delsarte Bound for Orthogonal Arrays and Combinatorial Designs

The basic problem of the coding theory is the construction of the maximum (on cardinality) -code. Together with this problem there exists another one of constructing the minimum (on cardinality) -design (or equivalently a code with dual distance $d^{0} =$ + 1). As it is proved in [5,2] this two problems are dual.

Following the notations in [7] we will consider **polynomial metric space** (PMS) with a given substitution (d) and a system of orthogonal (with respect to the measure (z)) polynomials $Q_i(z)$ of degree i, $i = 0;1;:::;s(M)$, the so called *zonal spherical functions* (ZSF). A polynomial metric space M is called **antipodal** if for every point $x \in M$ there exists a point $\overline{x} \in M$ such that for any point $y \in M$ we have $(d(x;y)) + (d(\overline{x};y)) = 0$: A code for which $(d(x;y))$ (d), where d is the minimum distance of C we call an $(M;jCj;\ )$-code. When M is nite the measure (z) is left continuous function and has $s+1$ steps at the point $z_i =$ (d_i) with positive step sizes w_i, $i = 0;1;:::;s(M)$, $\sum_{i=0}^{s} w_i = 1$.

For arbitrary $a;b \in \{0;1\}$ can be de ned the so called **adjacent** system of orthogonal polynomials $Q_k^{a;b}(z)$ (with respect to the measure $(1-z)^a(1+z)^b$ (z)) in a real z of degree $k;k = 0;1;:::;s(M) -$ $_{a;1} -$ $_{b;1}$, positive constant $c^{a;b}$ and $r_k^{a;b}$ [7]. The ZSF satisfy the recurrence formula $(z + m_i + c_i - 1)Q_i(z) = m_iQ_{i+1}(z) + c_iQ_{i-1}(z)$; for i 0, where $r_{-1} = m_{-1} = 0$, $m_i = \frac{a_{i;i}}{a_{i+1;i+1}}$, $c_i = \frac{r_{i-1}m_{i-1}}{r_i}$ and $Q_{-1}(z)$ 0, $Q_0(z)$ 1. Denote by $z_k^{a;b}$ the greatest zero of the polynomial $Q_k^{a;b}(z)$. We will introduce the notations $Q_k^{a;b}(z) = \sum_{i=0}^{k} a_{k;i}^{a;b} z^i$, $n_i^{a;b} = \frac{a_{i;i-1}^{a;b}}{a_{i;i}^{a;b}}$; $n_i^{a;b} = \frac{a_{i;i-2}^{a;b}}{a_{i;i}^{a;b}}$; $u_i^{a;b} = \frac{a_{i;i}^{a;b}}{a_{i;i}}$:

The linear programming bounds for codes and designs was obtained by using the following theorem [15].

Theorem 2.1. *Let C M be an $(M;jCj;\ )$-code (reps. -design) and let $f(z)$ be a real non-zero polynomial such that*

(A1) $f(z)$ 0; *for* -1 z , *(resp.* (B1) $f(z)$ $\sum_{i=0}^{k} f_iQ_i(z)$ *satisfy* $f_0 > 0$,
(A2) *the coe cients in the ZSF expansion* $f(z) = \sum_{i=0}^{k} f_iQ_i(z)$ *satisfy* $f_0 > 0$, f_i 0 *for* $i = 1;:::;k$. *(resp.* (B2) *the coe cients in the ZSF expansion* $f(z) = \sum_{i=0}^{k} f_iQ_i(z)$ *satisfy* $f_0 > 0$, f_i 0 *for* $i =$ + $1;:::;k$.)

Then, jCj $f(1)=f_0 = \Omega(f)$ *(resp. jCj* $f(1)=f_0$).

We denote by $B_{M;}$ the set of real polynomials which satisfy the conditions (B1) and (B2) and $B(M;\ ) = \max\{\Omega(f) : f(t) \in B_{M;}\}$: A polynomial

$f(z) \in B_{M;}$ is called $B_{M;}$-extremal if $\Omega(f) = \max\{\Omega(g) : g(z) \in B_{M;}; deg(g) \quad \deg(f)\}$.

Many authors obtained various pairs of bounds [15] for the cardinality of codes and designs in nite PMS which follow from Theorem 2.1. For our investigations the most important ones will be the Levenshtein bound $L_{2k-1+"}(M; \ )$ for codes and the Delsarte bound $D(M; \ )$ for -designs which can be presented as follows [7,3]: $jCj \quad L_{2k-1+"}(M; \ ) = (1 - \frac{Q^{1;0}_{k-1+"}(\ )}{Q^{0;"}_{k}(\ )}) \sum_{i=0}^{k-1+"} r_i$; where $" = 0$ if $z^{1;1}_{k-1} < z^{1;0}_{k}$ and $" = 1$ if $z^{1;0}_{k} < z^{1;1}_{k}$, resp. $jCj \quad D(M; \ ) = 2 \ c^{0;} \sum_{i=0}^{k} r^{0;}_{i}$; where $\in \{0; 1\}$ and $= 2k + \ $. This two bounds can be obtained by the polynomials $f^{(\ )}(z) = (z - \ )(z + 1)^{"}(\sum_{i=0}^{k-1} r^{1;"}_{i} Q^{1;"}_{i}(z) Q^{1;"}_{i}(\ ))^2$; and $f^{(\ )}(z) = (z + 1) \ ((Q^{1;}_{k} \ (z))^2$, respectively, in the Theorem 2.1.

PMS are nite metric spaces represented by P- and Q- polynomial association schemes as well as in nite metric spaces. The most famous examples of the nite spaces are the Hamming, Johnson, Grassmann space. We will consider only Hamming and Johnson spaces, presented by Q-polynomial association schemes. In these spaces -designs are known as orthogonal arrays and combinatorial designs, respectively. Analogously the Delsarte bound is in fact the Rao bound for orthogonal arrays and the Ray-Chaudhuri/Wilson bound for the combinatorial t-designs.

We consider the following linear functional $G \ (M; f) = \frac{f(1)}{D(M; \ )} + \sum_{i=1}^{k+ \ } \ ^{(\ )}_{i} f(\ _i)$, where $ \ _i$ are the zeros of $Q^{1;}_{k} \ (t)$, and $^{(\ )}_{i}$ are positive constants. This linear functional maps the set of real polynomials to the set of real numbers. Now we will give necessary and su cient conditions for improvement of the Delsarte bound.

Theorem 2.2. *[10,11] The bound $D(M; \ )$ can be improved by a polynomial $f(z) \in B_{M;}$ of degree at least $+ 1$, if and only if $G \ (M; Q_j) < 0$ for some $j \quad + 1$. Moreover, if $G \ (M; Q_j) < 0$ for some $j \quad + 1$, then $D(M; \ )$ can be improved by a polynomial in $B_{M;}$ of degree j.*

Theorem 2.3. *[11] Let M be non-antipodal PMS. Then, any $B_{M;}$-extremal polynomial of degree $+ 2 \ (\ = 2k + \ )$ has the form*

$$f^{(\ )}(z; \quad + 2) = (1 + z)^{1-} \ [q(z + 1) + (1 - z)][\ Q^{1;1-}_{k-1+} \ (z) + Q^{1;1-}_{k+} \ (z)]^2$$

where q and $\ $ are suitable constants.

Let us introduce the following notations: for $= 2k$

$$B_1 = \frac{u^{1;1}_{k}(n_{ \ +2} - 2n^{1;1}_{k})}{2u^{1;1}_{k-1} m_{k-1} c^{1;1} r^{1;1}_{k-1}}; \quad B_2 = \frac{1}{(c^{1;1})^2 r^{1;1}_{k-1} r^{1;1}_{k}}; \quad S_1 = \frac{4}{r_{k+1}} + \frac{4}{r_k}$$

$$S_2 = \frac{4(2n_{k+1} - n_{ \ +2})}{u^{1;1}_{k} u^{1;1}_{k-1} m_{k-1} m^2_{k}} - \frac{8}{u^{1;1}_{k} u^{1;1}_{k-1} m_{k-1} m_k}; \quad S_3 = \frac{4}{r_{k+1} r_k}$$

$$S_4 = \frac{4(2n_{k+1} - n_{\ +2})}{u_k^{1;1}u_{k-1}^{1;1}(m_k)^2 m_{k-1}r_k}; \quad S_5 = \frac{-4}{(u_k^{1;1}u_{k-1}^{1;1}m_k m_{k-1})^2}$$

and for $= 2k + 1$

$$B_1 = \frac{u_{k+1}^{1;0}(1 + n_{\ +2} - 2n_{k+1}^{1;0})}{4u_k^{1;0}m_k c^{1;0}r_k^{1;0}}; \quad B_2 = \frac{1}{(4c^{1;0})^2 r_k^{1;0}r_{k+1}^{1;0}}; \quad S_5 = \frac{-4(u_{k+1}^{0;1}u_k^{0;1})^2}{(u_{k+1}^{1;0}u_k^{1;0})^2}$$

$$S_1 = \frac{2}{c^{0;1}r_{k+1}^{0;1}} + \frac{2}{c^{0;1}r_k^{0;1}} \quad S_2 = \frac{4(u_{k+1}^{0;1})^2(1 + 2n_{k+1}^{0;1} - n_{\ +2})}{u_{k+1}^{1;0}u_k^{1;0}m_k} - \frac{8u_{k+1}^{0;1}u_k^{0;1}}{u_{k+1}^{1;0}u_k^{1;0}};$$

$$S_3 = \frac{1}{(c^{0;1})^2 r_{k+1}^{0;1}r_k^{0;1}} \quad S_4 = \frac{2(u_{k+1}^{0;1})^2(1 + 2n_{k+1}^{0;1} - n_{\ +2})}{c^{0;1}u_{k+1}^{1;0}u_k^{1;0}m_k r_k^{0;1}}:$$

Now taking into account that $S(M; \ ) = \Omega(f^{(\ )}(z; \ + 2))$ and using the notations above we obtain the following analytical form of the bound $S(M; \ )$.

Lemma 2.1. *[11] Let M be a non-antipodal PMS. Then the bound $S(M; \ )$ is equal to*

$$1 + \frac{S_1 + (B_1 + \sqrt{B_1^2 + B_2})S_2}{S_3 + (B_1 + \sqrt{B_1^2 + B_2})S_4 + (B_1 + \sqrt{B_1^2 + B_2})^2 S_5} \ D(M; \ - 2)$$

Corollary 2.4 *[11] Let M be a non-antipodal PMS and let be an integer. Then*

a) $B(M; \ ) \quad S(M; \ ) = \Omega(f^{(\ )}(z; \ + 2))$*:*
b) $S(M; \ ) > D(M; \ )$ *if and only if* $G \ (M; Q_{\ +2}) < 0$.

Theorem 2.5. *[11] Let M be antipodal PMS. Then, any $B_{M;}$ -extremal polynomial of degree $+ 3$ ($= 2k +$) has the form*

$$f^{(\ )}(z; \ + 3) = (1 + z)\,[q(z + 1) + (1 - z)][\ {}_1Q_{k-1}^{1;}(z) + \ {}_2Q_k^{1;}(z) + Q_{k+1}^{1;}(z)]^2$$

where q, ${}_1$ and ${}_2$ are suitable constants.

Corollary 2.6 *[11] Let M be an antipodal PMS and let be an integer. Then*

$$B(M; \ ) \quad S(M; \ ) = \Omega(f^{(\ )}(z; \ + 3))\text{:}$$

Let us consider the Hamming space $M = H_v^n \ (n; v = 2; 3; : : :)$. The orthogonal arrays are commonly denoted by $OA \ (\ ; n; v)$ and their cardinality satisfy $jCj = \ v$. The ZSF for the Hamming space are the Krawtchouk polynomials $K_k^{n;v}(z)$.

A stronger version of the Theorem 2.1 is the following:

Theorem 2.7. *[5] Let $C \subset M$ be an d-code (reps. -design) and let $f(z) = \sum_{i=0}^{n} f_i K_i^{n;v}(z)$ be a real non-zero polynomial such that*

(C1) $f(0) > 0$, $f(i) \geq 0$; *for* $i = d; \ldots; n$,
(resp. (D1) $f(0) > 0$, $f(i) \geq 0$; *for* $i = 1; 2; \ldots; n$*),*
(C2) $f_0 > 0$, $f_i \geq 0$ *for* $i = 1; \ldots; n$.
(resp. (D2) $f_0 > 0$, $f_i \geq 0$ *for* $i = d + 1; \ldots; n$.*)*

Then, $|C| \leq \min\Omega(f)$ *(resp.* $|C| \geq \max\Omega(f)$*), where* $\Omega(f) = f(0)/f_0$.

Let us denote by $A_v^?(n;d) = \min\{\Omega(f)$ for polynomials f satisfying the conditions (C1), (C2)$\}$; $B_v^?(n;d) = \max\{\Omega(f)$ for polynomials f satisfying the conditions (D1), (D2) $\}$; and $B_v^{??}(n;d) = \max\{\Omega(f)$: for polynomials f satisfying the conditions (D1); (D2) and $\deg f \geq d\}$.

Theorem 2.8. *[2,5] For any integers n, d, v $(1 \leq d \leq n + 1; v \geq 2)$,*

$$A_v^?(n;d)B_v^?(n;d-1) = v^n \tag{2}$$

Here we will present well known pairs of universal bounds, i.e. inequalities which are valid for all codes $C \subset H_v^n$. The first pair is the **Singleton bound** [15] for a code $C \subset H_v^n$

$$v \leq |C| \leq v^{n-d+1};$$

where any of the bounds is attained if and only if $d + d^0 = n + 2$, $(d^0 - 1 = \quad)$.

The second pair of bounds is formed by **Rao** and **Hamming** [15] bounds for a code $C \subset H_v^n$.

$$D(H_v^n; \quad) \leq |C| \leq \frac{v^n}{D(H_v^n; d-1)} \tag{3}$$

Codes, which cardinality is equal to the left-hand side or the right-hand side of (3) are called *tight* designs and *perfect* codes, respectively.

The third pair universal bounds for any code $C \subset H_v^n$ is the **Levenshtein** bound [5].

$$\frac{v^n}{L(H_v^n; \quad (\quad + 1))} \leq |C| \leq L(H_v^n; \quad (d))$$

First two pairs of bounds are obtained by means of combinatorial methods, but all of them can be obtained using Theorem 2.1 or Theorem 2.7.

Applying Theorem 2.8 for our bound we have

Theorem 2.9. *[12] For any code $C \subset H_v^n$*

$$S(H_v^n; \quad) \leq |C| \leq \frac{v^n}{S(H_v^n; d-1)} \tag{4}$$

In the Johnson space $X = J_w^n$ ($n = 2;3;\dots;w = 1;\dots;bn=2c$) designs are the classical $t-(v;k;\ )$ and codes are called constant weight codes. The ZSF are the Hahn polynomials $J_k^{n;w}(z)$. For J_w^n an analog of the Theorem 2.7 is also valid and there are known several pairs of bounds.

Theorem 2.10. *[12] For any design* $C\ \ J_n^w$

$$S(J_w^n;\ )\ \ jCj \tag{5}$$

3 Resilient and Correlation-Immune Functions

In [14] Stinson gave the connection between correlation-immune function, resilient function and orthogonal arrays.

Theorem 3.1. *[1] A function* $f : Z_v^n -! Z_w$ *is correlation-immune of order* t *if and only if* Z_v^n *is partitioned into* w *orthogonal arrays* $OA\ (t;n;v)$.

Theorem 3.2. *[1] A function* $f : Z_v^n -! Z_v^T$ *is resilient of order* t *if and only if* Z_v^n *is partitioned into* v^T *orthogonal arrays* $OA_{v^{n-T-t}}(t;n;v)$.

Note that in the rst theorem need not be identical. A *large set of orthogonal arrays* $LOA\ (t;n;v)$ is a set of $v^{n-t}=$ simple arrays $OA\ (t;n;v)$ such that all have the same value.

Corollary 3.3 *[1] There exists a function* $f : Z_v^n -! Z_v^T$ *that is resilient of order* t *if and only if there exists an* $LOA_{v^{n-T-t}}(t;n;v)$.

A necessary condition for the existence of a correlation-immune function of order t and for existence of a $(n;T;t)$-resilient function are as follows:

$$w\ \ \frac{v^n}{B_v^?(n;t)} = A_v^?(n;t+1);\quad \log_v(B_v^?(n;t))\ \ n-T: \tag{6}$$

One is concerned with developing upper bounds for the optimum value of t for a given n and T. It is easy to see that $n\ \ T+t$ and so the trivial upper bound is $t\ \ n-T$. If we substitute the Delsarte bound instead of $B_v^?(n;t)$ we obtain another upper bound for t [1]. The upper bounds based on the Delsarte (Rao) bound for orthogonal arrays are stronger than the ones obtained using the trivial bound. We can improve this bound using our previous result in Theorem 2.9.

Theorem 3.4. *Suppose there exists an correlation-immune function of order* t. *Then* $w\ \ \frac{v^n}{S(H_v^n;t)}$:

Theorem 3.5. *Suppose there exists a* $(n;T;t)$*-resilient function. Then*

$$\log_v(S(H_v^n;t))\ \ n-T$$

However, we can often do better by using Delsarte's linear programming bound. Let $W(n;t)$ be the optimal solution to the linear programming problem Theorem 2.7. In view of the equation (6), this implies that $\log_v(W(n;t)+1) \quad n-T$. For large values of t, the orthogonal array bounds obtained by the linear programming technique are usually much better than the Delsarte (Rao) bound and our new bound $S(H_v^n;t)$. The disadvantage of this method is that one needs to solve a di erent linear program for every parameter situation. Thus it is of interest to derive explicit bounds as corollaries of the linear programming bound. In the cases $v=2$, $t+1<n<2t+2$ and $v=2$, $t+1<n<2t+3$ the most important bounds are as follows :

Theorem 3.6. *[1,4] Suppose there exists a $(n;T;t)$-resilient function and $v=2$. Then* $t \quad b\frac{2^{T-1}n}{2^T-1}c-1;$ $t \quad 2b\frac{2^{T-2}(n+1)}{2^T-1}c-1:$

4 Designs in Product Association Schemes. Maximum Independent Resilient System of Functions

Let $(P; \ )$ be a partially ordered set (*poset*). If there exist constants $_0;:::; \ _t$ such that, for $0 \quad i \quad t$ and $8x \ 2 \ P^i$, $jfy \ 2 \ D : x \quad ygj = \ _i$ then the set $D \quad P^d$ is called a *poset t-design* in $(P; \ )$. For $1 \quad i \quad m$, let $(Y_i;A_i)$ be a d_i-class association scheme with adjacency matrices A_i. The *direct product* of these schemes is the association scheme $(X;A) = (Y_1;A_1) \quad ::: \quad (Y_m;A_m)$ defined by $X = Y_1 \quad ::: \quad Y_m$ and $A = f \bigotimes_{i=1}^{m} M_i : M_i \ 2 \ A_i; 1 \quad i \quad mg$ where $\bigotimes_{i=1}^{m} M_i$ is the m-fold Kronecker product of matrices. Assume that each component scheme $(Y_i;A_i)$ has an attached Q-poset $(P_i; \ _i)$. Consider the Delsarte T -design $(T = T - f0g)$ in $(\ Y_i;A)$ as poset designs in the product poset P_i where T is any downset in the product chains C. Let $(X;A)$ be the product of Q-polynomial association schemes $(Y_i;A_i);(1 \quad i \quad m)$. Each matrix $M \ 2 \ A$ may be expanded in the form $M = v \sum_{\underline{j} 2 C} \ _{\underline{j}} E_{\underline{j}}$. If we change the bases, we have $M = \sum_{\underline{i} 2 C} \ _{\underline{i}} A_{\underline{i}}$, where $_{\underline{i}} = \sum_{\underline{j} 2 C} Q_{\underline{i}\underline{j}} \ _{\underline{j}}$, $(\underline{i} \ 2 \ C)$. For $\underline{j} \ 2 \ C$, let $f_{\underline{j}} = rank E_{\underline{j}}$.

Theorem 4.1. *[9] Let $(X;A)$ be the product of Q-polynomial association schemes $(Y_i;A_i);(1 \quad i \quad m)$. Let T be a downset in C and let $D \quad X$ be a Delsarte T -design. Consider the matrices M satisfying the conditions*

(i) *M is non-negative matrix;*
(ii) $_{\underline{j}} \quad 0$ *for* $\underline{j} \ 2 \ T$ *;*
(iii) $_{\underline{0}} = 1$.

Then, the lower bound on the size of a T -design is $jDj \quad _{\underline{0}}$.

Theorem 4.2. *(*Delsarte bound*)[9] Let $(X;A)$ be the product of Q-polynomial association schemes $(Y_i;A_i);(1 \quad i \quad m)$. Let T be a downset in C and let $D \quad X$ be a Delsarte T -design. If $E \quad C$ satis es $(E+E) \setminus C \quad T$, then $jDj \quad \sum_{\underline{j} 2 E} f_{\underline{j}}$.*

Here are some examples from [9].

Mixed-level orthogonal arrays $OA(M; q_1^{n_1} \ldots q_m^{n_m}; t)$ of *strength* t are studied by Sloane and Stufken in [13]. This object is equivalent to the Delsarte T-design in the scheme $H(n_1; q_1) \times \ldots \times H(n_m; q_m)$, where $T = \{j : j_i \ldots t\}$.

Mixed t-designs [8] is the product of Johnson schemes $J(v_1; k_1) \times \ldots \times J(v_2; k_2)$, which is the Delsarte T-design for $T = \{(i_1; i_2) : i_1 + i_2 \ldots t\}$.

Fused orthogonal array design of strength t can be considered as a product scheme of the form $H(n; q) \times J(v; k)$.

Split orthogonal arrays $SOA(t; n; T; N; v)$ are introduced by Levenshtein [6]. The cardinality of $SOA(t; n; T; N; v)$ is v^{t+T}. Given $q; n; N; t; T$ we wish to find an $M \times (n + N)$ array with entries in $Z_q = \{0; \ldots; q-1\}$ such that, upon choosing any t columns from among the first n columns and any T columns from among the last N columns, all $(t + T)$-tuples over the alphabet Z_q occur equally often. This is equivalent to a T-design in the product scheme $H(n_1; q) \times H(n_2; q)$ where $T = \{(i_1; i_2) : 0 \ldots i_1 \ldots t_1; 0 \ldots i_2 \ldots t_2\}$. For such objects, the Delsarte linear programming bound is equivalent to the following: let $f(z) = 1 + \sum_{i=1}^{n} f_i K_i^{n;v}(z)$ and $g(z) = 1 + \sum_{j=1}^{N} g_j K_j^{N;v}(z)$ be polynomials satisfying the condition (D1) and $f_i g_j \ldots 0$ for $i \ldots t + 1$ or $j \ldots T + 1$ then $|D| \ldots f(0)g(0)$.

Theorem 4.3. *[6] If D is split orthogonal array then*

$$|D| \ldots \max(B_v^{?}(n; t)B_v^{??}(N; T); B_v^{??}(n; t)B_v^{?}(N; T))$$

Hence we have the following bound $|D| \ldots D(H_v^n; t)D(H_v^N; T)$. Using again Theorem 2.9 we obtain the next statement.

Theorem 4.4. *If D is split orthogonal array then*

$$|D| \ldots \max(S(H_v^n; t)D(H_v^N; T); D(H_v^n; t)S(H_v^N; T))$$

A system of N functions in n variables over Z_v is called *T-wise independent t-resilient* if any subset of T functions of the system forms a t-resilient system. Our goal is to find the maximum number N, such that there exists a T-wise independent t-resilient system. The connection between this cryptographic objects and the orthogonal arrays was studied by Levenshtein in [6].

Theorem 4.5. *[6]The existence of T-wise independent t-resilient system is equivalent to that of split orthogonal array $SOA(t; n; T; N; v)$ with ($= v^{n-t-T}$).*

Corollary 4.6 *We derive the inequality*

$$v^n \ldots \max(S(H_v^n; t)D(H_v^N; T); D(H_v^n; t)S(H_v^N; T)).$$

Summarizing the results our bounds (Theorems 3.4, 3.5, 4.4 and Corollary 4.6) give a necessary condition for the existence of the above considered cryptographic objects.

References

1. J.Bierbrauer, K.Gopalakrishnan, D.R.Stinson, Orthogonal arrays, resilient functions, error correcting codes and linear programming bounds, SIAM *J.Discrete Math.* 9, 1996, 424-452.
2. J.Bierbrauer, K.Gopalakrishnan, D.R.Stinson, A note on the duality of linear programming bounds for orthogonal arrays and codes, *Bulletin of the ICA* 22, 1998, 17-24.
3. P.Delsarte, An Algebraic Approach to Association Schemes in Coding Theory, *Philips Research Reports Suppl.*, 10, 1973.
4. J.Friedman, On the bit extraction problem. *Proc. 33rd IEEE Symp. on Foundations of Computer Science*, 1992, 314-319.
5. V.I.Levenshtein, Krawtchouk polynomials and universal bounds for codes and designs in Hamming spaces, *IEEE Trans. Inf. Theory* 41, 5, 1995, 1303-1321.
6. V.I.Levenshtein, Split orthogonal arrays and maximum independent resilient systems of functions, *Designs, Codes and Cryptography* 12, 1997, 131-160.
7. V.I.Levenshtein, Universal bounds for codes and designs, Chapter 6 in *Handbook of Coding Theory*, V.Pless and W.C.Hu man, 1998 Elsevier Science B.V., 449-648.
8. W.J. Martin, Mixed block designs, *J.Combin.Designs* 6, 2, 1998, 151-163.
9. W.J. Martin, Designs in product association schemes *Designs, Codes and Cryptography* 16, 3, 1999, 271-289.
10. S.I. Nikova, Bounds for designs in in nite polynomial metric spaces, Ph.D. Thesis, Eindhoven University of Technology, 1998.
11. S.I.Nikova, V.S.Nikov, Improvement of the Delsarte bound for -designs when it is not the best bound possible, submitted in *Designs Codes and Cryptography*.
12. S.I.Nikova, V.S.Nikov, Improvement of the Delsarte bound for -designs in nite polynomial metric space, to be published.
13. N.J.A. Sloane, J.Stufken, A linear programming bound for orthogonal arrays with mixed levels, *J. Stat. Plan. Inf* 56, 1996, 295-306.
14. D.R.Stinson, Resilient functions and large sets of orthogonal arrays,*Congressus Numer.* 92, 1993, 105-110.
15. F.J.MacWilliams, N.J.A.Sloane, *The Theory of Error-Correcting Codes*, North Holland, Amsterdam, 1977.

Further Results on the Relation Between Nonlinearity and Resiliency for Boolean Functions

Enes Pasalic and Thomas Johansson

Dept. of Information Technology
Lund University, P.O. Box 118, 221 00 Lund, Sweden
{thomas, enes}@it.lth.se

Abstract. A good design of a Boolean function used in a stream cipher requires that the function satisfies certain criteria in order to resist different attacks. In this paper we study the tradeoff between two such criteria, the nonlinearity and the resiliency. The results are twofold. Firstly, we establish the maximum nonlinearity for a fixed resiliency in certain cases. Secondly, we present a simple search algorithm for finding Boolean functions with good nonlinearity and some fixed resiliency.

1 Introduction

A Boolean function used in a stream cipher requires that the function satisfies certain criteria in order to resist different attacks. Here we study the tradeoff between two such criteria, the nonlinearity and the resiliency of the Boolean function. The *resiliency* is defined as the number of arbitrary input variables to the function that can be kept fixed without making the output unbalanced, when running through all the other input variables. This criteria is directly related to the class of correlation attacks [9,13]. The *nonlinearity* of a Boolean function is defined as the Hamming distance to the nearest affine function when we run through all the input variables. It has many times been pointed out [10,4] that using a Boolean function close to an affine function is probably not a good choice, although no direct attack related to this criteria is known to the authors.

The results in this paper are twofold. Firstly, for a Boolean function in n variables, we establish the maximum nonlinearity for a fixed resiliency in certain cases. We use linear programming as well as algebraic proofs. Interesting results that can be mentioned is that the maximum nonlinearity for a 1-resilient function on $n = 6$ variables is 24 (whereas it is known to be 26 for a balanced function and 28 for unbalanced functions (bent functions)). It is also shown that the maximum nonlinearity for an $(n-3)$-resilient function on n variables is 2^{n-2}. These results relate to the algebraic construction in [2], which provide optimal constructions for these cases.

Motivated by the fact that designers may avoid algebraic constructions due to the possibility of a (very) weak property, we consider in the second part of the paper random generation of a Boolean function with good properties. We

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 35{44, 1999.

present a simple search algorithm for finding Boolean functions with good nonlinearity and some fixed form of resiliency. Such search algorithms have previously been considered in [11,12]. Compared with the algorithm in [11], the proposed algorithm could find better functions (higher nonlinearity) in certain cases.

This paper is organized as follows. Section 2 provides basic definitions and briefly discuss the current knowledge regarding the nonlinearity of Boolean functions. The results from an algebraic construction [2] are reviewed, since many important observations follow from it. In Section 3 we establish, in certain cases, the maximum nonlinearity that can be obtained for a fixed resiliency using e.g. linear programming methods. Section 4 describes a simple search algorithm for finding highly nonlinear balanced Boolean functions, and first order resilient functions. Results and comparison with the genetic algorithm [11] are presented. Section 5 is a brief conclusion.

2 Preliminaries

Although, we will only study the properties of Boolean functions, it can be beneficial to consider its application in the nonlinear combining generator, which is a classic technique for utilizing linear feedback shift registers (LFSRs) in the construction of stream ciphers [10]. Here, the Boolean function $f(x) : \mathbb{F}_2^n \to \mathbb{F}_2$ takes as input the output sequences of n LSFRs.

The function $f(x)$ can be written in algebraic normal form (ANF), i.e.,

$$f(x_1,\ldots,x_n) = a_0 + a_1x_1 + \cdots + a_nx_n + \\ + a_{12}x_1x_2 + a_{13}x_1x_3 + \cdots + a_{12\ldots n}x_1x_2\cdots x_n. \qquad (1)$$

where addition and multiplication are in $\mathbb{F}_2$. The *truth table* of the function $f(x)$, denoted in this paper by f, is a vector of 2^n bits representing the output of the function for each input, i.e., $f = [f_0 f_1 \cdots f_{2^n-1}]^T$, where $f_i \in \mathbb{F}_2$. The *algebraic degree* of $f(x)$, denoted $deg(f)$, is defined to be the maximum degree appearing in the ANF.

As mentioned, a Boolean function $f(x)$ must fulfill certain properties in order to increase the time/space complexity of different attacks. Common attacks are Sigenthaler correlation attack [13] and Berlekamp-Massey linearity synthesis attack [10]. There are at least four main criteria that $f(x)$ should fulfill. These are: balancedness, high nonlinearity, high algebraic degree, and some correlation immunity. The definition of some of these criteria can be derived from the Walsh–Hadamard transform, which is a very convenient way to study the cryptographic properties of Boolean functions.

The Walsh–Hadamard transform of a Boolean function $f(x)$ is defined to be the real-valued function $F(\omega)$ over the vector space $\mathbb{F}_2^n$ given by

$$F(\omega) = \sum_x f(x)(-1)^{\omega \cdot x}; \qquad (2)$$

where a *dot product* of vectors x and $!$ is de ned as $x \; ! = x_1 !_1 + \quad + x_n !_n$. In matrix form the Walsh{Hadamard coe cients of the function can be expressed as

$$F = A \quad f; \tag{3}$$

where F is a column vector of Walsh{Hadamard coe cients, and A is a $2^n \quad 2^n$ matrix of 1 corresponding the terms $(-1)^{! \; x}$ in the Walsh{Hadamard transform for every possible choice of x and $!$ (a Hadamard matrix).

We say that the Boolean function $f(x)$ is *balanced* if $P(f(x) = 1) = P(f(x) = 0) = 0{:}5$. Alternatively, using the Walsh{Hadamard transform, the Boolean function $f(x)$ is *balanced* i $F(0) = 2^{n-1}$.

We de ne the *nonlinearity* of a Boolean function $f(x)$, denoted by N_f, as the Hamming distance to the nearest a ne function, i.e., $N_f = \min_{g 2 A_n} d_H(f; g)$. Here, f and g are the truth tables of $f(x)$ and $g(x)$, A_n is the set of a ne functions on n variables and $d_H(f; g)$ is the Hamming distance between two vectors f and g, i.e., the number of positions where f and g di er.

Alternatively, the nonlinearity of $f(x)$ can be obtained as

$$N_f = 2^{n-1} - \max_{!} jF(!)j; \quad ! \; \neq 0{:} \tag{4}$$

This means that the nonlinearity is determined by the largest absolute value in the transform vector F. Let F_n be the set of all Boolean functions in n variables. The maximal nonlinearity that is possible for a Boolean function $f(x) \; 2 \; F_n$ in n variables is denoted by $NL(F_n)$. The current knowledge about $NL(F_n)$ is summarized below.

- n even: $NL(F_n) = 2^{n-1} - 2^{\frac{n}{2}-1}$ and this nonlinearity is obtained by *bent functions*. We note however that bent functions are not balanced.
- n odd: $NL(F_n)$ is known, when $n = 3; 5; 7$. For $n \quad 9$, the exact value of $NL(F_n)$ is not known [6].

Note that if we restrict ourselves to *balanced* Boolean functions, the maximal nonlinearity will decrease when n is even. For $n = 6$ it is known to be 26, for $n = 8$ it is known to be either 116 or 118, etc [11].

Finally, the Boolean function $f(x_1; \ldots; x_n)$ is said to be m-th order *correlation immune* (m-CI) , if for any m-tuple of independent identically distributed binary random variables $X_{i_1}; \ldots; X_{i_m}$, it is valid that

$$I(X_{i_1}; X_{i_2}; \quad ; X_{i_m}; Z) = 0 \quad 1 \quad i_1 < i_2 < \quad < i_m \quad n; \tag{5}$$

where $Z = f(X_1; X_2; \ldots; X_n)$, and $I(X; Z)$ denotes the mutual information [5]. The condition above is equivalent to saying that any subset of m random variables chosen from $X_1; X_2; \ldots X_n$ is statistically independent of $Z = f(X_1; X_2; \ldots; X_n)$. An m-th order correlation immune function which is balanced is called an m-th order *resilient* (m-resilient) function. Using the Walsh{Hadamard transform, $f(x)$ is m-resilient i $F(!) = 0$ for any $! \; 2 \; \mathbb{F}_2^n$ such that $0 \quad j!j \quad m$, where $j!j$ denotes the weight of $!$.

Consider the combining generator. If the Boolean function $f(x)$ is 1-resilient, then the Sigenthaler correlation attack [13] cannot be performed on a single LFSR. The correlation attack can be modified to consider a pair of LFSRs (since $f(x)$ is not 2-resilient), but this is a much more difficult task than attacking a single LFSR.

As was observed by Sigenthaler [14], there exists a tradeoff between the algebraic degree and resiliency, that is, if the cryptosystem has a high resistance to a correlation attack then it will be less resistant to the linear complexity attack. The result states that if $f(x)$ is balanced, then in order to be m-resilient no product of $n - m$ or more variables can be present in the ANF of $f(x)$.

Among the different designs [4,2,1] of highly nonlinear balanced Boolean functions, the results obtained using an algebraic design [2] of resilient function with controllable nonlinearity provides the best known nonlinearity among all proposed methods. Briefly, in the method of [2], the nonlinearity of a k-th order resilient function is given by, $N_f = 2^{n-1} - 2^{l_1}$, where l_1, $k + 1 \quad l_1 \quad n$, is the smallest integer satisfying

$$\binom{l}{k+1} + \binom{l}{k+2} + \quad + \binom{l}{l} \quad 2^{n-l}.$$

The algebraic degree of $f(x)$ is given by, $deg(f) = n - l_1 - 1$ [2]. Some computed parameters for this construction are presented in Table 1.

	n					
CI	5	6	7	8	9	10
0	12	24	56	112	240	480
1	12	24	56	112	240	480
2	8	24	48	112	240	480
3	0	16	48	96	224	480
4	0	0	32	96	192	448
5	0	0	0	64	192	448
6	0	0	0	0	128	384

Table 1. Nonlinearity N_f obtained by the design in [2].

3 Determining the Maximum Nonlinearity for Fixed Resiliency in Certain Cases

In this section we first use linear programming to establish the maximum nonlinearity in certain cases. A linear programming problem (LP problem) given in a *standard form* is a maximization(minimization) problem of the so{called

objective function z, under a set of linear constraints A. In matrix notation an LP problem can be written as,

$$\begin{aligned} \min(\max)\, z &= c^T x; \\ Ax \quad & b; \qquad\qquad x \quad 0: \end{aligned} \tag{6}$$

There is an extensive literature on the topic [7]. In our case, the variables are binary and hence the problem is actually a 0{1 integer programming (0{1 IP) problem. 0{1 IP is commonly implemented using a *branch and bound* algorithm.

The problem of nding a Boolean function $f(x)$ that is m-resilient and having nonlinearity N_f is equivalent to $F(!)$ satisfying the following criteria:

$$\begin{aligned} F(0) &= 2^{n-1}; \quad \text{balancedness}; \\ F(!) &= 0; \quad 1 \quad j!j \quad m; \quad m\text{-resilient}; \\ jF(!)j \quad & 2^{n-1} - N_f; \quad ! \not\in 0; \quad \text{nonlinearity}: \end{aligned} \tag{7}$$

If we let the constraints in (6) correspond to those given in (7), then a solution to the 0-1 IP problem is equivalent to the existence of an m-resilient function with nonlinearity N_f. We illustrate the procedure with a simple example where we specify the constraints in (7). Consider the question of whether there exists 1-resilient functions on ve variables, $n = 5$, with nonlinearity $N_f = 12$. The set of constraints in (7) is given by

$$\begin{aligned} A_{(00000)} f &= 2^{n-1}; \\ A_{(00001)} f &= 0; \\ &\vdots \\ A_{(10000)} f &= 0; \\ A_! f \quad & 4; \qquad j!j \quad 1; \\ A_! f \quad & -4; \qquad j!j \quad 1; \end{aligned} \tag{8}$$

where $A_!$ denotes the row of the matrix A corresponding to $!$. The object function is not used in the problem description, since any solution is su cient to solve the problem.

Note, that since the 0-1 IP problem is solved by a global optimization technique, it means that if a solution is not found then there do not exist Boolean functions having the speci ed nonlinearity and correlation immunity. In general, 0{1 IP is an NP{hard problem and problems of input size of more than 100 variables are considered as infeasible. Thus, the complete results as obtained for $n = 5$ and $n = 6$ are not likely to be established for $n = 7$ (the number of variables is 128). When running the algorithm, we considered all instances of the problem that could not be solved in three weeks as \infeasible". The results are given in the Table 2, where ( ) indicates infeasible instances. For $n = 5$ the above results could also have been established by an exhaustive search through all 2^{32} possible values of f. For $n = 6$, this would not be possible. Most notable is the fact that $N_f = 24$ for a 1-resilient function when $n = 6$. We give this main result as a theorem.

	Resiliency					
n	0	1	2	3	4	5
5	12	12	8	0	0	0
6	26	24	24	16	0	0
7	*	*	*	*	*	0

Table 2. Maximum N_f obtained by solving a 0{1 IP

Theorem 1. *The maximum nonlinearity for a 1-resilient Boolean function on* $n = 6$ *variables is* 24.

There are many heuristically based arguments, which indicate that the algebraic construction presented in the previous section achieves the highest possible values of the nonlinearity for m-resilient functions, when $m \geq 0$. For instance, using the search algorithm to be described in the next section, we have found millions of functions with nonlinearity 114/116 when $n = 8$, none of which was 1-resilient, see Table 1. There is also an underlying regular structure between the values in Table 1. We formulate the following conjecture, which can be extended to other interesting cases.

Conjecture 1 *The maximum nonlinearity for a 1-resilient Boolean function* $f(x)$ *on* n *variables is given by*

$$N_f = \begin{cases} 2^{n-1} - 2^{\frac{n}{2}}; & n \text{ even} \\ 2^{n-1} - 2^{\frac{n-1}{2}}; & n \text{ odd:} \end{cases}$$

Finally, we provide an algebraic proof of the following fact.

Theorem 2. *The maximum nonlinearity for an* $(n-3)$*-resilient Boolean function* $f(x)$ *on* n *variables is given by*

$$N_f = 2^{n-2}:$$

Proof. (Sketch) From the result of Siegenthaler [14] we know that if $f(x)$ is $(n-3)$-resilient, then the algebraic degree of $f(x)$ is at most $n-(n-3)-1 = 2$. Hence we can assume that $f(x)$ is of degree 2. Through classical results by Dickson [3], any $f(x)$ can be linearly transformed to one of a number of normal forms. Since $f(x)$ must also be balanced, the number of possible normal forms reduces to one, namely

$$\sum_{i=1}^{v} x_i x_{v+i} + x_{2v+1}:$$

The Hamming distance to the a ne functions is 2^{n-2}. Hence, the possible values for the nonlinearity is either 0 (a ne functions) or 2^{n-2}.

We nally note that looking at Table 1, it seems likely that the maximum nonlinearity for an $(n-4)$-resilient function is $2^{n-1} - 2^{n-3}$.

4 A Novel Search Algorithm

In this section, we present a novel search algorithm for finding highly nonlinear balanced Boolean functions, referred to as the Directed Search Algorithm (DSA). The algorithm is then modified to search for 1-resilient functions, which is discussed later. For convenience, we use here the Walsh transform defined by

$$\hat{F}(\omega) = \sum_{x} (-1)^{f(x)}(-1)^{\omega \cdot x}. \tag{9}$$

Let $\hat{f}(x) = (-1)^{f(x)}$. Then the transform operations are comprehensively written in matrix form as $\hat{F} = A\hat{f}$, where $\hat{F}$ and $\hat{f}$ are column vectors of size 2^n, while A is the Walsh–Hadamard matrix as defined before. In the sequel, we call $\hat{f}$ the polarity truth table.

Note that the entries, in both A and $\hat{f}$, takes values in $\{+1, -1\}$. Using the Walsh transform a Boolean function $f(x)$ is *balanced* iff $\hat{F}(0) = 0$. The nonlinearity of $f(x)$ is now given by,

$$N_f = 2^{n-1} - \frac{1}{2} \max_{\omega} |\hat{F}(\omega)|, \quad \omega \neq 0. \tag{10}$$

The condition for m-resiliency remains the same, i.e.,

$$\hat{F}(\omega) = 0, \quad 0 \leq |\omega| \leq m.$$

We now briefly describe DSA. Assume that a function $f(x)$ is given. In order to increase the nonlinearity, DSA tries to decrease the entry with the maximum absolute value in $\hat{F}$. Let vectors $\hat{f}$ and $\hat{F}$ denote the polarity truth table and Walsh coefficients respectively, after the k-th entry in $\hat{f}$ has been complemented. In vector form, this operation can be viewed as a simple addition of vectors

$$\hat{f} = \hat{f} + f^k, \tag{11}$$

where f^k denotes a vector, which is all-zero except in its k-*th* entry $f_k \in \{2, -2\}$. Complementing the k-*th* entry in $\hat{f}$ will change each entry in $\hat{F}$ by a constant value of +2 or -2. A new vector of Walsh coefficients, $\hat{F}$, is obtained by

$$\hat{F} = A\hat{f} = A\hat{f} + Af^k = \hat{F} + A_{:k} f_k, \tag{12}$$

where $A_{:k}$ denotes the k-*th* column of A.

Assume now that the m-th entry of $\hat{F}$ has maximum absolute value in $\hat{F}$, denoted F_{max}, that is, $|\hat{F}_m| = F_{max} = \max_{1 \leq l \leq 2^n} |\hat{F}_l|$. Denote by c a vector of length 2^n with entries $c_j \in \{0, 1\}$, where $c_j = 1$ if entry f_j causes F_{max} to decrease by 2 when complemented, otherwise $c_j = 0$.

Thus, any entry s in vector $\hat{f}$ can be complemented if s is chosen such that $c_s = 1$. Having flipped a single position in $\hat{f}$, we are assured that $|\hat{F}_m|$ is decreased. But if more than one entry in $\hat{F}$ are equal to $|\hat{F}_m|$, there is no guarantee

that these coe cients are decremented when a particular position in f is complemented. Furthermore, the entries in F with value $F_{max} - 2$ will be a ected unpredictably and these coe cients will be increased or decreased in a random manner. To avoid this random oscillation around F_{max}, the algorithm sorts indices according to $|F_m|$, and then chooses to complement the bit that causes as many as possible of the $|F_m|$ with high value to decrease. The algorithm then continues with the same procedure and complements new bits in the function's truth table until it reaches the point when the same $f(x)$ has appeared before. This exception is called a *cycle*. The directed search algorithm can be described as follows:

1. Generate a random balanced Boolean function $f(x)$.
2. Compute F and sort the entries in F, s.t. $|F_{i_1}| \quad |F_{i_2}| \qquad |F_{i_{2^n}}|$.
3. Find the entry s that maximizes v, where v is the integer such that $|F_{i_1}|, |F_{i_2}|, \ldots, |F_{i_v}|$ all decreases when the s-th bit is complemented. Complement the s-th bit in $f(x)$.
4. Check the nonlinearity of the new $f(x)$. If a cycle is detected go to step 1, otherwise go to step 2.
5. Output the best obtained function.

Note that the output function generally is not balanced, but this is not a problem because the existence of just a single zero in the vector F allows us to obtain a balanced function.

Let $f(x)$ be a nonbalanced function on $\mathbb{F}_2^n$. Suppose that $F(\omega_0) = 0$ for some vector $\omega_0 \neq 0$ in $\mathbb{F}_2^n$. Then $f'(x) = f(x) + x \cdot \omega_0$ de nes a new function on $\mathbb{F}_2^n$, which is balanced having the same nonlinearity and algebraic degree as $f(x)$. It is obvious that the nonlinearity and the algebraic degree remains the same, since function $f'(x)$ di ers from $f(x)$ just in linear terms. Recalling the de nition of the Walsh transform (9), we have

$$F'(0) = \sum_x (-1)^{f'(x)} = \sum_x (-1)^{f(x)} (-1)^{\omega_0 \cdot x} = F(\omega_0) = 0.$$

After showing how to obtain a balanced function, we now consider how to obtain 1-resilient function. For a xed $f(x)$, let W^0 denote the set of points in $\mathbb{F}_2^n$ for which the transform value is zero, i.e., $W^0 = \{\omega : F(\omega) = 0; \omega \in \mathbb{F}_2^n\}$. Let B be an $n \quad m; m \quad n$ matrix whose rows are linearly independent vectors from W^0. Obviously, if $m = n$, then it is always possible to move these vectors to some desired positions, using a linear transformation matrix C. The Boolean function $f(x)$, in order to be 1-resilient, must have zeros in $F(\omega)$ for $|\omega| = 1$. Hence, we essentially move some points from W^0 to these n points with $|\omega| = 1$ using a linear transformation.

Let $m = n$ and $C = B^{-1}$. Starting with a function $f(x_1, \ldots, x_n)$, we are able to obtain a new function by

$$f'(x) = f(C \cdot x).$$

We prove that $f(x)$ is a 1-resilient function,

$$\hat{F}(!) = \sum_x (-1)^{f(C\,x)}(-1)^{!\,x} = \sum_x (-1)^{f(x)}(-1)^{!\,B\,x} = 0; \quad \text{for } j!j = 1:$$

In the algorithm, we will hence test whether W^0 contains n independent vectors, whenever we obtain a good function.

Because of the algorithm's low complexity, a huge pool of random functions can be examined in order to obtain the highest nonlinearity with or without requirement on resiliency. For example, to process a random input function $f(x)$ in 10 variables, $n = 10$, did not take more than 0.7 seconds, while the nonlinearity on average is not worse than 476.

Another search algorithm is the one presented in [11], which is a quasi{ random technique based on combinatorial optimization methods. This algorithm, also called GA{HC, is in fact an improved version of the genetic algorithm (GA) [12].

	Nonlinearity			
n	GA{HC nonCI	DSA nonCI	GA{HC 1-CI	DSA 1-CI
8	116	116	112	112
9	236	236	232	**236**
10	484	482	476	**480**
11	980	978	976	976

Table 3. $N_f(n; CI)$ obtained by GA{HC and DSA

The highest nonlinearities obtained by GA{HC and DSA are presented in Table 3. The conclusion is that the DSA improves the performance of GA{ HC for certain sizes of input, when modi ed to seek for 1-resilient functions. All instances of DSA was simulated no longer than two weeks. The algebraic degree of the functions was also checked. All the best 1-resilient functions in Table 3 meet the Siegenthaler inequality regarding the algebraic degree. This is not the case for the algebraic construction in [2]. Hence it seems that the obtained nonlinearity as in Table 3 is better than all known constructions with maximal algebraic degree, see [8].

5 Conclusion

Some theoretical results on the nonlinearity of resilient Boolean functions have been presented. The results could be established only for certain cases, which leave a lot of open problems. They also indicate the strength of the algebraic construction in [2].

In the second part of the paper, a new search algorithm was presented. The algorithm has proved to be very e cient in nding highly nonlinear Boolean function with or without the resiliency of order 1. It is possible to extend the algorithm to consider m-resilient functions, but the e ciency is unclear.

References

1. P. Camion, C. Carlet, P. Charpin and N. Sendrier, \On Correlation-Immune functions", *Advances in Cryptology - CRYPTO'91, Lecture Notes in Computer Science*, 1233, pp. 422{433, Springer-Verlag, 1997.
2. S. Chee, S. Lee, D. Lee, S. H. Sung, \On the correlation immune functions and their nonlinearity", *Advances in Cryptology - ASIACRYPT '96, Lecture Notes in Computer Science*, 1163, pp. 232{243, Springer-Verlag, 1996.
3. L. E. Dickson (1900), *Linear Groups with an Exposition of the Galois Field Theory*, Teubner, Leipzig 1900; Dover, New York, 1958.
4. E. Filiol and C. Fontaine, \Highly Nonlinear Balanced Boolean Functions with a Good Correlation-Immunity" *Advances in Cryptology - EUROCRYPT'98, Lecture Notes in Computer Science*, 1403, pp. 475{488, Springer-Verlag, 1998.
5. R. Gallager, *Information theory and reliable communication,* 1968.
6. X. D. Hou, \On the Norm and Covering Radius of the First-Order Reed{Muller Codes". *IEEE Transactions on Information Theory*, 43(3), pp.1025{1027, 1997.
7. B. Kolman and R. E. Beck, *Elementary Linear Programming with Applications,* Academic Press, 1995.
8. S. Maitra and P. Sarkar, \Highly Nonlinear Resilient Functions Optimizing Siegenthaler's Inequality" *Advances in Cryptology - CRYPTO'99, Lecture Notes in Computer Science*, 1666, pp. 198{215, Springer-Verlag, 1999.
9. W. Meier, and O. Sta elbach, \ Fast correlation attacks on certain stream ciphers", *Advances in Cryptology{EUROCRYPT'88, Lecture Notes in Computer Science*, 330, pp. 301{314, Springer-Verlag, 1988.
10. A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography,* CRC Press, 1997.
11. W. Millan, A. Clark and E. Dawson, \Heuristic design of cryptographically strong balanced Boolean functions" *Advances in Cryptology{EUROCRYPT'98, Lecture Notes in Computer Science* , 1403, pp. 489{499, Springer-Verlag, 1998.
12. W. Millan, A. Clark and E. Dawson, \An e ective genetic algorithm for nding highly nonlinear Boolean functions", In *First International Conference on Information and Communications Security* , *Lecture Notes in Computer Science*, 1334, pp. 149{158, 1997.
13. T. Siegenthaler, \ Decrypting a class of stream ciphers using ciphertext only". *IEEE Trans. Comput.*, vol. C-34, pp. 81{85, 1985.
14. T. Siegenthaler, \Correlation immunity of nonlinear combining functions for cryptographic applications", *IEEE Transactions on Information Theory*, vol. IT{30, pp. 776{780, 1984.

Combinatorial Structure of Finite Fields with Two Dimensional Modulo Metrics*

Edgar Martínez-Moro[1], F. Javier Galan-Simon[2], Miguel A. Borges-Trenard[3], and Mijail Borges-Quintana[3]

[1] Dpto. Matematica Aplicada Fundamental,
Universidad de Valladolid. Valladolid, 47002 Spain
edgar.martinez@ieee.org
[2] Dpto. Organizacion y Gestion de Empresas,
Universidad de Valladolid. Valladolid, 47002 Spain
javi@tita.emp.uva.es
[3] Departmento de Matematicas. Facultad de Ciencias,
Universidad da Oriente. Santiago de Cuba, 90500 Cuba
fmborges,mijailg@csd.uo.edu.cu

Abstract. This paper shows the connection between the combinatorial structure of two dimensional metrics over finite fields (Shortly, Mannheim and Hexagonal metrics) and some group actions defined over them. We follow the well known approach of P. Delsarte [9] to this problem through the construction of association schemes. Association schemes based on this distances are the basic tools we propose to deal with the metric properties of codes defined over two dimensional metrics and their parameters. We note that some examples of cyclotomic association schemes (which we call M schemes and H schemes respectively) fit properly as weakly metric schemes for these metrics.

1 Introduction

In this section we review briefly some basic facts about Gaussian integers and Einsestein-Jacobi integers as well as two dimensional modulo metrics defined over finite fields, for an extensive account see [7,8]. The interest on complex integers in coding theory arises from the fact that they allow us to code QAM(Quadratic Amplitude Modulation) signal spaces. Some of the constructions shown in this paper can be seen also in [14] for the Mannheim metric.

1.1 Gaussian and Einsestein-Jacobi Numbers

In this paper we consider a simple generalization of an integer. We say an algebraic number is an **algebraic integer** if it is a root of a monic polynomial whose

* First and second authors are supported by Junta de Castilla y Leon project "Construcciones criptograficas basadas en codigos correctores", first one is also supported by Dgicyt PB97-0471.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 45–55, 1999.

coe cients are rational integers. Most common examples of algebraic integers are the roots of the equations: $i^2 + 1 = 0$; $^2 + + 1 = 0$

The set of **Gaussian integers** is a subset of complex numbers whose real and imaginary part are integers, ie. $\mathbb{Z}[i]$, indeed they are quadratic integers since $= a + bi \ 2 \ \mathbb{Z}[i]$ is a root of $^2 - 2a - a^2 - b^2 = 0$. The set of **Einsestein-Jacobi integers** are just the subset of the complex numbers given by $\mathbb{Z}[(-1 + i\sqrt{3})=2] = \mathbb{Z}[\ ]$. Again they are quadratic integers since $= a + bi \ 2 \ \mathbb{Z}[\ ]$ is the root of the equation: $^2 - (2a - b) + a^2 - ab + b^2 = 0$.

For $2 \ \mathbb{C}[i]$, we will denote its conjugate by . The square norm of and element is just $k \ k = $. A basic fact in arithmetic is that the units on $\mathbb{Z}[i]$ are $f \ 1; \ ig$ and the units on $\mathbb{Z}[\ ]$ are $f \ 1; \ ; \ (1 + \ )g$ and both are unique factorization domains. We say that two numbers are associated if they di er in the product by one unit. Any number whose norm is a prime is also a prime (the converse is false). For a reference to these facts see [6].

1.2 Embedding Finite Fields in 2-dim Metrics

Case A: Gaussian Integers Let be an element whose norm is a prime integer p, and $p = 2$ or $p \ 1 \mod 4$. It is well known (Fermat's two square theorem) that p can be written as:

$$p = a^2 + b^2 = \quad \text{where} \quad = a + ib \text{ (not unique)}: \tag{1}$$

If we denote by $\mathbb{Z}[i]$ the set of Gaussian integers modulo , we de ne the modulo function $: \mathbb{Z}[i] \ ! \ \mathbb{Z}[i]$ associating to each class in $\mathbb{Z}[i]$ its representant with smallest norm :

$$( ) = r \text{ where } = q + r \text{ and } krk = \min fk \ k j \ = \mod g \tag{2}$$

This can be done because $\mathbb{Z}[i]$ is and Euclidean domain. The quotient q can be calculated as $[\frac{}{p}]$ where $[x]$ denotes the Gaussian integer with closest real and imaginary part to x. Taking the carrier set of $GF(p)$ as $f0; 1; : : : ; p - 1g \ \mathbb{Z}$, we can restrict to $GF(p)$ the application so that it induces an isomorphism [7] $: GF(p) \ ! \ \mathbb{Z}[i]$ given by:

$$\text{For } g \ 2 \ GF(p) \quad (g) = g - \frac{g}{p} \tag{3}$$

Therefore $GF(p)$ and $\mathbb{Z}[i]$ are mathematically equivalent but $\mathbb{Z}[i]$ o ers technical advantages for coding two-dimensional signal constellations [7].

Remark 1 *In the case* $p \ 3 \mod 4 = p$ *and the isomorphism above does not apport any relevant information over* $GF(p)$. *For this type of primes* -1 *is a quadratic non residue of* p, *hence we get the following isomorphism between* $GF(p^2)$ *and* $\mathbb{Z}_p[i]$ *where :*

$$\mathbb{Z}_p[i] = \{ k + il \ j \ k; l \ 2 \ \frac{-(p-1)}{2}; : : : ; -1; 0; 1; : : : ; \frac{(p-1)}{2} \}^{o} \tag{4}$$

constructing $GF(p^2)$ *with the irreducible polynomial* $x^2 + 1$.

Case B: Einsestein-Jacobi Integers In this case it is enough note that the equation (2) is valid if we denote by [ ] the operation rounding to the closest Eisenstein-Jacobi integer. Clearly the size of the eld must be either $GF(p)$ for $p \equiv 1 \bmod 6$ or $GF(p^2)$ for $p \equiv 5 \bmod 6$

1.3 Metrics over $\mathbb{Z}[\,]]=\mathbb{Z}[\,]$

From now on will denote either i or and a prime in the respective set of integers.(Note that must be a square of a prime in the exceptional cases given by $p \equiv 3 \bmod 4$ in Mannheim case and $p \equiv 5 \bmod 6$ in hexagonal case.)

Mannheim Metric Let $; \in \mathbb{Z}[i]$ and let $\gamma = - \bmod$. The Mannheim weight of γ is de ned as: $!_M(\gamma) = j<(\gamma)j + j=(\gamma)j$ and the Mannheim distance between and , is $d_M(\ ;\ ) = !_M(\gamma)$. In other words, if we consider the units $f\ 1;\ ig$ on the ring as \unit steps", the Mannheim distance between two elements is just the number of unit steps from one element to another.

Let $GF(q)$ be an extension of the elds above, ie. a nite eld of odd characteristic with either $q = p^m$ with $p \equiv 1 \bmod 4$ or $q = p^{2m}$ with $p \equiv 3 \bmod 4$. The Mannheim weight of $\gamma \in GF(q)$ is obtained by adding the m weights in a representation of γ [7,8].

Let us consider the vector space $(\mathbb{Z}[i]\ )^n$, and a vector x = $(x_0;\dots;x_{n-1})$. We de ne the Mannheim weight of x as:

$$!_M(\mathrm{x}) = \sum_{j=0}^{n-1} !_M(x_j) \tag{5}$$

This de nes a distance in $(\mathbb{Z}[i]\ )^n$ given by $d_M(\mathrm{x};\mathrm{y}) = !_M(\mathrm{x} - \mathrm{y})$.

Hexagonal Metric As in subsection above we de ne the hexagonal weight of an Einsenstein-Jacobi number γ as the smallest number of unit steps in the set $f\ 1;\ ;\ (1 +)g$ from 0 to γ. Extensions to elds of size p^m for $p \equiv 1 \bmod 6$ or p^{2m} for $p \equiv 5 \bmod 6$ are done as above as well as extending the distance to the vector space $(\mathbb{Z}[\ ]\ )^n$.

2 Association Schemes

2.1 Transitive Actions

In the following discussion we will need some notation on permutation groups acting on nite sets. For an reference on this topic see [1]. For a given permutation group G of elements of a nite set X we denote the orbit of an element $x \in X$ as $(G)(x) = fgx\ j\ g \in Gg$. Two orbits are either identical or disjoint. We denote by $O(G\ j\ X)$ the set of all the orbits of the action. We denote the stabilizers

by $G_x = fg\ 2\ G\ j\ gx = xg$. It is well known the relation between orbits and stabilizers given by:

$$\begin{array}{c} G\ (x)\ !\ G=G_x \\ gx \quad 7! \quad gG_x \end{array} \tag{6}$$

Let $h\ i$ $(\mathbb{Z}[\ ]\ )$ be the group generated by . We shall call this group rotations. Clearly in $\mathbb{Z}[\ ]$ multiplication by an element in $h\ i$ is an isometry for the appropriate distance. So are the elements in the group of translations T of integers in $(\mathbb{Z}[\ ]\ )$. We form the semi direct product of both groups $H = h\ i \ltimes T$. Roughly speaking we will also denote by H the permutation group on $\mathbb{Z}[\ ]$ generated by the permutation ($7!$) and the translations in T . Consider now S_n the permutation group on a set of n elements; we de ne the wreath product $H \wr S_n$ as a permutation group on the set $(\mathbb{Z}[\ ]\ )^n$ de ned by:

$$H \wr S_n = H^n \quad S_n = \quad (\mathrm{h};\ )\ j\ \mathrm{h}\ 2\ H^n;\quad 2\ S_n \tag{7}$$

with the multiplication rule: $(\mathrm{h};\ )(\mathrm{h}^0;\ ^0) = (\mathrm{h}\mathrm{h}^{0};\ \ ^0)$ where $(\mathrm{h}\mathrm{h}^0\ )_j = \mathrm{h}_j\,\mathrm{h}^0\ _{(i)}$

$H \wr S_n$ acts transitively on $(\mathbb{Z}[\ ]\ )^n$ in a very natural way, suppose $(\mathrm{h};\ )\ 2$ $H \wr S_n$ and x 2 $(\mathbb{Z}[\ ]\ )^n$, then:

$$(\mathrm{h};\ )(\mathrm{x}) = \mathrm{x};\ \text{where } x_i = \mathrm{h}_i(x\ _{-1(i)}) \tag{8}$$

The action is easily explained as permuting the input word with and after "flipping" each component j according to the isometry h_j. Clearly it acts as an isometry on the words.

The following lemma [12] will be of importance below. It reduces the action of the wreath product $W \wr S_n$ where W is a permutation group on a nite set X, to the action of the group S_n on the set of all n-uples of orbits of W acting on X

Lemma 1 (Lehmann's Lemma) *Let S_n, W as above, then S_n acts on the set $(Orb(WjX))^n$ in the following way:*

$$(O)_j = (O)\ _{-1(j)} \qquad O\ 2\ (Orb(WjX))^n \tag{9}$$

and the mapping:

$$\begin{array}{c} : Orb(W \wr S_n jX^{\,n}) \ !\ Orb(S_n j(Orb(WjX)^n) \\ (W \wr S_n)(\mathrm{h}) \quad 7! \quad S_n(H) \end{array} \tag{10}$$

is a bijection, where $H\ 2\ (Orb{:}(WjX))^n$ is given by $H_j = (W)(h_i)$.

2.2 Preliminaries on Association Schemes

We will follow the de nition of association scheme given by E. Bannai and T. Ito [2]

Definition 1 *An association scheme with d classes is a pair $S = (X; \{R_i\}_{i=0}^{d})$ of a finite set X and a set of relations $\{R_i\}_{i=0}^{d}$ on X satisfying the following rules:*

1. $R_0 = \{(x;x) \mid x \in X\}$ *(the diagonal relation)*
2. $\{R_i\}_{i=0}^{d}$ *is a partition on* $X \times X$.
3. $\forall i \in \{0,\ldots,d\} \ \exists j \in \{0,\ldots,d\}$ *s.t.* $R_i^t = R_j$, *where* $R_i^t = \{(y;x) \mid (x;y) \in R_i\}$
4. *For each election of* $i;j;k \in \{0,\ldots,d\}$, *the number:* $p_{ij}^k = |\{z \in X \mid (x;z) \in R_i \ \wedge \ (z;y) \in R_j\}|$ *is constant for all* $(x;y) \in R_k$

We can rewrite the above conditions in as matrix relations in the usual way: consider the set of square matrices of order $v = |X|$ given by:

$$i = 1,\ldots,d \qquad D_i = [D_i(x;y)]_{0 \le x,y < v} \qquad D_i(x;y) = \begin{cases} 1 & \text{if } (x;y) \in R_i \\ 0 & \text{elsewhere} \end{cases}$$

Therefore conditions $1-4$ above are equivalent to:

1'. $D_0 = Id$ *(identity matrix)*
2'. $\sum_{k=0}^{d} D_k = J$, *where* J *is the matrix where all entries are* 1.
3'. $\forall i \in \{0,\ldots,d\} \ \exists j \in \{0,\ldots,d\}$ *such that* $D_i^t = D_j$
4'. $D_i \cdot D_j = \sum_{k=0}^{d} p_{ij}^k D_k$

The set of matrices $\{D_i\}_{i=0}^{d}$ is the generating set of a semi simple algebra B over $\mathbb{C}$ called Bose-Mesner algebra. All matrices in the set are linearly independent by condition 2^0, hence B has dimension $d+1$. If the scheme is commutative, it is clear that $D_i D_j = D_j D_i$, and also the algebra is commutative. In this case the algebra is diagonalizable and there are a unique set of primitive idempotents $\{E_0 = \frac{1}{|X|}J; E_1,\ldots,E_d\}$ with J the all ones matrix. The matrix P of change of basis matrix from $\{A_0; A_1,\ldots,A_d\}$ to the set of idempotents is called character table or first eigenmatrix of the association scheme. $Q = |X| P^{-1}$ is called second eigenmatrix. We say that the scheme is symmetric if $R_i^t = R_i$, hence, since all the A_i's are symmetric, the entries of the character table are real numbers.

2.3 Constructing the Mannheim Scheme

In this section we construct and association scheme associated to Mannheim metric over $(\mathbb{Z}[i])^n$. We follow the idea of P. Sole in [17]. We say that a symmetric commutative association scheme over the set X is weakly metric if there is a quasi-distance on X which is constant on all the classes of the scheme. This means that exist a function d from $\{0;1,\ldots,t\}$ to the nonnegative integers, such that: $aR_k b \ , \ d(a;b) = d(k)$

Also Patric Sole proposed a construction for building some of this schemes (See "An all purpose construction" [17]) as follows: Given a finite set X and G a group of isometries on X such that it acts transitively. If we consider the induced action of G in $X \times X$, then the orbits of such action build a weakly metric association scheme. (Usually this is known as the collection of 2-orbits of a transitive permutation group)

From the above discussion is clear that if we consider the orbits of the action:

$$H \wr S_n \quad ((\mathbb{Z}[i]\ )^n \quad (\mathbb{Z}[i]\ )^n)\ !\ ((\mathbb{Z}[i]\ )^n \quad (\mathbb{Z}[i]\ )^n) \tag{11}$$

induced by the transitive action in equation (8), we build an weakly metric association scheme for the Mannheim metric.

From Lehmann's lemma it is clear that orbits can be obtained by considering the action: $H \quad (\mathbb{Z}[i] \quad \mathbb{Z}[i]\ )\ !\ (\mathbb{Z}[i] \quad \mathbb{Z}[i]\ )$, and the pairs in each class of the Mannheim scheme are given by:

$$(x;y)\ 2\ R_k\ ,\ \ (x;y)\ 2\ O_k \tag{12}$$

where the O_k are the orbits of the previous action. Since H is a transitive permutation group, if we take G_0 (i.e. those permutations that x 0), it is known [11] that we have the coset decomposition: $H = G_0\ p_0\ [\ \ [\ G_0\ p_t$, where p_i is the permutation transforming 0 in some complex number belonging to the coset. Therefore orbits can be rewritten as:

$$(x;y)\ 2\ R_k\ ,\ \ x - y\ 2\ G_0\ \ p_k \tag{13}$$

This shows a great resemblance with Clark-Liang schemes [17,13]

Example 1 *We recall the example in [8]. Consider* $GF(13)$ *represented as* $\mathbb{Z}[i]_{3+2i}$. *We have a pictorial representation of it as in gure 1.*

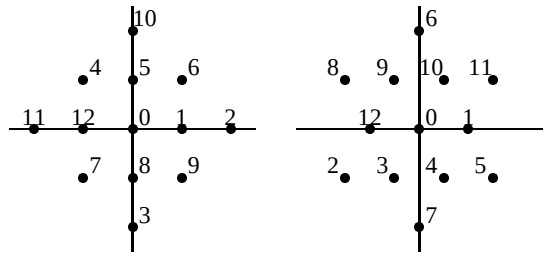


Fig. 1. $\mathbb{Z}[i]_{3+2i}$ and $\mathbb{Z}[\ ]_{3+4}$

Clearly the orbits are given by:

$$G_0\ \ p_0 = f0g; G_0\ \ p_1 = f1; 5; 12; 8g; G_0\ \ p_2 = f6; 4; 7; 9g; G_0\ \ p_3 = f2; 10; 11; 3g$$

And using the de nition in (13) above the relations are given by:

$$D_0 = [1;0;0;0;0;0;0;0;0;0;0;0;0] \quad D_1 = [0;1;0;0;0;1;0;0;1;0;0;0;1]$$
$$D_2 = [0;0;0;0;1;0;1;1;0;1;0;0;0] \quad D_3 = [0;0;1;1;0;0;0;0;0;0;1;1;0]$$

Note that each matrix D is represented only by its rst row since they are circulant, and hence their eigenvalues are calculated easily (see [3]), and they are:

Matrix	*Eigenvalues.*	
	Eigenvalues	*Multiplicity*
D_0	*1*	*13*
D_1	*4*	*1*
D_2	*a*	*4*
D_3	*b*	*4*
	c	*4*

where: $\begin{cases} a = \sum_{i \in G_0 p_1} !^j \\ b = \sum_{i \in G_0 p_2} !^j \\ c = \sum_{i \in G_0 p_3} !^j \end{cases}$ $! = exp(\frac{2\ i}{13})$.

2.4 Hexagonal Schemes

Clearly, if we take now = with the construction in the previous section we get and association scheme, and we will call it **hexagonal scheme**.

Example 2 *Let us represent* $GF(13)$ *as* $\mathbb{Z}[\ ]_{3+4}$. *We have a pictorial representation of it as in gure 1. the orbits are given by:*

$$G_0\ \ p_0 = f0g; G_0\ \ p_1 = f1; 10; 9; 12; 3; 4g; G_0\ \ p_2 = f5; 11; 6; 8; 2; 7g$$

the relations are:

$D_0 = [1; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0]$, $D_1 = [0; 1; 0; 1; 1; 0; 0; 0; 0; 1; 1; 0; 1]$,
$D_2 = [0; 0; 1; 0; 0; 1; 1; 1; 1; 0; 0; 1; 0]$

Matrix	*Eigenvalues.*	
	Eigenvalues	*Multiplicity*
D_0	*1*	*13*
D_1	*4*	*1*
D_2	*a*	*12*

where: $\begin{cases} a = \sum_{i \in G_0 p_1} !^j \\ ! = exp(\frac{2\ i}{13}) \end{cases}$

3 Patterns

We present here a valuable tool for describing the orbits of our association schemes. For our purpose we have to review some well known results in combinatorics on cycle sums and patterns. Most of the material in this section can be found in [4]. First we will x our notation, let Γ a group of permutations on a nite set D (domain) and R (range) another nite set. We will call the elements in the domain places and the ones in the range gures. The functions R^D are called con gurations. If we consider the action: $\Gamma \quad R^D \ ! \ R^D$ where $(\ ; f) \ 7! \ (\ ^? f) = f(\ d) \ \ d \ 2 \ D$ Then $\Gamma^? f \ ^? j \ \ 2 \ \Gamma g$ is a group isomorphic to Γ acting on R^D. Finally a pattern is an equivalence class of con gurations under $\Gamma^?$. Let now $\Gamma = S_n$:

De nition 2 *We call error patterns of the Mannheim scheme or the Hexagonal scheme to the equivalence relations of the action given in (8) where the group* $H = hii$ *or* $H = h \ i$ *respectively.*

Lemma 2 *The error patterns of the Mannheim scheme or the Hexagonal scheme are completely determined by the orbits of the action:*

$$\begin{array}{ccc} h\ i & \mathbb{Z}[\ ] & -! \ \mathbb{Z}[\ ] \\ (\ ^{k};x) & 7! & \ ^{k}x \end{array} \tag{14}$$

Proof. It follows directly from the discussion above and Lehmann's lemma.

We recall the set of gures the orbits of the action above $O_1;:::;O_l$. Hence two error gures given by $(x_1;:::;x_n)$ and $(y_1;:::;y_n)$ are in the same pattern if there is a permutation $2\ S_n$ and gures $(O_{i_1};:::;O_{i_n})$ such that $x_j - y_{\ (j)}\ 2\ O_{i_j}$. Of course for $n = 1$ the gures are the patterns.

Clearly from this setting we are looking for properties of the set of orbits of $S_n^?$ on the range $O_1;:::;O_l$ of orbits of the action in (14).

De nition 3 *Let A an arbitrary commutative ring and let $! : R \to A$ a function. We call $!(r)$ the weight of gure r. The k-th gure sum is de ned as the sum $s_k = \sum_{r2R} !(r)^k$, and the weight con guration of $f\ 2\ R^D$ is given by $!(f) = \prod_{a2D} !(f(a))$*

Clearly two equivalent con gurations under $\Gamma^?$ have the same weight, so we de ne $!(O_i^?) = !(f)$, where $f\ 2\ O_i^?$. [1] and the pattern sum as: $S = \sum !(O_i^?)$ Next theorem allow us deriving of the pattern sum:

Theorem 1 *Let $\Gamma^? = S_n^?$*

$$\sum !(O_i^?) = \sum_{j_1+2j_2+\ \ +nj_n=n} \frac{1}{\prod_{k=1}^{n} k^{j_k} j_k!} s_1^{j_1} s_2^{j_2} :::s_n^{j_n}$$

Proof. For a proof see [4] pp. 281 theorem 23 and pp.290 example 65.

Example 3 *If we let $!(r) = 1$ for all r we get the number of possible error patterns, indeed s_k is the number of orbits in the action (14). From Burnside's lemma we get the number of orbits in (14) and then we compute the number of error patterns. In a Mannheim scheme the number of orbits of (14) is just $\frac{1}{4}(jXj + 3)$ and in Hexagonal scheme $\frac{1}{6}(jXj + 5)$, where respectively:*

$$jXj = \begin{cases} p & \text{if } p \ \ 1 \mod 4 \\ p^2 & \text{if } p \ \ 3 \mod 4 \end{cases} \quad or \quad jXj = \begin{cases} p & \text{if } p \ \ 1 \mod 6 \\ p^2 & \text{if } p \ \ 5 \mod 6 \end{cases}$$

and hence the number of $S_n^?$-orbits for dimension n is just:

$$\sum_{j_1+2j_2+\ \ +nj_n=n} \frac{1}{\prod_{k=1}^{n} k^{j_k} j_k!} \ \ \frac{1}{4}(jXj + 3)^{\ j_1+j_2+\ \ +j_n}$$

[1] We denote by $O^\star$ the orbits of $\Gamma^\star$ and O the orbits of action (14).

and

$$\sum_{j_1+2j_2+\cdots+nj_n=n} \frac{1}{\prod_{k=1}^{n} k^{j_k} j_k!} \left(\frac{1}{6}(jXj+5)\right)^{j_1+j_2+\cdots+j_n}$$

respectively.

Example 4 *If we let* $!(r) = x_i$ *if* $r \in O_i$ *we get* $s_k = \sum_{i=1}^{k} x_i$*, the resulting pattern sum is an homogeneous polynomial in* $x_1, \ldots, x_l$*. For example, consider* $GF(13)$ *represented as* $\mathbb{Z}[i]_{3+2i}$ *(see example 1), and consider the case* $n = 2$*. The pattern sum is:*

$$S = \sum_{j_1+2j_2=2} \frac{1}{\prod_{k=1}^{2} k^{j_k} j_k!}(x_0+x_1+x_2+x_3)^{j_1}(x_0^2+x_1^2+x_2^2+x_3^2)^{j_1}$$
$$= \frac{1}{2}(x_0+x_1+x_2+x_3)^2 + \frac{1}{2}(x_0^2+x_1^2+x_2^2+x_3^2)$$

Indeed, we get no much information since we get all degree two monomials as the possible patterns of errors, i.e. all the combinations of two errors taken from the orbits. Following example gives us some more information:

Example 5 *If we let* $!(r) = x_{d(i)}$ *if* $r \in O_i$*. For example, consider* $GF(13)$ *represented as* $\mathbb{Z}[i]_{3+2i}$ *(see example 1), and consider the case* $n = 2$*. The pattern sum in this case is:*

$$S = \frac{1}{2}(x_0+x_1+2x_2)^2 + \frac{1}{2}(x_0^2+x_1^2+4x_2^2)$$

For example, an error has weight x_0x_2 *if it as error pattern of distance two, thus the number of such patterns is the coe cient of the monomial* x_0x_2 *in* S

Note that many other weight functions can be proposed, and indeed, there is a close relationship in last two examples with weight enumerators of codes. Also in those examples seems reasonable let $x_0 = 1$ since it denotes no error has been made. If we are concerned only with distance patterns, the usual weight imposed is: $!(r) = x^{d(i)}$ if $r \in O_i$. Therefore last example becomes:

$$S = \frac{1}{2}(1+x+2x^2)^2 + \frac{1}{2}(1+x^2+4x^2)$$

4 Conclusions

As we have seen, 1-dimensional schemes de ned here are translation invariant, therefore their eigenvalues are easy to compute (see [16]) as partitions on the set. Also Lehmman lemma gives us a tool for calculating the eigenvalues of higher dimensional schemes giving us descriptions of the orbits (they are also translation

invariant). Therefore we can de ne all the parameters of a code de ned in two dimensional metrics as functions of its weight distribution and the eigenvalues of the scheme as well as de ne the Lloyd polynomial [17,18]. Also MacWilliams theorem can be recover from them in a natural way, not only for linear codes [8], but also for a general code following the construction on [17]. Moreover, the construction in this paper seems to be more natural for two dimensional metrics that the use of complete weight enumerator (see [8]). Anyway, there is a clear relationship between the scheme de ned above an the composition schemes for the Abelian groups $f\ 1;\ ig$, $f\ 1;\ ;\ (1 +)g$ of unit steps (see [5,10]) and also with group characters [1,11].

Further investigations points towards the classi cation of perfect codes over these metrics which must ful ll the constrains and equalities on the eigenvalues [17,18] and on the Lloyd polynomials[2]. Also the isometry classes of these codes are proposed for a deeper investigation.

References

1. J.L. Alperin , R.B. Bell, *Groups and representations*, Graduate Text in Mathematics 162, Springer Verlag (1995).
2. E. Bannai, T. Ito *Algebraic combinatorics I: Association Schemes* Benjamin Cummimgs Publishers (1983)
3. N. Biggs, *Algebraic Graph theory*, 2nd. edition, Cambridge University Press (1993).
4. B. Bollobas, *Modern graph theory*, Graduate text in Mathematics. Springer. New York (1998).
5. P. Camion, *Codes and Association Schemes: Basic Properties of Association Schemes Relevant to Coding*, in " Handbook of Coding Theory vol 2", pp. 1441{ 1566, Ed. V.S. Pless, W.C. Hu man, North-Holland 1999
6. Hardy,G.H., Wright,E.M. *And introduction to the theory of numbers.* Oxford Science Publications. 5th. edition. (1979)
7. K. Huber, *Codes over Gaussian integers.*, IEEE Trans. on Inf. Theory, 40 (1), 207-216 (1994).
8. K. Huber, *The Mac Williams Theorem for Two-dimensional modulo metrics.*, AAECC, 8 (1), 41-48 (1997).
9. P. Delsarte, *An algebraic approach to the association schemes of coding theory.*, Technical report, Philips Research Laboratory, 1973
10. P. Delsarte,V.I. Levenshtein, *Association schemes and coding theory.*, IEEE Trans. on Inf. Theory, vol. 44, 6, pp.2477{2504. October 1998
11. W. Ledermann, *Introduction to group characters.*, Cambridge University Press, 1986
12. H. Lehmann, *Ein vereinheitlichender Ansatz für die Red eld-Polya-de Bruijnsche Abzähltheory.*, PhD. Thesis Universität Giessen, 1976
13. E. Mart nez, F.J. Galan: *Combinatorial structure of arithmetic codes.*, Winter School on Coding and Information Theory 1998 (Ebeltoft, Denmark)

[2] For a computer algebra construction of these polynomials see [15]

14. E. Mart nez,M.A. Borges, M. Borges : *Combinatorial structure of rings of complex integers with Mannheim metric.*, 1st. Workshop on Combinatorics, Geometry, Coding Theory and related areas. CIMAF'99 La Habana, Cuba, March 1999.
15. E. Mart nez: *Computations on character tables of association schemes.*, Computer Algebra in Scienti c Computing 99, pp. 293{307, Springer-Verlag, 1999.
16. H. Tarnanen , *On Abelian Schemes*, TUCKS Technical Report no. 88. Turku Centre for Computer Science 1996
17. P.Sole, *On the parameters of codes for the Lee and modular distance.*, Discrete Mathematics 89 (1991), pp. 185{194.
18. P.Sole, *A Lloyd theorem in weakly metric association schemes.*, Europ. J. of Combinatorics. 10 189{196 (1989)

A New Method for Generating Sets of Orthogonal Sequences for a Synchronous CDMA System

Helen Donelan and Timothy O'Farrell

University of Leeds, Leeds, LS2 9JT, UK
eenhmd@electeng.leeds.ac.uk

Abstract. A new, systematic method of generating orthogonal sets of sequences with good correlation properties is described. An orthogonal set is de ned as a collection of n sequences, of length n chips, that are mutually orthogonal. Although there are many possible combinations of sequences forming orthogonal sets of a speci ed length, few have been identi ed with a structured method of generation such as Walsh codes and orthogonal Gold codes. The application of the new sequences discussed is orthogonal spreading codes in a synchronous code division multiple access (CDMA) system and their correlation properties are considered accordingly.

1 Introduction

Orthogonal sequences are utilised in many speci cations, in particular CDMA spread spectrum systems to improve the bandwidth e ciency. The most common orthogonal sequences and those employed in or proposed for today's communications systems are Walsh codes[1] and more recently orthogonal Gold codes[2]. The new algorithm proposed is related to that used to generate orthogonal Gold codes but produces large numbers of di erent orthogonal sets with favourable crosscorrelation values between sets of the same size. The procedure generates $(n-1)$ distinct, orthogonal sets of n sequences of length n. Sequences are represented by the notation given in (1).

$$\{x_i\} = (x_0; x_1; x_2 \cdots\cdots x_{n-2}) \tag{1}$$

The sets of sequences are represented by the notation given in (2).

$$\left\{x^{(k)}\right\} = \left\{x_i^0\right\}; \left\{x_i^1\right\} \cdots \left\{x_i^{n-1}\right\} \tag{2}$$

The sequences contain elements of the alphabet $\{1; -1\}$ or equivalent de nitions can be used by mapping $\{1 \rightarrow 0\}$ and $\{-1 \rightarrow 1\}$ and replacing multiplication operations between elements with modulo-2 addition. The second and third sections of this paper detail the novel construction method used to create the orthogonal sets. The fourth and fth sections outline some of their properties, the set cross correlation and mean-square correlation parameters which have been described in relation to their consideration in a CDMA spread spectrum system.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 56{62, 1999.

2 Method of Construction

The orthogonal sequences are developed from a set of sequences created using the Gold sequence construct. Gold sequences are constructed from a preferred pair of maximal length sequences, by the element-by-element multiplication of one sequence with every phase shift of the second sequence. Orthogonal Gold sequences can then be constructed from this family of Gold sequences by appending an additional '1' on the end of each sequence. Although, for optimum periodic crosscorrelation the two m-sequences should be preferred pairs[3], this construct can be applied to any pair of m-sequences of the same length to produce orthogonal sets of sequences. Using the new method, orthogonal sequences are developed from a family of sequences generated using the Gold sequence construct, i.e. sequences generated by the multiplication of one m-sequence with all shifts of a second m-sequence. The two sequences are not necessarily preferred pairs. Two m-sequences of length $(n-1)$ are represented by $\{a_i\}$ and $\{b_i\}$ where:

$$\{a_i\} = (a_0; a_1; a_2 \ldots\ldots a_{n-2}) \tag{3}$$

$$\{b_i\} = (b_0; b_1; b_2 \ldots\ldots b_{n-2}) \tag{4}$$

Using the Gold construct method on these two sequences forms the set of sequences given in (5).

$$\left\{g^{(k)}\right\} = \begin{cases} \{a_i\} \quad T^k\{b_i\} & \text{for } 0 \quad k < n-1 \\ \{a_i\} & \text{for } k = n-1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Where $T^k\{b_i\}$ represents a cyclic shift of $\{b_i\}$ by k chips and is the element by element multiplication. The nal member of the set is one of the original m-sequences $\{a_i\}$. For future reference, the set of sequences de ned by (5) will be referred to as a Gold constructed set of sequences. Performing the following procedure on the above Gold constructued sequences produces a set of orthogonal sequences.

Step 1: Make the entry in the rst column a '1' (the rst chip of the sequence).
Step 2: If the rst chip of the sequence was already a '1' and has not therefore been altered by step 1 then a x a '-1' on the end of the sequence.
Step 3: If the rst chip of the sequence was a '-1' to begin with and has therefore, been altered by step 1 then a x a '1' on the end of the sequence.
Step 4: Repeat steps 1 to 3 for all sequences of the Gold constructed set.

This procedure can be represented as follows:
Let $\left\{u^{(k)}\right\}$ be the set of sequences $\left\{g^{(k)}\right\}$ with the rst chip, $g_0^{(k)}$ of every sequence removed. Then, the orthogonal set of sequences can be represented by (6).

$$\left\{v^{(k)}\right\} = \left\{1; \left\{u^{(k)}\right\}; -g_0^{(k)}\right\} \tag{6}$$

The set of sequences $v_{(k)}$ is a set of n sequences of length n that are orthogonal to each other. By following the same procedure with the same m-sequences but with the m-sequences in a di erent initial phase shift, an entirely di erent set of orthogonal sequences is generated.

$$\{a_i\}_2 = (a_1; a_2 \ldots\ldots\ldots a_{n-2}; a_0) = T^1\{a_i\} \tag{7}$$

$$\{b_i\}_2 = (b_1; b_2 \ldots\ldots\ldots b_{n-2}; b_0) = T^1\{b_i\} \tag{8}$$

For each of the initial phases of the m-sequences, where the circular shift is the same on both m-sequences, there is a di erent orthogonal set. These have been called base sets and are distinct from each other. No sequence appears in more than one set. For a pair of m-sequences of lengths $(n - 1)$ there exists $(n - 1)$ base sets of n sequences of length n.

3 Alternative Constructs

By changing the initial phases or shifts of the m-sequences, the base sets produced are di erent. The $(n - 1)$ base sets can be used as a basis to study the influence of the initial shift of the m-sequences on the properties of the orthogonal sets produced by them. Firstly, interchanging $\{a_i\}$ and $\{b_i\}$ so that the Gold constructed family is derived from $\{b_i\}$ multiplied by all shifts of $\{a_i\}$ produces, as before, a Gold constructed family for each of the $(n - 1)$ phase shifts. From these, new base sets are formed. The sequences produced are the same as those in the original base sets, as the same combinations of shifted sequences are used in the construction, but in this case they appear in a di erent order and are therefore grouped into di erent base sets. These sets are still orthogonal, so the order of the m-sequences is irrelevant to the generation of orthogonal sets. Secondly, shifting the initial phase of one of the m-sequences with respect to the other, produces the same combination of sequences grouped into the same base sets but the order of the sequences within the base sets is dependent on the size of the circular shift on the one m-sequence.

Investigations were carried out to look at the e ects of changing a di erent column to all ones (step 1) other than the rst column. Excluding the original base sets, an additional $[(n - 1) \quad (n - 2)]$ orthogonal sets can be produced. These sets are not completely distinct from the original base sets, some of the sequences from the base sets were repeated within these new sets.

All the above variations on the new method still produce orthogonal sets of sequences. A pair of m-sequences produces $(n - 1)$ base sets and all variations above, concerning the initial phase shift of the m-sequences and changing di erent columns, are related back to these base sets.

4 Correlation Properties

In CDMA systems, orthogonal sequences are used to separate users sharing the same bandwidth. At the receiver, the signal is correlated with the user's unique

sequence to recover the information conveyed. Due to the orthogonality between the required user's sequence and all other sequences, the correlation with all other users is zero, therefore there is no interference. Orthogonal spreading can only be used if all the users are synchronised, as the crosscorrelation value between sequences at di erent time shifts is not zero[4].

In the TIA IS-95 CDMA system the forward channel is synchronous and therefore orthogonal spreading can be used. 64 Walsh codes are used to provide orthogonality between users in the same cell. A user's data is rst spread by one of the 64 Walsh codes and then masked by a long pseudo-noise (PN) sequence unique to the cell so that the same 64 Walsh codes can be reused in each cell. In this case the interference between users of another cell behaves like any long PN code. The new method presented here, generates more than one set of distinct sequences for a given sequence length. To explore the possibility of using more than one set simultaneously, ie. a di erent set allocated to each cell, the maximum interference between sequences of di erent cells must be quanti ed. This can be represented as the peak crosscorrelation value at zero time shift between sequences.

All base sets of sizes $n \quad n$ for values of $n = 8; 16; 32; 64; 128$ and 256 were created and the peak crosscorrelation value at zero time shift between all combinations of sequences of the same length measured and displayed in Table 1. The results improve in relation to set size as the set size increases. Sets of size 32 and above exhibit good set cross correlation properties as the maximum value is less than half the sequence length, Such sequence sets would therefore o er low intercell interference levels, thereby enhancing the capacity of a CDMA cellular system.

Table 1. Values of set crosscorrelation

Set Size	Number of Sets	Peak Correlation Value
8× 8	7	4
16× 16	15	8
32× 32	31	8
64× 64	63	16
128× 128	127	20
256× 256	256	32

5 Mean Square Correlation Parameters

In CDMA systems it is desirable to nd a set of sequence with low crosscorrelation between sequences for all shifts to minimise multiuser interference in a multipath environment. Also it is desirable to have low autocorrelation sidelobes for each sequence for all shifts except the in-phase position for antifading capability and synchronisation purposes. However, in general if a set of sequences have

good autocorrelation properties then the crosscorrelation properties are not very good and vice versa[5]. Therefore some trade o between the two is required. There are many correlation parameters that can be investigated when searching for 'good' sets of sequences. For CDMA systems the mean square value of the aperiodic correlation is considered a reasonable measurement[6].

The mean-square crosscorrelation parameter of a sequence l with each of the other $K - 1$ sequences of the set is given by (9).

$$c^{(l)} = \sum_{k=0;k\neq l}^{K-1} \sum_{m=1}^{N-1} |C_{S_l S_k}(m)|^2 \tag{9}$$

Where, number of sequences in a set (ie. possible number of simultaneous users) is K and length of each of the sequences in the set is N.

Therefore the mean-square crosscorrelation parameter of each sequence of the set with every other sequence of the set (ie. all possible combinations of sequences) is given by (10).

$$c = \sum_{l=0}^{K-1} c^{(l)} \tag{10}$$

Similarly, the mean-square autocorrelation parameter of a sequence l is given by (11).

$$a^{(l)} = 2 \sum_{m=1}^{N-1} |C_{S_l S_l}(m)|^2 \tag{11}$$

This is the sidelobe energy across the whole of the autocorrelation period, except at zero phase shift ie. $m = 0$. Therefore the mean-square autocorrelation parameter of all the sequences of the set is given by (12).

$$a = \sum_{l=0}^{K-1} a^{(l)} \tag{12}$$

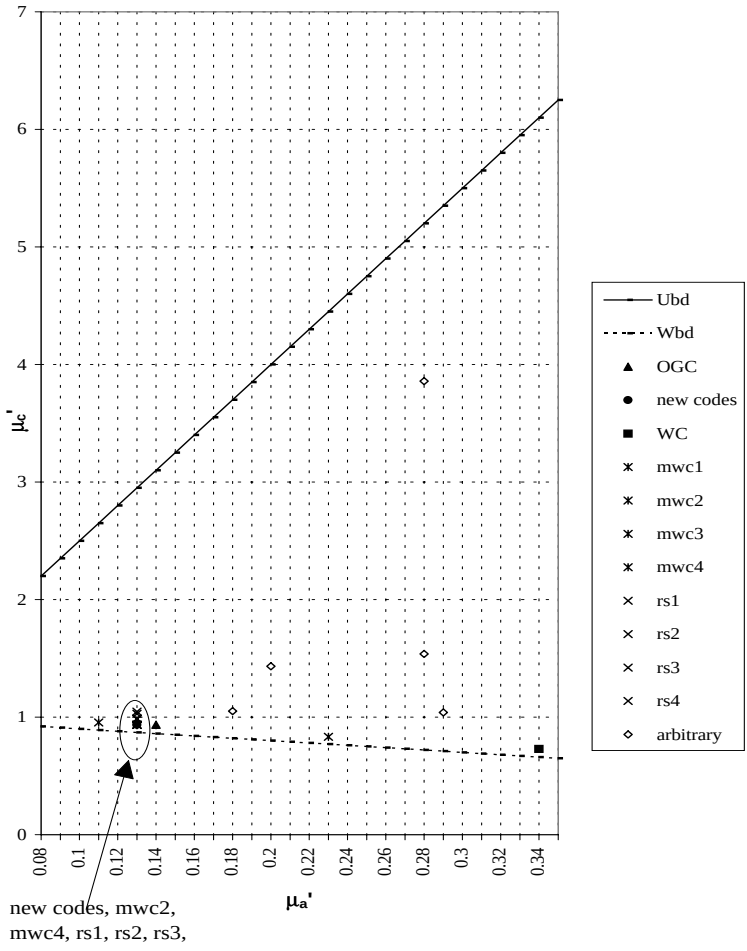
Plotting a^{0} and c^{0} as given in (13) , gives a representation of a set of sequences' mean square cross and autocorrelation characteristics[7] as illustrated in Fig.1.

$$a^{0} = \frac{a}{K(K-1)N^2} \text{ and } c^{0} = \frac{c}{K(K-1)N^2} \tag{13}$$

The values of a^{0} and c^{0} are bounded by the Welch bound, upper bound and maximum and minimum sidelobe energy bounds as illustrated in Fig.1. The sequences plotted are of length 16 and are compared to other sets of sequences of the same length and number. Some are orthogonal sets, ie. Walsh codes(WC), several di erent sets of masked Walsh codes(mwc1-4) and orthogonal Gold codes (OGC), some are randomly selected sequences(rs1-4) and others are sets of sequences that are expected to give relatively bad results(arbitrary), such as some sequences being cyclic shifts of other sequences in the same set. It can be seen that for optimum mean-square autocorrelation a trade-o has to be made with

mean-square crosscorrelation values, emphasizing that both values may not be optimised simultaneously.The new codes exhibit good cross and autocorrelation properties in comparison to the other sets of sequences illustrated.

Fig. 1. Mean-square correlation parameters for di erent sets of sequences, size 16 by 16



6 Conclusion

In summary, a novel method for generating sets of orthogonal sequences has been described. For a pair of m-sequences of length $(n-1)$, a total of $(n-1)$ distinct base sets can be produced. The investigation into the properties of the sets included measuring the peak crosscorrelation between sequences of di erent sets of the same size and measuring mean-square correlation parameters of an individual set. Results were favourable for the set crosscorrelations for the larger sets (sizes 32x32 and above) promoting the idea of using all sequences of the same size ($n \quad (n-1)$) sequences in total, simultaneously, for orthogonal spreading sequences in a synchronous CDMA spread spectrum system environment. Also the mean square auto and crosscorrelation values were calculated, for an indication of how the sequences would perform in a multipath environment. For a set of sequences of size 16x16 the new set of sequences exhibited good mean-square correlation parameters.

References

1. Garg, V.K., Smolik, K., Wilkes, J.E.: Applications of CDMA in Wireless/Personal Communications. Prentice Hall (1997)
2. Tachikawa, S.: Recent Spreading Codes for Spread Spectrum Communication Systems. Elec. and Comm. in Japan. Vol.75. No.6. (1992) 41{49
3. Popovic, B.M.: E cient despreaders for multi-code CDMA systems. Proc. ICUPC. (1997) 516{520
4. Dinan, E.H., Jabbari, B.: Spreading codes for direct sequence CDMA and wideband CDMA cellular networks. IEEE Comms. Mag. (1998) 48-54
5. Sarwate, D.V.: Bounds on crosscorrelation and autocorrelation of sequences. IEEE Trans. on Communications. Vol.IT-25. No.6 (1979) 720{724
6. Pursley, M.B.: Performance evaluation for phase-coded spread-spectrum multiple-access communications - Part I: System analysis. IEEE Trans.on Communications. Vol.COM-25. No.8. (1977) 795{799
7. Schotten, H.: Tutorial: Sequenzen und ihre Korrelationseigenschaften. University of Ulm.(1998)

New Self-Dual Codes over GF(5)

Stelios Georgiou and Christos Koukouvinos

Department of Mathematics,
National Technical University of Athens,
Zografou 15773, Athens, Greece.

Abstract. Self-dual codes and orthogonal designs have been studied for a long time as separate research areas. In the present paper we show a strong relationship between them. The structure of orthogonal designs is such as to allow us a much faster and more systematic search for self-dual codes over GF(5).
Using our method we constructed the following linear self-dual codes over $GF(5)$:(i) [4,2,2], (ii) [8,4,4], (iii) [12,6,6], (iv) [16,8,6], (v) [20,10,8], (vi) [24,12,9], (vii) [28,14,10]. The codes (i), (ii), (iii), (v) are extremal. A [28,14,10] code is constructed here for the first time.

Key words and phrases: Self-dual codes, construction, orthogonal designs.

1 Introduction

We first give some basic definitions which are needed in order to explain our method for the construction of self-dual codes. Self-dual codes are important because many of the best codes known are of this type and have a rich mathematical theory. An *orthogonal design* of order n and type $(s_1; s_2; \ldots; s_u)$ $(s_i > 0)$, denoted $OD(n; s_1; s_2; \ldots; s_u)$, on the commuting variables $x_1; x_2; \ldots; x_u$ is an $n \times n$ matrix D with entries from the set $\{0, \pm x_1, \pm x_2, \ldots, \pm x_u\}$ such that

$$DD^T = \left(\sum_{i=1}^{u} s_i x_i^2\right) I_n$$

Alternatively, the rows of D are formally orthogonal and each row has precisely s_i entries of the type x_i. In [2], where this was first defined, it was mentioned that

$$D^T D = \left(\sum_{i=1}^{u} s_i x_i^2\right) I_n$$

and so our alternative description of D applies equally well to the columns of D. It was also shown in [2] that $u \leq \rho(n)$, where $\rho(n)$ (Radon's function) is defined by $\rho(n) = 8c + 2^d$, when $n = 2^a b$, b odd, $a = 4c + d$, $0 \leq d < 4$. For more details and construction methods of orthogonal design see [3].

In this paper we restrict our attention in two variable orthogonal designs, i.e in the case where $u = 2$.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 63{69, 1999.

For our consideration we also need some facts from coding theory. Our terminology and notation follow [6]. Let $F = GF(q)$ be the field with q elements where q is a prime power. An $[n; k]$ linear code C over F is a k-dimensional vector subspace of F^n. In particular, codes over $GF(2)$ and $GF(3)$ is said binary and ternary codes, respectively. The elements of C are called codewords and the weight of the codeword is the number of its non-zero coordinates. A minimum weight is the smallest weight among non-zero codewords. An $[n; k]$ code with a minimum weight d is called an $[n; k; d]$ code. Two binary codes are equivalent if one can be obtained from the other by a permutation of the coordinates.

The dual code $C^?$ of C is defined as $C^? = \{x \in F^n \mid x \cdot y = 0 \text{ for all } y \in C\}$. If $C \subseteq C^?$, C is called a self-orthogonal code. C is called self-dual if $C = C^?$. Furthermore C is called doubly-even if the weights of all codewords of C are a multiple of four. A self-dual code is called singly-even if there exists at least one codeword whose weight is $2 \pmod 4$.

A self-dual code C is called extremal if C has the largest possible minimum weight. The known bounds of d for $q = 2, 3, 4$ are given in [7] and [8]. In particular the following theorem is known.

Theorem 1 *([8]) The minimum distance d of a self-dual* $[n; n/2]$ *code* C *satisfies*

$$d \leq \begin{cases} 2\left\lfloor \frac{n}{8} \right\rfloor + 2 & \text{if } q = 2 \text{ and } C \text{ is singly-even} \\ 4\left\lfloor \frac{n}{24} \right\rfloor + 4 & \text{if } q = 2 \text{ and } C \text{ is doubly-even} \\ 3\left\lfloor \frac{n}{12} \right\rfloor + 3 & \text{if } q = 3 \\ 2\left\lfloor \frac{n}{6} \right\rfloor + 2 & \text{if } q = 4 \text{ and } C \text{ is even.} \end{cases}$$

For each length, the details of the largest possible minimum weight is listed in Table I in [1]. Conway and Sloane [1] also gave a list of the possible weight enumerators of binary extremal self-dual codes. The existence of some extremal self-dual codes is an open question in [1].

2 The Method

In this section we will show how we can use an orthogonal design in order to obtain a linear self-dual code over $GF(5)$.

We consider an orthogonal design $OD(n; s_1; s_2)$. Then we replace the first variable by 1 and the second variable by 2. This replacement of course does not affect the orthogonality of the rows, and let us denote the derived matrix by A. We shall take the elements of GF(5) to be either $\{0, 1, 2, 3, 4\}$ or $\{0, \pm 1, \pm 2\}$ using whichever form is more convenient.

On the other hand since there are more orthogonal matrices with elements from GF(5) than orthogonal designs with elements from a set of commuting variables, we use both of them in order to construct the desired codes.

Lemma 1 *If we say* $c = \sum_{i=1}^{u} s_i x_i^2$, *where in our case* $u = 2$, *then the matrix* $C = [aI_n; A]$ *is the generator matrix of a* $[2n; n; d; 5]$ *linear self-dual code if and only if* $c + a^2$ *is divisible by 5.*

Proof. We have that

$$CC^T = [aI_n; A][aI_n; A]^T = (c + a^2)I_n:$$

Thus if $c + a^2$ is divisible by 5 then $CC^T = 0_n$ over $GF(5)$, where 0_n is the $n \times n$ matrix whose entries are all zero, and then the matrix $C = [aI_n; A]$ is the generator matrix of a $[2n; n; d; 5]$ linear self-dual code. On the other hand if the matrix $C = [aI_n; A]$ is the generator matrix of a $[2n; n; d; 5]$ linear self-dual code then $CC^T = 0_n$ over $GF(5)$ and then $c + a^2$ is divisible by 5. □

Example 1 *We consider the following orthogonal design* $OD(8; 2; 6)$.

$$D = \begin{bmatrix} b & a & a & -b & b & b & -b & b \\ a & b & -b & a & b & b & b & -b \\ -a & b & b & a & -b & b & -b & -b \\ b & -a & a & b & b & -b & -b & -b \\ -b & -b & b & -b & b & a & a & -b \\ -b & -b & -b & b & a & b & -b & a \\ b & -b & b & b & -a & b & b & a \\ -b & b & b & b & b & -a & a & b \end{bmatrix}:$$

Then we replace the first variable by 1 and the second variable by 2. This replacement of course does not affect the orthogonality of the rows, and let us denote the derived matrix by A. We shall take the elements of GF(5) to be $\{0; 1; 2; 3; 4\}$. *Then* $[2I_8; A]$ *is the generator matrix of a* $[16; 8; 6; 5]$ *linear self-dual code where A is the following matrix.*

$$A = \begin{bmatrix} 2 & 1 & 1 & 3 & 2 & 2 & 3 & 2 \\ 1 & 2 & 3 & 1 & 2 & 2 & 2 & 3 \\ 4 & 2 & 2 & 1 & 3 & 2 & 3 & 3 \\ 2 & 4 & 1 & 2 & 2 & 3 & 3 & 3 \\ 3 & 3 & 2 & 3 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 2 & 1 & 2 & 3 & 1 \\ 2 & 3 & 2 & 2 & 4 & 2 & 2 & 1 \\ 3 & 2 & 2 & 2 & 2 & 4 & 1 & 2 \end{bmatrix}:$$

Its weight enumerator is

$$\begin{aligned} W(z) = {} & 1 + 160z^6 + 192z^7 + 2880z^8 + 5568z^9 + 26848z^{10} + \\ & + 37824z^{11} + 89568z^{12} + 84480z^{13} + 91392z^{14} + \\ & + 39936z^{15} + 11776z^{16}: \end{aligned}$$

It is obvious that any orthogonal design with two variables can give a linear code over $GF(5)$ and if there exist $a \in GF(5)$ such that Lemma 1 holds then this code is self-dual, but in order to find a large enough minimum weight d we must try a lot of orthogonal designs and orthogonal matrices. From the description of our method it is clear that this can also be applied in the construction of self-dual codes over GF(2) and GF(3). Thus we are able to construct a series of the previously known linear self-dual codes over GF(2) and GF(3) by this method.

Example 2 *We consider the following orthogonal design* $OD(4;2;2)$.

$$D = \begin{bmatrix} a & b & a & -b \\ b & a & -b & a \\ -a & b & a & b \\ b & -a & b & a \end{bmatrix}.$$

Then we replace both variables by 1. This replacement of course does not affect the orthogonality of the rows, and let us denote the derived matrix by A. We shall take the elements of GF(3) to be $\{0;1;2\}$. *Then* $[I_4;A]$ *is the generator matrix of a* $[8;4;4;3]$ *linear extremal self-dual code where* A *is the following matrix.*

$$A = \begin{bmatrix} 1 & 1 & 1 & 2 \\ 1 & 1 & 2 & 1 \\ 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix}.$$

Its weight enumerator is

$$W(z) = 1 + 24z^4 + 16z^5 + 32z^6 + 8z^8.$$

3 The Results

In this section we present the results that we find using either orthogonal design or orthogonal matrices. In particular we construct the following linear self-dual codes over $GF(5)$:1. [4,2,2], 2. [8,4,4], 3. [12,6,6], 4. [16,8,6], 5. [20,10,8], 6. [24,12,9], 7. [28,14,10]. The codes (1), (2), (3), (5) are extremal. Self-dual codes over $GF(5)$ with same parameters were constructed, but with a different method, in [4] and [5]. A [24,12,9] code was also constructed in [4]. A [28,14,10] code is constructed here for the first time. Although it has not been proved yet if this code is extremal or not its minimum distance is quite large.

1. The matrix $[I_2;A]$ is the generator matrix of an $[4;2;2]$ extremal singly-even self-dual code where A is the following matrix.

$$A = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}.$$

   Its weight enumerator is

$$W(z) = 1 + 8z^2 + 16z^4.$$

2. The matrix $[2I_4; A]$ is the generator matrix of an $[8; 4; 4]$ extremal self-dual code where A is the following matrix.

$$A = \begin{bmatrix} 3 & 3 & 2 & 3 \\ 3 & 3 & 3 & 2 \\ 3 & 2 & 3 & 3 \\ 2 & 3 & 3 & 3 \end{bmatrix}.$$

Its weight enumerator is

$$W(z) = 1 + 48z^4 + 32z^5 + 288z^6 + 128z^7 + 128z^8.$$

3. The matrix $[I_6; A]$ is the generator matrix of an $[12; 6; 6]$ extremal self-dual code where A is the following matrix.

$$A = \begin{bmatrix} 3 & 3 & 4 & 3 & 4 & 0 \\ 4 & 3 & 3 & 0 & 3 & 4 \\ 3 & 4 & 3 & 4 & 0 & 3 \\ 2 & 0 & 1 & 3 & 4 & 3 \\ 1 & 2 & 0 & 3 & 3 & 4 \\ 0 & 1 & 2 & 4 & 3 & 3 \end{bmatrix}.$$

Its weight enumerator is

$$W(z) = 1 + 440z^6 + 528z^7 + 2640z^8 + 2640z^9 + 5544z^{10} + 2640z^{11} + 1192z^{12}.$$

4. The matrix $[2I_8; A]$ is the generator matrix of a $[16; 8; 6]$ self-dual code where A is the following matrix.

$$A = \begin{bmatrix} 2 & 1 & 1 & 3 & 2 & 2 & 3 & 2 \\ 1 & 2 & 3 & 1 & 2 & 2 & 2 & 3 \\ 4 & 2 & 2 & 1 & 3 & 2 & 3 & 3 \\ 2 & 4 & 1 & 2 & 2 & 3 & 3 & 3 \\ 3 & 3 & 2 & 3 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 2 & 1 & 2 & 3 & 1 \\ 2 & 3 & 2 & 2 & 4 & 2 & 2 & 1 \\ 3 & 2 & 2 & 2 & 2 & 4 & 1 & 2 \end{bmatrix}.$$

Its weight enumerator is

$$\begin{aligned} W(z) = 1 &+ 160z^6 + 192z^7 + 2880z^8 + 5568z^9 + 26848z^{10} + \\ &+ 37824z^{11} + 89568z^{12} + 84480z^{13} + 91392z^{14} + \\ &+ 39936z^{15} + 11776z^{16}. \end{aligned}$$

5. The matrix $[I_{10}; A]$ is the generator matrix of an $[20; 10; 8]$ extremal self-dual code where A is the following matrix.

$$A = \begin{bmatrix}
3&3&4&1&1&3&0&0&0&2\\
1&3&3&4&1&2&3&0&0&0\\
1&1&3&3&4&0&2&3&0&0\\
4&1&1&3&3&0&0&2&3&0\\
3&4&1&1&3&0&0&0&2&3\\
2&3&0&0&0&3&1&1&4&3\\
0&2&3&0&0&3&3&1&1&4\\
0&0&2&3&0&4&3&3&1&1\\
0&0&0&2&3&1&4&3&3&1\\
3&0&0&0&2&1&1&4&3&3
\end{bmatrix}.$$

Its weight enumerator is

$$W(z) = 1 + 2280z^8 + 23408z^{10} + 72960z^{11} + 241680z^{12} + 437760z^{13} + \\ + 1203840z^{14} + 1586880z^{15} + 2229840z^{16} + 1901520z^{17} + \\ + 1418160z^{18} + 528960z^{19} + 118336z^{20}.$$

6. The matrix $[2I_{12}; A]$ is the generator matrix of a $[24; 12; 9]$ self-dual code where A is the following matrix.

$$A = \begin{bmatrix}
4&1&1&1&1&1&1&1&4&0&1&4\\
1&4&1&1&1&1&1&4&1&1&4&0\\
1&1&4&1&1&1&4&1&1&4&0&1\\
4&4&4&4&1&1&1&0&4&4&4&1\\
4&4&4&1&4&1&0&4&1&4&1&4\\
4&4&4&1&1&4&4&1&0&1&4&4\\
4&4&1&4&0&1&4&1&1&1&1&1\\
4&1&4&0&1&4&1&4&1&1&1&1\\
1&4&4&1&4&0&1&1&4&1&1&1\\
0&4&1&1&1&4&4&4&4&4&1&1\\
4&1&0&1&4&1&4&4&4&1&4&1\\
1&0&4&4&1&1&4&4&4&1&1&4
\end{bmatrix}.$$

Its weight enumerator is

$$W(z) = 1 + 1056z^9 + 11088z^{10} + 36960z^{11} + 212352z^{12} + 591360z^{13} + \\ + 2382336z^{14} + 5287040z^{15} + 13796640z^{16} + 23037696z^{17} + \\ + 39528720z^{18} + 46163040z^{19} + 49252896z^{20} + 35604800z^{21} + \\ + 20240352z^{22} + 6832320z^{23} + 1161968z^{24}.$$

7. The matrix $[2I_{14}; A]$ is the generator matrix of a $[28; 14; 10]$ self-dual code where A is the following matrix.

$$A = \begin{bmatrix}
3&3&3&0&3&1&3&3&3&4&3&1&1&0\\
3&3&3&3&0&3&1&0&3&3&4&3&1&1\\
1&3&3&3&3&0&3&1&0&3&3&4&3&1\\
3&1&3&3&3&3&0&1&1&0&3&3&4&3\\
0&3&1&3&3&3&3&3&1&1&0&3&3&4\\
3&0&3&1&3&3&3&4&3&1&1&0&3&3\\
3&3&0&3&1&3&3&3&4&3&1&1&0&3\\
2&0&4&4&2&1&2&3&3&1&3&0&3&3\\
2&2&0&4&4&2&1&3&3&3&1&3&0&3\\
1&2&2&0&4&4&2&3&3&3&3&1&3&0\\
2&1&2&2&0&4&4&0&3&3&3&3&1&3\\
4&2&1&2&2&0&4&3&0&3&3&3&3&1\\
4&4&2&1&2&2&0&1&3&0&3&3&3&3\\
0&4&4&2&1&2&2&3&1&3&0&3&3&3
\end{bmatrix}.$$

Its weight enumerator is

$$\begin{aligned}W(z) = 1 &+ 3500z^{10} + 9856z^{11} + 99820z^{12} + 362152z^{13} + 1938224z^{14} + \\ &+ 6041504z^{15} + 22861496z^{16} + 57103424z^{17} + 154245868z^{18} + \\ &+ 300198752z^{19} + 575888012z^{20} + 833084840z^{21} + 1106507192z^{22} + \\ &+ 1116111808z^{23} + 955594024z^{24} + 598184608z^{25} + 281403808z^{26} + \\ &+ 81992064z^{27} + 11884672z^{28};\end{aligned}$$

References

1. J.H.Conway and N.J.A.Sloane, A new upper bound on the minimal distance of self-dual codes, *IEEE Trans. Inform. Theory*, 36 (1990), 1319-1333.
2. A.V.Geramita, J.M.Geramita, and J.Seberry Wallis, Orthogonal designs, *Linear and Multilinear Algebra*, 3 (1976), 281-306.
3. A.V.Geramita, and J.Seberry, *Orthogonal designs: Quadratic forms and Hadamard matrices*, Marcel Dekker, New York-Basel, 1979.
4. M.Harada, Double circulant self-dual codes over GF(5), *Ars Combinatoria*, to appear.
5. J.S.Leon, V.Pless, and N.J.A.Sloane, Self-dual codes over $GF(5)$, *J. Combin. Theory Ser. A*, 32 (1982), 178-194.
6. F.J.MacWilliams and N.J.A.Sloane, *The theory of error-correcting codes*, North-Holland, Amsterdam, 1977.
7. E.Rains and N.J.A. Sloane, Self-dual codes, in *Handbook of Coding Theory*, eds. V.Pless et al., Elsevier, Amsterdam, 1998.
8. V.D.Tonchev, Codes, in *The CRC Handbook of Combinatorial Designs*, ed. C.J.Colbourn and J.H.Dinitz, CRC Press, Boca Raton, Fla., 1996, 517-543.

Designs, Intersecting Families, and Weight of Boolean Functions

Eric Filiol*

Ecoles Militaires de Saint-Cyr Coëtquidan
DGER/CRECSC/DSI
56381 Guer Cedex
efiliol@mailhost.esm-stcyr.terre.defense.gouv.fr

Abstract. Determining the weight of Boolean functions is of exponential complexity. By using combinatorial results, it is proved that from their algebraic normal form (ANF), it is possible to have polynomial time results on the weight, for some classes of functions. As a result, the structure of the majority functions MAJ_{2^q-1} is given.

1 Introduction

The weight of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is defined by

$$wt(f) = |\{x \in \mathbb{F}_2^n | f(x) = 1\}|$$

and a balanced Boolean function is such that $wt(f) = 2^{n-1}$.

One important problem for these functions is precisely : computing their weight or at least identifying functions which are balanced or not balanced. This problem occurs recurrently in different areas such as cryptology [12] (since unbalancedness means predictability that is to say weakness), in coding theory [15], in logic circuit design, circuit testing (fault testing [7], built-in self testing [8]).

The applications are very numerous, and their theoretical role very important. Among many, we can take the example of the propagation criterion. It is precisely defined by means of balanced Boolean functions [16]. Another very important example (in cryptology and coding theory) is that of bent functions. They are known to have strong resistance against some cryptanalysis but they are not balanced. Different attempts have successfully succeeded in bypassing this drawback. Either partially-bent functions have been defined [5] or balancedness has been obtained only with high nonlinearity, from bent functions [11].

Unfortunately, determining or computing the weight of Boolean functions is of exponential complexity $O(2^n)$ where n is the number of variables. So as soon as n has large values, the task becomes unfeasible. Consequently, constructing

* also INRIA - Projet Codes - Domaine de Voluceau - Rocquencourt - BP 105 78153 Le Chesnay Cedex France - Eric.Filiol@inria.fr

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 70–80, 1999.

Boolean functions with given balancedness is as complex as computing their weight, except for some easy instances.

In this paper, we will show how to construct some classes of functions with xed balancedness, by applying combinatorial results on the algebraic normal form (ANF) of the Boolean functions. Conversely, considering this ANF, it is possible in polynomial time to have information on the weight. Such information is important in itself for using these functions in the construction of balanced functions [6]. The chosen representation for Boolean functions is their *Algebraic Normal Form* (ANF). It has the advantage that it describes directly the function and yields a compact representation. Moreover, this representation is generally preferred in as di erent areas as cryptology and logic testing for example. Recently [19], the ANF has been used to study the nonlinearity from a new point of view, by considering the number of terms.

The paper is organized as follows. Section 2 contains a brief description of the combinatorial theory we need and some basic notation and de nitions on Boolean functions theory. Section 3 presents a new combinatorial view of the ANF by linking it with combinatorial objects such as designs and exposes results on the weight of some functions. Section 4 gives a combinatorial characterization of the ANF structure of majority functions and proves they are very bad cryptographic functions when n is small.

An extended version of this paper is available by contacting the author. It contains construction and characterization algorithms to build balanced Boolean functions with totally controllable ANF structure.

2 Preliminaries

In all this paper, the addition will be done in the nite eld $\mathbb{F}_2$ (modulo 2), unless otherwise stated. We limit ourself to the functions without constant since $wt(1 + f) = 2^n - wt(f)$.

2.1 Boolean Functions

Since we exclusively focus on the ANF of Boolean functions, some notation need to be set. Let f a Boolean function, $f = \mathbb{F}_2^n \to \mathbb{F}_2$. Its ANF will be described by

$$f(x_1; x_2; \ldots; x_n) = \sum_{2\mathbb{F}_2^n} a\ x \qquad a\ 2\ \mathbb{F}_2$$

where $x = (x_1; x_2; \ldots; x_n)$ and $x = x_1{}^{1} x_2{}^{2} \ldots x_n{}^{n}$ with $_i\ 2\ \mathbb{F}_2$. The coe - cients a of the ANF can be obtained by the Möbius transform [15] :

$$a = g( ) = \sum f(_1; _2; \ldots; _n)$$

where describes the partial ordering of the subset-lattice of $\mathbb{F}_2^n$. Otherwise said, , $_i$ $_i$ $8i$. Since the Möbius transform is involutive we also

have $f(x) = \sum_{x} g(\ )$. In a binary context, we can denote any monomial of f by and the ANF itself as the set :

$$A(f) = \{ \ 2 \mathbb{F}_2^n | a \ = 1\}$$

This set yields a more convenient way to manipulate the ANF. The *minimum degree* of f, $d_{min}(f)$ of the ANF is de ned as :

$$d_{min}(f) = \min_{2A(f)} \{wt(\ )\}$$

where *wt(.)* denotes the Hamming weight. The minimum degree is then the minimum total degree of the di erent monomials composing the ANF. The degree of f, seen as a multivariate polynomial, will be noted $deg(f)$. We will use the notion of *support* of $2 \mathbb{F}_2^n$:

$$supp(\ ) = \{i \ 2 \mathbb{N}_n \ | \ _i = 1\}$$

So $supp(\ )$ is a subset of $\mathbb{N}_n = \{1;2;:::;n\}$ when we consider non-zero coordinates of in its base 2 decomposition. Equivalently, $supp(\ )$ is identi ed to itself, since the context is binary.
We will then need the notion of *covering* of a monomial and we denote it

$$cov(\ ) = \{ \ 2 A(f) | supp(\ ) \ supp(\ )\}$$

When $2 A(f)$ then $|cov(\ )| \ 1 \bmod 2$ and is equal to the Möbius coe cient a . The covering of will be said odd if $|cov(\ )| \ 1 \bmod 2$ and even otherwise. Finally, x is said to cover $2 A(f)$ if $supp(\ ) \ supp(x)$ or equivalently denoted if x. We will note $<>$ to describe the fact that and are non comparable elements for this partial ordering (antichain). Let us sum up the previous notation by a short example.

Example 1 *Let $f(x_3;x_2;x_1) = x_1 + x_1x_2 + x_2x_3 + x_1x_2x_3$ over $\mathbb{F}_2^3$. Then we have $A(f) = \{(0;0;1);(0;1;1);(1;1;0);(1;1;1)\}$, and $supp(111) = \{1;2;3\}$ and covers all the monomials. Moreover $d_{min}(f) = 1$.*

The notation $f(x)$ and $f(A)$ for some $A \ \mathbb{N}_n$ and $A = supp(x)$ will be equivalent. Evaluating one of them consists in counting the monomials x (by involutivity of the Möbius transform). At last, $f(x) = f(A)$ will denote $f(\mathbb{N}_n \ nA) = f(x + 1)$ where 1 stands for $(1;1;1;:::;1;1)$.

It is a known result [17] that ANF of balanced Boolean functions have degree $< n$. We now give such a characterization with the minimal degree.

Proposition 1 *Let f be a Boolean function on $\mathbb{F}_2^n$. If $d_{min}(f) > d\frac{n}{2}e$ then f is underbalanced.*

Proof. If $d_{min}(f) > d\frac{n}{2}e$, then $f(x) = 1$ if and only if $wt(x) \ d_{min}$. According to the parity of n, by using the binomial expansion of 2^n, it is easy to see that less than 2^{n-1} such values give $f(x) = 1$. The proposition is proved. ∎

2.2 Designs and Intersecting Families

Definition 1 *A $t-(v;k;\ )$ design is a pair $(V;B)$ where V is a v-element set of points and B is a collection of k-element subsets of V (blocks) with the property that every t-element subset of V is contained in exactly blocks.*

When $t = 2$ such a design is called a *Balanced Incomplete Block Design* (BIBD). In this case, two other parameters are of great use : r the replication number, that is to say the number of blocks in which each point is contained and b which is the number jBj of blocks of the design.

Some necessary conditions on the parameters must be satis ed for the existence of such an object (*admissible parameters*) :

- $r(k-1) = (v-1)$
- $bk = vr$
- $v \quad b$ (Fisher's inequality)

Working on $\mathbb{F}_2$, we will consider only simple designs, in other words, designs with no repeated blocks. Additionally, a design will be called complete if it is simple and contains $\binom{v}{k}$ blocks. A BIBD is *symmetric* (SBIBD for short) when $b = v$ or equivalently $k = r$. Then is the constant number of points in every intersection of two blocks.

An arc is a s-element subset of V containing no block of B and a s-arc will be said complete if it is not properly contained in a $(s+1)$-arc.

In order to unify the notation between combinatorial world and that used for Boolean functions, we will replace v by n where n denote the number of variables of the function. So we will talk about $(n;k;\ )$ design since we will use this kind of object to study Boolean functions.

The other combinatorial concept which will be important for us is that of intersecting family.

Definition 2 *[4] Let F be a family of subsets of a n-set. F is said intersecting if*

$$8A\ 2\ F;\ 8B\ 2\ F;\quad A \setminus B\ \not\in\ ;$$

The interesting property of this kind of combinatorial objects, very important for characterizing the balanced Boolean functions, is now given :

Proposition 2 *Let F be an intersecting family of subsets of a n-set. Then*

$$jFj \quad 2^{n-1}:$$

There exist intersecting families reaching this upper bound.

Intersecting families of size 2^{n-1} are too numerous to even try to class them. See [4] for some examples.

For an extended presentation of these preceding combinatorial objects see [2,4,10,13].

3 ANF and SBIBD

The strong regularity and symmetry properties of SBIBD make them hopefully good candidates as construction objects for balanced Boolean functions. Indeed, balancedness can itself be seen as a regularity property. We will limit ourself in a rst time to the SBIBD, these combinatorial objects being the most strongly regular but all the results can be easily generalized to other $t-(v;k;\ )$ designs.

First of all, we need to de ne precisely the links between designs (and more generally combinatorial structures) and the ANF of a Boolean function.

De nition 3 *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ a Boolean function. Let $B = \{supp(\ ) \mid \in A(f)\}$ and $V = \mathbb{N}_n$. Then f and the structure $(V;B)$ will be said associate. When $(V;B)$ is a $t-(n;k;\ )$ design for some parameters n, k and , it is called the associate design of f.*

The structure $(V;B)$ can be any of those known (PIBD, projective planes, designs, orthogonal arrays, ...[10]). So in case of $t-(n;k;\ )$ designs, the ANF contains only monomials of total degree k, that is to say, $d_{min} = k = deg(f)$. Each block is in fact the support of one monomial of the ANF. Hence, evaluating $f(x)$ for some $x \in \mathbb{F}_2^n$ consists in considering how many blocks are included in $supp(x)$.

Example 2 *The ANF $f(x_3;x_2;x_1) = x_1x_2 + x_1x_3 + x_2x_3$ is associated with a $(3;2;1)$ (complete) symmetric design with $V = \{1;2;3\}$ and $B = \{\{1;2\};\{1;3\};\{2;3\}\}$. Then $f(111) = 1$ since $supp(111) = \{1;2;3\}$ contains the three blocks (monomials).*

A rst property concerning SBIBD must be established for some further results.

Proposition 3 *In a $2-(n;k;\ )$ SBIBD, with $n \neq k$, for all $(b_i;b_j) \in B \times B$ we have*

$$b_i \cup b_j = V \quad , \quad k = n-1$$

Proof. By de nition $|b_i| = |b_j| = k$ and by symmetry property $|b_i \setminus b_j| =$. If $b_i \cup b_j = V$ then we have $= 2k-n$. With the symmetry property again we also have $= \frac{k(k-1)}{n-1}$. We easily obtain the following equation in unknown k : $k^2-(2n-1)k+(n^2-n) = 0$. The only possible solution gives $k = n-1$. The converse is straightforward to prove. □

Remark : the SBIBD relating to proposition 3 are the $(n;n-1;n-2)$ complete symmetric design.

To illustrate this combinatorial approach, we can di erently characterize a very simple result, generally easily proved by polynomial approach :

Proposition 4 *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ a Boolean function associated with an $(n;n-1;n-2)$ symmetric design. Then f is never balanced except for $n \in \{2;3\}$ and its weight is given by :*

$$wt(f) = \begin{cases} n \text{ if } n \text{ even} \\ n+1 \text{ if } n \text{ odd} \end{cases}$$

Proof. By de nition, the ANF of f only contains monomials of degree $n-1$. It follows that only the $x \in \mathbb{F}_2^n$ of weight n and $n-1$ could give $f(x)=1$ (by application of proposition 3). Those of weight $n-1$ do. There are n such values corresponding to the number of blocks. It su ces to check for the sole $x=1$ of weight n and we get the result on the weight. For f to be balanced, we must solve $2^{n-1}=n$ (n even) or $2^{n-1}=n+1$ (n odd) which corresponds to $n \in \{2;3\}$. □

The only balanced Boolean functions associate to such SBIBD are then the linear function on two variables and the sole function of degree 2 which is the function of the Example 2.

In order to generalize the result of Proposition 4, we will use the following proposition :

Proposition 5 *Let be a $(n;k;\ )$ SBIBD. Then the number of subsets of $\mathbb{N}_n$ containing at least one block of the SBIBD is lower or equal to 2^{n-1} .*

Proof. We use the fact that the blocks of a SBIBD consists of an intersecting family (since $\neq 0$). It follows that the subsets containing at least one of them consists of an intersecting family too. The result is obtained by applying Proposition 2. □

Remark : It is not possible to forecast equality to 2^{n-1}. For example the $(7;3;1)$ SBIBD (projective plane of order 2) reaches it but its complementary design $(7;4;2)$ does not.

We now can give the general result :

Theorem 1 *Let be a $(n;k;\ )$ SBIBD with $(n;k) \neq \{(2;1);(3;2)\}$ and f its associate Boolean function. Then f is underbalanced that is to say*

$$wt(f) < 2^{n-1}$$

Proof. We will take $v > k$ since for this case the associate function is obviously underbalanced (its weight is moreover 1). Let us consider $U = \{supp(x) | f(x) = 1\}$ and let F denote the intersecting family, which consists of all subsets of $\mathbb{N}_n$ containing at least one block. Clearly $U \subset F$ ($f(x)=1$ if there exists an odd number of blocks contained in $supp(x)$).
If $|F| < 2^{n-1}$ the result is immediate.
Suppose $F = 2^{n-1}$ and consider $a = b_i \cup b_j$ the union of any two blocks of B ($|a| = 2k-\ $). We have $a \in U$ if and only if the number of blocks included in a is odd. Let us now show that the union of any two blocks contains only two blocks. So suppose that there exists a block b_l such that $b_l \subset (b_i \cup b_j)$ with $i;j;k$ all distinct. This is equivalent to one of the three following cases coming from the fact that $|b_i \cup b_j \cup b_l| = 2k- \ = 3k-3\ + |b_i \setminus b_j \setminus b_l|$:

- $2k- \ = 3k-2\ (|b_l \setminus b_j \setminus b_i| = \ )$. This lead to the non-valid solution $v=k$.
- $2k- \ = 3k-3\ $ (all two-block intersection are disjoint). This equation yields a $(2k-1;k;\frac{k}{2})$ SBIBD, or, since $2k-1$ odd and k even to $(4q-1;2q;q)$ SBIBD for some q (which exists : consider the Brück-Ryser Chowla

theorem [10] and take $q = p^2$). Now consider $U = \{x | f(x) = 0\}$. Thus $|U| \sum_{i=0}^{k-1} \binom{n}{i} + \underbrace{\binom{n}{k}}_{>2^{n-1}} - b$ where $b = \frac{n(n-1)}{k(k-1)}$ is the number of blocks. Thus $|U| = 2^n - |U| < 2^{n-1}$.

{ $2k - = 3k - 2^{\,0} - 3^{\,00}$ ($|b_1 \cap b_j \cap b_i| = {}^{0}$ and $= {}^{0} + {}^{00}$ with ${}^{0} \neq 0$ and ${}^{00} \neq 0$; recall that in a SBIBD represents the number of common points in any two blocks). This easily yields the inequality $< k < 2$. (using ${}^{0} <$). Since $= \frac{k(k+1)}{n-1}$ it equivalent to $k < n < 2k - 1$ that is to say $k > \frac{n+1}{2}$. In this case, the function is underbalanced since the minimal degree d_{min} is too high.

The theorem is proved.□

Remark : if b is even the result is immediate since $f(1) = 0$ whence $U < 2^{n-1}$. The following de nition will be useful later :

De nition 4 *Let f a Boolean function whose ANF satis es the intersecting property (i.e. any two blocks have non empty intersection). The associate intersecting family F is the intersecting family, each subset of which contains the support of at least one monomials of $A(f)$.*

Corollary 1 *Let be a $(n; k;$) SBIBD and f its associate Boolean function. Then we have :*

$$\sum_{i=0}^{k-1} bA^{i}_{n-k} < wt(f) < 2^{n-1}$$

where $b = \frac{n(n-1)}{k(k-1)}$ is the number of block of the SBIBD.

Proof. The upper bound comes from Theorem 1. The lower bound expresses the number of x covering only one block (so $f(x) = 1$) the weight of which is ranging from k to $2k - - 1$ (since the intersection of any two blocks has points).□

To get a more optimal lower bound, we thus should take account of the number of s-arcs ($2k - s m$, m standing for the value of maximality for an arc in a SBIBD). Enumerating these arcs is still an open problem since only existence results are known (particularly for partial geometries; for details see [3,14,18])

This corollary gives an interesting characterization of a family of underbalanced Boolean functions (and of course of overbalanced ones by taking $1 + f(x)$) with bounded weight, only with knowledge of the structure of their ANF. Theorem 1, contrary to what one can think, is not disappointing at all. Indeed, it reinforces a very important principle of cryptology : *structure is often equivalent to weakness*. Here the strong properties of the SBIBD give to its associate Boolean function imbalance, which constitutes a cryptographic weakness. If we give up the symmetry property of SBIBD, we have the very strong following result :

Theorem 2 *Let be* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *with* $n = 2^q - 1$, *whose ANF describes a* $(2^q - 1; 2^{q-1}; \binom{2^q-3}{2^{q-1}-2})$ *design (associate design). Then* f *is balanced.*

Proof. Such a function (since associate to a complete design) contains $\binom{2^q-1}{2^{q-1}}$ monomials (blocks). Hence the associated intersecting family has clearly size of 2^{n-1} where $n = 2^q - 1$. For any two distinct blocks b_i and b_j of this design we have $|b_i \setminus b_j| \in \{1; 2; \ldots; \frac{n-1}{2}\} = K$ and $|b_i \cup b_j| = |b_i| + |b_j| - |b_i \setminus b_j|$ otherwise said $|b_i \cup b_j| = 2\frac{n+1}{2} - k = n + 1 - k$ where $i \in K$. So f will be balanced if the union of two blocks covers an odd number of blocks (without loss of generality it suffices to consider the union of only two blocks; in fact in proof of theorem (6) we will show that considering only this case is sufficient) or equivalently said if

$$\forall k \in K \qquad \binom{n+1-k}{\frac{n+1}{2}} \equiv 1 \mod 2$$

Since $supp(\frac{n+1}{2}) \subseteq supp(n + 1 - t)$, we get the result by applying Lucas' theorem [1]. □

4 ANF Structure of the Majority Functions

The designs of theorem 2 are the complete design of order $2^q - 1$. The associate function also are the MAJ_{2^q-1}. To prove it, we first give this following strong result on the structure of the MAJ_n functions (balancedness of these functions is a known result [6]). Without loss of generality, we limit ourself to the case of n odd. When n is even, $\binom{n}{\frac{n}{2}}$ functions are to be considered but the proof is the same, yet slightly more technical.

The MAJ_n functions are widely used and thus very interesting : in conception of common logic circuits like n-bit adder, in logic circuits testing, cryptography (like in MD5,SHA or as local combining functions in stream ciphers), in coding theory (majority decoding where these functions can be implemented directly in hard [9, Meggit decoder,pp 1581]). But their use and implementation, for n high, is limited by the exponential complexity of computing their ANF.

Proposition 6 *Let* f *be a* MAJ_n *function. Its ANF satisfies the following conditions and then is balanced :*

1. $\forall \alpha \in A(f), \forall \beta \in A(f), supp(\alpha) \setminus supp(\beta) \neq \emptyset$ (intersecting property)
2. $\forall \alpha \in A(f), |cov(\alpha)| \equiv 1 \mod 2$ (odd covering parity property)
3. $\forall \alpha \in A(f), \forall \beta \in A(f), |cov(\alpha) \cup cov(\beta)| \equiv 1 \mod 2$
   (odd union covering parity property)

Proof. by definition of f and according to the parity of n, we will have $\binom{n}{\frac{n+1}{2}}$ (n odd) or at least $\frac{1}{2}\binom{n}{\frac{n}{2}}$ (n even) monomials. So by taking all the k-subsets of $\mathbb{N}_n$ such that $k \geq d_{min}$ ($d_{min} = \frac{n+1}{2}$ or $\frac{n}{2}$), the family F of all subsets containing at least the support of one monomial of $A(f)$ is intersecting and has size 2^{n-1} (by application of the intersecting property of the ANF and by the number of monomials of total degree d_{min}). $\forall A \in F$ we have two cases to consider :

- $\{ 9 \ 2 A(f)$ such that $supp(\ ) = A$. The odd covering parity ensures that A covers an odd number of so $f(x) = 1$ with $supp(x) = A$
- $\{ A \ 2 A(f)$. By maximality of the prebalancedness, A can be obtained by the union of the supports of some $2 A(f)$ of weight $\frac{n+1}{2}$ (or $\frac{n}{2}$ if n even) and the odd union covering parity property ensures that $f(x) = 1$ for $supp(x) = A$.

In fact, for the last case, we should prove that for any $I \ A(f); jIj = i; jcov(\ _{2I})j \ 1 \bmod 2$ but in fact it su ces to prove it for $i = 2$. If $I_{;} = f \ 2 A(f)j \ [\ g$ by considering $I_{;;}; I_{;;}; I_{;}$ and $I_{;;}$ and supposing that $I_{;;}; I_{;}$ and $I_{;}$ are all three of odd cardinality (that is to say we consider the property true for $i = 2$), we easily show that $I_{;;}$ is of odd cardinality too, by application of the Möbius inversion formula, since

$$jI_{;;}j = jI_{;}j + jI_{;}j + jI_{;}j - \sum_{i;j;k} jI_{i;j} \setminus I_{j;k}j + jI_{;} \setminus I_{;} \setminus I_{;}j$$

The generalization to any i monomials of $A(f)$ ($i > 3$) is immediate. By construction it is easy to see that these functions are the MAJ_n functions since for all x such that $wt(x) \ d_{min} \ f(x) = 1$.

Remark : By application of Theorem 2 and Proposition 6, we have totally de ned the structure of the ANF of the MAJ_{2^q-1} functions. They consists of all the monomials of total degree 2^{q-1}. This result is very important for logic circuits design. Indeed, we no longer need to compute their ANF (exponential complexity) to completely know their structure.

The properties 2 and 3 of Proposition 6 ensure that the intersecting family containing $A(f)$ has not the *union property* ($F \ [\ F' \ \mathbb{N}_n \ ; \ 8F \ 2 \ F; 8F' \ 2 \ F$). Otherwise, according to the Daykin-Lovasz-Schönheim theorem [13], we would have $jFj < 2^{n-2}$.

Unfortunately, the MAJ_n functions are not good cryptographic functions, except for asymptotic values of n.

Proposition 7 *The Boolean functions MAJ_n have correlation order 1 and*

$$P[MAJ_n(x) = x_i] = \frac{1}{2} + \frac{\binom{n-1}{\frac{n-1}{2}}}{2^n}$$

Proof. In fact, by using the fact that $MAJ_n(x) + MAJ_n(x + \overline{1}) = 1$ (self-dual functions [6]) it is easy to prove the correlation order by means of the Walsh transform but this technique don't give the correlation value. So we use a di erent approach, yet close to that of Walsh transform, to prove the complete result. Once again, we limit ourself to n odd, without loss of generality. We have correlation order 1 if $P[MAJ_n(x) = x_i] \ \frac{1}{2}$. So let us compute $jIj = jfx \ 2 \ \mathbb{F}_2^n jMAJ_n(x) = x_i gj$ for some arbitrary i. Two di erent cases are to be considered :

- $x_i = 0$ and $MAJ_n(x) = 0$. Once x_i is fixed to 0, we have to choose at most $\frac{n-1}{2}$ ones among $n-1$ positions . Then x will be of weight $< \frac{n+1}{2}$ and $MAJ_n(x) = 0$. There are $\sum_{j=0}^{\frac{n-1}{2}} \binom{n-1}{j}$ such values.
- $x_i = 1$ and $MAJ_n(x) = 1$. In a similar way, we have $\sum_{j=\frac{n-1}{2}}^{n-1} \binom{n-1}{j}$ such values.

Finally $|I| = \sum_{j=0}^{\frac{n-1}{2}} \binom{n-1}{j} + \sum_{j=\frac{n-1}{2}}^{n-1} \binom{n-1}{j}$ that is to say $|I| = 2^{n-1} + \binom{n-1}{\frac{n-1}{2}}$. To speak in terms of probability, we normalize by 2^n and get the result. $\sharp$

Remark : It is easy to prove in the same way that $P[MAJ_n(x) = \sum_i x_i] = \frac{1}{2}$ when the number of x_i is even. When $n \gg 1$ and with the Stirling approximation of $n!$, we easily see that the second term (deviation from $\frac{1}{2}$) tends to 0. This means that asymptotically, $MAJ_n(x)$ is a good cryptographic function.

5 Conclusion

By using combinatorial results and objects, we have shown that it easy to have a direct knowledge on the weight and the structure of some Boolean functions. Particularly, by linking the ANF of these functions with combinatorial objects like designs, it has been possible to build Boolean functions with given balancedness in polynomial time. Characterization of the structure of some MAJ_n functions becomes also easy, by the same approach. However, and generally speaking, these latter have been proven to be bad for cryptographic use, except for an very high number of variables. These results yields effective means of practical constructions of these different functions, as long as characterization from their ANF. The algorithms are presented in the extended version.

6 Acknowledgements

I would like to thank Anne Canteaut and Pascale Charpin from INRIA. Their valuable comments helped me very much in writing this paper. I thank both and other members of the Codes project at INRIA for their kindness and their welcome.

References

1. E. R. Berlekamp, Algebraic Coding Theory, McGraw-Hill, New-York, 1968
2. Th. Beth, D. Jungnickel, H. Lenz, Design Theory, Cambridge University Press, Cambridge, 1986
3. A. Beutelspacher, *Classical Geometries* in CRC Handbook of Combinatorial Designs, C. J. Colbourn, J. H. Dinitz eds., CRC Press, 1996
4. P. J. Cameron, Combinatorics : Topics, Techniques, Algorithms, Cambridge University Press, 1996
5. C. Carlet, *Partially Bent Functions*, Designs, Codes and Cryptography, 3, 135-145, 1993

6. K. Chakrabarty, J.P. Hayes, *Balanced Boolean Functions*, IEE Proc.-Comput. Digit. Tech., Vol. 145, No. 1, January 1998
7. K. Chakrabarty, J.P. Hayes, *Balance testing and balance-testable design of logic circuits*, J. Electron. Testing : Theory Appl., 1996, 8, pp 81{86
8. K. Chakrabarty, J.P. Hayes, *Cumulative balance testing of logic circuits*, IEEE Trans. VLSI Syst., 1995, 3, pp 72{83
9. R.E. Blahut, *Decoding of cyclic codes and codes on curves*, in Handbook of Coding Theory, V. S. Pless and W. C. Hu man Editors, North-Holland, 1998
10. C. J. Colbourn, J. H. Dinitz eds. The CRC Handbook of Combinatorial Designs, CRC Press, 1996.
11. H. Dobbertin, *Construction of Bent Functions and Balanced Boolean Functions with High Nonlinearity*, Fast Software Encryption, 1994, Lecture Notes in Computer Sciences, Vol. 1008, Springer Verlag.
12. E. Filiol, C. Fontaine, *Highly Nonlinear Balanced Boolean Functions with a good Correlation-Immunity*, Advances in Cryptology - Eurocrypt'98, Vol 1403, Lecture Notes in Computer Science, Springer Verlag, 1998
13. P. Frankl, *Extremal Set Systems*, in Handbook of Combinatorics , R. L. Graham, M. Grötschel and L. Lovasz (eds), Elsevier, 1995
14. J. W. P. Hirshfeld, Projective Geometries over Finite Fields, Oxford University Press, 1979
15. F. J. MacWilliams, N. J. A. Sloane. The Theory of Error-Correcting Codes, North-Holland Mathematical library, North-Holland, 1977.
16. B. Preenel, W. V. Leekwijck, L.V. Linden, R. Govaerts, J. Vandewalle, *Propagation Characteristics of Boolean Functions*, in Advances in Cryptology, Eurocrypt'90, vol 437, Springer Verlag, 1991
17. T. Siegenthaler, *Correlation Immunity of Nonlinear Combining Functions for Cryptographic Applications*, IEEE Transactions on Information Theory, Vol. IT 30, N. 5, 1984, pp 776{780
18. J. A. Thas, *Partial Geometries*, in CRC Handbook of Combinatorial Designs, C. J. Colbourn, J. H. Dinitz eds., CRC Press, 1996
19. Y. Zeng, X.M. Zhang, H. Imai, *Restriction, Terms and Nonlinearity of Boolean Functions*, to appear in the Special Issue on Cryptography, Theoretical Computer Science, in honour of Professor Arto.

Coding Applications in Satellite Communication Systems
[Invited Paper]

Dr Sean McGrath

University of Limerick
Ireland
sean.mcgrath@ul.ie

Abstract. This paper provides a brief insight in satellite communication systems from the perspective of coding applications. CDMA based systems for use in Low Earth Orbit (LEO) satellite systems is the focus of the paper. The code-division-multiple-access (CDMA) format is emerging as a dominant air interface technology for cellular, personal-communications-services (PCS) as well as satellite installations. This transmission technology relies on a combination of spread-spectrum modulation, Walsh coding, and sophisticated power-control techniques. In a typical CDMA transmitter, a data signal is encoded using a Walsh code and then mixed with the RF carrier, which has been spread using a pseudorandom-noise (PN) source. In a base-station transmitter, multiple data signals are assigned unique Walsh codes and combined. In the CDMA receiver, the signal is filtered and fed to a correlator, where it is despread and digitally filtered to extract the Walsh code. The paper examines some weaknesses of such systems.

LEO System

The systems of Low-earth orbiting (LEO) satellites provide mobile voice, data and facsimile and other mobile satellite services for both domestic and international subscribers. The systems consists typically consist of a space segment, a user segment and a Ground segment, which connects to the terrestrial telephone network. The space segment consists of any thing from 10 to 66 satellites orbiting the earth at an altitude of over 1000Km. The user segment is composed of hand-held, mobile and fixed terminals. The ground segment consists of the satellite control center and Gateways. The systems of Low-earth orbiting (LEO) satellites provide mobile voice, data and facsimile and other mobile satellite services for both domestic and international subscribers. The systems consists typically consist of a space segment, a user segment and a Ground segment, which connects to the terrestrial telephone network. The space segment consists of any thing from 10 to 66 satellites orbiting the earth at an altitude of over 1000Km. The user segment is composed of hand-held, mobile and fixed terminals. The ground segment consists of the satellite control center and Gateways.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 81-83, 1999.

CDMA

A CDMA spread spectrum signal is created by modulating the radio frequency signal with a spreading sequence known as a pseudo-noise (PN) digital signal because they make the signal appear wide band and "noise like". The PN code runs at a higher rate than the RF signal and determines the actual transmission bandwidth. Messages can also be cryptographically encoded to any level of secrecy desired with direct sequencing as the entire transmitted/received message is purely digital. An SS receiver uses a locally generated replica pseudo noise code and a receiver correlator to separate only the desired coded information from all possible signals. A SS correlator can be thought of as a specially matched filter -- it responds only to signals that are encoded with a pseudo noise code that matches its own code. Thus an SS correlator (SS signal demodulator) can be "tuned" to different codes simply by changing its local code. This correlator does not respond to man made, natural or artificial noise or interference. It responds only to SS signals with identical matched signal characteristics and encoded with the identical pseudo noise code.

Air-Interface

CDMA was selected due to its interference tolerance as well as its security inherent in the modulation scheme. CDMA is able to provide good voice quality while operating at relatively low RF power levels. Path diversity is employed using rake receivers to receive and combine the signals from multiple sources. In the forward direction the use of diversity brings substantial gain if one of the satellites is obstructed. However, the reverse direction, because this is non-coherent diversity combining the gain is not as good.

Assignment of the code channels transmitted by a gateway. Out of the 128 code channels the forward channel consist of pilot channel, one sync channel, up to seven paging channels, and a number of Forward Traffic Channels. Multiple Forward channels are used in a Gateway by placing each Forward channel on a different frequency, namely the forward link pilot, sync and paging channel.

The pilot channel will generate an all zeros Walsh Code. This combined with the short code used to separate signals from different Gateways and different satellites. The pilot channel is modulo 2 added to 1.2288 Mc/s short code and is then QPSK spread across the 1.23 MHz CDMA bandwidth.

The Sync Channel is interleaved, spread and modulated spread spectrum signal. The sync channel will generate a 1200b/s data stream that includes time, gateway identification and assigned paging channel. This convolutionally encoded and block Interleaved to combat fast fading. The resulting 4800 symbols per second data stream is modulo two added to the sync Walsh code at 1.2288Mc/s which is then modulated using QPSK across the 1.23MHz CDMA bandwidth.

The paging Channel is used to transmit control information to the user terminal. The paging channel is convolutionally encoded at rate =1/2, constraint length K =9

and block interleaving. The resulting symbol rate is combined with the long code. The paging channel and long code are modulo two added, which is then modulo two added to the 1.2288Mc/s Walsh Code.

Modulation & Spreading

The spreading sequence structure for a CDMA channel comprised on an inner PN sequence pair and a single outer PN sequence. The inner PN sequence has a chip rate of 1.2288 Mcps and a length of 1024, while the outer PN sequence has an outer chip rate of 1200 outer chips per second and a length of 288. The outer PN sequence modulates the inner PN sequence to produce the actual spreading sequence lasting 240msec. Exactly one inner PN period is contained within a single outer PN chip.

Other parameters such as Link delay are important end-to-end parameters. The LEO satellites will provide a much more benign delay than the more common synchronous orbit satellites. Delay is held to 150ms in each direction. The vocoder uses a Code Exited Linear Prediction (CELP) algorithm which is similar to that used by the IS-96 coder.

Conclusion

This paper provides an insight and overview LEO systems and the application of this current area and the possible next generation systems. Underlining focus was on coding used in satellite applications and in particular to CDMA systems. The discussion has involved coding in all aspects of the satellite system, from user terminals to ground stations are discussed. The implementation of the various blocks are discussed. Finally the paper looks at future satellite systems and examines the coding requirements.

A Uni ed Code

Xian Liu[1], Patrick Farrell[2], and Colin Boyd[3]

[1] Communications Research Group, School of Engineering, University of Manchester, Manchester M13 9PL, UK
mbgeexl2@fs1.eng.man.ac.uk
[2] Communications Research Centre, Lancaster University, Lancaster LA1 4YR, UK
p.g.farrell@lancaster.ac.uk
[3] School of Data Communications, Queensland University of Technology, Brisbane Q4001, Australia
boyd@fit.qut.edu.au

Abstract. We have proposed a novel scheme based on arithmetic coding, an optimal data compression algorithm in the sense of shortest length coding. Our scheme can provide encryption, data compression, and error detection, all together in a one-pass operation. The key size used is 248 bits. The scheme can resist existing attacks on arithmetic coding encryption algorithms. A general approach to attacking this scheme on data secrecy is di cult. The statistical properties of the scheme are very good and the scheme is easily manageable in software. The compression ratio for this scheme is only 2 % worse than the original arithmetic coding algorithm. As to error detection capabilities, the scheme can detect almost all patterns of errors inserted from the channel, regardless of the error probabilities, and at the same time it can provide both encryption and data compression.

1 Introduction

Data compression, cryptologic algorithms, and error control coding are the central applications in information theory and are the key activities in a communication system. In fact, e ciency and reliability are the main concerns in a communication system. Data compression increases the e ciency by reducing the transmission and storing sizes without losing information signi cantly; cryptologic algorithms denies the unauthorised users trying to read or modify the messages being transmitted or stored; error control coding provides protection against channel errors. For error control, there are two basic strategies: forward error correcting (FEC) and automatic repeat request (ARQ). FEC works with an appropriate error correcting code and can, within the code's ability, automatically recover the inverted bits resulting from channel errors at the receiver end. ARQ applies a suitable error detecting code so that the decoder at the receiver end is within the code capability able to detect if the encoded le received has been damaged by channel errors and request the sender to retransmit the le. The essential fact in error control coding is that appropriate redundancy is introduced in the encoded le.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 84{93, 1999.

Arithmetic coding provides an e ective mechanism for removing redundancy in the encoding of data. It can achieve theoretical compression ratio bounds so it has gained widespread acceptance as an optimal data compression algorithm. The rst practical implementation for arithmetic coding was provided by Witten, Neal, and Cleary [1, 12] in 1987 (which is called the WNC implementation in this paper). Since then, many di erent implementations of arithmetic coding with di erent models have appeared. The authors of this paper have investigated the possibilities of providing cryptology and error control based on arithmetic coding and proposed a scheme providing both encryption and data compression [8], a scheme providing both error correction and data compression [10], and a scheme providing encryption, data integrity, and data compression, all together in a one-pass operation [9]. In this paper we will propose a uni ed code that can provide encryption, error detection, and data compression all together in a one-pass operation. The e ciencies in both encryption and data compression are the same as our previous schemes but also the scheme can detect almost all error patterns inserted from the channel, regardless of the error probabilities.

2 Arithmetic Coding

Arithmetic coding is based on the fact that the cumulative probability of a sequence of statistically independent source symbols equals the product of the source symbol probabilities. In arithmetic coding each symbol in the message is assigned a distinct subinterval of the unit interval of length equal to its probability. This is the encoding interval for that symbol. As encoding proceeds, a nest of subintervals is de ned. Each successive subinterval is de ned by reducing the previous subinterval in proportion to the current symbol's probability. When the message becomes longer, the subinterval needed to represent it becomes smaller, and the number of bits needed to indicate that subinterval grows. The more likely symbols reduce the subinterval by less than the unlikely symbols and thus add fewer bits to the message. This results in data compression. When all symbols have been encoded, the nal interval has length equal to the product of all the symbol probabilities and can be transmitted by sending any number belonging to the nal interval. That means if the probability of the occurrence of a message is p, arithmetic coding can encode that message in $-\log_2 p$ bits, which is optimal in the sense of the shortest length encoding. The pseudocode of arithmetic coding is as follows:

```
/* In the model, symbols are numbered 1, 2, 3, ... */
/* The cum_prob[ ] stores the                      */
/* cumulative probabilities of symbols with        */
/* cum_prob[i] increasing as i                     */
/* decreases and cum_prob[0]=1. The encoding       */
/* transmits any value in the final [low, high)    */
/* when it is finished.                            */
```

```
Encoding: The initial encoding interval [low, high) = [0, 1)
EncodeSymbol (symbol, cum_prob)
range = high - low;
high = low + range * cum_prob[symbol-1];
low = low + range * cum_prob[symbol];

Decoding:  The initial decoding interval [low, high) = [0, 1)
DecodeSymbol (cum_prob)
find symbol such that cum_prob[symbol]
      <= (value - low)/(high - low)
      < cum_prob[symbol - 1];
range = high - low;
high = low + range * cum_prob[symbol - 1];
low = low + range * cum_prob[symbol];
return symbol;
```

The WNC implementation for arithmetic coding [1, 12] was the first practical algorithm and is widely accepted. The algorithm is provided with either a static model or a first-order adaptive model. The algorithm realises integer arithmetic and incremental transmission. The arithmetic precision is 16-bit. In their first-order adaptive model, all the frequencies are initialised to 1. If the current model exceeds the maximum cumulative frequency, the model reduces all frequencies by half and recalculates cumulative frequencies. If necessary the model reorders the symbols to always put the current one in its correct rank in the frequency ordering. Adaptation is performed by incrementing the proper frequency count and adjusting cumulative frequencies accordingly. Due to the limit of the first-order adaptation, the compression ratio is 50% to 70% according to the size and type of the file. However it can be greatly improved by using a higher-order adaptive model.

3 Our Basic Scheme

Of course, the purpose of data compression is to reduce the redundancy in the message. On the other hand, redundancy contained in the output of a cryptosystem is usually one of the main resources to be used by the cryptanalyst. Also in an adaptive model, the current state in the model is related to the initial state and all of the messages that have been encoded so far since the model was initialised. Based on these facts Witten et al [11] suggested that an adaptive arithmetic coding algorithm may provide high level security. They also indicated that to use an adaptive modelling compression algorithm as an encryption algorithm it was enough to transmit the initial state in the model as a key over a secure channel.

3.1 Witten-Cleary Proposal

In 1988 [8] Witten and Cleary suggested two ways to insert the key into arithmetic coding:

Method 1: The initial model is used as the key in which an array of single-character frequencies in the range of 1-10 would do.

Method 2: A constant initial model is used and before transmission begins both the encoder and decoder assimilate a short secret message into the model.

Their further suggestion is that the adaptive links should be maintained over long periods of time; i.e. the nal model of encoding the current message will become the initial model to encode the next message.

Aiming at Method 1 in Witten-Cleary proposal with WNC adaptive implementation, Bergen and Hogan suggested a chosen plaintext attack on rst-order adaptive arithmetic coding in 1993 [2]. Instead of trying to recover the initial model the Bergen-Hogan attack tries to take control of the model and reduce it to a manageable form. If the encoder does not initialise its model, the attacker can decrypt any message transmitted after the attack is done. To be successful, in the Bergen-Hogan attack an associate as well as an attacker are necessary. The associate needs to send 2^{18} symbols and the attacker needs to try decoding the test string 2^{14} times. Up until now the Bergen-Hogan attack is the only feasible attack on the adaptive arithmetic coding encryption algorithm.

3.2 Our Basic Scheme

In [8, 9] we proposed our initial scheme to provide both encryption and data compression, and the scheme to provide encryption, data integrity, and data compression all together in a one-pass operation. In this section we will summarise some of the related results and the further results on the updated scheme providing both encryption and data compression. The key point is that we found without any exception that if the state in the adaptive model in the decoder is only slightly di erent from that in the encoder the decoder is unable to work at all and if the current interval in the decoder is only slightly di erent from that in the encoder the decoder is also unable to work at all.

The Scheme

1. Select an initial frequency count for every symbol randomly, which acts as the initial state in the model. The only restriction is that these numbers are all larger than 0.
2. Select the initial interval within the full range randomly, but the length of the initial interval should not be less than $(2^{16} - 1)/4$.
3. Select a secret 16-bit substitution with key size 16 bits, which is used to substitute for the rst 16-bit output of the encoding.
4. Choose two secret parameter pairs $(''^0_l; ''^0_h)$ and $(''^1_l; ''^1_h)$ which are used to shrink the current interval controlled by a random 64-bit string cyclically, where the four parameters are di erent.

3.3 The Key Size

Firstly, 96 bits can be used to indicate the initial state. There are 96 symbols with exact meaning in the extended ASCII set in text compression, so 96 bits

are enough to indicate the initial state. If the bit is 0 set the count of the symbol to 2, otherwise to 3. Set the counts of the remaining 160 symbols to be 1. The total number of di erent initial states in the model is 2^{96}. Secondly, 32 bits can be used to indicate the initial interval. The initial interval can be indicated by determining *low* and *high*, or *low* and *range*, so 16 bits are used to indicate one of them and 32 bits for both. The number of all of the valid initial intervals is 2^{30}. Thirdly, 40 bits can indicate the shrinking parameters. $''^{i}_{l}$ is chosen to be 4 decimal digits in the form of 0.0*** and $''^{i}_{h}$ is chosen to be 4 decimal digits in the form of 0.9***. The unknown part in every parameter ranges from 0 to 999, so 10 bits are necessary to indicate each of them. And then, we use 64 bits for the random control string. Finally, 16 bits are used for the 16-bit substitution key size. A secret 16-bit substitution with key size 16 bits is preferable. So the total key size for the scheme is 248 bits. It should also be pointed out that in our basic scheme, the second shrinking for the *low* is based on the new interval resulting from the rst shrinking for the *high*.

3.4 Communication Protocol

Protocol 1: The nal state in the model during encoding of the current message becomes the initial state of the model to encode the next message, and during the lifetime of using the same key the model will be initialised regularly. When encoding the current message is nished, shift the cyclic shift register storing the 64-bit random string one step and the next bit will become the rst control bit to control the rst shrinking in encoding the next message. Whenever nishing the encoding of the current message, all of the rest will be initialised.

This protocol has the advantage that the initial model to encode the next message is relevant to all of the messages which have been sent since initialisation.

Protocol 2: The only change compared with protocol 1 is that whenever encoding is nished, the model is re-initialised.

Protocol 2 on its own de nitely denies the Berger-Hogan attack because the attack is unable to nd the initial state in the model. Until now the Bergen-Hogan attack is the only powerful approach to attacking arithmetic coding encryption algorithms, so protocol 2 is preferable.

3.5 The Strength

In the Bergen-Hogan attack the attacker knows he matches the keying materials only when he successfully decodes the test string. To use the Bergen-Hogan attack on our proposal, the associate's strategy is the same as that to attack with the Witten-Cleary proposal, but the attacker's work will be increased dramatically. In order to decode the test string the attacker has to nd the rst symbol's frequency count in the standard form, the initial interval, the substitution, the pairs $(''^{0}_{l}; ''^{0}_{h})$ and $(''^{0}_{l}; ''^{0}_{h})$, and the 64-bit control string all together, instead of just trying the rst symbol's frequency count in the standard form 2^{14} times in breaking with the Witten-Cleary proposal. The attacker has to try to decode the test string 2^{14} 2^{30} 2^{16} 2^{19} 2^{19} $2^{64} = 2^{168}$ times. Partially nding

the keying materials is also very di cult. The reasonably simplest way for the attacker would be to rstly nd the rst symbol's frequency count in the standard form together with the initial interval. For this purpose the attacker only needs to decode the rst symbol in the test string, but he has to try decoding 2^{14} 2^{30} 2^{16} 2^{19} = 2^{79} times.

It has been shown in [8, 9] that compared with the WNC rst order adaptive implementation, the compression ratio of our scheme is only 2% worse and the running time is slightly less than double of that of the WNC implementation. Furthermore, the encoded les with our scheme have very good randomness. Changing any number of bits in the le to be encoded results in the fact that from the position in the encoded le that the rst changed bit corresponds to, then in the subsequent output, if this encoded le is compared with the encoded le resulting from the totally unchanged original le to be encoded, the changed bits and unchanged bits take the probabilities 0.5, and distribute uniformly and randomly. Furthermore, the outputs of our scheme and the les of the bitwise modulo 2 addition of the output of our scheme and the outputs of our scheme with the key being changed randomly, as well as the le of the bitwise modulo 2 addition of the output of our scheme with the le to be encoded in which one bit near the beginning was inverted and the output of our scheme with the unchanged le to be encoded have passed the frequency test, the binary derivative test, the change point test, the poker test, the runs test, the sequence complexity test, the linear complexity test, and Maurer's universal test (the statistical test software Crypt-X [5] is from the Information Security Centre at the Queensland University of Technology), and also there is no statistical di erence between the output from our modi ed scheme and that from the DES. So good plaintext di usion and ciphertext avalanche are achieved. Also, in the modi ed scheme, the keying materials have very good e ects on balance, di usion, completeness, and avalanche.

It has also been shown in [8, 9] that a general approach to attacking our scheme is di cult and our scheme can resist other related attacks [4, 6] to arithmetic coding encryption algorithms. In fact, if the model our scheme works with is a xed binary model with known symbol probabilities, and the initial substitution is ignored, and also the scheme works with theoretical arithmetic coding, nding the key is equivalent to solving two polynomial equations with 70 variables and degree 256. However, this analysis is based on a much simpli ed xed binary model and with ideal theoretical arithmetic coding. Practically, our scheme works with the Witten-Cleary rst order adaptive model with alphabet size 256. So far, there has not been any method to trace the evolution of the adaptive model. Also our scheme works with Witten-Cleary implementation for arithmetic coding. Quite a few practical strategies in the implementation make the coding procedure much more di cult to trace than in theoretical arithmetic coding. In fact, there are a number of main di erences between the scheme with theoretical arithmetic coding with a xed binary model and the scheme with WNC rst order adaptive arithmetic coding. Firstly, for the WNC rst adaptive arithmetic coding, there is no way to nd the exact current state in the

adaptive model. One may argue that as the uncertainty of the current state in the adaptive model, if a chosen plaintext attack is used and the secret shrinking parameters, the initial interval, and the secret control string as well are known, totally depends on the secret initial state in the model, after encoding a huge known le the e ect from the initial model is trivial; i.e., it can be converted from any known initial model. However, this is not true with WNC rst order adaptive model. Approximation does not make any sense unless the two current states are the same in arithmetic coding. Secondly, WNC adaptive arithmetic coding uses 16-bit nite precision. That means it has to expand the current interval after encoding (decoding) one symbol. Such expansions are unpredictable and untraceable with our scheme. Therefore, such a regular relation between the input and the output in the encoder de nitely does not hold in our scheme with WNC adaptive arithmetic coding. One thing clear is that if a mathematical relation exists in WNC adaptive arithmetic coding it must be much more complicated.

4 A Scheme Providing Encryption, Error Detection, and Data Compression All Together

In arithmetic coding, the decoding is successful only when the current interval in the decoding is identical to that in the encoding and the current state in decoder's model is the same as that in encoder's model. In case the current interval or the current state in the model in the decoder is not the same as that in the encoder, the whole subsequent encoded le would be unable to be recovered. That is, even a single bit error appearing in transmission would probability corrupt the remainder of the le. So the problem with the compressed data is that it is highly susceptible to transmission errors. The better the compression achieved, the more serious the e ect will be. In practice, error control techniques will have to be used to prevent transmission errors and to provide arithmetic coding with a completely noise free channel.

In fact, it is not di cult to introduce redundancy into arithmetic coding. An obvious way to introduce redundancy into arithmetic coding is to shrink the current interval in some way: after encoding a symbol or periodically. This method is equivalent to the method demonstrated as follows, because to add the same amount of redundancy periodically is the same as to encode an extra symbol with xed probability periodically. Besides the adaptive compression model, we use an additional xed model, in which there are only two symbols: the check symbol and the forbidden symbol. The check symbol is encoded periodically after one or several symbols from the adaptive compression model is encoded, and the forbidden symbol is never encoded. In decoding, the compression and check models are used alternately. If the forbidden symbol is decoded, an error has occurred. Redundancy can be controlled by varying the probability of the check symbol. Like convolutional codes, the redundancy is spread evenly through the message and errors may be detected soon after they occur. The error control performance will be related to how frequent the check symbol is encoded as well

as how much the current interval is shrunk or the probability of the check symbol arranged in the xed check model.

4.1 The Scheme

The third author of this paper proposed a scheme [3] which provides both error detection and data compression. The strategies he used are to shrink the current interval by a xed amount after encoding (decoding) every symbol, to add an extra nal check symbol after encoding (decoding) the EOF symbol to detect errors at the end of the le, and to exclusive-or every adjacent pair of output bits to be able to detect single isolated error bits. It was declared in [3] that the scheme could detect all single isolated error bits.

In order to provide error detection in our basic scheme as well, it is de nitely necessary to introduce some kind of redundancy into the encoding procedure. In our basic scheme we shrink the current interval twice after encoding (decoding) a symbol, which is one of our main steps to insert the secret key. It is our opinion that we have already added appropriate redundancy in the coding procedure even though it was not intended for error detection. We also think that it is not necessary to exclusive-or adjacent pair of output bits, but we do need to encode an extra nal check symbol at the end of each le.

Compared with our basic scheme we only need to encode an extra nal check symbol after encoding the EOF symbol for each le to achieve encryption, data compression, and error detection all together in a one-pass operation, with almost no extra price.

4.2 Performance

We have done exhaustive tests for the performance of the error detection abilities of our scheme. The decoder always succeeds in detecting the rst error bit and stops after a short delay. We have also found that the extra nal check symbol is very e ective in detecting any errors at the end of the le. The error detection performance of this scheme is independent of the error probabilities.

In the formal test, the le tested is book1.html from the Calgary Corpus with size 768771 bytes, which is a typical English technical report, and the compressed size is 451729 bytes; the compression ratio is 59%. Random errors are inserted to construct four groups of error probabilities: fi=1000000ji = $1; 2; : : : ; 999g$, fi=100000ji = $1; 2; : : : ; 999g$, fi=10000ji = $1; 2; : : : ; 999g$, and fi=1000ji = $1; 2; : : : ; 1000g$, which result in about 4000 di erent values of error probability ranging from 10^{-6}, step by step, to 100%. According to these error probabilities, random errors are inserted into the encoded le, resulting in about 4000 error corrupted encoded les. The decoder succeeds in detecting the rst error bit in all of the 4000 les after a short delay. The mean value of the delay is 92.88 bits and the standard deviation of the delay is 95.16 bits. That means, in the exhaustive test, the decoder always nds the rst error inserted after a delay of around 92.88 bits.

The experimental results from the exhaustive test allow us to predict that the scheme can detect most if not all error patterns inserted from the channel, independent of the error probabilities. This is in contrast to the fact that in the original arithmetic coding scheme the decoder cannot usually detect errors.

It is necessary to compare the performance of our scheme with the performances of dedicated standard error detection codes. Traditionally, the dedicated error detection codes are cyclic redundancy check codes (CRC), such as IEC TC57, IEEE WG77.1, ANSI, IBM-SDLC, and CCITT X.25, because cyclic codes are very e ective in detecting burst errors [7].

The CCITT X.25 CRC code has the generator polynomial:

$$G(x) = x^{16} + x^{15} + x^{12} + 1.$$

The minimum distance of this code is 4. In a block length up to 32768 bits it can detect: all triple or fewer random errors, all odd numbers of errors, all bursts of length up to 16, and 99% of all other longer bursts. However, there are many combinations of error patterns of even weight that the code cannot detect and that means it can only detect 50% of all of errors. So the error detection performance of our scheme is even better than that of the dedicated standard cyclic redundancy check codes. However, in addition, our scheme can provide both encryption and data compression.

5 Conclusions

In this paper we present a scheme that can provide data encryption, data compression and error detection all together in a one-pass operation. The scheme is based on WNC implementation for arithmetic coding in which a rst order adaptive model is used. The total key size is 248 bits. The statistical properties of our scheme are very good. Attacking this scheme is di cult. The compression ratio is about 2% worse than WNC implementation for arithmetic coding. The scheme can detect almost all patterns of errors inserted from the channel, independent of the error probabilities.

References

1. Bell T., Cleary J., and Witten I.: *Text compression*, Prentice Hall, 1990.
2. Bergen H. and Hogan J.: \A chosen plaintext attack on an adaptive arithmetic coding compression algorithm", Computers and Security, Vol.12, 1993, pp.157-167.
3. Boyd C., Clearr J., Irvine S., Rinsma-Melchert I., and Witten I.: \Integrating error detection into arithmetic coding", IEEE Trans. COM, Vol.45, No.1, 1997, pp.1-3.
4. Cleary J., Irvine S., and Rinsma-Melchert I.: \On the insecurity of arithmetic coding", Computers and Security, Vol.14, 1995, pp.167-180.
5. Crypt-X, Statistical Package Manual, Measuring the Strength of Stream and Block Ciphers. Information Security Research Centre, Queensland University of Technology, 1990.

6. Irvine S. and Cleary J.: \The subset sum problem and arithmetic coding", private communication, 1995.
7. Klove T. and Korzhik V.: *Error Detecting Codes, General theory and their application in feedback communication systems*, Kluwer Academic Publishers, 1995.
8. Liu X., Farrell P., and Boyd C.: \Resisting the Bergen-Hogan attack on adaptive arithmetic coding", LNCS-1355, Cryptography and Coding, Springer, December, 1997, pp.199-208.
9. Liu X., Farrell P., and Boyd C.: \Arithmetic coding and data integrity", Proceedings of WCC'99, pp.291-299, Paris, 11th-14th January, 1999.
10. Liu X. and Farrell P.: \Arithmetic coding with error correction", Proceedings of PREP'99, pp.330-333, Manchester, 5th-7th January, 1999.
11. Witten I. and Cleary J.: \On the privacy a orded by adaptive text compression", Computers and Security, Vol.7, 1988, pp.397-408.
12. Witten I., Neal R. and Cleary J.: \Arithmetic coding for data compression", Communications of the ACM, Vol.30, No.6, 1987, pp.520-540.

Enhanced Image Coding for Noisy Channels

Paul Chippendale, Cagri Tanriover, Bahram Honary

Department of Communication Systems
Lancaster University, Lancaster LA1 4YR, United Kingdom
{p.chippendale, c.tanriover, b.honary}@lancs.ac.uk
http://www.dcs.lancs.ac.uk

Abstract. This paper explores the application of a combined error resilient coding scheme to image transmission over time-varying noisy channels. To improve performance at low signal-to-noise ratios, turbo coding is incorporated into the system. Demonstrated through simulations, this novel combination of source and channel coding is shown to correct and restrict errors incurred during transmission over Additive White Gaussian Noise (AWGN) and Rayleigh Fading channels. The error correcting capability of the coding scheme also illustrated with compressed and uncompressed image transmission results which are comparable in terms of their visual quality.

APEL Coding

Absolute addressed Picture ELement coding (APEL) [1], [2] is a lossless, robust image coding system which translates variable sized pixel areas of pre-defined dimensions into independent picture blocks (pels). Each pel is issued with two co-ordinates, *x* and *y*, establishing an absolute location with respect to an origin.

As the APEL coding technique operates on a binary level, the encoding of *n*-bit grey-scale or colour images employs a Bit Plane Coding (BPC) [2] stage. The BPC stage furnishes the APEL encoder with a colour coding sequence to represent a given source image in *n* binary planes.

Taking each extrapolated binary plane in turn, a recognition algorithm searches through each image looking for square areas of black pixels; starting with large square pels during the first scan, then repeating this process in multiple passes selecting pels of decreasing magnitude. The maximum size of the initial pel is limited according to the anticipated nature of the channel, consequently less information is lost should corruption occur. Once all of the square pels of an efficient size have been removed from the plane, run-lengths of various geometries are used to encode the residue. Fig. 1 illustrates an APEL encoded section of an image. Here, it can be seen how (*x*,*y*) co-ordinates are assigned to pels of various geometries.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 94-103, 1999.

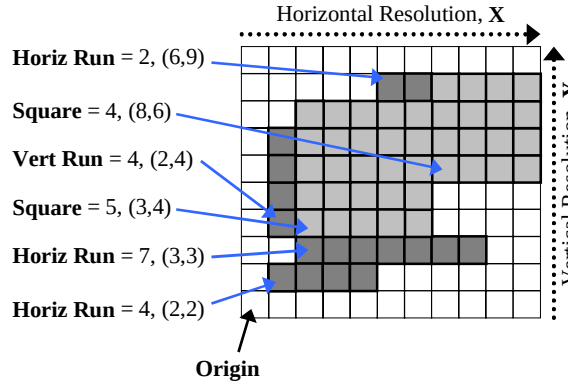


Fig. 1. APEL coded section of an image

The data-stream created from this process can be pictured as a succession of (*x*,*y*) addresses, grouped according to pel size and interspersed with control symbols (see Fig. 2). These symbols not only serve to provide synchronisation markers, but in addition convey pel geometry metrics to the decoder [2].

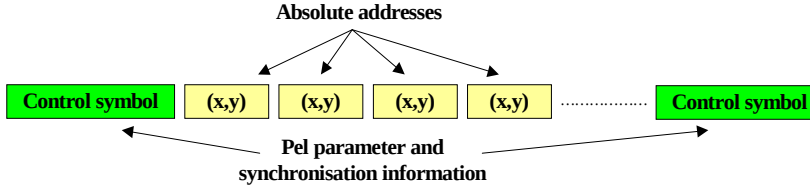


Fig. 2. Breakdown of APEL data stream

The APEL scheme alleviates the need for End Of Line (EOL) symbols and, as each codeword is independent, offers a solution to the problems of horizontal and vertical error propagation. Additionally, as each pel has its own address, it is possible to interleave them within the transmitted data-stream. This versatility can be utilised in many ways, for example: pels pertaining to important image detail can either be dispersed throughout the data-stream or transmitted at the start depending on channel conditions or operator preference.

Application of Turbo Coding to APEL

Turbo codes [3] are forward error correction schemes which employ concatenated component codes, interleaving and iterative decoding principles to achieve bit error rate performance close to the Shannon limit. Decoding is performed by the sub-optimal log-Maximum Aposteriori Probability (MAP) algorithm [4], which improves

the accuracy of the decoded information symbols through a set of iterations where soft extrinsic information is passed between the component decoders.

In this paper, a turbo encoder with parallel concatenation is incorporated into an APEL system. The turbo codec implemented is composed of two recursive systematic convolutional component codes, of rate ½ and constraint length 3. In general, the use of systematic convolutional codes provides robustness against decoding errors by decreasing the minimum free distance of the code. As a consequence of minimising the free distance, the error correction capability of the system improves. It is this feature of systematic convolutional codes that prevents catastrophic error propagation.

It should also be noted that the turbo decoder used in this coding scheme is designed to correct random errors only, hence the majority of burst errors encountered during transmission cannot be corrected.

Results

To demonstrate the benefits gained and also to provide benchmarks for comparison, in addition to incorporating turbo coding into an APEL system, we also concatenated the aforementioned turbo codec onto JPEG [5] and bitmap (BMP) file formats. Simulations over AWGN channel, at various signal-to-noise ratios, attest to the excellent performance of the APEL-turbo combination compared to the application of turbo coding onto JPEG and BMP file formats. The results presented in this paper were obtained from simulations conducted using an interleaver length of 8000 bits and performing 16 decoding iterations.

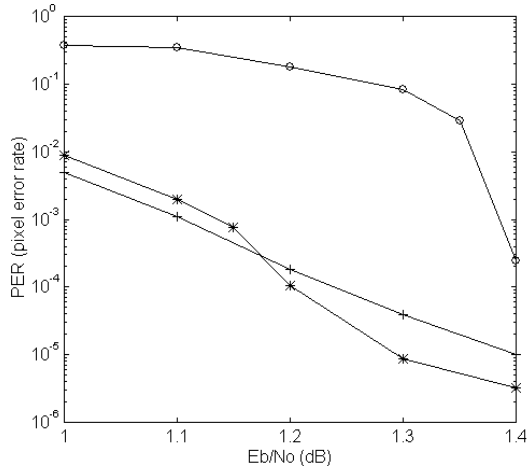


Fig. 3. Turbo coded image transmission over AWGN channel *Turbo coded JPEG* (*o*), *Turbo coded BMP* (+), *Turbo coded APEL* (*)

Visual effects of bit errors can be assessed in terms of pixels. Therefore, the error performance of the images was presented in a ratio called “Pixel Error Rate (PER)”, which indicates the degree of image degradation.

Through the analysis of the i[th] received pixel’s variance from its transmitted value, a measure of visual disturbance, Δ_i, can be quantified as in (1), where t_i and r_i represent the transmitted and received pixel colours respectively, for an n colour image.

$$\Delta_i = \frac{|r_i - t_i|}{n} \quad . \tag{1}$$

From (1) it follows that the PER is calculated as in (2),

$$PER = \frac{1}{XY} \sum_{i=0}^{XY} \Delta_i \quad . \tag{2}$$

where X and Y are the horizontal and the vertical resolution of the image, respectively.

As Fig. 3 shows, the performance of the Turbo-JPEG scheme is very poor in the 1.0 – 1.4 dB range. This is due to the inherent fragility of the JPEG structure and its inability to correct or restrict the propagation of any errors.

As expected, the performance of the BMP-turbo scheme is good throughout the range. This results from the complete independence of all pixels from one another. Hence, when errors cannot be repaired by the turbo decoder, only pixels with corrupted bits are affected.

Finally, as Fig. 3 clearly indicates, performance close to, and, as the channel improves, surpassing that of the BMP-turbo model is achieved by the turbo coded APEL. In the region after 1.175 dB, the post-processing techniques employed by APEL [2] recover many of the damaged pixels which could not be corrected in the case of BMP.

To observe the visual impact of data errors, samples of the various file formats have been decoded at a signal-to-noise ratio of 1.175 dB (Fig. 5-Fig. 7). To provide a qualitative reference for comparison, an uncoded version of the BMP file transmitted over the same channel has been included (Fig. 4).

In this example, although the PER is the same for images in Fig. 6 and 7, the subjective quality of the latter is slightly better. This results from the less frequent and clustered nature of the pixel errors in the APEL image, and the effects can be seen in more detail in the magnified areas in these figures.

Since the APEL image coding scheme is lossless, the compression level relating to the JPEG format was reduced to provide a fair comparison, although even at this minimal level of compression the JPEG image was found to be greatly modified pixel-wise from that of the source. The attained compression ratio for these two formats, APEL and JPEG, was nevertheless still around 3 to 1, offering a substantial reduction over uncompressed BMP data.

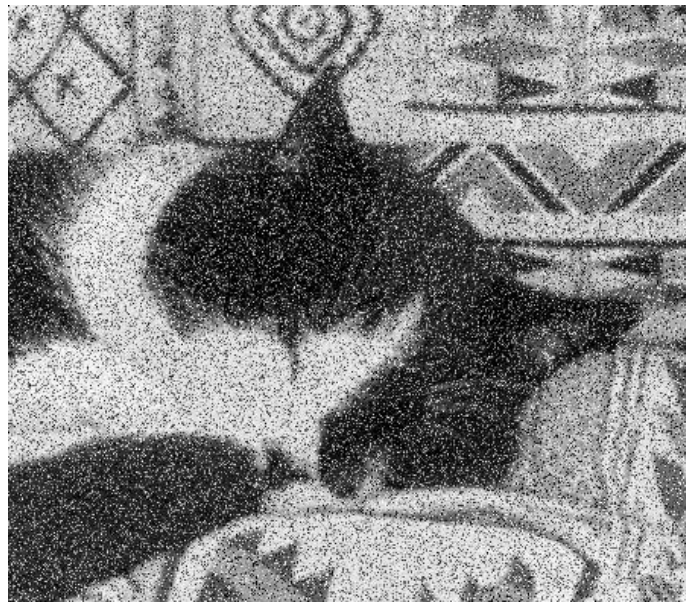


Fig. 4. Uncoded BMP image transmitted through AWGN channel at 1.175 dB

Fig. 5. Turbo coded JPEG image transmitted through AWGN channel at 1.175 dB

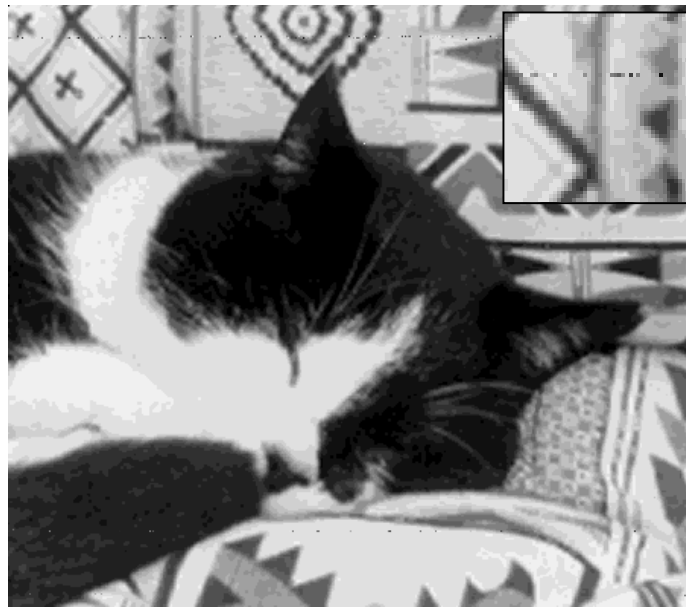


Fig. 6. Turbo coded BMP image transmitted through AWGN channel at 1.175 dB

Fig. 7. Turbo coded APEL image transmitted through AWGN channel at 1.175 dB

Turbo coded JPEG, bitmap and APEL images were also transmitted over a Rayleigh channel with a maximum of 200 burst errors introduced randomly in each interleaver block. Figures 9 through 11 illustrate the system performance in the presence of burst errors. The uncoded BMP image has also been included to provide an insight into channel conditions (Fig. 8). Unlike the Gaussian channel, errors in the more severe Rayleigh case made for unreliable and inconsistent PER plots. It was observed that the dynamic range of the decoded image quality was wide in this case.

Due to the severe effects of burst errors, Turbo-JPEG fails to maintain data integrity and synchronisation after decoding (Fig. 9). The fragile structure of Huffman coding stage makes it almost impossible to withstand such channel conditions. In addition, since the turbo decoder is unable to correct burst errors, image transmission with Turbo-JPEG becomes very unreliable.

As illustrated in Fig. 8, channel errors are introduced as both randomly and in bursts. The Turbo-BMP scheme (Fig. 10) was observed to eliminate the majority of random errors effectively, however the majority of burst errors remained uncorrected. In other words, this scheme behaved similar to a 'burst-pass filter', where erroneous pixels appeared as trails of various lengths after decoding.

Turbo-APEL (Fig. 11) performance in the presence of burst errors, is visually comparable to that of Turbo-BMP (recall that the APEL image has 3:1 compression!). Channel errors which affect pels from various bit planes can be corrected through an analysis of the other planes. In other words, the post-processing techniques introduced by APEL coding, provide a powerful means of interpolating pixels using valid image information. Hence, the output of the 'burst-pass filter' can be further processed to correct more pixel errors than in the other cases. However, as the number of burst errors per information block is increased, distortion in APEL images, as anticipated, remain noticeable despite the post-processing.

Secondly, the interleaving stage in APEL coding, distributes pixel errors across the entire image (Fig. 11). Visually, small clustered errors are less disturbing to the eye than erroneous pixel trails (Fig. 10).

Conclusions

We have proposed the combination of APEL and turbo coding in order to produce an enhanced image transmission system for low signal-to-noise ratios. Moreover, images in Fig. 6 and 10 require more bits to encode and transmit than images in Fig. 7 and Fig. 11; further underlining the advantages of the APEL turbo scheme outlined here.

Even though APEL coding is used with a Turbo decoder that is not powerful enough to correct burst errors, the second interleaving stage in APEL is seen to minimise the visual impact of errors. This visual improvement is achieved by the spreading of burst errors across different bit planes, which provides an interleaver gain at the decoder.

Fig. 8. Uncoded BMP image transmitted through Rayleigh channel at 4.0 dB

Fig. 9. Turbo coded JPEG image transmitted through Rayleigh channel at 4.0 dB

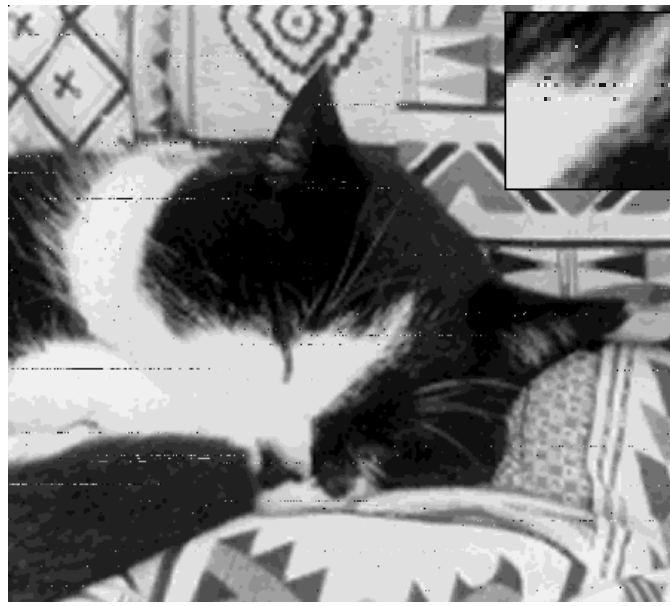


Fig. 10. Turbo coded BMP image transmitted through Rayleigh channel at 4.0 dB

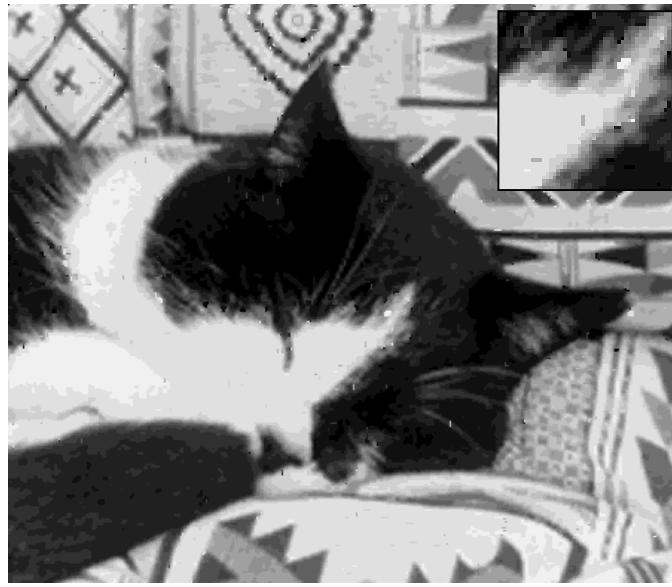


Fig. 11. Turbo coded APEL image transmitted through Rayleigh channel at 4.0 dB

In addition, within the APEL image, whilst the majority of bit errors are corrected via iterative decoding, any which evaded detection (and thus perhaps falsely inserted as erroneous pixels) are restricted as a result of the robust data structure.

The novel combination of source/channel image coding technique, in this case APEL, with additional channel protection provides not only a resilience to Gaussian type errors, but also offers a powerful tool for the restriction and correction of burst errors.

To conclude, this approach can be further explored to develop integrated coding techniques, which could provide more reliable image communication means for noisy channels.

Acknowledgements

The authors wish to thank DERA Malvern and NDS Ltd for their financial and technical support.

References

1. Chippendale, P., Honary, B., Arthur, P. and Maundrell, M.: International Patent Ref.: PCT GB 98/01877, 'Data Encoding System', 1999
2. Chippendale, P.: 'Transmission of images over time-varying channels', PhD Thesis, August 1998
3. Berrou, C., Glavieux, A., Thitimajshima, P.: 'Near Shannon Limit Error Correcting Coding and Decoding: Turbo-Codes', IEEE Proc. ICC '93 Geneva, Switzerland, May 1993, pp. 1064-1070
4. Hagenauer, J., Offer, E., Papke, L.: 'Iterative Decoding of Binary Block and Convolutional Codes', IEEE Transactions on Information Theory, Vol. 42, No. 2, March 1996
5. International Organisation for Standardisation.: 'JPEG Digital Compression and Coding of Continuous-Tone Still Images'. Draft ISO 10918, 1991
6. NewScientist: 'The sky's the limit', No.2193, 03.07.1999, pp. 6

Perfectly Secure Authorization and Passive Identi cation for an Error Tolerant Biometric System

George I. Davida[1] and Yair Frankel[2]

[1] Center for Cryptography, Computer, and Network Security,
University of Wisconsin-Milwaukee, USA.
davida@cs.uwm.edu
[2] CertCo Inc., New York, NY, USA.
yfrankel@cs.columbia.edu, yfrankel@cryptographers.com

Abstract. A biometric identi cation system was recently developed and analyzed as a secure mechanism for user authentication. The system provided for the con dentiality, without the use of cryptographic encryption, of the user's biometric information stored in public veri cation templates.

Here we demonstrate that the use of majority decoding can enhance the prior techniques in several ways. One enhancement allows the biometric authentication system to leak no information about a user's biometric when using the proper computational assumptions. Another enhancement is a passive identi cation system.

1 Introduction

An Iris scan is a biometric technology which uses the human iris to authenticate users [BAW96, HMW90, Dau92, Wil96]. This technology produces a 2048 bit user biometric template such that any future scan of the same user's iris will generate a \similar" template. By similar, we mean having an acceptable Hamming distance within a prede ned range, usually up to ten percent of the size of the code (e.g., Hamming distance between original reading and future reading is set to be in the range from 20 to 200). Moreover, the Hamming distance for the biometric readings of two di erent users has been shown to be much higher, about 45 percent (or 921 bits).

One can think of a biometric reading of a user as a faulty communication channel which may introduce a limited number of errors. Informally the typical biometric system works in the following manner. A user's biometric template is registered. A future reader compares the newly generated template with the registered template to test for closeness. With respect to iris scan technology closeness is measured by the Hamming Distance.

In the work [DFM98], the feasibility of protecting the privacy of a user's biometric and other security features were studied. It was suggested that providing additional privacy for the user's biometric may provide for stronger user acceptance. (For instance, an iris template may be used to determine some medical

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 104{113, 1999.

conditions by an insurance company instead of the legitimate identi cation process the user submitted to.) The objective in [DFM98] was to allow protection of a user's biometric information in unprotected devices (such as a magnetic strip) or in a publicly accessible database (such as in a public key certi cate). To address scalability concerns, private keys by the user or the reader was not used. Also, encryption is prone to loss of the cryptographic keys from the reader (i.e., the loss of a single key compromises every user).

Providing for authorization bound to a biometric template appears to be inherently di cult in this model, because the user's biometric template cannot exist in the clear on the storage device. To eliminate the need for storage of the biometric (in the clear or encrypted form), a new veri cation algorithm had to be developed.

The primary tools of the work of [DFM98] were error correcting codes (including majority decoding) and cryptographic methods. Later, the e ectiveness of majority decoding was further analyzed by [DFMP]. Due to the e ectiveness of majority decoding we are able to even further improve the error correcting codes (ECC)/Crypto based biometric system.

1.1 The Result

We enhance the system of [DFM98] in two ways:

- Though a biometric reading contains signi cant errors with respect to the original reading, we show how to use a biometric scan as an index in order to provide a scalable passive user identi cation system. By passive identi cation system we mean it can uniquely identify a user from a set of registered users through the use of only a biometric scan and no other input.
- We develop a system with perfect information preservation. The scheme in [DFM98] leaks approximately as many bits, in an information theoretical sense, as there are redundancy (error correcting) bits to correct errors in a biometric template. However, there was no way to quantify what kinds of bits are leaked and under what assumptions. Here we demonstrate that in a computational model, majority decoding can be used in a manner that no information is leaked with su ciently high probability.

2 Background

2.1 Error Correcting Codes

Majority decoding: In the rest of the paper, we will consider only binary error correcting codes. We will denote by $a||b$ the concatenation of two strings a,b.

Let $v_i = \langle v_{i;1}; v_{i;2}; \ldots; v_{i;n} \rangle$ be n bit code vectors. Given odd M vectors v_i, a majority decoder computes vector $V = \langle V_1; V_2 \ldots; V_n \rangle$, where

$$V_j = \langle \text{majority}(v_{1;j}; \ldots; v_{M;j}) \rangle;$$

i.e., V_j is the majority of 0's or 1's of bit j from each of the M vectors. We shall use majority decoding primarily to get the best biometric reading possible, thus reducing the Hamming distance between successive $final$ readings V.

In the biometric authentication protocol, described in Section 2.2 the biometric being measured will be estimated by sampling since the actual unique iris is not measured with precision. The samples that are taken of the iris will converge to the actual unique individual biometric, with majority decoding, with high probability.

Error correction: An $[n; k; d]$ code [Ber68, MS78, PW88] is a code of n bit codewords (vectors) where k is the number of information digits and d is the minimum distance of code. Such a code can correct at least $t = (d - 1)/2$ errors.

Note: Bounded distance decoding: In the rest of the paper, we assume that the decoding performed at the point of veri cation is to correct at most $(d - 1)/2$ errors. This is necessary to ensure that no bogus biometric is decoded into a valid one. Bounded distance decoding can be readily implemented through a simple count of the Hamming weight of the error vector computed. In some decoding schemes, the error locations that are computed are the roots of some polynomial $\sigma(z)$ over $GF(2^m)$ of degree $t' = degree(\sigma(z))$. If $t' > t = (d - 1)/2$ then the biometric is rejected.

2.2 An Error Correcting Based Biometric System

The primary observation of [DFM98] is that a user's biometric template can be viewed as the information bits of an error correcting code. Now instead of storing the biometric template only the error correction bits are necessary on the storage device, a magnetic strip card for simplicity[1]. Since only the check bits are stored on the user's card, the available information about the biometric template is reduced. On the other hand, the reader can take a new reading of the user's biometric template, append the ECC check bits, remove the errors using bounded distance decoding, and nally, with high probability, reproduce the original template, which can be veri ed with the signature on the token.

One other hurdle has to be overcome to provide security. The signature may itself leak the user's template. Observe that $\langle M; \mathrm{SIG}(M)\rangle$ is a signature for message M which leaks all bits of M, yet is a valid signature of M. To resolve this problem, special hash functions were used in [DFM98].

Here is a brief summary of the basic o -line biometric protocol presented in [DFM98].

System Setup: The authorization center generates its public and private keys and disseminates its public key to the biometric readers. The system also sets up an $[n; k; d]$ code.

User Initialization: To register, M biometric templates of length k are independently generated for the legitimate user. Majority decoding is then applied to

[1] A smartcard, a database record, a public certi cate can be stored as well.

the M biometrics to obtain the user's k bit template T. Given the k information digits T, an n digit codeword $T||C$ is constructed, where C are the check digits, in the $[n; k; d]$ code defined at system setup. A storage device is constructed with the following information:

1. Name of the individual, NAME.
2. Other public attributes ATT, such as the issuing center and a user's access control list.
3. The check digits C, of the biometric.
4. Sig(Hash(NAME; ATT; $T||C$)) where Sig(x) denotes the authorization officer's signature of x, and Hash( ) is a partial information hiding hash function [Can97] (e.g., Sig(Hash( )) is a content-hiding signature) or a random oracle (See [BR93]).

Biometric verification process: When a user presents herself/himself and the card with the information described above, M biometric templates are independently generated for the user. Majority decoding is applied to the M biometric vectors to obtain the user's k bit template T'. Error correction is performed on codeword $T'||C$ to obtain the corrected biometric T''. The signature Sig(Hash(NAME; ATT; $T''||C$)) is then verified. Successful signature verification implies the user passed the identification step. For simplicity of exposition, we assume that occasional rejection of a valid user is acceptable (the user would simply repeat the scan). In applications where rejection of a valid user is not acceptable, the parameters of the system can be changed so that such an event has negligible probability. Determining the correct parameters in such a case involves bounding the area under the tail of a binomial distribution (or a Normal approximation to the binomial distribution via the Central Limit Theorem).

Proof of security and in particular the choice of hash functions were discussed in [DFM98]. Moreover, the usefulness of majority decoding to detect impostors was discussed in [DFMP]

3 New Techniques

We now discuss the two new techniques: perfect confidentiality and passive identification. The new techniques are based on the following observation: Given a 2048 bit iris code, majority decoding is used on a sufficient number of samples to reduce the expected number of errors to a small number, e.g. 1 per block of 2048 bits.

No. of scans	Per bit prob. of error	Expected no. of errors in a 2Kb scan
1	0.1	205
3	0.028	58
11	0.000306	1
21	0.00000135	.002

Once a reduced-error iris code is obtained using majority decoding, we construct D indices $I_j; 1 \quad j \quad D$ from I-SIZE subsets of T with the GEN-Index function as follows:

Step 1 Set $j = 0$

Step 2 Set $j = j + 1$

let $X_j = PermutedChoice(j; T)$ be I-SIZE bits of the biometric T selected with schedule PC, where I-SIZE is chosen so that the entropy of the X_j bits is su cient(e.g. For an iris scan as described above if I-SIZE=600 bits, then, assuming that the biometric has an entropy of 160, the entropy of X_j, on average, satis es $H(X_j) = 53$).

let $I_j = \text{hash}(X_j)$

Step 3 if $(j < D)$ goto 2 else exit.

These indices I_j are pointers to the database locations where the user templates are stored. Collisions with other iris codes is dealt with by performing the checks to be described later.

Two important observations can be made. First, with high probability at least one index is error free (See Majority Decoding in section 2.1). Second when hash( ) has same the information hiding property as those used in [DFM98] (see [Can97] as an example) and X_j has su cient entropy, the I_j leak no useful information about the iris.

3.1 Passive Identi cation

In a passive identi cation system the user is uniquely identi ed with only the biometric reading and without any other inputs from the user. Hence the user does not provide an ID number or other inputs via a keyboard or a smart card. Once the user's biometric is read, the user must be uniquely identi ed to obtain a user id. This may be done by a linear search through a database of registered biometric/user attributes relationship database. However, in practice a linear search is not scalable for applications with a large user base.

In a biometric system, such as iris scan, there exists variances from the original registered reading with a later acquired reading. Because of the variances, it is not possible, in general, to use biometric systems as a scalable passive identi - cation systems. Scalability becomes di cult because if the reading is faulty and lacking any other input from the user due to the passive nature of the identi cation scheme, the biometric can no longer be an index into a registered template database and therefore only linear searches are generally possible.

As discussed above, we note that using majority decoding with iris scan technology one is able to reduce the number of errors to a negligible amount. This is based on observations that the errors in successive readings of a biometric di er in positions that are randomly distributed over the iris code, with about 10 percent Hamming distance between success readings, on the average. Assuming

that the errors are random over the code: they can be reduced through majority decoding of M independently read iris code vectors.

Let T be the template for an individual who presents to the authorization center. For each such user, we construct D indices $I_j; 1 \quad j \quad D$, of size I-SIZE as described above, which are pointers to the location of the record. Standard hashing techniques can be used to produce the indices.

We now de ne the following identi cation system:

To register, M biometric templates of length k are independently generated for the legitimate user. Majority decoding is then applied to the M biometrics to obtain the user's k bit template T. Given the k information digits T, an n digit codeword $TjjC$ is constructed, where C are the check digits, in the $[n; k; d]$ code de ned at system setup. In the secure database the following information is stored:

1. Name of the individual, NAME.
2. Other public attributes ATT, such as the issuing center and a user's access control list.
3. The check digits C, of the biometric.
4. Hash(NAME; ATT; $TjjC$) where Hash( ) is a partial information hiding hash function [Can97][2].

The database is set up so that the indices $I_1; : : : ; I_D$ created from T with the GEN-Index function link to the created record.

Passive identi cation process: During veri cation, when a user presents herself/himself, the veri cation unit performs the following steps

Step 1 Set $i = 0$, M biometric templates are independently generated for the user. Majority decoding is applied to the M biometric vectors to obtain the user's k bit template T^0.

Step 2 Set $i = i + 1$, Construct index I_i^0 with the GEN-Index function on input T^0.

Step 3 The records pointed to by indices I_i^0, containing the check digits and hash value, are requested. Let C_i be the check digit in record indexed by I_i^0. Each set of check digits C_i is then used along with T^0 to produce a new corrected biometric T_i^{00}.

Step 4 The hash value Hash(NAME; ATT; $T_i^{00}jjC$) is then compared for equality with the hash value received.

Step 5 If success, exit (success)

Step 6 If $i < D$ go to 2 else exit(failure)

Successful veri cation implies the user passed the identi cation step. The NAME and the ATT elds identify the user uniquely. Observe that with overwhelming probability at least one of the indices will be correct. In fact, there

[2] A signature, private key authentication, hash, etc. can be used as well depending on the security model.

will most likely be multiple indices pointing to the same record. To reduce the number of queries into the database those records pointed by the most indices should be tested first.

3.2 Perfect Secrecy

User acceptance is vital for any biometric system to be effective. However, most systems reveal information about the user in the registration template. Systems based on the iris measurements may be particularly sensitive to revealing health information in the template.

In [DFM98, DFMP], the protocol presented leaks only as much information as the error checking bits included in the template. However, this is an information theoretical analysis and it does not say anything about information leakage in a computational sense.

What we desire is an identification system that achieves perfect secrecy, without storing the biometric. Informally, perfect secrecy means that an polynomial time adversary given a registration template is unable to compute any information about the user biometric related to the template.

Here we use a technique very similar to the passive identification. Define $PRF(:;:)$ to be a Pseudo Random Function with two inputs. We obtain D indices as before but this time we store on the user token D tuples $\hbar R_j; C_j i$, for $1 \quad j \quad D$, where $C_j = PRF_{I_j}(R_j) \quad C$ and R_j is a random string. This in essence encrypts C under each of the keys I_j.

System Setup: The authorization center generates its public and private keys and disseminates its public key to the biometric readers. The system also sets up an $[n; k; d]$ code.

User Initialization: To register, M biometric templates of length k are independently generated for the legitimate user. Majority decoding is then applied to the M biometrics to obtain the user's k bit template T. Given the k information digits T, an n digit codeword $TjjC$ is constructed, where C are the check digits, in the $[n; k; d]$ code defined at system setup. Let $I_1; :::; I_D$ be the D indices chosen as described above. A record (stored on a token to be carried by the user) is constructed with the following information:

1. Name of the individual, NAME.
2. Other public attributes ATT, such as the issuing center and a user's access control list.
3. $\hbar R_j; C_j i$, $1 \quad j \quad D$, where $C_j = \mathrm{PRF}_{I_j}(R_j) \quad C$, I_j are D indices of size I-SIZE and R_j is a random string.
4. $\mathrm{Sig}_j = Sig(\mathrm{Hash}(\mathrm{NAME}; \mathrm{ATT}; TjjC_j))$, $1 \quad j \quad D$, where $\mathrm{Sig}(x)$ denotes the authorization officer's signature of x, and Hash( ) is a partial information hiding hash function [Can97] (e.g., Sig(Hash( )) is a content-hiding signature) or a random oracle (See [BR93]).

Biometric verification process: When a user presents herself/himself and the card with the information described above, the following steps are performed

Step 1 set $j = 0$
M biometric templates are independently generated for the user. Majority decoding is applied to the M biometric vectors to obtain the user's k bit template T^0.

Step 2 $j = j + 1$
Compute I_j^0 with the GEN-Index function on input T^0.
Compute $C_j^0 = \mathrm{PRF}_{I_j'}(R_j) \quad C_j$.
Apply error correction on codeword $T^0 jj C_j{}^0$ to obtain the corrected biometric T_j^{00}.

Step 3 The signature[2] $\mathrm{Sig}_j = Sig(\mathrm{Hash}(\mathrm{NAME}; \mathrm{ATT}; T_j^{00} jj C_j))$ is then checked.
A successful signature veri cation implies the user passed the identification step.
exit(success)

Step 4 If $i < D$ go to 2 else exit(failure)

Informally, the reasons this scheme attains perfect secrecy are: Observe that $hC_1; R_1 i; : : : ; hC_D; R_D i$ are multiple encryptions each of C with a key (index) with su cient entropy. That is each key has around 53 bits entropy, as discussed above, but more can be added. Now each of the keys (indices) I_j operates on a random R_j to provide independence amongst the tuples. If a random oracle rather than pseudo-random function is used then the random values R_j are not necessarily needed.

3.3 Passive Identi cation with Untrusted Veri er

In the passive identi cation protocol above the reader performed the nal veri cation process. That is it veri ed the signature. If it is desired that this veri cation step be performed by the central database holder, without leaking information about the user's biometric, then using a random oracle model we can solve this problem by combining the presented techniques.

As in the passive identi cation, indices are generated and a user's information is stored in a manner which allows the indices to point to the appropriate data. However, this time the user information is di erent:

System Setup: The authorization center generates its public and private keys and disseminates its public key to the biometric readers. The system also sets up an $[n; k; d]$ code.

User Initialization: To register, M biometric templates of length k are independently generated for the legitimate user. Majority decoding is then applied to the M biometrics to obtain the user's k bit template T. As in the GEN-Index function, let $X_j = PermutedChoice(j; T)$ be the I-SIZE random bits of the vector T. Now, for RO( ), a random oracle (see [BR93]), let $T_j = T \quad \mathrm{RO}(\backslash 0" jj X_j)$

[2] Hashes, private key authentication etc. can be used instead depending on the security model.

and I_j = RO(\1"jjX_j). Given the k information digits T_j, an n digit codeword T_jjjC_j is constructed, where C_j are the check digits, in the $[n;k;d]$ code de ned during setup. In the database we store at a location pointed to by indices I_j:

1. Name of the individual, NAME.
2. Other public attributes ATT, such as the issuing center and a user's access control list.
3. The check digits of the encrypted biometric: C_j, $1 \quad j \quad D$.
4. D hashes Hash(NAME; ATT; $TjjC_j$), $1 \quad j \quad D$, where Hash( ) is a partial information hiding hash function [Can97][3].

Biometric veri cation process: When a user presents herself/himself, M biometric templates are independently generated for the user. Majority decoding is applied to the M biometric vectors to obtain the user's k bit template T^0. As in the GEN-Index function, let X_j^0 be the I-SIZE bits of T' selected using schedule PC, as described above. The reader sends to the database server tuples $hI_j^0; T_j^0 i$ where $T_j^0 = T^0 \quad$ RO(\0"jjX_j^0) and I_j^0 = RO(\1"jjX_j^0). The server nds the user's records from the I_j^0. Error correction is performed, for each i, on codeword $T_i^0jjC_i$ to obtain the corrected biometric T_i^{00} by the database server. The hashes Hash(NAME; ATT; $T_i^{00}jjC_i$) are then checked. Successful veri cation implies the user passed the identi cation step. For simplicity of exposition, we assume that occasional rejection of a valid user is acceptable (the user would simply repeat the scan). In applications where rejection of a valid user is not acceptable, the parameters of the system can be changed so that such an event has negligible probability. The reader is then informed of the success or failure of the veri cation by the central server.

Observe that the C_j leak no information because all possible T_j are equally likely given that RO( ) is a random oracle. For correctness, observe that for valid user u with subsequent reading T^0 has an error vector $E = T \quad T^0$. Suppose X_j is the \index" without any errors. Then performing error correction on $T^0jjC_j = T \quad E \quad$ RO(\0"jjX_j)jjC_j returns $T_j^{00}jjC_j$ because E has low Hamming weight. Also note that we use two di erent random oracles RO(\1";), for the indices, and RO(\0";), for the keys to encrypt a users template. This allows us to use the same bits of the template in two ways without leaking the key (i.e., RO(\0"jjX_j)) for a key and index RO(\1"jjX_j)).

Computationally Simple Passive Identi cation: Using the same idea as described above a computationally simpler and heuristicly secure mechanism can be constructed. At the setup process vectors $T_j = T \quad$ RO(\0"jjX_j) and I_j = RO(\1"jjX_j), $1 \quad j \quad D$, are stored. In the biometric veri cation process T^0 is obtained as before. Vectors $T_j^0 = T' \quad$ RO(\0"jjX_j^0) and I_j^0 = RO(\1"jjX_j)^0, where X_j^0 is created from T^0 as before, using a Permuted Choice schedule, are now created. Acceptance occurs when there exists a T_j^0 whose Hamming weight is su ciently close to the retrieved vector T_j, retrieved with I_j^0. Observe in all the perfect security schemes no additional information is leaked if di erent

[3] A signature can be used as well.

authorization centers uses di erent parameters (e.g., PermutedChoice, $[n; d; k]$ code, random oracle, etc.).

In the non-passive case when the user is allowed to provide some information to the reader (e.g., a magnetic strip card containing error correction bits for the user's template), to provide for an untrusted veri er then as in [DFM98] a hashed biometric template regenerated by the reader can then be used as a key. This key can be used as an authentication key for a challenge response where challenge is generated by the untrusted veri er. That is, let $K = \mathrm{RO}(T)$ be stored by the untrusted veri er. Veri cation is response $f_K(C)$ where C is a challenge from untrusted veri er or generated by a random oracle, at the reader, with some one-time tag (i.e., using inputs such as time, date, random values, names, etc.).

References

[BAW96] F. Bouchier, J. S. Ahrens, and G. Wells. Laboratory evaluation of the iriscan prototype biometric identi er. Technical Report SAND96-1033, Sandia National Laboratories USA, April 1996.

[Ber68] E. R. Berlekamp. *Algebraic Coding Theory*. McGraw-Hill, 1968.

[BR93] M. Bellare and R. Rogaway. Random oracles are practical: a paradigm for designing e cient protocols. In *Proceedings of the 1st ACM conference on Computers and Communications Security*, 1993.

[Can97] R. Canetti. Towards realizing random oracles: Hash functions which hide all partial information. In *Advances in Cryptology. Proc. of Crypto'97*, pages 455{469, 1997.

[Dau92] J. Daugman. High con dence personal identi cations by rapid video analysis of iris texture. In *IEEE International Carnahan Conference on Security Technology*, pages 50{60, 1992.

[Dau93] J. Daugman. High con dence personal identi cations by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):648{656, November 1993.

[DFM98] G. I. Davida, Y. Frankel, and B. J. Matt. On enabling secure applications through o -line biometric identi cation. In *1998 IEEE Symposium on Security and Privacy*, pages 148{157, 1998.

[DFMP] G. I. Davida, Y. Frankel, B. Matt and R. Peralta, ' On the relation of error correction and cryptography to an o ine biometric based identi cation scheme. In *Proceedings of the Workshop on Codes and Cryptography 1999*.

[HMW90] J. P. Holmes, R. L. Maxell, and L. J. Wright. A performance evaluation of biometric identi cation devices. Technical report, Sandia National Laboratories, July 1990.

[MS78] F. J. MacWilliams and N. J. A. Sloane. *The theory of error{correcting codes*. North { Holland Publishing Company, 1978.

[PW88] W. W. Peterson and E. J. Weldon. *Error Correcting Codes*. The MIT Press, 1988.

[Wil96] G. O. Williams. Iris recognition technology. In *IEEE International Carnahan Conference on Security Technology*, pages 46{59, 1996.

An Encoding Scheme for Dual Level Access to Broadcasting Networks

Thumrongrat Amornraksa, David R.B. Burgess, and Peter Sweeney

CCSR, University of Surrey, Guildford, GU2 5XH, U.K.
t.amornraksa, d.burgess, p.sweeney@ee.surrey.ac.uk

Abstract. In this paper, we propose an encoding scheme which gives two levels of access to a broadcast encrypted signal. Watermarking-type techniques based on direct-sequence spread spectrum communications are implemented to add specific information to the signal within the bandwidth allocated for broadcasting. This is beneficial to both the service providers and all subscribers in the network since the information added can advertise programmes which many are not yet authorised to access.

1 Introduction

An advantage of communications over the broadcasting network is that the transmitted signal from a source station can be received simultaneously by many destination stations. Digital TV broadcasting is one of the applications that uses this advantage. Since some digital TV programmes are pay-TV services, they will be encrypted before transmitting to every subscriber. Only the authorised subscribers who pay an extra fee can get access to those programmes. This technique does not give any value at all to other subscribers who have not paid for that particular programme. The allocated bandwidth is only used for broadcasting the encrypted signal to the authorised subscribers, which may be a small group compared to all subscribers in the network. It will be more efficient if we can devise an encoding scheme in which the authorised subscribers can access the encrypted signal and, at the same time, the other subscribers can receive something on the same channel, such as an advertisement. However, the scheme should not extend the existing allocated bandwidth.

In this paper, we propose such an encoding scheme which gives two levels of access to the subscribers in the network. Watermarking-type techniques based on direct-sequence spread spectrum communications are implemented to add specific information (i.e. advertisements) to the access-limited signal, which is protected by encryption techniques. With this scheme, the allocated bandwidth for broadcasting is utilised more efficiently and more benefit is given to both the service providers (through advertising) and all subscribers in the network (since there will be programmes which they are not authorised to access but can see advertised).

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 114-118, 1999.

2 Description of the Scheme

In spread spectrum (SS) communications [1], a low level wideband signal can easily be hidden within the same spectrum as a high power signal where each signal appears to be noise to the other. The heart of these SS systems is a pseudo-random binary sequence (PRBS). For these direct sequence SS systems, the original baseband bit stream is multiplied by the PRBS to produce a new bit stream. Only those receivers equipped with the correct PRBS can decode the original message. At the receiver, the low level wideband signal will be accompanied by noise, and by using a suitable detector/demodulator with the correct PRBS, this signal can be squeezed back into the original narrow baseband. Because noise is completely random and uncorrelated, the wanted signal can easily be extracted.

Several watermarking techniques, such as those proposed in [2, 3], are based on these ideas. By spreading the information bits and modulating them with a PRBS, the watermark signal can be obtained. This signal is then embedded in the video signal below the threshold of perception. The recovery of the embedded watermark signal can be accomplished by correlating the watermarked video signal with the same PRBS that was used in the process of constructing the watermark signal. Correlation here is demodulation followed by summation over the width of the chip-rate (the number of blocks over which each information bit is spread). If the peak of the correlation is positive (or, respectively, negative), the recovered information bit is a +1 (or -1).

Using a similar technique to the above (particularly like that proposed in [3]), the information signal will be added to the encrypted signal (after the channel coding process) to give the signal for transmission. Given a key to reproduce the same PRBS at the receiver's side, the information signal can be recovered. Then the encrypted signal can be recovered by subtracting the information signal from the transmitted signal. Any errors which occur at this stage (both communication channel errors and any resulting from the need to ensure that the signal for transmission uses the same block size for its symbols as does the encrypted signal) will be detected and corrected by the channel decoder. The operation of the encoding scheme is shown in Figure 1 below.

We now describe the basic steps of adding the information signal to the encrypted signal. We denote by (m_j), $m_j \in \{-1, 1\}$ a sequence of information bits we want to add to the encrypted signal. This discrete signal is spread by a large factor cr, the chip-rate, to obtain the spread sequence (b_i)

$$b_i = m_j,\ j.cr \le i < (j+1).cr \tag{1}$$

The spread sequence (b_i) is then modulated with a PRBS (p_i), $p_i \in \{-1, 1\}$ and added to the encrypted signal s_i, each s_i block containing k bits, yielding the following signal for the modulation process:

$$s'_i = s_i + p_i.b_i \tag{2}$$

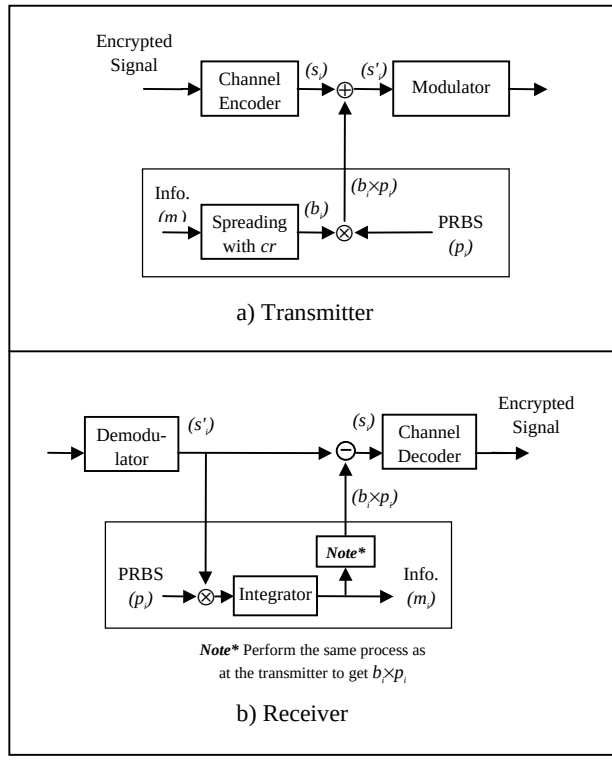


Fig. 1. The operation of the encoding scheme

At the receiver, the recovery of the added information is easily accomplished by multiplying the transmitted signal with the same PRBS (p_i) that was used in the encoder. The summation over the correlation window i.e. cr is as follows:

$$r_j = \sum_{i=j.cr}^{(j+1).cr-1} p_i . s'_i = \sum_{i=j.cr}^{(j+1).cr-1} p_i . s_i + \sum_{i=j.cr}^{(j+1).cr-1} p_i^2 . b_i \tag{3}$$

The first term on the right-hand side of (3) vanishes if p_i and s_i are uncorrelated and $\sum_{i=j.cr}^{(j+1).cr-1} p_i = 0$. However, we account for a different number of -1's and 1's in p_i over the interval [$j.cr$, (j+1).cr-1] by including the term

$$\Delta = \left(\sum_{i=j.cr}^{(j+1).cr-1} p_i \right) . (\bar{s'_i}) \tag{4}$$

Then r_j ideally becomes

$$r'_j = \sum_{i=j.cr}^{(j+1).cr-1} p_i \,.\, s'_i \; - \; \Delta \;\approx\; cr \,.\, m_j \tag{5}$$

and the recovered information bit $m'_j = sign\,(r'_j)$.

As an example, let the bit-rate of the encrypted signal be 3Mb/s, the chip-rate cr = 500 and let the block size k be 4 bits. Then, the rate at which information bits can be added after the channel coding process is 1.5kb/s. With this bit-rate, the information signal can be an image signal in compression form transmitted every 30s or so, and we can transmit the total bit-rate of 3.0015Mb/s within the existing bandwidth allocation of 3Mb/s. To increase the bit-rate of the information signal, the chip-rate and the block size should be reduced. However, a smaller block size implies a greater likelihood that subtracting the information signal from the transmitted signal will not give the encrypted signal. In addition, a smaller chip-rate implies a greater likelihood of error in decoding the information bits. To reduce this latter likelihood of error an error correcting code can be applied to the information bits before the spreading process.

3 Experimental Results

Experiments were carried out using the programming language C. The block size was varied from 2-5 bits to represent up to 32 values. In the experiments, the smallest chip-rate which gave no errors after the decoding process was 41, 95, 470, 1300 for a block size of 2, 3, 4, 5 respectively. For these block sizes, other values of the chip-rate considered gave different values of Bit Error Rate (BER) in the decoded data, and these values and the underlying trend line are shown in Figure 2.

From Figure 2, it can be seen that a larger block size needs a higher chip-rate to retain the same BER. In addition, since one single bit error in the decoded information signal causes error propagation in the encrypted signal, anything other than a large value of the chip-rate will result in a large value of BER. This means that implementing the scheme with a large block size, will lead to low efficiency. According to the experimental results, a block size of 3 is the optimum for the scheme.

In our experiments, the encoding scheme was performed in an error-free communication channel. That is, the errors which occurred in the decoded data came solely from the need to remain within the bandwidth of the broadcast channel. Although the proposed scheme has not been fully explored, it shows an idea of how to utilise the existing broadcast bandwidth in a more efficient way. Further work can be carried out by simulating the scheme in channel models e.g. AWGN. Error correcting codes can be applied in the scheme to improve its reliability.

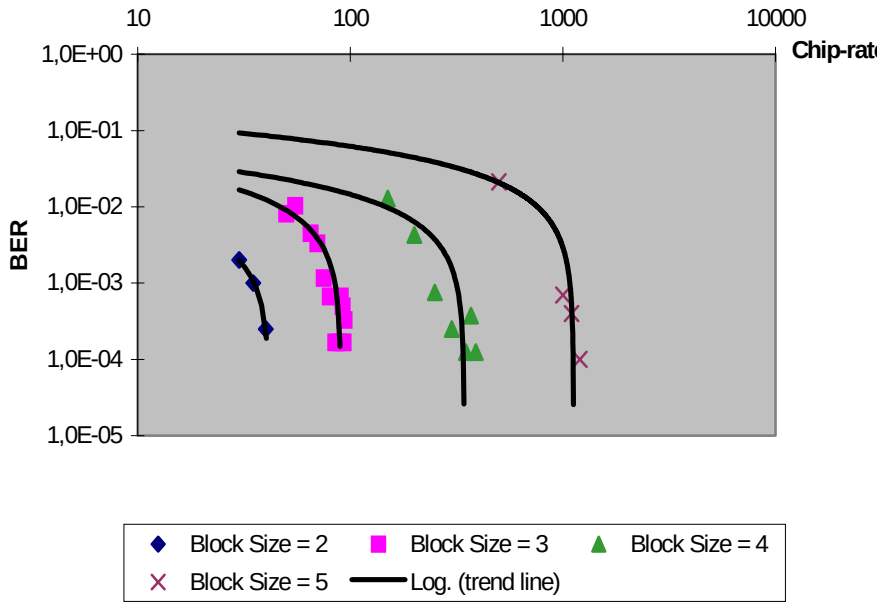


Fig. 2. Chip-rate vs. bit error rate of decoded data at different block sizes

References

1. Pickholtz, R., Schilling, D. and Millstein, L.: Theory of Spread Spectrum Communications. A Tutorial. IEEE Transaction on Communication, Vol. COMM-30 (1982) 855-884
2. Cox, I., Kilian, J., Leighton, T. and Shamoon, T.: Secure Spread Spectrum Watermarking for Multimedia. IEEE transactions on Image Processing, Vol. 6, No. 12 (1997) 1673-1687
3. Hartung, F. and B. Girod, B.: Watermarking of Uncompressed and Compressed Video. Signal Processing, Vol. 66, no. 3 (Special issue on Watermarking) (1998) 283-301

Photograph Signatures for the Protection of Identi cation Documents

Bruno Bellamy[1], John S. Mason[2], and Michael Ellis[3]

[1] ENIB, Brest, France
[2] Department of Electrical & Electronic Engineering,
University of Wales Swansea, SA2 8PP, UK
[3] Dynjab Technologies, Canberra, Australia
J.S.D.Mason@swansea.ac.uk

Abstract. This paper investigates a photo-signature approach to protecting personal identi cation documents such as passports. The approach is based on that described in a recent publication by O'Gorman and Rabinovich [1] which uses encoded data derived from comparisons of image sub-blocks across the photograph of the document. The encoded data is generated and stored at the time of document creation, and used subsequently to test document authenticity. Here we report on experiments which corroborate the fundamental ndings of [1], namely that it is possible to usefully encode the photograph information in only tens of bytes of data.

Furthermore we show that new block structures can improve the e - ciency of the encoded data. This is important since the encoding ef- ciency, measured in terms of number of bytes versus discriminating performance, is particularly important when storing data on a small document such as a passport or ID card. We show that a step structure is measurably better than the original octal structure used in [1] when there are only a small number of bytes (20 to 30) in the photo-signature.

1 Introduction

Photograph-based identi cation documents (PID's) such as passports and ID cards are practical and universally accepted means of verifying a person's identity. A good illustrative example is the international passport. In presenting a passport at an international border control a person is seeking entry to another country. Implicit in the act of presentation is the claim that:

- the presenter is the true owner of the passport,
- the passport is valid, eg not expired or withdrawn,
- the passport is authentic, being issued by a recognised agency, and
- the passport is in its original form and has not been tampered with in a fraudulent manner.

O cers at the border control are tasked with verifying these conditions, and this paper investigates a photograph encoding process aimed at helping in the case of the last two checks: the passport is authentic and the passport is tamper-free.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 119{128, 1999.

Given that a crucial check is based on photographic likeness, then counterfeiters often target a valid document (lost or stolen) and change the photograph to one belonging to the intended illegal recipient. Strategies to counter this form of tampering include sophisticated production methods with enhanced images and materials, in a manner analogous to the printing of bank notes. Automated authentication can then check for these manufacturing details (intaglio, microprinting, holograms etc). However, while these methods provide a deterrent to fraud, experience suggests that they by no-means prevent imitations or photograph substitutions completely, and further measures are needed.

The idea considered here is to extract information from the photograph of a PID at time of issue and store it for subsequent consistency checks. If this encoded photograph data, termed a *photo-signature*, is to be stored on the PID itself, then clearly the size of the encoded data must be quite small: constrained in practice to perhaps just a few tens of bytes.

This paper considers the photo-signature idea recently described by O'Gorman and Rabinovich [1] aimed exactly at this task of automatic PID authentication. The focus here is on e cient encoding strategies, given the need to print the data on the document. Thus a key factor in assessment is system discrimination performance against number of data bytes needed for storage.

2 Photo-Signatures against Counterfeiting

A photo-signature is de ned as an encapsulation of a photograph by a process which enables subsequent discrimination between photographs of di erent people. The proposed approach of [1] harnesses the discriminating properties of the photograph by tying them to the document details, thereby making it much more di cult to successfully tamper with photographs on PID's. The principle is to compress and encode a digital representation of the photograph. Testing repeats the operation and compares the two signatures.

The goal of such a system is clear and is common with all such anti-counterfeiting measures: simply, it is to make the creation of illegal, fake PID's more di cult. The photo-signature focuses on important discriminating feature of the PID, namely the photograph itself. So whether generating (illegally) a fake PID from the beginning or compromising an otherwise genuine document, a photo-signature corresponding to the photograph is needed. Thus further obstacles confront the counterfeiters. In addition now, rst they need to know the encoding algorithm and second they need to be able to write the code in an appropriate form. The task is therefore signi cantly more di cult than merely physically swapping a photograph.

2.1 Passport Protection

The case of passports is an interesting one which makes it stand out from other PID's. The passport's raison d'etre is to facilitate international border control,

and nations world-wide have their own passport designs, issuing o ces, control and anti-fraud measures. This variety makes the detection of counterfeit passports less e cient than it might otherwise be: ideally, the same practices should span national boundaries. For if some secret encoding process were to be incorporated in to the passports of just one country it could have only limited anti-counterfeiting bene ts. Holders of such passports would travel to and from that one country with increased passport security. For other destinations and other travelers, there would be no bene t from this particular measure. Furthermore, determined counterfeiters might eventually learn of the existence of security feature and, if mastering it proved too di cult, they could simply turn to 'softer' targets of other countries. Ironically, as the list of participating countries grows, so does the possibilities for counterfeiters to gain access to the encoding methods and emulate the security measures embedded in the manufacturing process. However, this seems a price that must be paid.

The inclusion of a system based on photo-signatures would seem to be most appropriate for passports since it would deter photograph substitution, a practice counting for a signi cant percentage of illegal passports. A distinct merit of a system like that proposed by O'Gorman and Rabinovich [1] would be the minimal physical changes to the passport itself: just the storage of the encoded photo-signature. Equally importantly, there would be only negligible marginal manufacturing costs. Enhanced security would stem from the automated detection of photograph tampering. If the passport photograph were to be changed, then the old photo-signature would also need to be changed in order that the two components tally. Of course, it is conceivable that a knowledge of the encoding algorithm could be gained by the criminal fraternity, and the likelihood of this is increased by the algorithm being known to the necessarily large number of passport issuing o ces. However these drawbacks are more than o set by the obvious increased di culties for the counterfeiters to produce and print the new photo-signature that matches the substitute photograph. In addition public-key cryptography is available as a further robust security measure.

2.2 The Photo-Signature of [1]

Merits of photo-signatures stem from the important principle that details of the photograph are tied to the document itself. So if one changes, then so must the other in order to maintain consistency. The proposed photo-signature of O'Gorman and Rabinovich [1] has additional merits, particularly pertinent to the case of passports. The rst relates to passport production. The inclusion of the photo-signature would introduce minimal demands for physical changes: merely storage of the encoded photo-signature. This in no way compromises or interferes with other security measures. The second relates to authentication in practice. The associated algorithms are un-demanding computationally and hence checks at border control can be performed rapidly by relatively low-cost machines. Finally, and perhaps most importantly, the signature is small, occupying just a few tens of bytes.

The procedure to derive the photo-signature of [1] can be summarised as:

1. low-pass, 2-D lter and sub-sample in successive powers of two the original grey-scale image, see Figure 1.

Fig. 1. An example of smoothing and sub-sampling for multi-resolution levels of encoding

2. at some or all levels t a grid to the sub-sampled image, see Figure 2.
3. at each grid intersection encode the relative intensities of the 8 neighbouring sub-image regions or blocks, see Figure 3. We refer subsequently to this as octal coding.

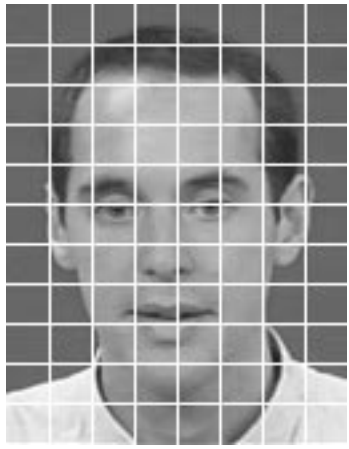
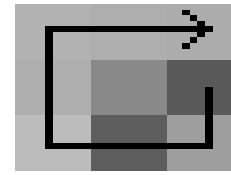


Fig. 2. Grid determination of the centres of comparison



Feature byte example: 11111010

Fig. 3. Octal coding: eight neighbouring intensities are compared with that of the central block

4. repeat the previous stage at di erent resolution levels. The nal signature comprises one byte for each grid intersection.

Clearly in the above case emphasis is given to a multi-resolution approach, combining di erent levels of smoothing and di erent grid sizes. One merit of this is the increased complexity making 'reverse engineering' of the encoding process that much more di cult for potential counterfeiters. It is also likely to increase security in that di erent levels of averaging will normally contain di erent information. However it does raise the question of which levels to combine. Here we avoid this question and examine in some detail the performance of di erent resolution levels, each treated singly rather than in combination.

3 Experiments

The goals of the experiments reported here are to assess photo-signature discrimination performance in terms of:

- sub-image block squares of dimensions K pixels,
- block separation distance, d, and
- di erent structures other than octal

3.1 Assessment

In the experiments reported below photo-signatures are compared one-with-another (inter-class), and with degraded forms of the same image (intra-class). The rst gives an indication of discrimination at time of production, that is, when the photo-signatures are rst generated from the photographs. The second gives an indication of robustness against subsequent natural changes in the photograph. This intra-class variation can be attributed to:

- changes in the original document due to naturally occurring degradations, fading, scratches etc.
- di erences in the scanning operation.

Below, intra-class and inter-class experiments are brought together to assess system performance.

The database used contains 34 pictures, 130 x 170 pixel sized corresponding to a scan at 150 dpi for common PID photographs. To simulate picture degradations, we add Gaussian noise to each test photograph with standard deviation equal to 30 over 256 grey-levels. Such noise is also used in [1].

In-class testing is performed with the same original photo-scan, plus the noise, so here there are 34 comparisons for each noise condition. Out-of-class testing uses 33 comparisons against the 34th, and so results are an average of 561 tests.

Experimental results are reported here in terms of bit di erences in the encodings of two images, a Hamming distance. These are presented separately

Fig. 4. Example of image degradation by additive Gaussian noise

for in-class and out-of-class comparisons. The latter aim to show that di erent photographs give di erent signatures, while in-class comparisons aim to show robustness against degradation. Coding of random images would lead to an average bit di erence of 50% in the out-of-class case and perfect operations would lead to zero bit di erence for the in-class. In practice, statistics from in-class and out-of-class tests would be used to set acceptance thresholds.

3.2 Experimental Results: Octal Coding

Figure 5) shows the geometry and fundamental variables associated with an octal coding scheme. The size of sub-image blocks is K by K pixels and the centres are d pixels apart. Clearly K controls the amount of smoothing and the combination of K and d controls the amount of block overlap.

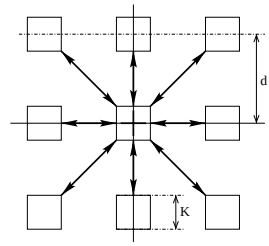


Fig. 5. Geometry of octal coding structure: sub-image blocks of size K by K with centres spaced at d

In the initial experiments we hold K and d constant at 10, and vary the grid dimensions in order to obtain results for di erent photo-signature data sizes. Each grid centre contributes one byte. Results are plotted in Figure 6. All four

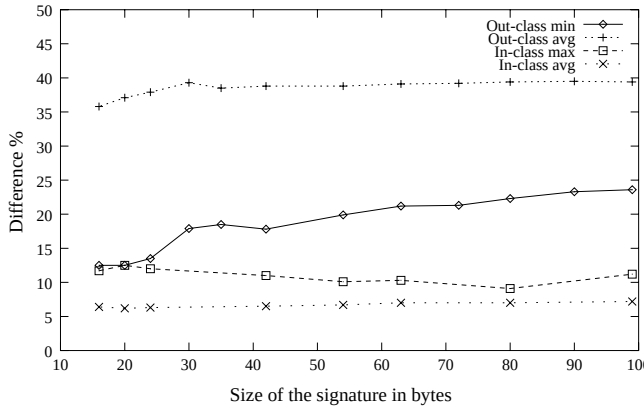


Fig. 6. Octal coding results

pro les represent bit di erences between photo-signatures, in percentage terms. The top two pro les relate to the inter-class experiments and show the average bit di erence (top pro le) and the minimum (second pro le down). The bottom two pro les relate to the intra-class experiments with this time the lower one representing the average and the second up representing the maximum.

The gap between the two middle pro les (maximum inter-class and minimum intra-class) is a measure of goodness. The gap is seen to grow steadily with larger signatures, up to 80 bytes long. Interestingly this is the value chosen by [1].

Next we hold the size of the photo-signature constant at 80 bytes and systematically vary the distance d with K constant at 10 pixels followed by the complement: K is varied while d is held at 10. The results are shown in Figures 7 and 8 respectively. The four pro les represent the same parameters as before. The performance is seen to be relatively insensitive to variations in both d and K. A value of d of 5 or above for K set to 10 gives good results and a value of 7 or above for K (d held at 10) also gives good results. So the conclusion is that performance is relatively insensitive to a range of values for the block smooth parameter, K, and the block separation distance, d.

Next we consider alternative geometries to the octal form. The motivation is as follows. In the octal case, 8 bits are derived for each grid centre. This could place undue emphasis on these centres, since each central block would feature 8 times in the resultant signature. A number of alternative structures have been considered, and here we present the results for the step structure shown in Figure 9.

This structure reduces the number of comparisons per grid centre from 8 down to 2. Results are shown in Figure 10. Unlike in the octal case, it is seen that good discrimination is now obtained down at signature sizes of 20 to 30 bytes, leaving scope for multi-resolution coding should spare capacity of storage be available.

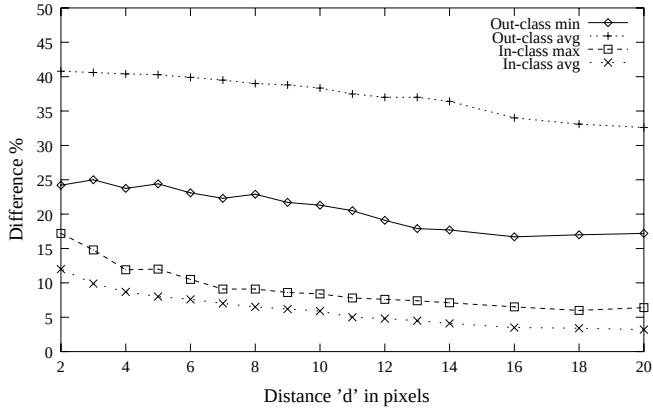


Fig. 7. Octal coding: influence of 'd', 'K' xed to 10 pixels, 80 byte signature

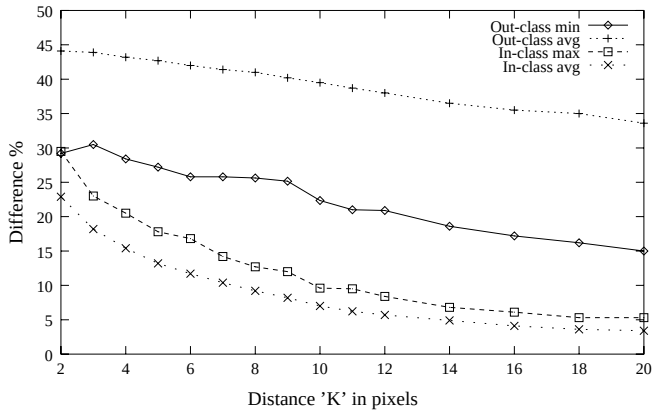


Fig. 8. Octal coding: influence of 'K', 'd' xed to 10 pixels, 80 byte signature

4 Conclusions

A photo-signature based approach to ID document protection proposed by [1] has been investigated in terms of encoded data size and discrimination properties. The approach is particularly appropriate for the case of international passports since it is shown that the signature can be small, in the order of a few tens of bytes, making it practicable to be written on to the document in the form of a

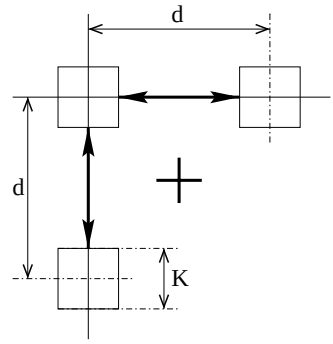


Fig. 9. Step coding structure

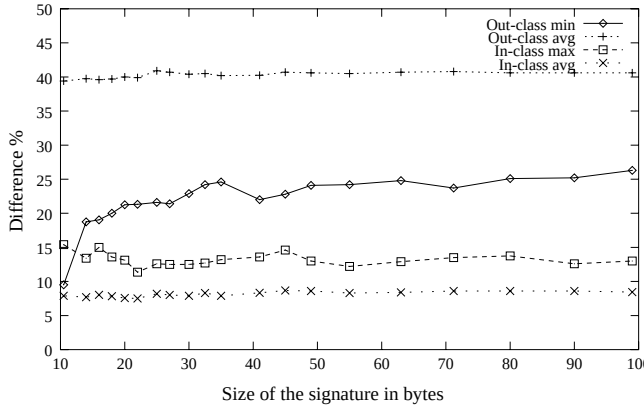


Fig. 10. Step coding results

2-D bar code, for example. It would require no other physical modi cations, and any additional production cost would be negligible.

Results here corroborate the ndings of [1] in demonstrating that signatures of 80 bytes can give good discrimination when using the octal coding structure. Furthermore, it is shown that a step structure in place of the octal form can give an even more e cient signature, with similar discrimination for as little as 20 to 30 bytes. The reason for this improvement is that now there are more centres of comparison spread evenly across the photograph. Other geometries with this characteristic are found to perform equally well. The obvious bene t of this improved e ciency is now any spare capacity in storage on the PID could then be used for multi-resolution coding, thereby improving discrimination.

References

[1] I. Rabinovich L. O'Gorman. \secure identi cation documents via pattern recognition and public-key cryptography". *IEEE trans. pattern analysis and machine intelligence*, pages 1097{1102, Oct. 1998.

An Overview of the Isoperimetric Method in Coding Theory (Extended Abstract)

[Invited Paper]

Jean-Pierre Tillich[1] and Gilles Zemor[2]

[1] Universite Paris-Sud,
LRI, bâtiment 490, 91405 Orsay, France
[2] Ecole Nationale Superieure des Telecommunications,
46 rue Barrault, 75634 Paris 13, France

Abstract. When decoding a threshold phenomenon is often observed: decoding deteriorates very suddenly around some critical value of the channel parameter. Threshold behaviour has been studied in many situations outside coding theory and a number of tools have been developped. One of those turns out to be particularly relevant to coding, namely the derivation of isoperimetric inequalities for product measures on Hamming spaces. we discuss this approach and derive consequences.

1 Background

Let C be a binary linear code with parameters $[n; k; d]$. Denote by $R = k=n$ its rate and $= d=n$ its relative minimum distance. We shall be mainly concerned with the asymptotic behaviour of C so that R and should be thought of as xed quantities as opposed to growing n. Traditionally, two approaches to coding theory have coexisted over the years. One approach consists of looking for codes with the largest possible minimum distance for a given length and dimension, based on the simple statement that if less than $d=2$ errors occur then (disregarding complexity issues) the original codeword can always be recovered. The other approach consists of studying the probability of a decoding error after a codeword has been corrupted by some channel. To take one of the most studied examples, the binary symmetric channel with transition probability p, the probability of a decoding error can be written as :

$$f(p) = 1 - \ _p(W) = 1 - \sum_{x \in W} p^{jxj}(1-p)^{n-jxj} \qquad (1)$$

where jxj denotes the Hamming weight of x. The set of vectors W is a *decoding region*, i.e. the set of error vectors that will be correctly decoded by a maximum-likelihood decoding scheme. Practitioners and theorists alike have been asking for the behaviour of $f(p)$: many results exist on the *typical* behaviour of $f(p)$ when C is chosen randomly from an ensemble of codes. In particular, if C is chosen randomly from all linear codes of (large) length n and rate R, then we

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 129{134, 1999.

know that $f(p)$ jumps suddenly from almost zero to almost one around the critical value $= H^{-1}(1-R)$, this is essentially Shannon's theorem. The typical value of $f(p)$ was then further studied in the sixties by Gallager who showed that $f(p)$ $e^{-nE(R;p)+o(n)}$ for a function $E(R;p)$ that he computed and that is positive whenever $p <$.

One frustrating fact is that the typical critical value of $f(p)$ actually equals the typical relative minimum distance . This means that even if there are many more than =2 corrupted bits, only very few error patterns will actually result in a decoding error. This well-known phenomenon has again been highlighted recently with the progress of iterative decoding techniques, e.g. the advent of turbodecoding. In that case the code C is chosen among a much more speci c and smaller ensemble of codes, and maximum-likelihood decoding is replaced by some suboptimal decoding scheme (this can be seen as replacing the decoding region W in (1) by a considerably smaller set of vectors). Experiments show that $f(p)$ still tends to behave in a threshold manner, for impressive values of , although the codes have poor minimum distance properties. However in these cases, for $p <$, the quantity $f(p)$ does not decay exponentially with the block length n any more.

Our approach is to try and bridge the gap between the two approaches to coding, the study of the minimum distance and the study of $f(p)$. Suppose we are given a code with minimum distance d : what can we say about $f(p)$? About its threshold behaviour ? How can we upperbound $f(p)$? This time we do not ask for typical behaviour but want a result valid for *any* code with minimum distance d.

2 Threshold E ects and the Isoperimetric Method

In 1974 a very elegant result of Margulis [2] went largely unnoticed by the coding community because its implications were not fully apparent.

For any two vectors x and y of the Hamming space $\mathbb{F}_2^n$, let us write x y if for any $i = 1;2;:::;n$ $x_i = 1$ implies $y_i = 1$. We shall say that W is *increasing* if for any $x \in W$, x y implies that y is also in W.

Margulis introduced the quantity :

$$h_W(x) = 0 \qquad \text{if } x \notin W$$
$$h_W(x) = card\{y \in W; d(x;y) = 1\} \text{ if } x \in W$$

where $d(x;y)$ denotes the Hamming distance between x and y. Denote by (W) the smallest nonzero value of $h_W(x)$. Let $_p$ denote the product probability measure on the Hamming space de ned by

$$_p(X) = \sum_{x \in X} p^{jxj}(1-p)^{n-jxj}$$

for any set of vectors X. Margulis's theorem is a very general statement about increasing sets.

Theorem 1 (Margulis 74) *For any > 0; > 0, there exists $m > 0$ such that for any increasing set W satisfying (W) m, the set of p's for which $_p(W)$ takes values between and $1 -$ is an interval of length smaller than .*

This emphasizes the threshold nature of the function $p \mapsto {}_p(W)$. The larger (W), the quicker $_p(W)$ jumps suddenly from almost zero to almost one.

Why is this relevant to coding ? Because it applies almost immediately to the decoding error probability $f(p)$ in (1). In this case the decoding region W is not increasing, but it is a *decreasing* set, i.e. $x \in W$ and $y \leq x$ imply $y \in W$: Margulis's theorem will also apply. Furthemore, the quantity (W) is directly dependent on the minimal distance of the code d, we have namely (W) $d/2$. The consequence is that the decoding error probability $f(p)$ behaves in a threshold manner, i.e. jumps suddenly from almost zero to almost one, and that the \jump" narrows as the minimum distance grows.

Deriving such a result is not straightforward and Margulis's method is especially interesting. It relies on the following identity, later to become known in percolation theory as Russo's identity, which states that for any increasing set W :

$$\frac{d\ {}_p(W)}{dp} = \frac{1}{p} \int_W h_W(x)\, d\ {}_p(x). \tag{2}$$

Margulis then goes on to lower bound the quantity $\int_W h_W(x)\, d\ {}_p(x)$ by a function of ${}_p(W)$. Integrating the resulting di erential inequality then yields the threshold behaviour. The method can be named isoperimetric because the integral $\int_W h_W(x)\, d\ {}_p(x)$ can be thought of as a measure of the \boundary" of W and is lower bounded by a function of its \volume" ${}_p(W)$.

Margulis's theorem was made much more explicit by Talagrand [3] who showed that the estimation of ${}_p(W)$ can be made more precise by considering a modi ed measure of the boundary of W, namely $\int p\sqrt{h_W}\, d\ {}_p$.

Talagrand's isoperimetric inequalities were re ned by Bobkov and Goetze [1], and improved again in [4]. After integration, these inequalities yield the following result for increasing sets.

Theorem 2 (Tillich Zemor 99) *Let W be an increasing set of vectors of $\mathbb{F}_2^n$, and let $=$ (W). Let be de ned by $(W) = 1/2$. Then ${}_p(W)$ satis es :*

$$ {}_p(W) \qquad \frac{p}{2}\left(\sqrt{-\ln} - \sqrt{-\ln p}\right) \quad \textit{for} \quad 0 < p < \tag{3}$$

$$ {}_p(W) \qquad \frac{p}{2}\left(\sqrt{-\ln} - \sqrt{-\ln p}\right) \quad \textit{for} \quad < p < 1. \tag{4}$$

where $(x) = \frac{1}{\sqrt{2}} \int_{-1}^{x} e^{-t^2/2} dt$.

Applied to coding, this yields the following theorem.

Theorem 3 (Tillich-Zemor 99) *Let C be a binary linear code with minimum distance d and any length. Over the binary symmetric channel with transition probability p, the probability of decoding error $f(p)$ associated to C satis es :*

$$f(p) \quad 1 - \left[p\overline{d} \quad p\frac{}{-\ln(1-\ )} - p\frac{}{-\ln(1-p)} \right] \quad \text{for} \quad 0 < p <$$

$$f(p) \quad 1 - \left[p\overline{d} \quad p\frac{}{-\ln(1-\ )} - p\frac{}{-\ln(1-p)} \right] \quad \text{for} \quad < p < 1:$$

where is de ned by $f(\ ) = 1=2$.

This theorem makes the threshold behaviour of $f(p)$ very precise. Note that for xed $a < 0$ and growing d the quantity $(a\overline{p\ d})$ is equivalent to $\frac{}{a\ p\frac{1}{2}d}e^{-da^2=2}$, so that theorem 3 really gives an upper bound of the form

$$f(p) \quad \exp(-dg(\ ;p))$$

where $g(\ ;p) > 0$ for $p <$: in other words $f(p)$ is exponentially small in d. In particular, families of codes with minimal distance growing linearly with their length n have a probability of decoding error which decreases exponentially with n, as long as $-p$ stays bounded below by some $" > 0$. We now know that this holds for *all* such codes, not just that it is typical behaviour.

3 Locating the Threshold

The isoperimetric method gives precise results on the behaviour of the decoding error probability $f(p)$ given d but does not say anything about the whereabouts of the threshold probability . As mentioned earlier, randomly chosen codes with large length have $= = d=n$, but what is the (asymptotic) situation for any code with prescribed relative distance ? More precisely, denoting by (C) the threshold value for a code C, we would like to determine

$$= \liminf \ (C)$$

where the lim inf is de ned over any sequence enumerating the set of all codes C such that $d \quad n$. What is the best lower bound on as a function of ? This is an interesting open question. It should be clear that we must have =2. If it were true that = , this would imply that the Varshamov-Gilbert bound is tight. The best lower bound on known to us makes use of averaging arguments together with bounds on the highest possible rate of constant weight codes [4] : here are some numerical values.

Table 1. as a function of

	0.1	0.2	0.3	0.35	0.4	0.45	0.5
lower bound on	0.053	0.123	0.212	0.267	0.330	0.385	0.5

4 The Erasure Channel

Theorem 2 is very general and should nd applications in a variety of situations. Its applications to coding are especially interesting in the context of the erasure channel. Let C be again a binary linear code with parameters $[n; k; d]$. This time, when a codeword of C is transmitted its symbols are erased independently with probability p. Let $e \in \mathbb{F}_2^n$ be the *erasure vector*, i.e. the characteristic vector of the set of erased positions. The probability $f(p)$ that we are now interested in is the probability that the initial codeword cannot be recovered from the set of received symbols : it is straightforward to check that this happens exactly when $c \quad e$ for some nonzero codeword c. We have therefore :

$$f(p) = \quad _p(W)$$

where W now stands for the set of vectors x for which there exists $c \in C$, $c \neq 0$ such that $c \quad x$. Every vector in W has weight at least d, from which we have the well-known fact that C can always correct up to $d - 1$ erasures. But it might very well be true (actually it is true, this is our point) that C can, with high probability, correct many more erasures.

It is clear that W is an increasing set of vectors. Furthermore, it is not di cult to show that, because C is linear, $(W) = d$. Therefore the isoperimetric method applies, and theorem 2 translates directly into a theorem of a nature similar to that of theorem 3 (see [4]).

Actually, Margulis's initial motivation for deriving his theorem was the study of this function $f(p)$ in the particular situation when C is the cocycle code of a graph. In this case $f(p)$ represents the probability that a random set of edges disconnects the graph.

As in the case of the binary symmetric channel, we would like to lower bound the threshold (de ned by $f( ) = 1=2$) by a function of . De ning again $= \liminf (C)$ where C runs over all codes such that $d \quad n$, we have trivially that . But interestingly, in this case the isoperimetric approach can be pushed further to yield nontrivial lower bounds on .

The idea is to de ne the sequence of sets

$$W_1 = W \quad W_2 \quad \cdots \quad W_t \quad \cdots$$

where W_t is the set of vectors x such that

$$C_x = \{c \in C \mid c \quad x\}$$

is a subcode of C of dimension t. The threshold probabilities associated to each W_t form a sequence

$$_1 = \quad _2 \quad \vdots\vdots \quad _t \quad \vdots\vdots$$

The isoperimetric method will show that the di erences $_{t+1} - _t$ must tend to zero as the minimum distance tends to in nity. We can then argue that a code of length $_t n$, dimension t, and minimum distance d must exist, so that these parameters must not contradict existing bounds. This argument gives [5,4]

$$2$$

and can be pushed further to yield improved lower bounds : this is the object of forthcoming work.

References

1. Bobkov, S., Goetze, F. (1996) Discrete Isoperimetric and Poincare-type inequalities. Technical report SFB 343 University of Bielefeld 96-086. ftp://ftp.mathematik.uni-bielefeld.de/pub/papers/sfb343/pr96086.ps.gz
2. Margulis, G. (1974) Probabilistic characteristics of graphs with large connectivity. Problemy Peredachi Informatsii. 10, 101{108
3. Talagrand, M. (1993) Isoperimetry, logarithmic Sobolev inequalities on the discrete cube, and Margulis' graph connectivity theorem. Geometric and Functional Analysis. 3, 295{314.
4. Tillich, J-P., Zemor, G. (1999) Discrete inequalities and the probability of a decoding error. Submitted to Combinatorics, Probability & Computing. http://www.infres.enst.fr/ zemor/isoperimetric.ps
5. Zemor, G. (1994) Threshold e ects in codes. In Algebraic coding, Springer-Verlag, LNCS 781 278{286.

Rectangular Basis of a Linear Code

Johannes Maucher[1], Vladimir Sidorenko[2], and Martin Bossert[1]

[1] Department of Information Technology, University of Ulm,
Albert-Einstein-Allee 43, 89081 Ulm, Germany,
joma,boss@it.e-technik.uni-ulm.de
[2] Institute for Information Transmission Problems,
Russian Academy of Science,
B.Karetnyi per.19 101447, Moscow GSP-4, Russia,
sid@iitp.ru

Abstract. A rectangular code is a code for which there exists an unique minimal trellis. Such a code can be considered to be an algebraically closed set under the rectangular complement operation. The notions of rectangular closure and basis were already de ned. In this paper we represent a method to construct a rectangular basis of a linear code from a given linear basis.

1 Introduction

Each code C can be represented in a trellis $T(C)$. This trellis representation is applied in decoding algorithms, for example in the Viterbi decoding algorithm. For a given code there exists a large variety of corresponding trellises. Obviously, for most applications a trellis with a minimal complexity is preferred, however there exists di erent complexity measures. It was shown in [3] that there exists a class of codes for which there exists a unique minimal trellis, i.e. a trellis which minimizes all ordinary complexity measures. This set of codes is called the set of rectangular codes. It can easily be shown that each linear code is a rectangular code. Hence, in most of the previous works people investigated which nonlinear codes are rectangular. In recent works [8], [7] the algebraic structure of rectangular codes is studied. It is shown in these papers that for any nonrectangular code there exists a unique rectangular closure. Moreover, an algorithm is proposed which computes for each rectangular code a rectangular basis. For a given nonrectangular code the trellis of the corresponding rectangular closure has a smaller complexity, than the trellis of the nonrectangular code itself. Therefore the decoding complexity decreases if a nonrectangular code is decoded in the trellis of its rectangular closure. Moreover, as shown in [9], for some iterative decoding algorithms which use a set of low weight codewords of a linear code, complexity decreases and performance increases if the rectangular closure of these low weight codewords is used. The merit of a rectangular basis is that it provides a quite compact description of a rectangular code, i.e. in applications in which a rectangular set must be stored on a device it is su cient to store only its rectangular basis and generate the whole set from this basis, whenever it is needed.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 135{143, 1999.

In this paper we investigate the relation between rectangular bases and linear bases of linear codes. In particular we show how a rectangular basis of a linear code can be derived from its generator matrix in trellis oriented form.

2 Notations and De nitions

2.1 Trellis Representation of a Code C

A trellis $T = (V; E; A)$ is an edge labeled directed graph. In a trellis of length n the set of vertices is $V = V_0 [\quad [V_n$ and the set of edges is $E = E_1 [\quad [E_n$. Each edge $e \; 2 \; E_i$ connects a vertex $v \; 2 \; V_{i-1}$ with a vertex $v' \; 2 \; V_i$. The initial vertex of e is then $i(e) = v$ and its nal vertex is $f(e) = v'$. The edge e is said to be incident from $i(e) = v$ and incident to $f(e) = v'$. A path of length l consists of a sequence of edges $[e_{i_1} ::: e_{i_l}]$, such that $f(e_{i_j}) = i(e_{i_{j+1}})$ for all $j \; 2 \; f1; ::: ; l - 1g$. Each edge $e \; 2 \; E_i$ is labeled by an element from the nite set A_i. Then each path of length n from a vertex in V_0 to a vertex in V_n is labeled by an element from the cartesian product $A = A_1 \quad A_n$. In a trellis $T(C)$ of a code C there exists only one vertex v_r in V_0 and only one vertex v_g in V_n. For each codeword c 2 C there exists a path pa(c) from v_r to v_g which is labeled by c. Usually trellises $T(C)$ are considered in which there exists for each c only one path pa(c) labeled by c. Among all possible trellises $T(C)$ of a given code C, the minimal trellis of C is the one which minimizes the number of vertices in each V_i simultaneously [5]. There exists an unique minimal trellis for C, i C is rectangular [3].

2.2 Rectangular Codes

In general a rectangular code of length n can be considered to be a subset of the cartesian product $A = A_i \quad A_n$ of arbitrary nite sets A_i. However, since we consider in this paper linear codes we restrict throughout this paper rectangular codes to be subsets of the $n-$ dimensional vector space $GF(q)^n$. Any codeword c $= (c_1; ::: ; c_n)$ 2 C can be partitioned in an arbitrary depth $t \; 2 \; f1; ::: ; ng$ into its $t-$head $p = (c_1; ::: ; c_t)$ and its $t-$tail $f = (c_{t+1}; ::: ; c_n)$. We denote by $P_t(C)$ the set of all $t-$heads and by $F_t(C)$ the set of all $t-$tails of code C.

De nition 1 *Let $p_1; p_2 \; 2 \; P_t(C)$ and $f_1; f_2 \; 2 \; F_t(C)$. Then C is called t-rectangular, i*

$$(p_1; f_1); (p_1; f_2); (p_2; f_1) \; 2 \; C \text{ implies } (p_2; f_2) \; 2 \; C: \tag{1}$$

If C is t-rectangular in all depths $t \; 2 \; f1; ::: ; ng$, then C is rectangular.

Example 1 *The code $C = f110; 101; 111g$ is $1-$rectangular, but not $2-$rectangular, because the vector $v = 100$ is not contained in C.*

All linear codes are rectangular because they are closed under addition and subtraction of codewords: For the codewords $c_1 = (p_1; f_1)$; $c_2 = (p_1; f_2)$ and $c_3 = (p_2; f_1)$ of the linear code C, the vector

$$c_4 = c_2 - c_1 + c_3 = (p_1; f_2) - (p_1; f_1) + (p_2; f_1) = (p_2; f_2)$$

is again a codeword of C. Thus the de nition relation (1) is ful lled for all depths $t \in \{1; \ldots; n\}$.

For any set of vectors $Y \subseteq GF(q)^n$ the *rectangular closure* $S = R(Y)$ is de ned to be the smallest rectangular set which contains Y. The rectangular closure of a given set Y is unique. $R(Y)$ is closed under the rectangular complement operation, which is de ned as follows:

De nition 2 *Let p_i be the $t-$head and f_i be the $t-$tail of vector c_i. Let $\{c_1; c_2; c_3\}$ be a set of three vectors with $f_1 = f_2$ and $p_2 = p_3$. Then the $t-$rectangular complement*

$$z := r_t(c_1; c_2; c_3)$$

of $\{c_1; c_2; c_3\}$ is $z = (p_1; f_3)$.

Note that the rectangular complement of a set of three vectors is de ned, i within this set there exists a pair of vectors which has the same $t-$head and a pair of vectors which has the same $t-$tail. If for a given set of three vectors $\{c_1; c_2; c_3\}$ the rectangular complement is de ned in depth t and depth l, then $r_t(c_1; c_2; c_3) = r_l(c_1; c_2; c_3)$. Therefore the depth index in the rectangular complement operation can be omitted. A rectangular closed set S, together with the rectangular complement operation r constitutes an algebra $A = < S; r >$. Strictly speaking it is a partial algebra [2] since the operation r is not de ned on all triples of vectors.

Example 2 *The rectangular complement of the three vectors of C from Example 1 is not de ned in depth $t = 1$. However in depth $t = 2$ the rectangular complement is de ned:*

$$r(110; 101; 111) = 100.$$

The new set $S = C \cup \{100\}$ is rectangular, i.e. $S = R(C)$ is the rectangular closure of C.

A set of vectors Y is said to be a *rectangular independent* set if none of the vectors $y_i \in Y$ is contained in the rectangular closure of the other vectors in Y:

$$y_i \notin R(Y \cap \{y_i\}).$$

If S is the rectangular closure of Y, then Y is said to be a generating set of S. A generating set of S, which is rectangular independent is a basis of S. In the sequel we denote a bases by B and its elements by b.

3 Construction of a Rectangular Basis

Throughout this section C denotes a rectangular code in general, i.e. C is not restricted to be linear. We represent an algorithm which constructs a rectangular basis $B(T)$ of code C in the minimal trellis $T(C)$. This algorithm was introduced in [8]. The proof of Theorem 1 can be found in [4].

Coloring Algorithm:

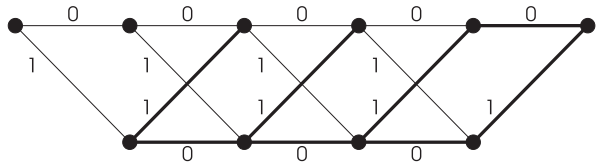
1. *For each vertex v in the minimal trellis $T(C)$ all edges, incident to v, except one edge, must be colored. All edges, incident to the goal vertex v_g must be colored. Set $B(T) = ;$.*
2. *For each colored edge e construct a path in $T(C)$ that goes from the root vertex v_r to the goal vertex v_g through the edge e and comes to the vertex $v = i(e)$ through noncolored edges. Join the codeword corresponding to this path to the set $B(T)$.*

Theorem 1 *If $T = (V; E; A)$ is a trellis of a rectangular code C, then $G(T)$ is a generating set of C and*

$$jB(T)j = jEj - jV j + 2:$$

If in addition the trellis T is the minimal, then $B(T)$ is a basis of C.

Example 3 *For the parity check code $C(5; 4; 2)$, the minimal trellis and a possible coloring according to item 1 of the coloring algorithm is shown in the picture below, where 'colored' edges are printed bold.*



From this colored trellis one can determine, according to item 2 of the coloring algorithm, for example the basis $B = f11000; 10100; 01100; 01010; 00110; 00101; 00011; 00000g$.

4 Construction of a Rectangular Basis for Linear Codes

The drawback of the coloring algorithm is that it constructs a rectangular basis in the minimal trellis and the construction of such a trellis may be quite complex for large codes. In the sequel we will represent a method to construct a rectangular basis of a linear code directly from its generator matrix. The merit of this new method is, that it is less complex than the coloring algorithm and it provides a

better understanding of the relation between rectangular and linear bases. In [1] Forney introduced the trellis oriented generator matrix. From the trellis oriented generator matrix of a code C one can directly determine some properties of the minimal trellis $T(C)$. In particular there exists an e cient method to construct the minimal trellis from the rows of this matrix [3]. On the other side we know a method to get a rectangular basis in the minimal trellis by the coloring algorithm. We will show how these two methods can be combined to construct a rectangular basis from the trellis oriented generator matrix such that the intermediate step of constructing a minimal trellis is not necessary.

4.1 Generator Matrix in Trellis Oriented Form

For a given codeword $c = (c_1; c_2; \ldots; c_n) \in C$ the *left index* $L(c)$ is de ned to be the smallest depth i for which $c_i \neq 0$ and the *right index* $R(c)$ is de ned to be the highest depth i for which $c_i \neq 0$. For example the left index of $c = (0100100)$ is $L(c) = 2$ and its right index is $R(c) = 5$. Vector c is said to be *active* within the interval $[L(c); \ldots; R(c) - 1]$ and *passive* in all depths outside this interval.

De nition 3 *Let G be the generator matrix of a linear code C, and denote the i:th row of G by g_i. Then G is said to be in trellis oriented form, if all pairs of rows $g_i; g_j$ have distinct left and distinct right indices:*

$$L(g_i) \neq L(g_j) \quad ; \quad R(g_i) \neq R(g_j) \tag{2}$$

Example 4 *A generator matrix of a parity check code $C(5; 4; 2)$ is*

$$G = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Since (2) is ful lled, G is in trellis oriented form.

In [1] it is mentioned, that for each linear code there exists a trellis oriented generator matrix.

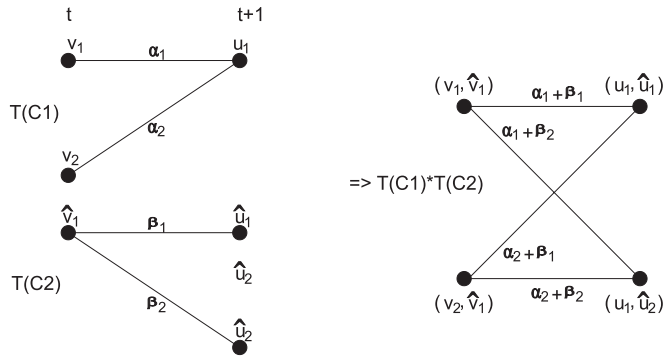
4.2 Minimal Trellis Construction Based on the Shannon Product

Given a pair of linear codes $C_1; C_2$ of the same length and corresponding code trellises $T(C_1)$, $T(C_2)$ the trellis $T(C)$ of their sum

$$C = C_1 + C_2 = \{c_1 + c_2 \mid c_1 \in C_1; c_2 \in C_2\}$$

can be obtained by computing the Shannon product $T(C_1) \times T(C_2) = T(C)$, which is de ned as follows:

Definition 4 *Let V_t and $\hat{V}_t$ be the set of vertices in depth t of trellises $T(C_1) = (V; E; A)$ and $T(C_2) = (\hat{V}; \hat{E}; A)$, respectively. The vertices of the Shannon product $T(C) = T(C_1) \; T(C_2)$ are then marked by pairs $(v; \hat{v})$ with $v \in V_t$ and $\hat{v} \in \hat{V}_t$. In $T(C)$ a vertex $(v; \hat{v})$ in depth t is connected to a vertex $(u; \hat{u})$ in depth $t + 1$, i in $T(C_1)$ vertex $v \in V_t$ is connected to vertex $u \in V_{t+1}$ and in $T(C_2)$ vertex $\hat{v} \in \hat{V}_t$ is connected to vertex $\hat{u} \in \hat{V}_{t+1}$. If is the label of the edge which connects v and u in $T(C_1)$, and is the label of the edge which connects $\hat{v}$ and $\hat{u}$ in $T(C_2)$, then the label of the edge which connects vertex $(v; \hat{v})$ with vertex $(u; \hat{u})$ in $T(C)$ is + :*



Let g_i be a row of a generator matrix of a linear $(n; k)-$code C, defined over $GF(q)$. We denote by $T(g_i)$ the minimal trellis of a subcode

$$C_i = hg_i i = fsg_i \quad j \quad s \in GF(q)g;$$

spanned by g_i. The minimal trellis $T(C)$ can be constructed from its trellis oriented generator matrix G by a stepwise calculation of the Shannon product of subcode trellises:

Theorem 2 *Let $fg_1; g_2; \ldots; g_k g$ be the rows of the generator matrix G of a linear code C. Then the Shannon product $T(g_1) \qquad T(g_k)$ is a minimal trellis of C, i G is in trellis oriented form.*

This Theorem is proved in [3] and [6].

5 Construction of a Rectangular Basis from the Trellis Oriented Generator Matrix

5.1 Subcodes of Dimension $k = 1$

The minimal trellis $T(g_i)$ has the property that in all depths l within the two intervals $[1; \ldots; L(g_i) - 1]$ and $[R(g_i); \ldots; n]$ all components of vector g_i are zero. Therefore in these depths l also all components c_l of all codewords in C_i are zero. From this follows property $P1$ and from $P1$ follow properties $P2; P3$ and $P4$ of the minimal trellis $T(g_i)$:

P1 All codewords of C_i have the same $(L(g_i) - 1)-$head, but distinct $l-$heads for $l \quad L(g_i)$. All codewords of C_i have the same $R(g_i)-$tail but distinct $l-$tails for $l < R(g_i)$:

P2 All paths $pa(c)$; $c \; 2 \; C_i$ go through the same vertex in depth l, if g_i is passive in depth l.

P3 For all pairs $c_a; c_b$ of distinct vectors from C_i the paths $pa(c_a)$ and $pa(c_b)$ go through distinct vertices in depth l, if g_i is active in depth l.

P4 In depth $l = R(g_i)$ all q distinct paths $pa(c)$ of codewords from C_i merge into a single vertex, denoted by $vr(i)$.

Note that $vr(i)$ is the only vertex in $T(g_i)$ in which merges more than one path - in particular q paths. The coloring in this trellis can be chosen for example such that all edges incident to $vr(i)$ which are labeled by a nonzero element from $GF(q)$ are colored. The set of $q - 1$ basis codewords b which belong to these colored edges in depth $l = R(g_i)$ is then the set of all nonzero codewords in C_i. However, this set is not the complete basis, since one must also assign a basis codeword b to the colored edge e in depth $l = n$, which is incident to v_g. Since this codeword must correspond to a noncolored path from v_r to $v = i(e)$, it can only be the allzero codeword from C_i. This proves the following Theorem:

Theorem 3 *The rectangular basis of a linear code C of dimension $k = 1$ is $B = C$.*

5.2 Linear Code of Dimension k

Let us now investigate how to determine colored edges, i.e. basis vectors which are assigned to colored edges, directly from the trellis oriented generator matrix of a linear code. By Theorem 2 the minimal trellis $T(C)$ of such a code is the Shannon product of the minimal trellises of the subcodes:

$$T(C) = T(g_1) \qquad T(g_k):$$

From De nition 4 follows that in trellis $T(C) = T(C_i) \quad T(C_j)$ there exists a vertex in which merges more than one edge in depth t, i in at least one of the trellises $T(C_1); T(C_2)$ there exists a vertex in depth t, in which merges more than one edge. This means that in $T(C) = T(g_1) \qquad T(g_k)$ there exist vertices in which merge more than one edge only in depths $R(g_1), R(g_2); : : : ; R(g_k)$. W.l.o.g. we assume that the rows of the trellis oriented generator matrix are ordered such that $L(g_i) < L(g_{i+1})$ and $R(g_i) < R(g_{i+1})$ for all $i \; 2 \; f1; : : : ; k - 1g$. In the sequel we will determine the codewords, which correspond to edges, which merge together in one of the depths $R(g_i)$.

We de ne $\overline{G}_i$ to be the matrix, which consists of all rows of G, except row g_i. The code generated by $\overline{G}_i$ is then the complement of subcode C_i in C:

$$\overline{C}_i = hg_1; : : : ; g_{i-1}; g_{i+1}; : : : ; g_k i:$$

For a xed codeword $\overline{\mathrm{c}} \in \overline{C}_i$ we de ne a coset $C_i(\overline{\mathrm{c}})$ of C_i as follows:

$$C_i(\overline{\mathrm{c}}) = \{\mathrm{c} + \overline{\mathrm{c}} \quad | \quad \mathrm{c} \in C_i\} = \{s\mathrm{g}_i + \overline{\mathrm{c}} \quad | \quad s \in GF(q)\}. \tag{3}$$

Using these de nitions we can generalize properties $P1$ to $P4$ as follows:

Q1 All codewords of $C_i(\overline{\mathrm{c}})$ have the same $(L(\mathrm{g}_i)-1)-$head but distinct $l-$heads for $l \quad L(\mathrm{g}_i)$: All codewords of $C_i(\overline{\mathrm{c}})$ have the same $R(\mathrm{g}_i)-$tail but distinct $l-$tails, for $l < R(\mathrm{g}_i)$.
Q2 All paths $pa(\mathrm{c}); \mathrm{c} \in C_i(\overline{\mathrm{c}})$ go through the same vertex in depth l, if g_i is passive in depth l.
Q3 For all pairs $\mathrm{c}_a; \mathrm{c}_b$ of distinct vectors from $C_i(\overline{\mathrm{c}})$ the paths $pa(\mathrm{c}_a)$ and $pa(\mathrm{c}_b)$ go through distinct vertices in depth l, if g_i is active in depth l.
Q4 In depth $l = R(\mathrm{g}_i)$ all q distinct paths $pa(\mathrm{c})$ of codewords from $C_i(\overline{\mathrm{c}})$ merge into a single vertex, denoted by $vr(i; \overline{\mathrm{c}})$.

We will now determine the set of distinct vertices $vr(i; \overline{\mathrm{c}})$ in depth $l_i = R(\mathrm{g}_i)$; in particular the corresponding codewords $\overline{\mathrm{c}} \in \overline{C}_i$ whose paths $pa(\overline{\mathrm{c}})$ go through vertex $vr(i; \overline{\mathrm{c}})$.

In each depth $l_i = R(\mathrm{g}_i)$ the generator matrix G can be partitioned in submatrices $G^{(l_i)}$ and $\overline{G}^{(l_i)}$ as follows: $G^{(l_i)}$ is de ned to be the matrix, which consists of the $k^{(l_i)}$ rows of the generator matrix G, which are passive in depth l_i and $\overline{G}^{(l_i)}$ is de ned to be the matrix, which consists of the $\overline{k}^{(l_i)} = k - k^{(l_i)}$ rows of the generator matrix G, which are active in depth l_i. The codes generated by $G^{(l_i)}$ and $\overline{G}^{(l_i)}$ are denoted by $C^{(l_i)}$ and $\overline{C}^{(l_i)}$, respectively. In the special case $\overline{k}^{(l_i)} = 0$ we de ne $\overline{C}^{(l_i)}$ to contain only the all-zero codeword.

For a xed codeword $\mathrm{u} \in \overline{C}^{(l_i)}$ we de ne a coset $C^{(l_i)}(\mathrm{u})$ of $C^{(l_i)}$ as follows:

$$C^{(l_i)}(\mathrm{u}) = \{\mathrm{a} + \mathrm{u} \quad | \quad \mathrm{a} \in C^{(l_i)}\}. \tag{4}$$

Note that in (4) all vectors $\mathrm{a} \in C^{(l_i)}$ are generated by all rows of G, which are passive in depth l_i. Therefore we have property $S1$: The vector u in (4) is a vector generated by active rows of G in depth l_i, which yields $S2$. Combining $S1$ and $S2$ yields $S3$:

S1 In depth $l_i = R(\mathrm{g}_i)$ all paths $pa(\mathrm{c}); \mathrm{c} \in C^{(l_i)}(\mathrm{u})$ go through the same vertex.
S2 For all pairs $\mathrm{u}_a; \mathrm{u}_b$ of distinct vectors from $\overline{C}^{(l_i)}$, the paths $pa(\mathrm{c}); \mathrm{c} \in C^{(l_i)}(\mathrm{u}_a)$ go through distinct vertices than the paths $pa(\mathrm{c}); \mathrm{c} \in C^{(l_i)}(\mathrm{u}_b)$.
S3 In depth l there exist $q^{\overline{k}^{(l_i)}}$ distinct vertices. Each path $pa(\mathrm{u}); \mathrm{u} \in \overline{C}^{(l_i)}$ goes through a distinct vertex in depth l_i.

From property $S3$ we know that in each depth $l_i = R(\mathrm{g}_i); i \in \{1; : : : ; k\}$ there exist $q^{\overline{k}^{(l_i)}}$ distinct vertices $vr(i; \mathrm{u})$, and from $Q4$ we know that in each of these vertices merge q distinct edges. These q distinct edges belong to the q distinct paths $pa(\mathrm{c}); \mathrm{c} \in C_i(\mathrm{u})$. W.l.o.g. the coloring of the edges which merge

into vertex $vr(i;\mathrm{u})$ can be chosen such that all $q-1$ edges which belong to paths $pa(\mathrm{c});\ \mathrm{c} \in C_i(\mathrm{u}) \setminus \{\mathrm{u}\}$ are colored. This means that all vectors $\mathrm{c} \in C_i(\mathrm{u}) \setminus \{\mathrm{u}\}$ must belong to the basis. If we apply the de ned coloring to all $q^{\overline{k}^{(l_i)}}$ distinct vertices $vr(i;\mathrm{u})$ in depth $l_i = R(\mathrm{g}_i)$ the set of basis vectors B_i, i.e. the basis vectors which belong to colored edges in depth i is:

$$\begin{aligned} B_i &= \{C_i(\mathrm{u}) \setminus \{\mathrm{u}\} \quad j \quad \mathrm{u} \in \overline{C}^{(l)}\} \\ &= \{s\mathrm{g}_i + \mathrm{u} \quad j \quad s \in GF(q) \setminus \{0\} \text{ and } \mathrm{u} \in \overline{C}^{(l)}\}; \end{aligned}$$

where $\overline{C}^{(l)}$ is the code generated by all rows of G, which are active in depth $l_i = R(\mathrm{g}_i)$. Taking into account that in depth $l = n$ all edges, incident to the goal vertex v_g must be colored and the basisvector, which corresponds to this additional colored edge can be chosen to be the all zero codeword 0; the rectangular basis B of code C is

$$B = \{0\} \cup \left[\bigcup_{i \in \{1;:::;k\}} B_i \right]:$$

Thus the basis B can be determined directly from the rows of the trellis oriented generator matrix, without constructing the minimal trellis $T(C)$.

Example 5 *From the generator matrix of the parity check code* $C(5;4;2)$, *represented in Example 4 we obtain the sets* $B_1 = \{11000;10100\}$, $B_2 = \{01100;01010\}$, $B_3 = \{00110;00101\}$, $B_4 = \{00011\}$. *The union of these sets together with the allzero codeword yields the same basis* B, *as calculated by the coloring algorithm in Example 3.*

References

1. G. Forney. Coset codes - part ii: Binary lattices and related codes. *IEEE Trans. Inform. Theory*, 34:1152{1187, 1988.
2. G. Graetzer. *Universal Algebra*. D. van Nostrand Company, Inc., London/Toronto/Melbourne, 1968.
3. F. Kschischang and V. Sorokine. On the trellis structure of block codes. *IEEE Trans. Inform. Theory*, 41:1924{1937, 1995.
4. J. Maucher. On the theory of rectangular codes. *Ph.D Thesis,Department of Information Technology, University of Ulm*, 1999.
5. D. Muder. Minimal trellises for block codes. *IEEE Trans. Inform. Theory*, 34:1049{1053, 1988.
6. V. Sidorenko, G. Makarian, and B. Honary. Minimal trellis design for linear codes based on the shannon product. *IEEE Trans. Inform. Theory*, 42:2048{2053, 1996.
7. V.Sidorenko, J.Maucher, and M.Bossert. Rectangular codes and rectangular algebra. *Proc. of 13.th AAECC Symposium, LNCS*, Nov. 1999.
8. V.Sidorenko, J.Maucher, and M.Bossert. On the theory of rectangular codes. *Proc. of 6.th International Workshop on Algebraic and Combinatorial Coding Theory*, Sept. 1998.
9. V.Sidorenko, J.Maucher, M.Bossert, and R.Lucas. Rectangular codes in iterative decoding. *Proc. of ITG Fachtagung*, Jan. 2000.

Graph Decoding of Array Error-Correcting Codes

Patrick G. Farrell[1], Seyed H. Razavi[2]

[1] Lancaster University, UK
P.G.Farrell@lancaster.ac.uk

[2] Curtin University of Technology, Perth, Australia

Abstract. The motivation for this paper is to report on concepts and results arising from the continuation of a recent study [1] of graph decoding techniques for block error-control (detection and correction) codes. The representation of codes by means of graphs, and the corresponding graph-based decoding algorithms, are described briefly. Results on the performance of graph decoding methods for block codes of the array and generalised array type will be presented, confirming the illustrative examples given in [1]. The main novel result is that the (7,4) Generalised Array Code, equivalent to the (7,4) Hamming Code, which has a graph which contains cycles, can be successfully decoded by means of an iterated min-sum algorithm.

1. Introduction

Graph decoding, using soft-decision methods, is potentially capable of providing simpler decoder implementations than other techniques. This applies particularly to list decoders, serial and parallel coding schemes, and iterative (eg, turbo) decoding algorithms. In these cases it is necessary to pass soft-decision information between the various stages of decoding, and graph-based decoding algorithms (eg, max- or min-sum and sum-product) are ideally suited for this purpose. In addition, they provide optimum symbol-by-symbol decoding as well as maximum and near-maximum likelihood codeword decoding.

Representation of a code by means of a graph was originally proposed by Tanner in 1981 [2]. Tanner used an iterative decoding algorithm previously discovered by Gallager in 1962 [3], and applied by him to the decoding of low-density parity-check codes. However, the power of this combination of a graphical representation and an iterative decoding algorithm was not fully realised until the work of Wiberg, Loeliger and Kotter in 1995 and 1996 [4,5]. This has led to a flurry of research into graph decoding, which is partly summarised in [1] and a paper by Forney [6]. It is interesting to note that very recent results on graph-based decoding have led to the creation of analog decoders, which outperform digital decoders by two orders of magnitude in speed and/or power consumption [8].

It turns out that array code and generalised array code (GAC) constructions [7] for block codes (and almost all optimum and well-known block codes can be constructed

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 144-152, 1999.

in this way) facilitate and simplify the graph decoding of block codes. There are two main reasons for this. Firstly, the coset decomposition structure of a GAC leads to a corresponding decomposition of the Tanner graph into several disjoint sub-graphs, which in many cases do not contain cycles. This is important because almost all interesting codes have Tanner graphs with cycles, which then makes it difficult to apply a graph-based decoding algorithm. This coset decomposition was demonstrated in [1], with illustrative examples. Results using a simplified version of the max-sum algorithm are given below. Secondly, array codes and GACs can relatively easily be characterised by sectionalised Tanner and state [1,6] graphs, in which each node represents more that one codeword symbol. This leads to a simplified graph, with fewer or no cycles, thus in turn simplifying the decoding algorithm. Some results were given in [1], and will also be the subject of a future paper.

Even with coset decomposition and/or sectionalisation, some code graphs or sub-graphs will turn out to contain cycles. It is therefore of interest to explore ways of applying the max-sum and other "belief propagation" algorithms [4,5] to graphs with cycles. The results of such a study are reported below for the particular case of the binary Hamming code with block length n = 7, k = 4 information bits and minimum distance d = 3, formulated as a GAC structure, using an iterative and simplified max-sum algorithm. This algorithm is not the one introduced in [1], which on further investigation turned out to be faulty. The present algorithm, however, can correct single hard errors in any position in the (7,4) codewords; simulation results for its performance under additive white Gaussian noise conditions (soft errors) also will be reported in a future paper.

2. The Tanner Graph and Max-Sum Algorithm

The Tanner graph [2] of a binary, block, error-correcting code is a bipartite graph specifying the parity check relationships of the code. Each position (bit) of a codeword in the code is represented by one of a first set of nodes in the graph. All the position nodes in a parity relationship are joined by edges to one of a second set of nodes called parity nodes. There are n position nodes and n-k parity nodes in the Tanner graph of an (n,k,d) code.

The Tanner graphs of the (7,3,3) Array code and the (7,4,3) GAC are shown in Figures 1 and 2. The (7,3,3) code is constructed using the (2,1,2) row code and the (4,3,2) column code, with the check-on-checks bit then removed [7]. The parity relations are therefore between positions 1 and 2, 3 and 4, 5 and 6, and 1,3,5and 7; as Figure 1 confirms. The graph does not contain cycles (ie, is a tree). This construction is also used as the basic Array code for the (7,4,3) code, but in addition a binary codeword (000 or 111) from the (3.1.3) repetition code is added to the bits in positions 2, 4 and 6. This then creates the (7,4,3) GAC code [7]. The generator matrix of this code is:

$$\mathbf{G} = \begin{pmatrix} 1100001 \\ 0011001 \\ 0000111 \\ 0101010 \end{pmatrix}$$

which leads to the parity check matrix:

$$\mathbf{H} = \begin{pmatrix} 1111000 \\ 0011110 \\ 1010101 \end{pmatrix}$$

as illustrated in Figure 2. It can be seen that this graph contains cycles.

Details of the max-sum decoding algorithm are given in [1,4,5,6]. Briefly, the Tanner graph of a code is realised as a set of position nodes linked by edges to adders which implement the parity nodes. Assuming that the graph does not contain cycles, the algorithm starts simultaneously from all the outer nodes, moves towards the root of the tree and then propagates back towards the outer nodes. The initial log-likelihood weights w(0) = log{prob(y/x = 0)} and w(1) = log{prob(y/x = 1)}of each received signal element y (representing the transmitted bit x plus noise and other possible impairments) are allocated to the corresponding nodes. As the algorithm propagates through the graph, the weights are processed as follows:

- at a node, the outgoing weights are the sums of the corresponding incoming and initial weights, the final weights at an inner node are the sums of the incoming and outgoing weights on any edge attached to the node, and the final weight of an outer node is the sum of the incoming and initial weights;
- an adder, the outgoing weights are the maxima of the sums of all the corresponding incoming weights, over the set of all possible incoming bit configurations.

In the binary case, only the difference in the weights matters during the algorithm computations, so only a single metric (which now may be negative as well as positive) given by w(0) - w(1), is used in the processing, as follows:

- at a node, the metric (difference) is the algebraic sum of the corresponding node metrics, combined as before;
- at an adder, the outgoing metric has magnitude given by the minimum value of the incoming metrics, and sign given by the sign of the product of all the incoming metrics; it is zero if any one of the incoming metrics is zero.

Thus, with this simplification, the max-sum algorithm becomes a min-sum algorithm [1,6]. The final result of using either decoding algorithm is the same, of course, except for the normalisation inherent in the simplified min-sum algorithm, and the same implications follow.

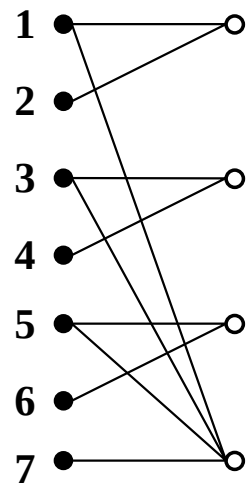


Fig. 1. Tanner graph of the (7,3,3) Array code

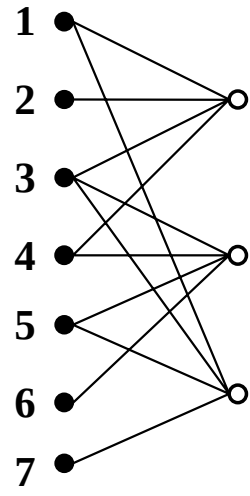


Fig. 2. Tanner graph of the (7,4,3) GAC

3. Decoding the (7,3,3) Array Code

The realisation of the graph of the code, given in Figure 1, is shown in Figure 3. If the initial metric at nodes 1,2, 4-7 is 4, corresponding to high confidence zeros, and is -2 at node 3, corresponding to a lowish confidence one, then applying the min-sum algorithm leads to final metrics of 10 for nodes 1,2,5,6 and 6 for the remaining nodes. The single error in node 3 is corrected, as expected, and the result also shows that positions 4 and 7 in the codeword are more affected by the error in position 3 than the other positions, as indicated by the lower confidence metric values.

4. Decoding the (7,4,3) GAC

As Figure 2 shows, the Tanner graph for the (7,4,3) code is not a tree, because it contains cycles or loops. Regardless of the way in which the code is constructed, its corresponding Tanner graph will always contain cycles. In order to use the min-sum decoding algorithm with this code, it is therefore necessary to find ways in which to avoid or overcome the problem of the presence of the cycles.

4.1 Using the Coset Graph

A first way of dealing with the problem is to take advantage of the GAC form of the code, as described above. This structure means that the codewords in the code can be classified into two cosets: the first coset comprises the codewords of the (7,3,3) Array code, and the second coset consists of the same codewords but with the 2^{nd}, 4^{th} and 6^{th} positions in the codewords inverted (complemented). The first coset is obtained when the 000 codeword in the (3,1,3) repetition code is added to positions 2, 4, and 6 in the basic (7,3,3) codeword (see Section 2 above); adding the 111 codeword then creates the second coset. Thus the (7,4,3) code can be represented by a pair of graphs, one for each coset. The graph corresponding to the first coset is identical to the graph for the (7,3,3) code, and the second graph is the same except that the bits in positions 2, 4, and 6 are inverted (which is equivalent to multiplying their metric values by -1) for the decoding process. The (7,4,3) GAC graph may therefore be drawn as in Figure 4; the min-sum algorithm is applied first with positions 2, 4, and 6 non-inverted, and then secondly with them inverted. In practice these two passes can be done serially or in parallel, as convenient. The results of each pass are then appropriately combined to obtain the final metrics for each bit.

For example, let a set of initial metrics be 4, -4, 4, 4, 4, -4, and 4 in bit positions 1, 2, …..,7 respectively. This corresponds to receiving the codeword 0101010 with a hard error in position 4. After combining the results of the first and second decoding passes, the final node metrics are 4, -4, 4, -4, 4, -4, 4. The error in position 4 is corrected, and all bits have the same final confidence value.

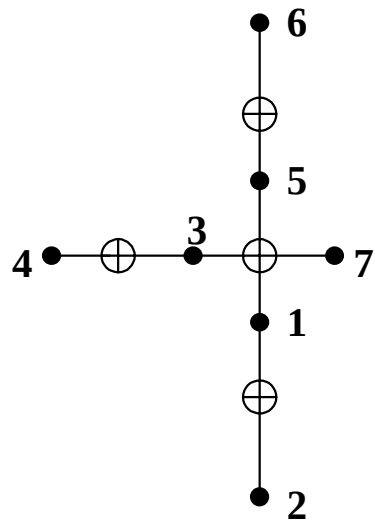


Fig. 3. Realisation of the (7,3,3) Array code graph

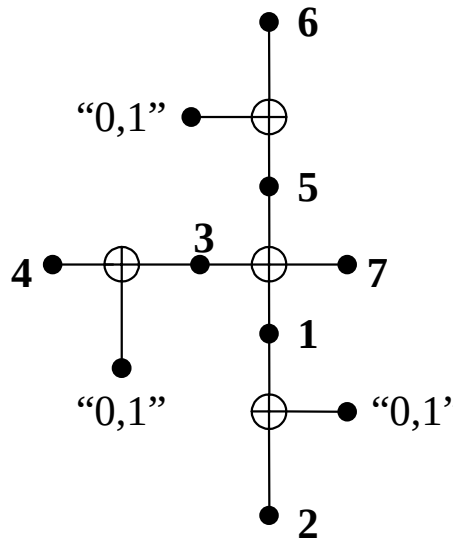


Fig. 4. (7,4,3) GAC coset graph

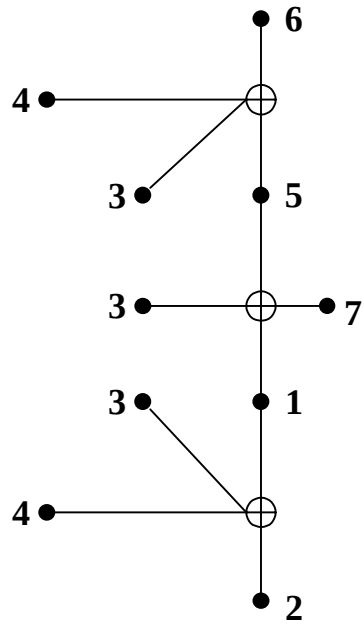


Fig. 5. Split graph for the (7,4,3) GAC

4.2 Using a Split Graph

The second way of dealing with the problem is to modify the graph so as to remove the cycles, and then to repeatedly apply the min-sum algorithm until the final metrics converge satisfactorily. The modification consists of splitting an appropriate set of nodes so as to remove the cycles and create a tree graph. In general there are several ways of doing this. Figure 5 illustrates one way of modifying the graph of the (7,4,3) code shown in Figure 2, by splitting node 3 into three nodes, and node 4 into two nodes. In effect, this has replicated bit positions 3 and 4 in the codewords of the code, which would give an unfair weight to the initial metrics of nodes 3 and 4. Therefore the initial metric on each of the three nodes corresponding to position 3 should be only one third of the original value, and one half of the initial value on each of the two nodes corresponding to position 4. With these modifications, the min-sum algorithm can be applied repeatedly, in a number of iterations.

For example, let a set of initial metrics be 4, -4, -4, -4, 4, -4, 4 in bit positions 1, 2, …,7 respectively. This corresponds to receiving the codeword 0101010 with a hard error in position 3. After splitting nodes 3 and 4, and reducing their initial metrics

accordingly, the set of initial values becomes 4, -4, -1.33, -1.33, -1.33, -2, -2, 4, -4, 4. The final metrics after two iterations of the min-sum algorithm are as follows:

- 1.33, -2.67, 2.67, -1.33, 1.33, -2.67, 2.67
- 2.89, -3.34, 6 , -3.11, 2.89, -3.34, 3.56

Note that the error in position 3 has been corrected, and that the metrics of the other positions are converging back to their original high confidence values after initial falls. Errors in other positions are similarly correctable, but may require up to four iterations.

5. Conclusions

Two methods which permit the use of the min-sum (or max-sum) graph decoding algorithm for block codes with Tanner graphs containing cycles have been described.

The coset graph method becomes computationally complex if the code has a large number of cosets, as many codes of practical interest do. It is therefore relevant to consider methods for limiting the number of cosets which have to be searched [9], though then the final metric values may not be very accurate. The method also depends on the basic coset having a cycle-free graph, and again this will not necessarily be so in many cases of interest. One way to increase the number of cycle-free coset graphs is to create suitable "sectionalised" Tanner graphs, with nodes representing more than one codeword position, and "adders" now operating on bit sequences rather than individual bits. State graphs [4,5,6] also seem very promising.

The split graph method warrants much further investigation. It is not clear, for example, when iteration should stop and how accurate the final metric values are after the last iteration. Computer simulations to determine the performance of the method for a range of codes and error conditions are also required. Is there an optimum way to split the cycle graph? Can state, coset and split graph techniques be combined in some way, to derive more effective soft decoding algorithms? These and other interesting questions will be addressed in subsequent papers.

References

[1] P.G. Farrell: Graph Decoding of Error-Control Codes; 5th Int. Symposium on DSP for Communication Systems, Scarborough, Perth, Australia, 1-4 February, 1999.

[2] R.M. Tanner: A Recursive Approach to Low-Complexity Codes; IEEE Trans Info Theory, Vol IT-27, No 5, pp533-547, Sept 1981.

[3] R.G. Gallager: Low-Density Parity-Check Codes; IRE Trans Info Theory, Vol IT-8, No 1, pp 21-28, Jan 1962.

[4] N. Wiberg, H.-A. Loeliger & R. Kotter: Codes and Iterative Decoding on General Graphs; Euro Trans Telecom, Vol 6, pp513-526, SEPT 1995.

[5] N. Wiberg: Codes and Decoding on General Graphs; PhD Dissertation, Linkoping University, Sweden, April 1996.

[6] G.D. Forney: On Iterative Decoding and the Two-Way Algorithm; Int Symp on Turbo Codes, Brest, France, Sept 3-5, 1997.
[7] P.G. Farrell: On Generalised Array Codes; in Communications Coding and Signal Processing, Eds B. Honary, M. Darnell and P.G. Farrell, Research Studies Press, 1997.
[8] H.-A. Loeliger, F. Tarkoy, F. Lustenberger & M. Helfenstein: Decoding in Analog VLSI; IEEE Comms Mag, April 1999, pp 99-101.
[9] I. Martin & B. Honary: Two-Stage Trellis Decoding of the Nordstrom-Robinson Code Based on the Twisted Squaring Construction, submitted to IEE Proceedings - Communications.

Catastrophicity Test for Time-Varying Convolutional Encoders

Conor O'Donoghue[1] and Cyril Burkley[2]

[1] Silicon & Software Systems, South County Business Park, Leopardstown, Co. Dublin, Ireland
conoro@s3group.com
[2] Dept. of Electronic Engineering, University of Limerick, Limerick, Ireland
cyril.burkley@ul.ie

Abstract. A new catastrophicity test for convolutional encoders whose rate and generator polynomials vary with time is presented. Based on this test computationally e cient algorithm to determine whether or not a time-varying convolutional encoder is catastrophic is derived. This algorithm is shown to be simpler than the catastrophicity test proposed by Balakirsky [1]. Furthermore, the algorithm can easily be generalised to rate k/n time-varying convolutional encoders.

1 Introduction

Let $\mathrm{F}_q[D]$ denote the ring of *polynomials* in the indeterminate D with elements $a(D) = \sum_{i=0}^{m} a_i D^i$, $m \geq 0$, and $a_i \in \mathrm{F}_q$, where F_q is some nite eld with q elements and q is a prime power. An n-vector of polynomials, $\mathbf{a}(D) = (a_1(D), a_2(D), \ldots, a_n(D))$, is an element of $\mathrm{F}_q[D]^n$. The degree of $\mathbf{a}(D)$ is de ned as the maximum degree of its components

$$\deg \mathbf{a}(D) = \max_i \{\deg a_i(D)\} \tag{1}$$

A rate k/n xed convolutional code $\mathcal{C}$ may be generated by any $\mathrm{F}_q[D]$-matrix

$$G(D) = \begin{bmatrix} g_{11}(D) & \cdots & g_{1n}(D) \\ \vdots & & \vdots \\ g_{k1}(D) & \cdots & g_{kn}(D) \end{bmatrix} = \begin{bmatrix} \mathbf{g}_1(D) \\ \vdots \\ \mathbf{g}_k(D) \end{bmatrix} \tag{2}$$

whose rows span $\mathcal{C}$. The ith constraint length and the overall constraint length of $G(D)$ are de ned as $\nu_i = \deg \mathbf{g}_i(D)$ and $\nu = \sum_{i=1}^{k} \nu_i$, respectively. The memory of $G(D)$ is de ned as $\nu_m = \max_i \{\nu_i\}$. Thus, we can write

$$G(D) = \sum_{i=0}^{\nu_m} G_i D^i \qquad G_i \in \mathrm{F}_q^{k \times n} \tag{3}$$

A convolutional encoder is said to be *catastrophic* if there exists some input sequence, $\mathbf{u}(D)$, with in nite Hamming weight which generates a code sequence,

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 153{162, 1999.

$y(D) = u(D)G(D)$, with finite Hamming weight. Such encoders are undesirable as a finite number of channel errors may give rise to an infinite number of errors in the decoded sequence. A necessary and sufficient condition for catastrophic convolutional encoders was first obtained by Massey and Sain [2] and generalised by Olsen [3]. Let $\Delta_i(D)$ be the ith full size minor of $G(D)$ and define

$$\Delta(D) = \gcd\{\Delta_1(D), \ldots, \Delta_L(D)\} \tag{4}$$

where $L = \binom{n}{k}$. Then $G(D)$ is noncatastrophic if, and only if, $\Delta(D) = D^d$ for some $d \geq 0$.

A generator matrix, $G(D)$, for C is *canonical* if its realisation in controller canonical form is minimal i.e. there are no encoders for C requiring fewer memory elements. If $G(D)$ has constraint lengths $\{\nu_i\}_1^k$ the high order coefficient matrix, $G_h \in \mathrm{F}_q^{k \times n}$, is the matrix whose ith row consists of the coefficients of D^{ν_i} in the ith row of $G(D)$. The following theorem, due to Forney [4,5], states when a generator matrix is canonical

Theorem 1 Let $G(D) \in \mathrm{F}_q[D]^{k \times n}$ be a generator matrix for some C with overall constraint length ν. Then the following statements are equivalent:

(a) $G(D)$ is a *canonical* generator matrix.
(b) (i) The gcd of the $k \times k$ minors of $G(D)$ is 1 and
    (ii) their greatest degree is ν.
(c) (i) $G(D)$ is noncatastrophic and
    (ii) G_0 and G_h have full rank. □

The constraint lengths of any canonical generator matrix are invariants of the code C and are called the *Kronecker indices* of C, and denoted by $\{\nu_i\}_1^k$. The sum $\nu = \sum_i \nu_i$ is simply referred to as the Kronecker index and is a measure of the complexity of C.

2 Time-Varying Convolutional Codes

Consider N convolutional codes of rates $1/n_1, \ldots, 1/n_N$ and constraint length at most ν defined by the generator polynomials

$$G^i(D) = \left[g_0^i(D)\ g_1^i(D) \cdots g_{n_i}^i(D) \right] \qquad 1 \leq i \leq N \tag{5}$$

We define a *selection function* $\tau(t)$ such that $\tau(t) \in \{1, 2, \ldots, N\}$ and

$$\tau(t + iT) = \tau(t) \qquad \forall t, i \tag{6}$$

As $\tau(t)$ is periodic it is completely specified by the T-tuple $\boldsymbol{\tau} = (\tau(0), \ldots, \tau(T-1))$. Consider an information sequence $u_0, u_1, \ldots$. Then the time-varying convolutional encoder output at time t is given by

$$y_t = \sum_{i=0} u_{t-i} G_i^{\tau(t)} \qquad y_t \in \mathrm{F}_q^{n_{\tau(t)}} \tag{7}$$

The encoder may be realised with a single shift register of length and time-varying connection vectors. In order to maximise utilisation of the available memory it is commonly assumed [6] that

$$G_0^i \neq 0 \quad \text{and} \quad G^i \neq 0 \qquad 0 \quad i < T \tag{8}$$

All the best time-varying codes reported in [7] and [8] satisfy this condition. Further justi cation for this assumption is provided at the end of this section.

The rate of the time-varying code is given by $R' = 1/n'$ where

$$n' = \frac{n'(0) + n'(1) + \quad + n'(T-1)}{T} \tag{9}$$

Note that n' may not be an integer. Nevertheless, writing R' in this form emphasises the fact that the time-varying code is described by a trellis corresponding to a rate $1/n$ xed code but with branch labels varying on successive trellis sections.

It is well known [7] that any periodic time-varying convolutional code with memory and period T is equivalent to a xed rate $T/n'T$ code with generator matrix

$$G(D) = \sum_{j=0}^{\nu_m} G_j D^j \qquad G_j \in \mathrm{F}_q^{T \times n'T} \tag{10}$$

where $\nu_m = \lceil \quad /Tn \rceil$ and

$$G_j = \begin{bmatrix} G'_{jT}(0) & G'_{jT+1}(1) & \cdots & G'_{(j+1)T-1}(T-1) \\ G'_{jT-1}(0) & G'_{jT}(1) & \cdots & G'_{(j+1)T-2}(T-1) \\ \vdots & \vdots & \ddots & \vdots \\ G'_{(j-1)T+1}(0) & G'_{(j-1)T+2}(1) & \cdots & G'_{jT}(T-1) \end{bmatrix} \tag{11}$$

where, by convention, $G_j^i = 0$ for $j < 0$ and for $j >$. Letting $= \nu_m T -$, the constraint lengths of $G(D)$ can be shown to be

$$\nu_i = \begin{cases} \nu_m - 1 & 1 \quad i \\ \nu_m & < i \quad T \end{cases} \tag{12}$$

The overall constraint length is, therefore, $\nu =$. This equivalence of time-varying and xed codes means that one can think of a time-varying code with period T and constraint length as a special case of a rate $T/n'T$ code which can be decoded with a simpler T-stage decoder with two additions and a binary comparison instead of a single stage decoder with 2^T additions and a 2^T-ary comparison for each state.

It is worth noting that using $'^0(t) = '(t + \ ), \quad \in \mathbb{Z}$, instead of $'(t)$ results in essentially the same time-varying code. However, the corresponding xed code will, in general, be di erent. Therefore, for every periodic time-varying code there are T equivalent xed convolutional codes. Now, consider the case where

$G_0^r = 0$ for some $1 \quad r \quad N$. Choose such that $'^0 = (:::;r)$. The generator matrix for the equivalent xed convolutional code has a low order coe cient matrix

$$G_0 = \begin{bmatrix} G_0^{'^0(0)} & G_1^{'^0(1)} & ::: & G_{T-1}^{'^0(T-1)} \\ 0 & G_0^{'^0(1)} & ::: & G_{T-2}^{'^0(T-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ::: & G_0^{'^0(T-1)} \end{bmatrix} \tag{13}$$

But $G_0^{'^0(T-1)} = G_0^r = 0$ and hence the last row of G_0 is zero. As a result the encoder has non-zero delay and therefore $<$. A similar argument may be used to show that, if there is some r for which $G^r = 0$, then the generator matrix of at least one of the equivalent xed codes will have constraint length $v <$. Since v it follows that $<$. Therefore all time-varying encoders not satisfying (8) are equivalent to xed codes with Kronecker index $<$ and therefore will have poorer distance properties. Consequently (8) is assumed throughout the remainder of this paper.

3 Catastrophicity Test

A time-varying encoder can be tested for catastrophicity by computing the gcd of the full size minors of $G(D)$, the generator matrix of the equivalent xed code.

Example 1 *Consider the time-varying code de ned by* $' = (0;1)$ *and the generator matrices*

$$G^0(D) = \begin{bmatrix} 1 + D^3 & 1 + D + D^2 \end{bmatrix} \qquad G^1(D) = \begin{bmatrix} 1 + D & 1 + D^3 \end{bmatrix}$$

The equivalent rate 2=4 *xed convolutional code has generator matrix*

$$G(D) = \begin{bmatrix} 1 & 1 + D & 1 & D \\ D^2 & D & 1 & 1 \end{bmatrix}$$

$(D) = 1$ *and hence the encoder is noncatastrophic.*

Example 2 *A time-varying code is de ned by* $' = (0;1;2)$ *and the generator matrices*

$$G^0(D) = \begin{bmatrix} 1 + D & 1 + D + D^2 \end{bmatrix} \qquad G^1(D) = \begin{bmatrix} 1 & 1 + D + D^2 \end{bmatrix}$$
$$G^2(D) = \begin{bmatrix} D & 1 + D^2 \end{bmatrix}$$

The generator matrix for the equivalent rate 3=6 *xed convolutional code is*

$$G(D) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & D & 1 & 1 & 1 & 0 \\ D & D & 0 & D & 0 & 1 \end{bmatrix}$$

In this case $(D) = 1 + D$ *and therefore the encoder is catastrophic.*

These examples demonstrate that catastrophicity is not inherited from the encoders $G^{i}(D)$, $1 \quad i \quad N$. In Example 1 both $G^{0}(D)$ and $G^{1}(D)$ are catastrophic but the time-varying encoder is noncatastrophic. The reverse is the case in Example 2. Note also that if $' = (0;2;1)$ is used in Example 2 the resulting time-varying code is noncatastrophic.

In general, the gcd method is not suitable for use in computer searches for good codes as it is not computationally e cient and the complexity grows exponentially with T. For example, a time-varying code with period $T = 4$ and $n' = 3$ requires computing 495 4 4 minors and then nding their greatest common divisor. As an alternative, Balakirsky [1] derived a necessary and su cient condition for catastrophicity based on the properties of autoregressive lters. However, it is not clear that the computational complexity of Balakirsky's test is signi cantly lower than that of the gcd method. In this section we present a new catastrophicity test for time-varying convolutional codes. A fast algorithm implementing the test is derived in Section 4.

Lemma 1 A periodic time-varying encoder with generator matrices $G^{i}(D)$, $1 \quad i \quad N$, and selection function $'(t)$ is noncatastrophic if, and only if, $G(D)$, the generator matrix for an equivalent xed code, is canonical. □

Proof. Let C be the rate $T=n'T$ code generated by $G(D)$. The low- and high-order coe cient matrices of $G(D)$ are given by

$$G_0 = \begin{bmatrix} G_0^{'(0)} & \cdots & G_{T-1}^{'(T-1)} \\ & \ddots & \vdots \\ 0 & & G_0^{'(T-1)} \end{bmatrix} \qquad G_h = \begin{bmatrix} 0 & I & & 0 \\ & \ddots & \ddots & \\ & & & I \\ I_{T-} & 0 & & \end{bmatrix} \begin{bmatrix} G^{'(0)} & & 0 \\ \vdots & \ddots & \\ G^{'(0)}_{-T+1} & \cdots & G^{'(T-1)} \end{bmatrix} \tag{14}$$

where $= Td =Te -$. Since $G_0^{'(i)} \not\equiv 0$ and $G^{'(i)} \not\equiv 0$ for all $1 \quad i < T$ it follows that both G_0 and G_h have full rank. Therefore, by statement (c) of Theorem 1, the time-varying encoder is canonical if, and only if, it is noncatastrophic. □

Thus we may test a time-varying encoder for catastrophicity using the following canonicity test for rate $k=n$ xed convolutional encoders [9, Theorem 6].

Theorem 2 Let $G(D) \; 2 \; \mathrm{F}_q^{k \quad n}$ be any generator matrix with overall constraint length v and memory v_m. Then $G(D)$ is canonical if, and only if,

$$\operatorname{rank} H \; [v] = (k+1)v \tag{15}$$

where $H \; [']$ is the $(' + v_m)k \quad 'n$ matrix

$$H \; [\,'\,] = \begin{bmatrix} G_0 & & 0 \\ \vdots & \ddots & \\ G_{v_m} & & G_0 \\ & \ddots & \vdots \\ 0 & & G_{v_m} \end{bmatrix} \tag{16}$$

□

With a view to reducing the number of computations required to determine the rank of $H[v]$ we will analyse the rank properties of this matrix in more detail. To do this we will use the following lemma due to Forney [4].

Lemma 2 Let C be any rate k/n fixed convolutional code with Kronecker indices $\{\nu_i\}_1^k$ and let C_ℓ denote set of polynomial codewords in C with degree strictly less than ℓ.

$$C_\ell := \{y(D) \in C \mid \deg y(D) < \ell;\ \mathrm{del}\, y(D) \neq 0\} \tag{17}$$

Then C_ℓ is a subspace of C over F_q with dimension

$$\dim_{\mathrm{F}_q} C_\ell = k\ell - \sum_{i:\ \nu_i > \ell} (\nu_i - \ell) \tag{18}$$

□

The rank of the matrix $H[\ell]$ is given by the following theorem.

Theorem 3 Let C be a rate k/n fixed convolutional code and let $G(D)$ be any generator matrix for C with constraint lengths $\{\nu_i\}_1^k$. Then

$$\mathrm{rank}\, H[\ell] = k\ell + \sum_{i:\ \nu_i^\perp > \ell} (\nu_i^\perp - \ell) \tag{19}$$

where $\{\nu_i^\perp\}_1^{n-k}$ are the Kronecker indices of $C^\perp$. □

Proof. Let $H(D)$ be any canonical polynomial generator matrix for $C^\perp$. Since C and $C^\perp$ are dual subspaces of $\mathrm{F}_q[D]^n$ it follows that

$$H(D)G^\prime(D) = 0 \tag{20}$$

where the dash denotes matrix transposition. Equivalently we can write $HG = 0$ where H and G are the semi-infinite block matrices

$$H = \begin{bmatrix} H_0 & H_1 & H_2 & H_3 & \cdots \\ 0 & H_0 & H_1 & H_2 & \cdots \\ 0 & 0 & H_0 & H_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{bmatrix} \qquad G = \begin{bmatrix} G_0^\prime & G_1^\prime & G_2^\prime & G_3^\prime & \cdots \\ 0 & G_0^\prime & G_1^\prime & G_2^\prime & \cdots \\ 0 & 0 & G_0^\prime & G_1^\prime & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{bmatrix} \tag{21}$$

Now let $G[\ell]$ denote the matrix consisting of the first ℓ block rows and $\ell + \nu_m$ block columns of G

$$G[\ell] = \begin{bmatrix} G_0^\prime & G_1^\prime & \cdots & G_{\nu_m}^\prime & 0 & 0 \\ & \ddots & \ddots & & \ddots & \\ 0 & 0 & G_0^\prime & G_1^\prime & \cdots & G_{\nu_m}^\prime \end{bmatrix} \tag{22}$$

The kernel of $G[\ell]$ is spanned by those rows of H that are zero in all but the first $n\ell$ columns. Since $H(D)$ is a canonical polynomial generator matrix these

rows are also a basis for $C^{?}$, the set of polynomial codewords in $C^{?}$ with degree less than $‘$. Therefore $\ker G[‘] = C^{?}$ and hence from Lemma 2

$$\dim \ker G[‘] = (n - k)‘ - \quad + \sum_{i:\ {}^{?}_{i} \quad ‘} ({}^{?}_{i} - ‘) \tag{23}$$

We may also write $\dim \ker G[‘] = n‘ - \operatorname{rank} G[‘]$. Substituting into (23) and re-arranging terms yields

$$\operatorname{rank} G[‘] = k‘ + \quad - \sum_{i:\ {}^{?}_{i} \quad ‘} ({}^{?}_{i} - ‘) \tag{24}$$

But inspection of (22) reveals that $G[‘]$ is the transpose of $H\ [‘]$. Therefore

$$\operatorname{rank} G[‘] = \operatorname{rank} H\ [‘] \tag{25}$$

Substituting the expression for $\operatorname{rank} G[‘]$ given by (24) yields the desired result. □

Combining Lemma 1 and Theorems 2 and 3 we obtain our main result.

Theorem 4 A periodic time-varying encoder with generator matrices $G^{i}(D)$, $1 \quad i \quad N$, constraint length $\ $, and selection function $’(t)$ is noncatastrophic if, and only if,

$$\operatorname{rank} H\ [‘] = T‘ + \quad ‘ \quad {}^{?}_{m} \tag{26}$$

where $G(D)$ is the generator matrix of an equivalent rate $T=n$, T xed code C and ${}^{?}_{m}$ is the largest Kronecker index of $C^{?}$. □

We note that $\frac{}{(n\ , -1)T} \quad {}^{?}_{m}$ and hence application of Theorem 4 may involve signi cantly fewer computations than computing the rank of $H\ [v]$. In the next section we will use Theorem 4 as the basis of a fast algorithm to test a time-varying encoder for catastrophicity.

4 Fast Algorithm

A computationally e cient algorithm to implement the catastrophicity test of Theorem 4 can be obtained by exploiting the banded and block Toeplitz structure of the matrix $H\ [‘]$. We begin by permuting the columns of the matrices $\{G_i\}_0^{v_m}$ such that $G_0 = [G_{00} | G_{01}]$ where $G_{00}\ 2\ \mathrm{F}_q^{T\ \ T}$ is a nonsingular upper triangular matrix. This is easy to do since there is at least one non-zero element in each of the block matrices on the diagonal of G_0. Multiplying the block rows of $H\ [‘]$ by G_{00}^{-1} and re-ordering columns yields the matrix

$$\hat{H}[\ell] = \left[\begin{array}{ccc|ccc} I & & 0 & S_0 & & 0 \\ R_1 & \ddots & & S_1 & \ddots & \\ \vdots & & I & \vdots & & S_0 \\ R_{\nu_m} & & R_1 & S_{\nu_m} & & S_1 \\ & \ddots & \vdots & & \ddots & \vdots \\ 0 & & R_{\nu_m} & 0 & & S_{\nu_m} \end{array}\right] \tag{27}$$

where $[R_i \mid S_i] = G_{00}^{-1} G_i$, $R_i \in \mathbb{F}_q^{T \times T}$, and $\nu_m = d - Te$. Using elementary row operations $\hat{H}[\ell]$ can be put in the form

$$\left[\begin{array}{ccc|ccc} I & & 0 & S_0 & & 0 \\ & \ddots & & \vdots & \ddots & \\ 0 & & I & S_{\ell-1} & \cdots & S_0 \\ \hline 0 & \cdots & 0 & S_\ell & \cdots & S_1 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & S_{\ell+\nu_m-1} & \cdots & S_{\nu_m} \end{array}\right] \tag{28}$$

where $S_i \in \mathbb{F}_q^{T \times (n-1)T}$, $0 \le i < \ell + \nu_m$. It can be shown that the matrices $\{S_i\}$ are given by the recursion formula

$$S_i = S_i + \sum_{j=0}^{i-1} R_{i-j} S_j \tag{29}$$

Note that $R_i = 0$, $\forall\, i > \nu_m$ and hence the computation of S_i requires at most ν_m matrix products. Having computed the $\{S_i\}$ we form the matrix

$$W[\ell] = W_1 W_2 \cdots W_\ell \tag{30}$$

where

$$W_i = \begin{bmatrix} S_i \\ \vdots \\ S_{i+\nu_m-1} \end{bmatrix} \tag{31}$$

From (28) it is easily seen that $\operatorname{rank} W[\ell] = \operatorname{rank} H[\ell] - T\ell$. Substituting the expression for $\operatorname{rank} H[\ell]$ given by (19) yields

$$\operatorname{rank} W[\ell] = -\sum_{i:\, i^* < \ell} (i^* - \ell) \tag{32}$$

and hence by Theorem 4 the encoder is non-catastrophic if, and only if,

$$\operatorname{rank} W[\ell] = \ell\, m^* \tag{33}$$

However, we have no a priori knowledge of $\stackrel{?}{m}$. This difficulty may be circumvented by noting that

$$\operatorname{rank} W[\ell + 1] - \operatorname{rank} W[\ell] \quad 0 \tag{34}$$

where the equality holds if, and only if, $\ell \quad \stackrel{?}{m}$. Hence we compute $\operatorname{rank} W[\ell]$ for $\ell = 1, 2, \ldots$ until either (i) $\operatorname{rank} W[\ell] =$ or (ii) $\operatorname{rank} W[\ell] - \operatorname{rank} W^{(\ell-1)} = 0$. In both cases $G(D)$ is non-catastrophic if, and only if, $\operatorname{rank} W[\ell] =$.

Finally, we may compute $\operatorname{rank} W[\ell]$, $\ell = 1, 2, \ldots,$ as follows. Using elementary column operations put $W[\ell]$ in column echelon form, denoted here by $W_c[\ell]$. The rank of $W_c[\ell]$ is determined by inspection. $W_c^{[l]}\ell + 1]$ is easily found from the augmented matrix $[W_c[\ell] \mid W_{\ell+1}]$ where $W_{\ell+1}$ is given by (31). The complete algorithm is summarised as follows:

Step 1 From the generator matrices $G^{i}(D)$, $1 \quad i \quad N$, and the function $'(t)$, construct the coefficient matrices $\{G_i g_0^{V_m}$.
Step 2 Compute G_{00}^{-1} and the matrices $\{R_i g_1^{V_m}$ and $\{S_i g_0^{V_m}$.
Step 3 for $\ell = 1$ to
  Compute W_ℓ using the recursion formula (29).
  Using elementary column operations obtain $W_c[\ell]$
  from $W_c[\ell - 1]$ and W_ℓ.
  if $(\operatorname{rank} W_c[\ell] - \operatorname{rank} W_c[\ell - 1] = 0)$ or $(\operatorname{rank} W_c[\ell] =)$
  then goto end
 next ℓ
Step 4 end. The encoder is noncatastrophic if $\operatorname{rank} W_c =$

4.1 Computational Complexity

For simplicity we assume binary codes. Computing the matrices $\{S_i\}$ requires $O(\stackrel{?}{m}\ T^2 n_\ell)$ binary operations. Computation of the rank of $W^{(\stackrel{?}{m})}$ requires $O(\stackrel{?}{m}\ ^2 T n_\ell)$ binary operations. Typically will be greater than T and consequently this latter step dominates the overall computational complexity of the algorithm.

5 Conclusions

We have presented a new algorithm for identifying rate $1/n_\ell$ catastrophic time-varying convolutional encoders. The algorithm requires no polynomial operations has a simple software implementation. The computational complexity is $O(\stackrel{?}{m} T n_\ell\ ^2)$ and is less complex than the algorithm proposed by Balakirsky. Furthermore, the algorithm presented here is easily generalised to rate k/n_ℓ time-varying codes.

References

1. V.B. Balakirsky, \A necessary and su cient condition for time-variant convolutional encoders to be noncatastrophic," *Lecture Notes in Computer Science*, No. 781, pp. 1-10, Springer-Verlag, 1993.
2. J.L. Massey and M.K. Sain, \Inverses of linear sequential circuits," *IEEE Trans. Computers*, Vol. C-17, No. 4, pp. 330-337, April 1968.
3. R.R. Olsen, \Note on feedforward inverses for linear sequential circuits," *IEEE Trans. Computers*, Vol. C-19, No. 12, pp. 1216-1221, Dec. 1970.
4. G.D. Forney, Jr., \Minimal bases of rational vector spaces, with applications to multivariable linear systems," *SIAM J. Control*, vol. 13, pp.493-520, May 1975.
5. R. Johannesson and Z.-X. Wan, \A linear algebra approach to convolutional encoders," *IEEE Trans. Inform. Theory*, vol. IT-39, No. 4, pp. 1219-1233, July 1993.
6. P.J. Lee, \There are many good time-varying convolutional codes," *IEEE Trans. Inform. Theory*, Vol. IT-35, No. 2, pp. 460-463, March 1989.
7. M. Mooser, \Some periodic convolutional codes better than any xed code," *IEEE Trans. Inform. Theory*, Vol. IT-29, No. 5, pp. 750-751, Sept. 1983.
8. R. Palazzo, \A time-varying convolutional encoder better than the best time-invariant encoder," *IEEE Trans. Inform. Theory*, Vol IT-39, No.3, pp. 1109-1110, May 1993.
9. C. O'Donoghue, and C.J. Burkley, \Minimality and canonicity tests for rational generator matrices for convolutional codes," in *Proc. 1998 IEEE Information Theory Workshop*, pp. 112-114, Killarney, 22-26 June, 1998.

Low Complexity Soft-Decision Sequential Decoding Using Hybrid Permutation for Reed-Solomon Codes

Min-seok Oh[1] and Peter Sweeney[2]

[1] CCSR, University of Surrey, Guildford, Surrey, GU2 5XH, UK
m.oh@ee.surrey.ac.uk
[2] CCSR, University of Surrey, Guildford, Surrey, GU2 5XH, UK
p.sweeney@ee.surrey.ac.uk

Abstract. We present a soft-decision decoding method for Reed-Solomon codes (RS codes) using both cyclic and squaring permutations. These permutations are used to provide a convenient sequence which is predicted to have relatively low complex error pattern with respect to a modified Fano sequential algorithm[1]. In order to preserve bit-level soft-decision values, each sequence of those permutation groups must keep equal weight distribution in symbol and bit level. Trellis construction is based on Wolf's method[2] and a binary systematic parity check matrix of RS codes is used for bit-level decoding[9]. In simulation results, it is shown that a hybrid of those two permutations can be used for low complexity decoding approaching maximum likelihood performance.

1 Introduction

Since Reed-Solomon codes[3] were introduced in 1960, many decoding methods have been developed. However, soft-decision decoding method could not be easily implemented because of complexity problem. Chase[5] and Forney[4] introduced interesting methods for soft-decision decoding of block codes. Their algorithms have tradeoffs between complexity and decoding performance for the application to RS codes. Later, some other approaches using trellis structure were developed and demonstrated some good results [6][7]. Despite such achievements, they did not fully use bit-level soft-decision information and could not solve complexity problem for long RS codes with a large field.

For the bit-level soft-decision decoding for RS codes, Vardy[8] presented a method using a union of costes being an interleaver of several binary BCH codes for representation of RS codes. Recently Oh and Sweeney presented another relatively simple method[9] which employs bit-level soft-decision information with low complexity. In this method a modified Fano algorithm was used with cyclic permutation of RS codes. In this work, although cyclic permutation contributes to considerable complexity reduction showing near-maximum likelihood performance(ML), it was not effective in decoding of a received sequence with widely distributed errors since a sequence given by a cyclic shift has a similar error pattern to the original one. In

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 163-172, 1999.

order to deal with this kind of error pattern, squaring permutation[13][16] can be useful, since it can provide a different set of sequences compared with the cyclic permutation. However, squaring permutation is generally inferior to the cyclic permutation because of smaller size of permutation group.

In this paper, we present hybrid permutation which is a combination of cyclic and squaring permutations. For (*n, k*) RS codes over $GF(2^m)$, since cyclic and squaring permutation generate *n* and *m* different sequences respectively, the hybrid permutation provides $m \cdot n$ different sequences from an original sequence. With this permutation, a sequential decoder can reduce complexity by performing a convenient sequence-first decoding.

Complexity characteristics of sequential decoding was well studied in [10][11]12]. In general the complexity depends on the error bits and error location in information block, it is reasonable to regard the sequence with the most reliable information block as the most convenient for sequential decoding. In this paper, we use a criterion which is represented by the sum of symbol confidences within information block of each sequence of permutation group. The symbol confidence is taken as the worst bit confidence within each symbol, since the decoding complexity will be affected by the worst one.

In section 2, we describe three permutation groups for RS codes: cyclic, squaring and hybrid permutation. Then, in section 3, we present *hybrid permutation sequential decoding* (HPSD) to achieve near-ML performance at reasonable complexity cost. Section 4 shows simulation results for HPSD in terms of error correcting performance and complexity.

2 Symbol Permutation of RS Codes

Permutation groups of RS codes provide many equivalent sequences which are useful for a low complexity decoding. When a received sequence contains some error, a permutation of the sequence may give a desirable effect that each permuted sequence can have different complexity due to changing the location of error bits. In this section we discuss three permutation techniques for RS codes: *cyclic*, *squaring* and *hybrid*.

2.1 Cyclic Permutation

We consider (*n, k*) Reed-Solomon codes over $GF(2^m)$ and denote a code word denote $c(x)$ as

$$c(x) = \sum_{i=0}^{n-1} c_i \cdot x^i \quad \text{for } c_i \in GF(2^m) . \tag{1}$$

The elements of *n* cyclic permutation group $T_c(c(x))$ are

$$\left(\sum_{i=0}^{n-1} c_i \cdot x^{i+\beta}\right) \bmod\left(x^n - 1\right) \text{ for } \beta \in (0,1,2,\cdots,n-1). \tag{2}$$

By cyclic permutation of a code word, *n* different sequences are obtained and each sequence is also a code word in which the bits and confidences are also shifted with. Therefore a certain error pattern of a received sequence is changed by the cyclic permutation. A decoder can firstly choose the sequence with the most convenient error pattern among *n* possible sequences. A decoding method using this kind of permutation has been shown in[9].

2.2 Squaring Permutation

Squaring permutation is a technique using the property that although the squaring of a code word polynomial changes the position of symbols constituting the code word, the squared result is also a code word. For (*n, k*) Reed-Solomon codes over $GF(2^m)$, we can get *m*-different sequences which have the different error pattern.

A decoding approach using squaring permutation decoding was previously described[13] based on algebraic decoding method. However, we need further consideration for the application to a bit-level sequential decoding since bit-level soft decision information should be preserved through squaring operation.

In this paper, each symbol for RS code is represented on *normal basis*[14] which is defined as a set of linearly independent roots with the form $\{\lambda_0, \lambda_1, \cdots, \lambda_{m-1}\}$ for $\lambda_i = \beta^{2^i}$. On this basis, since the result of squaring of each symbol is represented by just a cyclic shift, the bit level soft-decision can be completely preserved through the squaring process. Perlis[15] has shown that a necessary and sufficient condition for a normal basis such as

$$tr(\beta) = \sum_{i=1}^{m-1} \beta^{2^i} = 1. \tag{3}$$

Table 1. shows a normal basis satisfying the above condition.

Tabel 1. Basis Representation for RS codes

Field	$GF(2^3)$	$GF(2^4)$	$GF(2^5)$
Polynomial basis	$\alpha^2\alpha^1\alpha^0$	$\alpha^3\alpha^2\alpha^1\alpha^0$	$\alpha^4\alpha^3\alpha^2\alpha^1\alpha^0$
Normal basis	$\alpha^5\alpha^6\alpha^3$	$\alpha^9\alpha^{12}\alpha^6\alpha^3$	$\alpha^{17}\alpha^{24}\alpha^{12}\alpha^6\alpha^3$

The elements of the *m* squaring permutation group T_s for $s = 1,2,\cdots,m-1$. are

$$\left(c(x) = \sum_{i=0}^{n-1} c_i x^i\right)^{2^s} = \sum_{i=0}^{n-1} c_i^{2^s} x^{(2^s \cdot i) \bmod n} \tag{4}$$

Let c_i denote $c_i = \sum_{j=0}^{m-1} a_j \lambda_j$ for $a_j \in GF(2)$ on the normal basis, the coefficient $c_i^{2^s}$ is expressed by $c_i^{2^s} = \sum_{j=0}^{m-1} a_j \lambda_{(j+s) \bmod m}$. Consequently since squaring permutation using normal basis can preserve bit-level soft-decision values through the squaring operation, we can apply this technique to our bit-level sequential decoding. Moreover squaring of a symbol can be simply obtained by a bit cyclic shift within the symbol.

2.3 Hybrid Permutation

Hybrid permutation is the combination of the cyclic and squaring permutation. As we have examined in the previous section, since for (*n*, *k*) RS codes over $GF(2^m)$ the *m* squaring sequences can have *n* cyclic permuted sequences: a total of $n \times m$ permuted sequences can be obtained by combining the permutations. This means that an error pattern from a received sequence is also changed to other patterns with $n \times m$ different complexity. For RS codes with a large field, the number of possible sequences increases. Table 1 shows the number of possible sequences by the hybrid-permutation.

Table 1. Possible Sequences by Hybrid Permutation

Field	8	16	32	64
Possible Sequences	21	60	155	378

Fig.1 shows error pattern changes by hybrid-permutation for (15,9) RS codes. In the figure, it is shown that four sequences are obtained from squaring permutation and then 15 cyclic sequences are produced each corresponding to each one of the squaring permutation group. Thus total 60 sequences can be obtained from a received sequence and each sequence has the same symbol and binary weights as the original sequence because of the use of normal basis. The different thing in each sequence is the order of listed symbols. Since the complexity of the sequential decoder depends on the locations of errors, it is expected that the complexity of decoding the original sequence can also be changed with the permutation.

Hybrid permutation is very attractive to design an efficient permutation sequential decoder, since the decoder can choose the most convenient sequence from a greater variety of sequences than either of the cyclic or squaring permutation individually. This permutation gives a solution for individual drawbacks of cyclic and squaring techniques which are used for permutation decoding. Widespread errors can be rearranged by the squaring permutation so that cyclic permutation can effectively manage the rearranged sequence. Therefore we can improve the complexity and decoding performance simultaneously. In particular, in the application of RS codes over a large Galois field, hybrid permutation will be very powerful in reducing complexity and improving decoding performance.

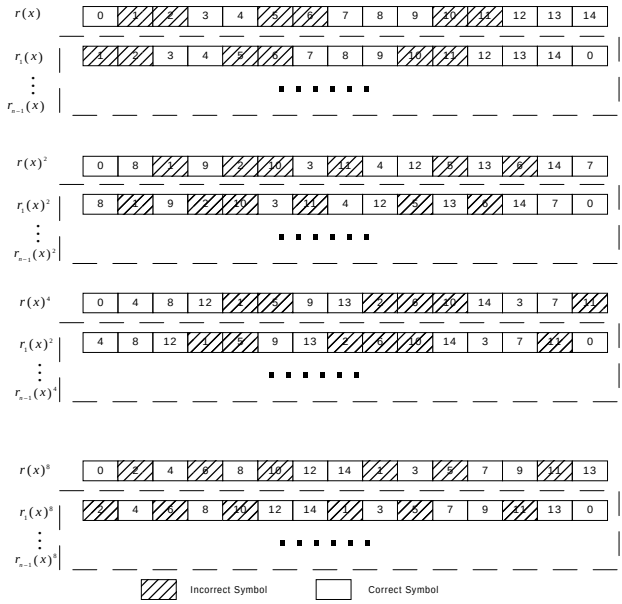


Fig. 1. Error Pattern Changes by Hybrid Permutation

3 Hybrid Permutation Sequential Decoder

We present *hybrid permutation sequential decoder* (HPSD) which uses a modified Fano algorithm with hybrid permutation. The *modified Fano algorithm*[9] has two additional functions, which are *path update function* and *decision rule*, to the original Fano algorithm[8]. The path update function updates a searched path whenever its path metric is greater than current one. By the path updating, the decoder can release the best path. On the other hand, the *decision rule* is to qualify searched paths in the case that the decoder has searched for a wrong path as if it were the correct path. With those two additional functions, the modified Fano algorithm approaches maximum likelihood performance only if the decoder has tried the correct path at least once.

For the efficient operation of the sequential decoding, the decoding parameters are optimized as the most proper value for computational limit L and threshold spacing step, ΔT. For the convenient sequence-first search, the decoder considers the convenience level as confidences with respect to the information part of a code word for the possible permuted sequences by hybrid-permutation. Those sequences are sorted by the sum of confidences of the information part and their priorities are assigned for decoding. The decoding procedure is explained as following:

(i) Obtain $m \times n$ candidates by cyclic and squaring permutation.
(ii) Assign the decoding priority of the candidates in order of IBC(information block confidence) of each candidate. Set *trial number* to 1.

(iii) Choose the sequence with the highest priority, which has the largest IBC.
(iv) Decode the chosen sequence by using the modified Fano algorithm (MFA).
(v) Check decoding result.
- If the decoder has found a valid path satisfying the *decision rule*, release the path and restore its sequence order.
- Otherwise store the best path which has been recorded so far by *the path update function*. Then go to the next step.

(vi) Increase *trial number*.
- If *trial number* is less than a given *maximum-trial-number*, choose the sequence with next priority and then go to step (iv).
- Otherwise release the best path which has been recorded so far and then restore the permuted sequence with respect to the path.

Fig.2. shows the flow chart of the hybrid permutation sequential decoder.

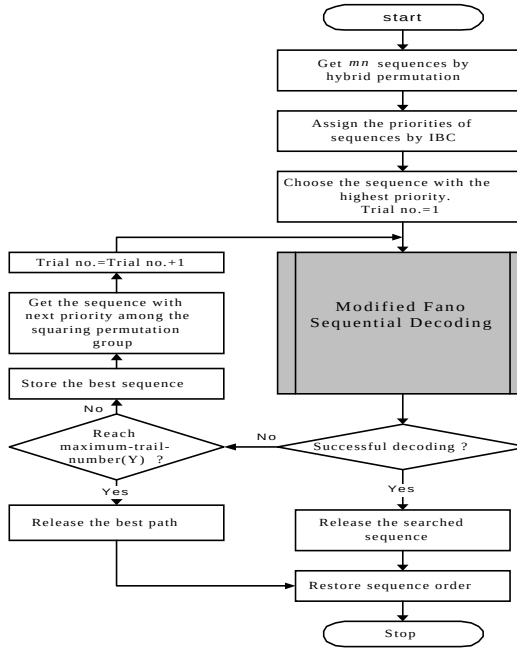


Fig.2. Hybrid Permutation Sequential Decoder

4 Simulation Results

Simulation was carried out on BPSK system with 8-level soft decision values over Gaussian channel. Decoding performance was obtained for (7,3), (15,9) and (31,27) RS codes in terms of complexity and decoding error rate as a function of Eb/No. The complexity was measured by average computations per information bit and error correcting performance was calculated by bit error rate(BER) with respect to Eb/No. For comparison with non permutation sequential decoding (NPSD), an equal value of

overall computational limit L was used for a same class of RS codes. The maximum number of trial, Y, in HPSD has the relation as

$$L = L_c \cdot Y \tag{5}$$

where L_c is a computational limit per each trial.

Fig.3 and Fig.4 are the comparison between Viterbi and *hybrid permutation sequential decoding* (HPSD) for (7,3) RS codes. In the figure, we can see that the decoding performance of HPSD was almost equal to that of Viterbi decoding. On the other hand, in Fig.4, the complexity of HPSD was much lower and it rapidly decreased as Eb/No increased. Thus it is well verified that the HSPD is very efficient decoding method achieving ML performance for (7,3) RS codes.

Fig.5 is the decoding performance comparison between non permutation decoding and hybrid permutation decoding. The hybrid permutation decoding produced considerable coding gain for (15,9) and (31,27) RS codes. Moreover more gain has been achieved for (31, 27) RS codes.

Fig.6 is the complexity comparison between HPSD and NPSD. In the figure, we can see that HPSD provides considerably low complexity for (7,3), (15,9) and (31,27) RS codes. In particular, when we consider the result obtained in Fig.5, the most cost-effective performance has been achieved for (31,27) RS codes. That is, HPSD provided around 1.0 dB gain with 1/3 complexity compared with NPSD. This results from the fact that more permutation group are available for (31, 27) RS codes than other two RS codes. Therefore HPSD will be more effective for long RS codes with large Galois field.

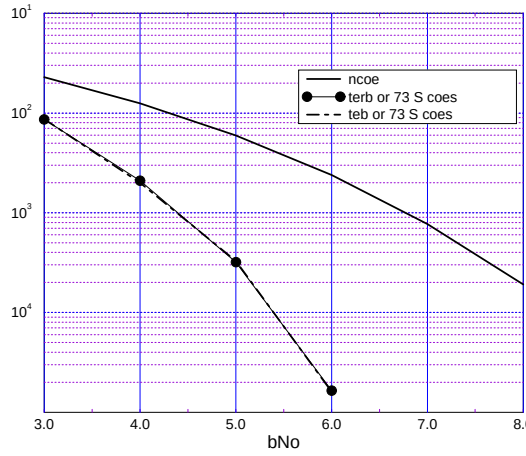


Fig.3. Performance Comparison with Viterbi decoder

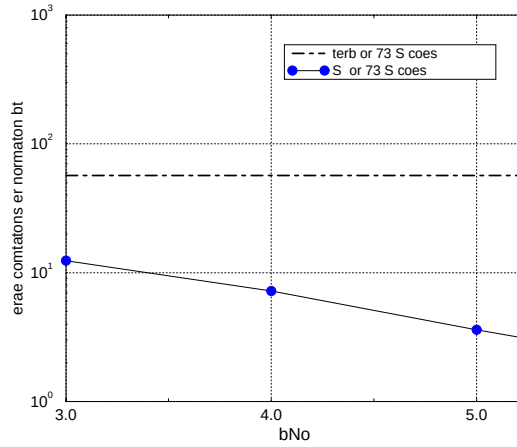


Fig.4. Complexity Comparison with Viterbi decodinG

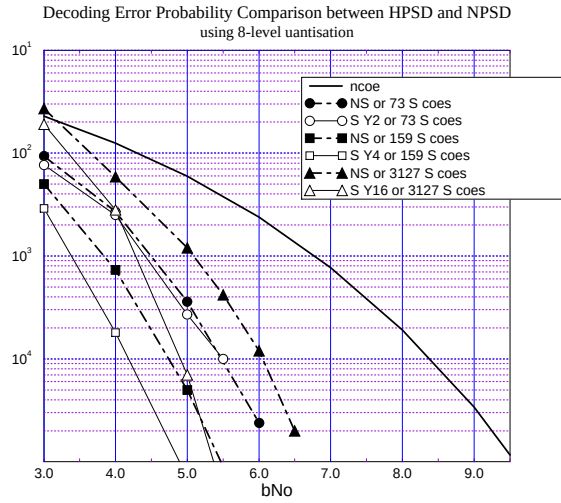


Fig.5. Decoding Performance by Permutations

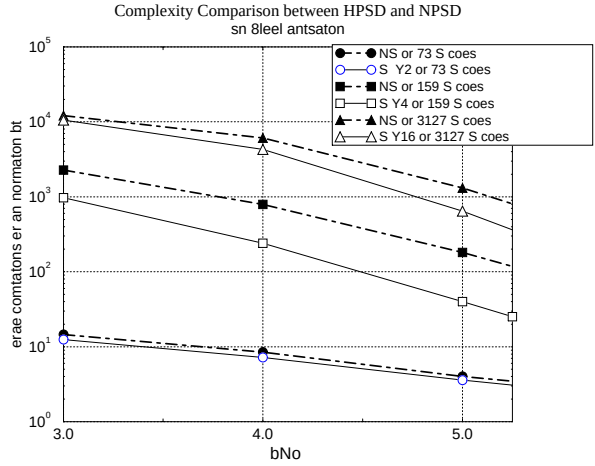


Fig.6. Complexity Comparison by Permutations

5 Conclusion

The use of the hybrid permutation gives a great improvement in decoding complexity and decoding performance. Since the complexity of the sequential decoding depends on the efficiency to search for the correct path at a given computational limit, the hybrid permutation decoding is very proper to drive the searching region of the decoder to the most likely one. Thus if the correct path has been tried at least once, this HPSD will always produce the maximum likelihood performance at low complexity. Furthermore this hybrid permutation can be useful to design a low complexity decoding for any block codes where the cyclic and squaring permutation are available.

References

1. Fano, R.: A Heuristic Discussion of Probabilistic Decoding. IEEE Trans. Inform. Theory. **IT-9**. (1963) 64-74
2. Wolf, J. K.: Efficient maximum likelihood decoding of linear block codes using a trellis. IEEE Trans. Inform. Theory. **IT-20** (1978) 76-80
3. Reed, I.S. and Solomon, G.: Polynomial codes over certain finite fields," SIAM Journal on Applied Mathematics. **8** (1960) 300-304
4. Forney, G.D.: Generalized minimum distance decoding. IEEE Trans. Inform. Theory, **IT-12** (1966) 125-131
5. Chase, D.: A class of algorithm for decoding clock codes with channel measurement information. IEEE Trans. Inform. Theory. **IT-18** (1972) 170-182,

6. Shin, S.: Trellis decoding of Reed-Solomon codes. Ph.D. Thesis, (1994)
7. Matis, K.R. and Modestino, J.W.: Reduced-search soft-decision trellis decoding of linear block codes. IEEE Trans. Inform. Theory. **39** (1991) 440-444
8. Vardy, A. and Be'ery, Y.: Bit level soft-decision decoding of Reed-Solomon codes. IEEE Trans. Inform. Theory. **IT-28** (1982) 349-355
9. Oh, M. and Sweeney, P.: Bit-level soft decision sequential decoding for RS codes. WCC'99. (1999) 111-120
10. Jacob, I. and Berlekamp, E..: A lower bound to the distribution of communication for Sequential Decoding. EEE Trans. Inform. Theory. **IT-13** (1967) 167 - 174
11. Savage, J.: The distribution of the sequential decoding computational time. IEEE Trans. Inform. Theory. **IT-12** (1966) 143- 147
12. Anderson, J.: Sequential decoding based on an error criterion. IEEE Trans. Inform. Theory. **40** (1994) 546 - 554
13. Martin, I., Honary, B., and Farrell, P.G.: Modified minimum weight decoding for RS codes. ELECTRONICS LETTERS. **31** (1995) 713-714.
14. Pei, D., Wang, C., and Omura, J.: Normal basis of finite field $GF(2^m)$. IEEE Trans. Inform. Theory. **IT-32** (1986) 285-287
15. Perlis, S.: Normal basis of cyclic fields of prime-power degree. Duke Math. J. **9** (1942) 507-517
16. Oh, M, and Sweeney, P.: Squaring permutation sequential decoding on normal basis for RS codes. ELECTRONICS LETTERS. **35** (1999) 1325-1326

On E cient Decoding of Alternant Codes over a Commutative Ring?

Graham H. Norton and Ana Salagean

Algebraic Coding Research Group, Centre for Communications Research
University of Bristol, U.K.
Graham.Norton@Bristol.ac.uk, Ana.Salagean@ntu.ac.uk

1 Introduction

Let R be a commutative ring e.g. the domain of p-adic integers or a Galois ring. We de ne alternant codes over R, which includes BCH and Reed-Solomon codes. We also de ne a corresponding key equation and concentrate on decoding alternant codes when R is a domain or a local ring. Our approach is based on minimal realization (MR) of a nite sequence [4,5], which is related to rational approximation and shortest linear recurrences. The resulting algorithms have quadratic complexity.

When R is a domain, the error-locator polynomial is the unique monic minimal polynomial of the nite syndrome sequence (Theorem 2), and can be easily obtained using Algorithm MR of [4] (which is division-free). The error locations and magnitudes can then be computed as over a eld. In this way we can e ciently decode any alternant code over a domain.

Recall that a Hensel ring is a local ring which admits Hensel lifting. (It is well-known that a nite local ring, such as a Galois ring, is a Hensel ring.) We characterize the *set* of monic minimal polynomials of a nite syndrome sequence over a Hensel ring (Theorem 3). It turns out that the monic minimal polynomials coincide modulo the maximal ideal M of R (Theorem 4) when R is a local ring. This yields an e cient new decoding algorithm (Algorithm 1) for alternant codes over a local ring R, once a monic minimal polynomial of the syndrome sequence is known. For determining the error locations, it is enough to nd the roots of the image of any such monic minimal polynomial in the residue eld R/M. After determining the error locations, the error magnitudes can be easily computed.

When R is a nite chain ring (e.g. a Galois ring) we invoke Algorithm MP of [5] to nd a monic minimal polynomial.

We note that a modi cation of the Berlekamp-Massey algorithm for $\mathbb{Z}_m$ was given in [8], where it was claimed [*loc. cit.*, Introduction] (without proof) to decode BCH codes de ned over the integers modulo m. An algorithm to decode BCH and Reed-Solomon codes over a Galois ring has also been given in [3]. However this algorithm may require some searching see [*loc. cit.*, Conclusions,

? Research supported in part by the U.K. Engineering and Physical Sciences Research Council under Grant L07680. The second author is now with Department of Mathematics, Nottingham Trent University, UK.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 173{178, 1999.

p. 1019] and their decoding algorithm requires root- nding in R itself, which is also less e cient.

For more details and proofs, we refer the reader to [7].

2 Alternant Codes over a Commutative Ring

Let R be a commutative ring with $1 \neq 0$ and let $N(R)$ denote the subset of R consisting of all elements which are *not* zero-divisors.

The following de nition of alternant codes over R generalises the de nition over elds.

De nition 1 (Alternant codes) *Let T be a subring of R and $d \geq 2$. Suppose that $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $y = (y_1, \ldots, y_n)$ are such that $y_i \in N(R)$ and $\alpha_i - \alpha_j \in N(R)$ for $1 \leq i < j \leq n$. If*

$$H = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \\ y_1\alpha_1 & y_2\alpha_2 & \cdots & y_n\alpha_n \\ y_1\alpha_1^2 & y_2\alpha_2^2 & \cdots & y_n\alpha_n^2 \\ \vdots & \vdots & & \vdots \\ y_1\alpha_1^{d-2} & y_2\alpha_2^{d-2} & \cdots & y_n\alpha_n^{d-2} \end{bmatrix} \tag{1}$$

then the alternant code of length n and alphabet T de ned by H is the T-module

$$A(\alpha, y, d) = \{c \in T^n : Hc^{\mathrm{tr}} = 0\}.$$

As usual, H is called the parity check matrix of $A(\alpha, y, d)$.

As in the case of elds, we have:

Theorem 1 *The minimum Hamming distance of $A(\alpha, y, d)$ is at least d.*

3 A Key Equation

For decoding alternant codes over a ring we follow the main steps for their algebraic decoding over a nite eld, except that we rely on minimal realization of a nite sequence which was introduced in [4]. For some advantages of the minimal realization approach, see [5, Introduction]. See also the expository account in [6], especially *loc. cit.* Section 8, which discusses the application to a nite sequence of syndromes over a nite eld.

Suppose that a codeword $c \in A(\alpha, y, d)$ is received as $r = c + e$. We have to nd the error vector e given the syndrome vector $Hr^{\mathrm{tr}} = He^{\mathrm{tr}}$.

We will henceforth assume that $d = 2t + 1 \geq 3$ and that the number of errors is $w = wt_H(e) \leq t$. Let $i_1, i_2, \ldots, i_w$ be the positions of the errors. As usual, $\alpha_{i_1}, \ldots, \alpha_{i_w}$ are called the error locations and $e_{i_1}, \ldots, e_{i_w}$ the error magnitudes. We write m for $1 - 2t$; note that $m \geq -1$.

De nition 2 (Syndrome sequence) *The syndrome sequence of the error e is the nite sequence $s_0; s_{-1}; \ldots; s_m$ over R, denoted sjm and de ned by:*

$$s_i = \sum_{k=1}^{n} e_k y_k \, {}_k^{-i} = \sum_{j=1}^{w} e_{i_j} y_{i_j} \, {}_{i_j}^{-i}:$$

for $i = 0; -1; \ldots; m$.

De nition 3 (Error polynomials) *We de ne the error-locator and error-evaluator polynomials by*

$$_e = \prod_{j=1}^{w} (X - {}_{i_j}) \text{ and } !_e = \sum_{j=1}^{w} e_{i_j} y_{i_j} \prod_{\substack{k=1;\ldots;w \\ k \neq j}} (X - {}_{i_k}):$$

Note that in the classical literature $_e$ and $X^{\deg(\ _e)-1-\deg(!_e)} !_e$ are called the error-locator and the error-evaluator polynomial respectively, where f denotes the reciprocal of $f \in R[X]$.

De nition 4 (Key equation) *Let $\Gamma = \sum_{i=m}^{0} s_i X^i \in R[X^{-1}]$. We say that $(f; h) \in R[X] \quad XR[X]$ is a solution of the key equation if f is monic,* $\deg(h) \quad \deg(f) \quad -m$ *and*

$$\Gamma \quad h/f \bmod X^{m-1}: \tag{2}$$

A solution $(f; h)$ is called minimal *if $\deg(f)$ is minimal.*

As in the classical case we easily obtain:

Proposition 1 *If $w \quad t$ then $(\ _e; X\,!_e)$ is a solution of the key equation.*

The minimality of the solution $(\ _e; X\,!_e)$ is not obvious, but will follow from Theorem 2 when R is a domain and from Theorem 4 when R is local.

We now recall some de nitions from [5]. For $f \in R[X]$ and $G \in R[X^{-1}]$, $f \quad G$ denotes their product in $R[X^{-1}; X]$ and $(f \quad G)_j$ is the coe cient of X^j in $f \quad G$. We write $\mathrm{lc}(f)$ for the leading coe cient of $f \in R[X] \cap f0g$.

De nition 5 *([5]) Let $r \in R \cap f0g$. The r{annihilator set of sjm is*

$$\mathrm{Ann}(sjm; r) = \{f \ : \ \mathrm{lc}(f) = r; \ (f \quad \Gamma)_j = 0 \text{ for } m + \deg(f) \quad j \quad 0g:$$

A polynomial f is said to be an *annihilating polynomial of the sequence sjm* if $f \in \mathrm{Ann}(sjm; r)$ for some r.

A non-zero polynomial in $\mathrm{Ann}(sjm; r)$ of minimal degree is called *a minimal polynomial of the sequence sjm*, and we write $\mathrm{Min}(sjm; r)$ for those minimal polynomials of sjm with leading coe cient r. (For the equivalence between minimal polynomials and shortest linear recurrences of a nite sequence, see [6, Corollary 2.3], which is valid for any R.)

Recall from [4] that for $f \in R[X]$, $(f; sjm) \in XR[X]$ is defined by

$$(f; sjm) = \sum_{j=1}^{\deg(f)} (f \quad \Gamma)_j X^j :$$

The connection between the key equation and minimal polynomials of sjm becomes clear from the following lemma:

Lemma 1 *The pair* $(f; h \in R[X] \quad XR[X]$ *is a minimal solution of the key equation (2) if and only if* $\deg(f) \quad -m;\ f \in \mathrm{Min}(sjm; 1)$ *and* $h = \quad (f; sjm)$.

4 Decoding over a Domain

Theorem 2 *If R is a domain then for all $r \in R \setminus \{0\}$,* $\mathrm{Min}(sjm; r) = \{r \quad _e\}$.

We can now develop a decoding algorithm for alternant codes over a domain. Algorithm MR of [4] computes a minimal polynomial f and the corresponding $(f; sjm)$ for *any sequence sjm over a domain*. But from Theorem 2, we know that for a syndrome sequence, such a polynomial f must be the error locator polynomial multiplied by some non-zero constant. Hence, after applying Algorithm MR to the sequence of syndromes, we simply divide the output polynomials f and $(f; sjm)$ by the leading coefficient of f, thus obtaining $_e$ and $X \quad !_e$. The algorithm has quadratic complexity. We then proceed as in the classical (field) case: we compute the error locations as the roots of $_e$ (which are of the form $_{i_1}; : : : ; \ _{i_w}$) and the error magnitudes as $e_{i_j} = \quad !_e(\ _{i_j})/(\ ^{0}_{e}(\ _{i_j}) y_{i_j})$.

This algorithm can decode, in particular, BCH and Reed-Solomon codes over the p-adic integers of [1].

5 Decoding over a Local Ring

We now assume that R is a local ring with maximal ideal M and residue field $K = R/M$. We extend the canonical projection $R \quad ! \quad K$ to a projection $R[X] \quad ! \quad K[X]$ and denote the image of $f \in R[X]$ under this projection by $\overline{f}$.

When R is a Hensel ring we can characterize the monic minimal polynomials of the syndrome sequence:

Theorem 3 *If R is a Hensel ring and $\overline{\ }_1; : : : ; \overline{\ }_n$ are distinct then*

$$\mathrm{Min}(sjm;) = \left\{ \prod_{j=1}^{w} (X - \ _{i_j} - z_j) : z_j e_{i_j} y_{i_j} = 0 \text{ for some } z_j \in R; j = 1; : : : ; w \right\} :$$

Our decoding algorithm is based on the following result:

Theorem 4 *If R is a local ring and $\alpha_1, \ldots, \alpha_n$ are distinct then $_e$ 2 Min(sjm; 1) and for any 2 Min(sjm; 1) we have*

$$\alpha = \alpha_e = \prod_{j=1}^{\Psi} (X - \alpha_{i_j}).$$

We can now develop a decoding algorithm for alternant codes over a local ring, provided we have an algorithm that computes a monic minimal polynomial for sjm. The latter can be achieved for sequences of syndromes of BCH and Reed-Solomon codes over $\mathbb{Z}_{p^a}$ (see [3], [8]), over nite local commutative rings (see [2]) and for any sequence over a nite chain ring (see [5]). A method of computing the error once we have a monic minimal polynomial f is discussed in [2,3]: (i) the roots of f in R are found and (ii) the ones that di er from some $_i$ by a zero-divisor are selected. Our method searches for the roots of $\overline{f}$ 2 $K[X]$ among $\alpha_1, \ldots, \alpha_n$ and is therefore more e cient.

Algorithm 1 (Decoding $\mathcal{A}(\ ; y; d)$ over a local ring)
Input: $r = (r_1, \ldots, r_n)$ containing at most $t = (d-1)/2$ errors, where t 1.
Output: $c = (c_1, \ldots, c_n)$, the nearest codeword.

0. Let $m = 1 - 2t$.
1. Compute the syndrome sequence sjm as $(s_0\ s_{-1}\ \cdots\ s_m)^{\mathrm{tr}} = Hr^{\mathrm{tr}}$. If $sjm = (0, \ldots, 0)$, return r.
2. Compute a monic minimal polynomial for the sequence sjm.
3. Compute the roots $\alpha_{i_1}, \ldots, \alpha_{i_w}$ of α in K. Then the errors occurred at positions $i_1, \ldots, i_w$.
4. Compute $_e = \prod_{j=1}^{w} (X - _{i_j})$.
5. Compute ^0_e and $!_e = (\ _e; sjm)/X$.
6. Set $e = (0, \ldots, 0)$ and for $j = 1, \ldots, w$, put $e_{i_j} = !_e(\ _{i_j})/(\ ^0_e(\ _{i_j}) y_{i_j})$. Return $r - e$.

Algorithm 1 can decode, in particular, BCH and Reed-Solomon codes over Galois rings.

Acknowledgement. The authors gratefully acknowledge nancial support from the U.K. Engineering and Physical Sciences Research Council (EPSRC). The second author was supported by EPSRC Grant L07680.

References

1. A. R. Calderbank and N. J. A. Sloane. Modular and p-adic codes. *Designs, Codes and Cryptography*, 6:21{35, 1995.
2. A. A. de Andrade and R. Palazzo, Jr. Construction and decoding of BCH codes over nite commutative rings. *Linear Algebra and its Applications*, 286:69{85, 1999.

3. J. C. Interlando, R. Palazzo, and M. Elia. On the decoding of Reed-Solomon and BCH codes over integer residue rings. *IEEE Trans. Inform. Theory*, 43(3):1013{ 1021, 1997.
4. G. H. Norton. On the minimal realizations of a nite sequence. *J. Symbolic Computation*, 20:93{115, 1995.
5. G. H. Norton. On minimal realization over a nite chain ring. *Designs, Codes and Cryptography*, 16:161{178, 1999.
6. G. H. Norton. On shortest linear recurrences. *J. Symbolic Computation*, 27:323{347, 1999.
7. G. H. Norton and A. Salagean. On the key equation over a commutative ring. *Designs, Codes and Cryptography*, 1999. To appear.
8. J. A. Reeds and N. J. A. Sloane. Shift-register synthesis (modulo *m*). *SIAM J. Computing*, 14:505{513, 1985.

Reduced Complexity Sliding Window BCJR Decoding Algorithms for Turbo Codes

Jihye Gwak[1], Sooyoung Kim Shin[2], Hyung-Myung Kim[3]

[1]Satellite Communications System Department, Electronics and Telecommunications Research Institute, 161 Kajong-Dong, Yusong-Gu, Taejon, 305-350, Korea
jihye@etri.re.kr
[2]Satellite Communications System Department, Radio & Broadcasting Technology Laboratory, ETRI, 161 Kajong-Dong, Yusong-Gu, Taejon, 305-350, Korea
dssy@satnet.etri.re.kr
[3]Department of Electrical Engineering, Korea Advanced Institute of Sience and Techonology, 373-1 Kusong-Dong, Yusong-Gu, Taejon, 305-701, Korea
hmkim@panda.kaist.ac.kr

Abstract. In decoding the turbo codes, the sliding window BCJR algorithm, derived from the BCJR algorithm, permits a continuous decoding of the coded sequence without requiring trellis termination of the constituent codes and uses reduced memory span. However, the number of computations required is greater than that of BCJR algorithm. In this paper, we propose an efficient sliding window type scheme which maintains the advantages of the conventional sliding window algorithm, reduces its computational burdens, and improves its BER performance by allowing the window to be forwarded in multi-step. Simulation results show that the proposed scheme outperforms the conventional sliding window BCJR algorithm with reduced complexity.

1 Introduction

The decoding of turbo codes is performed frame by frame assuming that the receiver knows the final states of each frame [1], [2]. It means that the turbo encoder requires trellis termination. However, the trellis termination of turbo codes is non-trivial unlike convolutional codes [3].

The sliding window (SW) BCJR algorithm for continuous decoding is proposed by Benedetto *et al.* which does not divide information bits into blocks and does not require trellis termination [2]. The SW BCJR algorithm has an advantage in the application where it requires a small delay such as speech transmission with short frames, since trellis termination usually requires another redundancy. However, the computational complexity of SW BCJR algorithm is even greater than that of the BCJR algorithm which also suffers from high computational burdens. Therefore it is essential to reduce the computational complexity of the SW BCJR algorithm.

In this paper, we propose an efficient sliding window type scheme which reduces the complexity by forwarding the window by C steps, where $C \geq 1$. The proposed algorithm resulted in enhanced performance compared to the SW BCJR algorithm with the same complexity.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 179-184, 1999.

In section II we describe SW BCJR algorithm compared to BCJR algorithm, and in section III we explain an efficient sliding window type algorithm to overcome the complexity problem of conventional SW BCJR algorithm. Section IV is dedicated to simulation results for different decoding algorithms. Finally conclusion is drawn in section V.

2 BCJR Algorithm and SW BCJR Algorithm

The BCJR algorithm estimates *a posteriori* probability (APP) of information bit to obtain log likelihood ratio (LLR) [1]. In this paper, we do not detaily describe the numerical expressions for LLRs, and we simply adopt the expressions used in [2]. LLRs typically represent soft outputs, which are used in an iterative decoding process. The BCJR algorithm calculates $\alpha_k(S_i), \beta_k(S_i)$ by forward and backward recursion respectively [2], and $\gamma_k(c)$ whenever the channel outputs of codewords are received [2], where k is a time index, S_i represents encoder state, and c is a codeword which is determined by encoder state and information bit. Then APPs can be obtained from $\alpha_k(S_i), \beta_k(S_i), \gamma_k(c)$.

The BCJR algorithm requires the whole sequence to be received before the decoding process is started, and trellis terminations of each frame is also required in prior to initialize the backward recusrion as shown in Fig. 1. Moreover, it is necessary for the BCJR algorithm to store all the values of $\alpha_k(S_i)$ and $\gamma_k(c)$ in one frame.

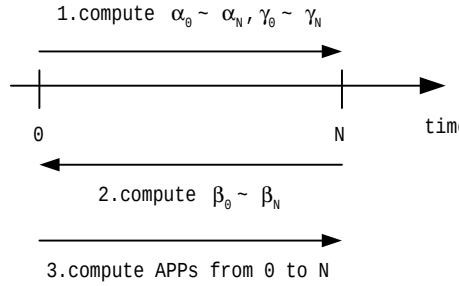


Fig. 1. The steps of the BCJR algorithm

In 1996, the SW BCJR algorithm is proposed by Benedetto *et al*. [2] which avoids the problem of trellis termination and operates on a fixed memory span. Forwarding the window of width *D*, the SW BCJR algorithm gives LLRs, but does not divide the received sequence into blocks, as shown in Fig. 2.

The SW BCJR algorithm initializes the backward recursion at time k using the value of $\alpha_k(S_i)$, and performs backward recursion from time k-1 back to time k-D and then computes APP at time k-D. After calculating APP, the SW BCJR algorithm forwards the window by 1 step and repeats the same operations at time k+1 to obtain APP at time k-D+1. Therefore the decoding process is not performed by frame basis, and also the trellis termination is not necessary. Moreover, the SW BCJR algorithm uses less

memory because it stores $\alpha_k(S_i)$s and $\gamma_k(c)$s for a corresponding window instead of a whole frame.

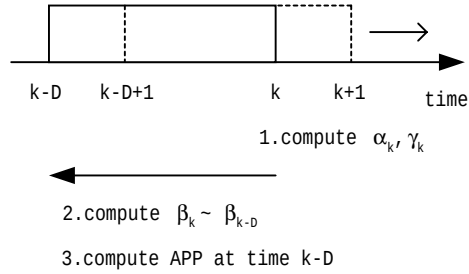


Fig. 2. The steps of the SW BCJR algorithm

3 Reduced Complexity SW BCJR Algorithm

The numbers of computations of $\alpha_k(S_i)$ and $\gamma_k(c)$ are equal both in the SW BCJR and in the ordinary BCJR algorithms, but the required computations of $\beta_k(S_i)$ of the SW BCJR algorithm are *D* times as many as those of the BCJR algorithm.

The SW BCJR algorithm performs backward recursion from *k* to *k-D* to obtain $\beta_{k-D}(S_i)$, and then the window of width *D* forwards by 1 step, as shown in Fig. 3-a). In this paper, we propose an efficient sliding window type scheme which reduces the number of computations of $\beta_k(S_i)$ by forwarding the window by *C* steps (Fig. 3-b)).

If the proposed algorithm uses the window of the same width as that of the SW BCJR algorithm and *C* is greater than 1, the complexity of the proposed algorithm reduces but the performance degrades. This is because the initialization of backward recusrion is less accurate than in the original SW BCJR. That is, the more backward recursion, the more accurate value of $\beta_k(S_i)$ could be achieved. With the same computational complexity, however, the performance of the proposed algorithm can be enhanced. It should be noted that we can lengthen the window width of the proposed algorithm compared to that of the original SW BCJR, resulting the same computational complexity.

Let us compare the complexities of the BCJR algorithm, SW BCJR algorithm, and the proposed algorithm. These algorithms have the same numbers of computations for $\alpha_k(S_i)$ and $\gamma_k(c)$, but the different numbers of computations for $\beta_k(S_i)$. The required computations of $\beta_k(S_i)$ of the SW BCJR algorithm are *D* times as many as those of the BCJR algorithm, and the number of computations of the proposed algorithm is *D/C* times as those of the BCJR algorithm as shown in Table 1, where k_0 represents parameter of (k_0, n_0) convolutional codes, and N_S is the number of states. In addition, the SW BCJR algorithm and the proposed algorithm have the same memory requirements, which are *D/N* times as many as those of the BCJR algorithm. Table 2 shows the memory requirement of each algorithms.

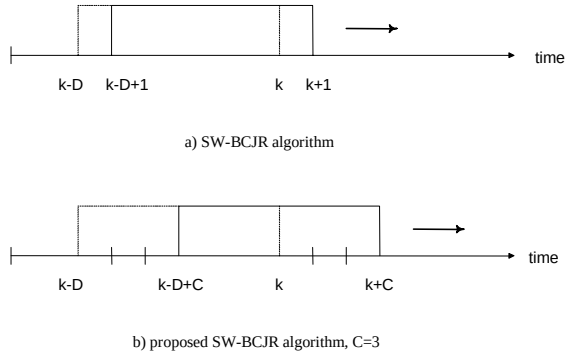


Fig. 3. The movement of window

Table 1. The number of computation to obtain $\beta_k(S_i)$

	the number of additions of 2^{k_0} numbers each	the number of multiplications
BCJR	N_S	$N_S \times 2^{k_0}$
SW BCJR	$D \times N_S$	$D \times N_S \times 2^{k_0}$
proposed	$\frac{D}{C} \times N_S$	$\frac{D}{C} \times N_S \times 2^{k_0}$

Table 2. The memory requirement of each algorithms

	the number of $\gamma_k(c)$ to be stored	the number of $\alpha_k(S_i)$ to be stored
BCJR	$N \times M$	$N \times N_S$
SW BCJR	$D \times M$	$D \times N_S$
proposed	$D \times M$	$D \times N_S$

4 Simulation Results

In this section, we have estimated the performances of the proposed algorithm in comparison to the SW BCJR algorithm. We carried out Monte Carlo simulations using rate 1/3 turbo encoders over AWGN channel. Two types of equal component codes were employed, generator polynomials $\{7,5\}_8$ with K=3 (code 1), and $\{17, 15\}_8$ with K=4 (code 2). The simulation results are shown in Fig. 4 and Fig. 5.

In the figures, the 'SW' denotes sliding window BCJR algorithm, 'PSW' denotes proposed algorithm, and D, C represent window width and window forwarding step

size respectively. The number of iterations in the decoding process is 2. The width of window used refered to the decoding depth of convolutional codes, which is about 5 times of the number of registers in encoder [4], [5].

Fig. 4 shows the performance comparison of the proposed algorithm for D=10, C=5 and D=15, C=5 to those of the SW BCJR algorithm for D=3, 10 with K=3. The SW BCJR algorithm with D=3 and the proposed algorithm with D=15, C=5 have same complexity. The proposed algorithm with D=15, C=5 shows the best performance.

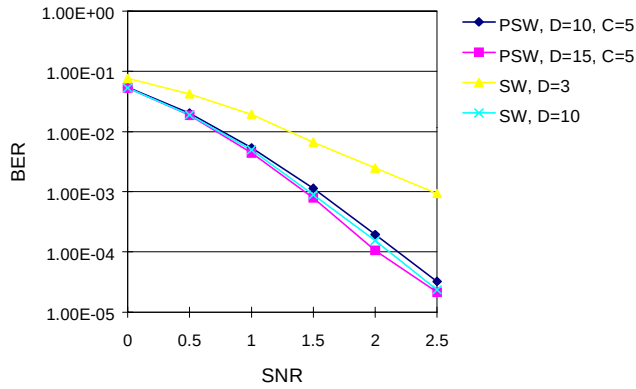


Fig. 4. BER performance for various decoders of code1

Fig. 5 compares the performances of the proposed algorithm with those of the SW BCJR algorithm for K=4.

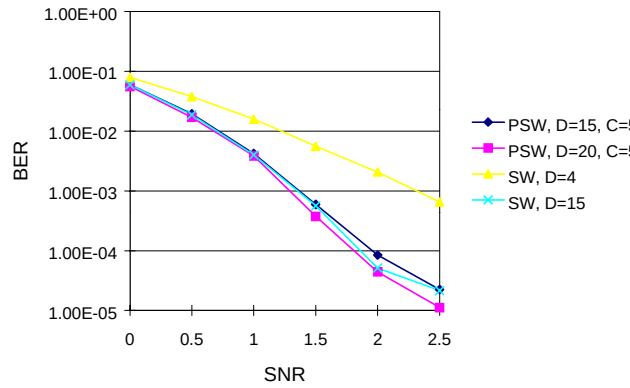


Fig. 5. BER performance for various decoders of code 2

5 Conclusions

In this paper, we propose an efficient sliding window type scheme which maintains the advantages of the conventional sliding window algorithm, reduces its computational burdens. We can improve performance and reduce complexity simultaneouly with proper choices of *C* and *D*.

References

1. C. Berrou, A. Glavieux, and P. Thitimajshma, „Near Shannon limit error-correction coding and decoding : Turbo-codes," in *Proc. ICC*, pp. 1064-1070, Geneva, Switzerland, May 1993.
2. S. Benedetto, D.Divsalar, G. Montorsi, and F. Pollara, „Soft-output decoding algorithms for continuous decoding of parallel concatenated convolutional codes," in *Proc. ICC*, pp. 112-117, Dallas, U. S. A., June 1996.
3. P. Robertson, „Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," in *Proc. GLOBECOM*, pp. 1298-1303, San Francisco, U. S. A., Nov. 1994.
4. F. Hemmati and D. J. Costello, Jr., „Truncation error probability in Viterbi decoding," *IEEE Trans. Commun.*, vol. 25, pp. 530-532, May 1977.
5. S. Lin and D. J. Costello, Jr., *Error Control Coding*. Prentice-Hall, 1983.

Advanced Encryption Standard (AES) - An Update

[Invited Paper]

Lars R. Knudsen

University of Bergen, Norway

Abstract. On January 2, 1997, the National Institute of Standards and Technology in the US announced that they intend to initiate the development of a new world-wide encryption standard to replace the Data Encryption Standard (DES). A call for candidates was announced world-wide with the deadline of 15th June 1998. Totally, 15 candidates were submitted from the US, Canada, Europe, Asia and Australia. The author is the designer of one of the candidates, and a codesigner of another proposal.

The AES proposals are required to support at least a block size of 128 bits, and three key sizes of 128, 192, and 256 bits. The hope of NIST is that the end result is a block cipher \with a strength equal to or better than that of Triple-DES and signi cantly improved e ciency."

In March 1999 the rst AES workshop was held in Rome, Italy. August 9, 1999, NIST announced the selection of ve candidates for a nal round of analysis. After a second AES workshop to be held in New York in April 2000, NIST intends to make a nal selection of one or two algorithms for the Advanced Encryption Standard during the summer of year 2000.

The ve algorithms selected to the nal round are MARS, RC6, Rijndael, Serpent, and Two sh, which also are the candidates predicted by the author in a letter to NIST.

The winner(s) of the AES competition are likely to be used widely and for many years to come. Therefore, it is important that a candidate is chosen with a high level of security not only now, but also in 25 years time or more. It is of course impossible to predict which of the ve candidates will survive attacks for such a long period, but this also speaks in favor of the choice of a candidate with a large security margin.

All AES candidates are iterated ciphers, where a ciphertext is computed as a function of the plaintext (and possibly some previous ciphertexts) and the key in a number of rounds. In the call for candidates NIST did not allow for a variable number of rounds. Although NIST allowed for possible \tweaks" (small changes), at the end of the rst round (April 15, 1999) none of the designers changed the number of rounds of their algorithms. In fact of the ve nal ones, only the MARS designers suggested a modi cation to overcome a small key-schedule problem.

In our opinion, the number of rounds xed by some of the designers is too small, and the algorithms will prove inadequate for long-term security. We believe that this narrows down the ve candidates to only a few.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 185{185, 1999.

The Piling-Up Lemma and Dependent Random Variables

Zsolt Kukorelly

Signal and Information Processing Laboratory, ETH Zürich
Sternwartstrasse 7, 8092 Zürich, Switzerland
kukorell@isi.ee.ethz.ch

Abstract. In a linear cryptanalysis attack, several assumptions are made by the attacker. One of them is that the threefold sums used in the attack are independent. This allows one to apply then the Piling-up Lemma to them. According to this lemma, the imbalance of a sum *modulo* 2 of independent, binary-valued random variables is equal to the product of their imbalances. It is shown here that in some cases, both quantities can di er considerably for dependent random variables, but that they are almost equal for virtually all binary-valued random variables when the sample space on which these are de ned is large enough.

1 The Piling-Up Lemma

In a linear cryptanalysis attack on iterated block ciphers, one identity important for the computation of the probability of success of the attack is Matsui's Piling-up Lemma, which states that for independent, binary-valued random variables $X_1,:::;X_n$, the probability that $X_1 \quad X_n = 0$ is $1=2+2^{n-1}\prod_{i=1}^{n}(P[X_i = 0]-1=2)$ [4]. Using the notation introduced by Harpes, Kramer and Massey [2], this can be written as $I(X_1 \quad X_n) = \prod_{i=1}^{n} I(X_i)$, where $I(X) = j2P[X = 0]-1j$ is the *imbalance* of the binary-valued random variable X.

Another important gure in this attack is that of an input/output sum. An *i-round input/output sum (I/O sum)* is an expression of the form $S^{1:::i} = f_0(X) \quad f_i(Y(i))$, where X is the plaintext, $Y(i)$ is the output of the i^{th} round of the cipher, and f_0, f_i are binary-valued balanced functions, that is, functions which take on each of the values 0 and 1 for half of their arguments.

Now one can also de ne imbalances based on conditional probabilities. For an i-round I/O sum $S^{1:::i}$, one de nes

- the *key-dependent imbalance* as the imbalance of $S^{1:::i}$ given xed values of the round keys, i.e., $I(S^{1:::i}jz_1;:::;z_i) := 2P[S^{1:::i} = 0j(Z_1;:::;Z_i) = (z_1;:::;z_i)]-1$, where $Z_1;:::;Z_i$ are the round keys;
- the *average-key imbalance* as the expected value, taken over all round keys, of the key-dependent imbalances, i.e., $\overline{I}(S^{1:::i}) := E[I(S^{1:::i}jZ_1;:::;Z_i)]$.

It turns out that, provided some assumptions, the probability of success of an attack using linear cryptanalysis, that is, the probability that the key found is

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 186{190, 1999.

the right one, is approximately proportional to the square of the average-key imbalance of the $(r-1)$-round I/O sum used in the attack [1,2]. Thus, it is important for the cryptanalyst to find balanced functions f_0 and f_{r-1} such that $\overline{I}(S^{1,\ldots,r-1})$ is as large as possible.

But it is usually infeasible to compute $\overline{I}(S^{1,\ldots,r-1})$ as it requires the computation of $Y(r-1)$ for all values of X and all values of the round keys. An efficient way out of this dead-end can be found in [1,2]: define $T_i = f_{i-1}(Y(i-1)) \oplus f_i(Y(i)) \oplus h_i(Z_i)$, where f_{i-1}, f_i and h_i are binary-valued functions the first two of which are balanced. Then $T_1 \oplus \cdots \oplus T_{r-1} = S^{1,\ldots,r-1} \oplus h_1(Z_1) \oplus \cdots \oplus h_{r-1}(Z_{r-1})$ and $I(T_1 \oplus \cdots \oplus T_{r-1}) = \overline{I}(S^{1,\ldots,r-1})$. Thus, finding $T_1,\ldots,T_{r-1}$ such that $I(T_1 \oplus \cdots \oplus T_{r-1})$ is large assures that the average-key imbalance of the corresponding I/O sum is large.

However, for the same reason as for $\overline{I}(S^{1,\ldots,r-1})$, it is also usually infeasible to compute $I(T_1 \oplus \cdots \oplus T_{r-1})$. If $T_1,\ldots,T_{r-1}$ were independent, then, by the Piling-up Lemma,

$$I(T_1 \oplus \cdots \oplus T_{r-1}) = \prod_{i=1}^{r-1} I(T_i). \tag{1}$$

The right side of the equation can be computed much more easily because T_i involves only the input, the output and the round key of a single round. The problem is therefore reduced to finding $T_1,\ldots,T_{r-1}$ independent. This is very difficult in practice. Thus, what one usually does is to assume that $T_1,\ldots,T_{r-1}$ are independent and to apply (1). We call that the *piling-up approximation*.

The ignorance whether the piling-up approximation is valid or not, i.e., whether $\prod_{i=1}^{n} I(T_i)$ is a fairly accurate or a very bad approximation of $I(T_1 \oplus \cdots \oplus T_{r-1})$, has never prevented anyone of using linear cryptanalysis. It is important only in the computation of the probability of success and, if the approximation is valid, it gives the cryptanalyst a clear conscience.

2 The Piling-Up Approximation Is Dangerous

In a sense, the piling-up approximation can be strongly misleading, as shows the following lemma.

Lemma 1. *For any binary-valued random variables $X_1,\ldots,X_n$, we have*

$$I(X_1) \cdots I(X_n) \geq 2^{-n}((n-1) + I(X_1 \oplus \cdots \oplus X_n))^n \tag{2}$$

with equality if and only if $I(X_i) = \frac{1}{2}((n-1) + I(X_1 \oplus \cdots \oplus X_n))$ for all i. Moreover, equality can occur in (2).

Proof. The steps in the proof are the following (details can be found in [3]):

1. $I(X_1) + I(X_2) \geq 1 + I(X_1 \oplus X_2)$ for all binary-valued random variables X_1 and X_2;

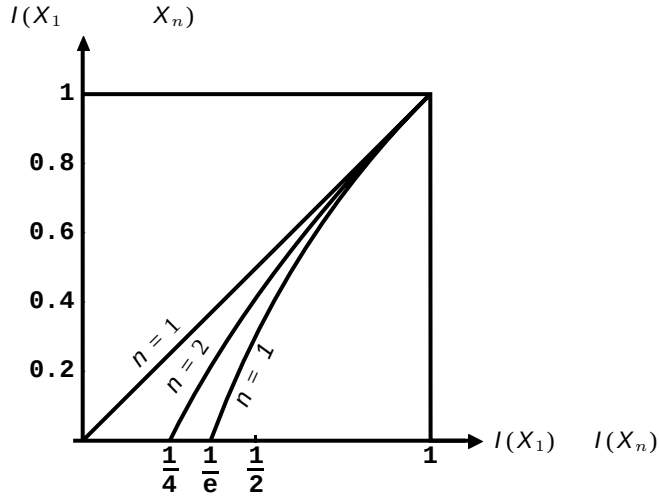


Fig. 1. For n 2, all points on or above the line are possible.

2. by induction, $I(X_1) +$ $+ I(X_n)$ $(n-1) + I(X_1$ $X_n)$ for all binary-valued random variables $X_1,:::;X_n$;
3. because a_1a_2 a_n $(\frac{a_1+ \ +a_n}{n})^n$ for any nonnegative numbers $a_1,:::;a_n$ with equality if and only if all a_i are equal, we have

$$I(X_1) \quad I(X_n) \quad n^{-n}(I(X_1) + \quad + I(X_n))^n$$
$$n^{-n}((n-1) + I(X_1 \quad X_n))^n;$$

with equality everywhere if and only if $I(X_i) = \frac{1}{n}((n-1) + I(X_1$ $X_n))$ for all i;
4. equality can occur in (2). ∎

Figure 1 visualises the inequality (2). The Piling-up Lemma says that if the random variables are independent, then we are always on the diagonal. Inequality (2) says that, for n 2, all points on or above the solid line are possible. Hence, for dependent random variables, the product of the imbalances can di er considerably from the imbalance of the sum and thus (1) is sometimes very far from being satis ed.

3 The Piling-Up Approximation Is Applicable

Fortunately for the cryptanalyst, on average, things are di erent.

3.1 Two Random Variables

Consider two random variables X_1 and X_2 de ned on some sample space Ω. Let Ω have an even number 2# of elements. (The case $j\Omega j$ odd is not interesting for

our purpose.) Then $I(X_1)$ and $I(X_2)$ are of the form $i_1/\#$ and $i_2/\#$, respectively, where i_1 and i_2 are integers, $0 \le i_1, i_2 \le \#$.

Now fix $i_1/\#$ and $i_2/\#$ and consider all different pairs of random variables (X_1, X_2), independent or not, for which $I(X_1) = i_1/\#$ and $I(X_2) = i_2/\#$. (Two random variables are different if they differ as functions from Ω to $\{0,1\}$, not if their probability distribution is different.) Then compute $I(X_1 \oplus X_2)$ for each pair (X_1, X_2). By some counting arguments, the average of $I(X_1 \oplus X_2)$ is equal to $f(i_1, i_2)$, where

$$f(i,j) := \frac{1}{\binom{2\#}{\#+j}} \sum_{m=i+j}^{\#-i} \binom{2m-\#-i-j}{m} \binom{\#+i}{m} \binom{\#-i}{m-i-j}, \quad 0 \le i \le j \le \#/2,$$

and $f(i,j) := f(j,i)$ if $i > j$.

One shows that $ij/\#^2 \le f(i,j) \le ij/\#^2 + (1+\varepsilon(\#))/\sqrt{\#}$ for all i, j, where $\lim_{\# \to \infty} \varepsilon(\#) = 0$. This means that the average of $I(X_1 \oplus X_2)$ is lower-bounded by $I(X_1)I(X_2)$ and upper-bounded by $I(X_1)I(X_2) + (1+\varepsilon(\#))/\sqrt{\#}$. Now if the sample space Ω on which X_1 and X_2 are defined is large, then the above average of $I(X_1 \oplus X_2)$ is close to $I(X_1)I(X_2)$. This is a first indication that the piling-up approximation might be valid after all.

The same average of $I^2(X_1 \oplus X_2)$ is equal to $h(i_1, i_2)$, where

$$h(i,j) := \frac{1}{\#^2 \binom{2\#}{\#+j}} \sum_{m=i+j}^{\#-i} \left(2m-\#-i-j\right)^2 \binom{\#+i}{m} \binom{\#-i}{m-i-j}, \quad 0 \le i \le j \le \#/2,$$

and $h(i,j) := h(j,i)$ if $i > j$. One has $h(i,j) = \frac{1}{2\#-1} + \frac{2\#}{2\#-1}\frac{i^2}{\#^2}\frac{j^2}{\#^2} - \frac{1}{2\#-1}\left(\frac{i^2}{\#^2} + \frac{j^2}{\#^2}\right)$, that is, the average of $I^2(X_1 \oplus X_2)$ is $\frac{1}{2\#-1} + \frac{2\#}{2\#-1} I^2(X_1)I^2(X_2) - \frac{1}{2\#-1}\left(I^2(X_1) + I^2(X_2)\right)$. If $\#$ gets large, this is approximately equal to $I^2(X_1)I^2(X_2)$. Thus, we can conclude:

Proposition 2. *Let Ω be some sample space with $2\#$ elements. Then, if $\#$ is large enough, $I(X_1 \oplus X_2) \approx I(X_1)I(X_2)$ for virtually all binary-valued random variables X_1 and X_2.* □

3.2 More than Two Random Variables

For more than two random variables, we can compute similar averages recursively. Take some integers $0 \le i_1, \ldots, i_n \le \#$ and consider all n-tuples $(X_1, \ldots, X_n)$ of random variables such that $I(X_1) = i_1/\#, \ldots, I(X_n) = i_n/\#$. Then compute $I(X_1 \oplus \cdots \oplus X_n)$. Denote by $P(\ell | i_1, \ldots, i_n)$ the empirical probability that $I(X_1 \oplus \cdots \oplus X_n) = \ell/\#$ given that $I(X_1) = i_1/\#, \ldots, I(X_n) = i_n/\#$; denote also by $f(i_1, \ldots, i_n)$ (resp. $h(i_1, \ldots, i_n)$) the average of $I(X_1 \oplus \cdots \oplus X_n)$ (resp. of $I^2(X_1 \oplus \cdots \oplus X_n)$), that is, $f(i_1, \ldots, i_n) = \sum_k k P(k | i_1, \ldots, i_n)$ and $h(i_1, \ldots, i_n) = \sum_k k^2 P(k | i_1, \ldots, i_n)$. One shows then that

- $P('ji_1;\ldots;i_n) = \sum_k P('jk;i_n)P(kji_1;\ldots;i_{n-1})$;
- $f(i_1;\ldots;i_n) = \sum_k f(k;i_n)P(kji_1;\ldots;i_{n-1})$;
- $h(i_1;\ldots;i_n) = \sum_k h(k;i_n)P(kji_1;\ldots;i_{n-1})$.

By induction, one also shows that

- $f(i_1;\ldots;i_n) \quad i_1 \quad i_n = \#^n$;
- $f(i_1;\ldots;i_n) \quad i_1 \quad i_n = \#^n + (n-1)(1+\varepsilon(\#))\sqrt{\#}$, where ε is the same function as above with $\lim_{\# \to \infty} \varepsilon(\#) = 0$;
- $h(i_1;\ldots;i_n) = \frac{1}{2\#-1} + \frac{2\#}{2\#-1}\frac{i_n^2}{\#^2}h(i_1;\ldots;i_{n-1}) - \frac{1}{2\#-1}\ h(i_1;\ldots;i_{n-1}) + \frac{i_n^2}{\#^2}$.

Again, if $\#$ is large but $i_1 = \#;\ldots;i_n = \#$ xed, then $f(i_1;\ldots;i_n) \quad i_1 \quad i_n = \#^n$, and from the recursion for h follows that $h(i_1;\ldots;i_n) \quad \prod_1^n i_n^2 = \#^{2n}$. Thus, if $\#$ is large, then the average of $I(X_1 \quad X_n)$ is close to $\sum_{i=1}^n I(X_i)$ and the average of $I^2(X_1 \quad X_n)$ is close to $\sum_{i=1}^n I^2(X_i)$. Hence, we have:

Theorem 3. *Let Ω be some sample space with $2\#$ elements. If $\#$ is large enough, then $I(X_1 \quad X_n) \quad \sum_{i=1}^n I(X_i)$ for virtually all binary-valued random variables $X_1;\ldots;X_n$ de ned on Ω.* □

3.3 Implication for the Piling-Up Approximation

Let n be the text blocklength of the cipher and k be the length of the round keys. The plaintext X and the round keys Z_i are usually considered to be independent random variables uniformly distributed on $\{0;1\}^n$ and $\{0;1\}^k$, respectively. Because the round functions yield an invertible function when one xes the value of the round key, the output $Y(i)$ of the i^{th} round of the cipher is also a random variable uniformly distributed on $\{0;1\}^n$. Thus, $(Y(i-1);Y(i);Z_i)$ is a random variable with values on $\{0;1\}^{2n+k}$ and $T_i = f_{i-1}(Y(i-1)) \quad f_i(Y(i)) \quad h_i(Z_i)$ is a binary-valued random variable with sample space $\Omega = \{0;1\}^{2n+k}$ with $2\,2^{2n+k-1}$ elements. In practical ciphers, $\# \approx 2^{2n+k-1}$ is fairly large. Thus, by the above Theorem, $I(T_1 \quad T_{r-1}) \quad \sum_{i=1}^{r-1} I(T_i)$ in virtually all cases, that is, the piling-up approximation is valid.

References

1. Carlo Harpes, *Cryptanalysis of Iterated Block Ciphers*, Vol. 7 of ETH Series in Information Processing, Ed. J.L. Massey, Hartung-Gorre Verlag, Konstanz, 1996. ISBN 3-89649-079-6.
2. Carlo Harpes, Gerhard G. Kramer, and James L. Massey, \A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma", in *Advances in Cryptology { Eurocrypt'95*, Lecture Notes in Computer Science 921, pp. 24-38, Springer 1995. ISBN 3-540-59409-4.
3. Zsolt Kukorelly, *On The Validity of Some Hypotheses Used in Linear Cryptanalysis*, Vol. 13 of ETH Series in Information Processing, Ed. J.L. Massey, Hartung-Gorre Verlag, Konstanz, 1999. ISBN 3-89649-470-8.
4. Mitsuru Matsui, \Linear cryptanalysis method for DES cipher", in *Advances in Cryptology { Eurocrypt'93*, Lecture Notes in Computer Science 765, pp. 386-397, Springer 1993. ISBN 3-540-57600-2.

A Cryptographic Application of Weil Descent

Steven D. Galbraith[?1] and Nigel P. Smart[2]

[1] Mathematics Department, Royal Holloway University of London, Egham, Surrey TW20 0EX, U.K.
s.galbraith@rhbnc.ac.uk
[2] Hewlett-Packard Laboratories, Filton Road, Stoke Gi ord, Bristol, BS12 6QZ, U.K.
nigel_smart@hpl.hp.com

Abstract. This paper gives some details about how Weil descent can be used to solve the discrete logarithm problem on elliptic curves which are de ned over nite elds of small characteristic. The original ideas were rst introduced into cryptography by Frey. We discuss whether these ideas are a threat to existing public key systems based on elliptic curves.

1 Introduction

Frey [4] introduced the cryptographic community to the notion of \Weil descent", which applies to elliptic curves de ned over nite elds of the form $\mathbb{F}_{q^n}$ with $n > 1$. This paper gives further details on how these ideas might be applied to give an attack on the elliptic curve discrete logarithm problem. We also discuss which curves are most likely to be vulnerable to such an attack.

The basic strategy consists of the following four stages (which will be explained in detail in the remainder of the paper).

1. Construct the Weil restriction of scalars, A, of $E(\mathbb{F}_{q^n})$.
2. Find a curve, C, de ned over $\mathbb{F}_q$ which lies on the variety A.
3. Pull the discrete logarithm problem back from $E(\mathbb{F}_{q^n})$ to $Jac(C)(\mathbb{F}_q)$.
4. Solve the discrete logarithm problem on $Jac(C)(\mathbb{F}_q)$ using an index calculus method.

We must emphasize that this paper does not represent a fully developed attack on the elliptic curve discrete logarithm problem. Nevertheless, we believe that there are examples of elliptic curves over nite elds of the form $\mathbb{F}_{q^n}$ for which the method of this paper would be faster than any previously known method for solving the elliptic curve discrete logarithm problem.

In Sections 2 and 3 we give some details about abelian varieties, curves and the \Weil restriction" (which is an abelian variety). We also provide more details about the rst two stages above. The ideas in these sections are well-known in algebraic geometry, but they are not well understood among the cryptographic community.

* This author thanks the EPSRC for support.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 191{200, 1999.

In Section 4 we describe a possible method for solving Stage 3 above. In Section 5 we give a down to earth explanation of the whole approach with a very simple example. In this example we construct the Weil restriction over $\mathbb{F}_{2^m}$ of an elliptic curve E over $\mathbb{F}_{2^{4m}}$.

In Section 6 we discuss how the discrete logarithm problem in the divisor class group of a curve can be solved in heuristic sub-exponential time relative to the genus of the curve.

In the nal section we discuss some open problems and give an argument of why we do not expect most elliptic curves to be vulnerable to a Weil descent attack. In particular we explain why we believe composite values of n may be weaker than prime values, thus possibly explaining the choice of some standards bodies in precluding composite values of n.

2 Curves, Divisor Class Groups, and Jacobians

We gather a few relevant facts from algebraic geometry. The main references are Hartshorne [9], Mumford [12] and Milne [11].

A *curve* C over a eld k is a complete, non-singular variety of dimension one over k. We often deal with curves which are given as a speci c a ne model, which can cause problems. For instance in this setting the curves are often singular, but we will nevertheless still call them curves. The usual approach for handling singularities on a curve, C/k, is to take a sequence of blow-ups, to produce a non-singular curve. This process may involve enlarging the ground eld k. Nevertheless, for any C there is a non-singular curve, C', called the normalisation (see [9], Ex. II.3.8), which is de ned over the same eld k, and a degree one rational map $C' \to C$.

The *genus* g of a non-singular curve is an important invariant of the curve. For a singular curve one may de ne the *geometric genus* (see [9], Section II.8), which is a birational invariant, as the genus of the normalisation of the curve.

The *Jacobian variety* $Jac(C)$ of a non-singular curve C of genus g, over a eld k, is an abelian variety over k of dimension g. One construction of the Jacobian is as a subset of the gth symmetric power, $C^{(g)}$, of the curve (see [11]). We gather two important facts about the Jacobian of a curve.

Proposition 1 *Suppose C is a non-singular curve over a eld k with a point P de ned over k. The following properties hold.*

1. *(Canonical map from C into $Jac(C)$) There is a canonical map $f^P : C \to Jac(C)$ which takes the point P to the identity element of $Jac(C)$.*
2. *(Universal property) Suppose A is an abelian variety over k and suppose there is some mapping of varieties $\phi : C \to A$ such that $\phi(P) = 0_A$, then there is a unique homomorphism $\psi : Jac(C) \to A$ such that $\psi = \phi f^P$.*

The divisor class group $Pic^0_k(C)$ of a curve is the group of degree zero divisors on C, which are de ned over k, modulo principal divisors. If C is a non-singular curve over k with a k-point, P, then $Jac(C)$ and $Pic^0_k(C)$ are isomorphic as

abelian varieties (see [11], Theorem 7.6). It is convenient to view prime divisors on the normalisation as places of the function field. Since the function field of a singular curve is isomorphic to the function field of its normalisation it follows that we can define the divisor class group of a singular curve C and that it will be isomorphic to the divisor class group of its normalisation.

An abelian variety A over k is *simple* if it has no proper abelian sub-varieties defined over k. An abelian variety is *absolutely simple* if, even when considered over the algebraic closure k, it is simple. An *isogeny* is a mapping of abelian varieties and so is, in particular, a group homomorphism with finite kernel.

The Poincare complete reducibility theorem (see Mumford [12] Theorem 1, page 173) implies that every abelian variety is isogenous to a product of simple abelian varieties in a unique way (up to ordering).

We now give an application of this property (also see [11], p. 199). Suppose that A is a *simple* abelian variety of dimension d and that we are given a curve C and a map $\phi : C \to A$ (in this paper the maps we will consider will usually have degree one and we will use the phrase "C lies on A" to represent this situation). Since the image of the map $\phi : Jac(C) \to A$ is an abelian subvariety of A it follows, from the fact that A is simple, that the map ϕ is surjective and that A is an abelian subvariety of $Jac(C)$. In other words, one has the following result.

Proposition 2 *Let A be a simple abelian variety of dimension d over a field k. Suppose we have a map $\phi : C \to A$ from a non-singular curve C to A. Then the genus of C is at least d. Furthermore, $g(C) = d$ if and only if A is isogenous to the Jacobian of C.*

3 Weil Descent

Let k denote a finite field and K denote a Galois extension of degree n. For example, we could have $k = \mathbb{F}_2$ and $K = \mathbb{F}_{2^n}$ or $k = \mathbb{F}_{2^m}$ and $K = \mathbb{F}_{2^{nm}}$, which are the two most important cases for the applications. Let E denote some elliptic curve over K, we assume we wish to solve a discrete logarithm problem in $E(K)$ given by $P_2 = [\ ]P_1$, with $P_1, P_2 \in E(K)$. We include the case where E is defined over k in our discussion, this is the case of Koblitz curves. Koblitz curves are used in real life situations since they produce efficient cryptographic systems.

The *Weil restriction of scalars* of E over K is an abelian variety $W_{K/k}(E)$ of dimension n over the smaller field k.

The proof that such an object exists is fairly deep. Nevertheless, we can easily show how $W_{K/k}(E)$ can be constructed in our case (a specific example will be given later). First take a basis of K over k and expand the coordinate functions on the affine curve $E(K)$ in terms of the new basis, thus using $2n$ variables over k. Expanding out the formulae for the elliptic curve $E(K)$ and equating coefficients of the basis of K over k, we obtain n equations linking our $2n$ variables. This variety is an affine model for $W_{K/k}(E)$ and the group law is induced from the group law on the elliptic curve.

The following result is stated in [4];

Lemma 1 *If E is de ned over k then $W_{K=k}(E) = E(k)$ V, where V is an abelian variety of dimension $n-1$. If n is coprime to $\#E(k)$ then we have,*

$$V = fP \ 2 \ W_{K=k}(E) : Tr_{K=k}(P) = Og$$

where the trace is computed using the mapping from $W_{K=k}(E)$ to $E(K)$.

Proof. If E is de ned over k then it is clearly an abelian subvariety of $W_{K=k}(E)$. By the Poincare complete reducibility theorem it follows that there is an abelian subvariety B over k such that $W_{K=k}(E)$ is isogenous to E B.

The construction of the Weil restriction implies that a generic point of $W_{K=k}(E)$ is $(\ ; \ (\ ); : : : ; \ ^{n-1}(\ ))$ where is a generic point of $E=K$. It follows that the subvariety V of $W_{K=k}(E)$ has codimension 1.

Finally, one sees that V is an abelian subvariety of $W_{K=k}(E)$ and, since n is coprime to $\#E(k)$, the subvariety $V(k)$ has trivial intersection with $E(k)$. Therefore, V is isogenous to B and the Lemma is proved.

We let A denote the 'interesting' abelian variety, de ned over k, on which the discrete logarithm problem on $E(K)$ actually lies. In other words

De nition 1 *De ne A by*

i) If E is not de ned over k, then set $A = W_{K=k}(E)$. Hence $\dim A = n$.
ii) If E is de ned over k, then set $A = V$, from Lemma 1. Hence $\dim A = n-1$.

In general we expect the abelian variety A to be simple. Indeed, since the original elliptic curve will have order divisible by a large prime, it is clear by considering the number of points on A that there must be a large simple abelian subvariety of A.

We may now give a sketch of the Weil descent attack on the elliptic curve discrete logarithm problem: Given an elliptic curve E over K construct the abelian variety A over k as above. Next nd a (possibly singular) curve C de ned over k lying on A such that C has a k-point P_0 at the point at in nity of A. By the universal property of Jacobians there is a mapping of abelian varieties $: Jac(C^0) \ ! \ A$, where C^0 is the normalisation of C.

The points P_1 and P_2 of the discrete logarithm problem in $E(K)$ correspond to points on $A(k)$ in an obvious way, and these points may be pulled-back under to obtain divisors D_1 and D_2 in $Pic^0_k(C)(k)$ (whose support is only on the non-singular points of C) such that $(D_i) = P_i$. Finally, the discrete logarithm problem of D_2 with respect to D_1 on $Pic^0_k(C)$ can be solved using an index calculus method.

There are three main problems which must be overcome in order to apply this method.

1. It is necessary to nd curves of small genus on A.
2. It is necessary to pull back points on A to divisors on C.
3. It is necessary to have an index calculus method for general divisor class groups.

4 Pulling Back Along ψ

We shall need to describe the mapping more explicitly. Let C be a curve of genus g over k and let $: C \to A$ be the mapping of C into the abelian variety A. Suppose P_0 is the k-point on C (which we shall assume lies at in nity) which maps under $f = f^{P_0}$ to the identity element of A. Recall that elements of $Pic^0_k(C)$ may be represented in the form $D = D_{eff} - d(P_0)$ where $D_{eff} = \sum_{i=1}^{d}(Q_i)$ is an e ective divisor of degree d and where the Q_i are points on $C(k)$ such that, as a divisor, D_{eff} is de ned over k. Note that one usually restricts to $d \geq g$ but the process described below works for arbitrary values of d.

Proposition 3 *The map $: Pic^0_k(C) \to A(k)$ is given by*

$$(D_{eff} - d(P_0)) = \sum_{i=1}^{d} (Q_i)$$

where the addition on the right hand side is addition on the abelian variety A (which can be e ciently computed via the addition law on $E(K)$).

Proof. The divisor $D_{eff} - d(P_0)$ is equal to the sum (on $Pic^0_k(C)$) of the divisors $(Q_i) - (P_0)$. The canonical map $f : C(k) \to Pic^0_k(C)$ has the property that $f(Q_i) = (Q_i) - (P_0)$. The mapping $: Pic^0_k(C) \to A$ has the universal property that $= f$ and so $((Q_i) - (P_0)) = (Q_i) \in A(k)$. Since is a group homomorphism which preserves the action of the Galois group $Gal(k/k)$ the result follows.

In practice we will be using a singular equation for the curve C. The mapping above still gives a complete description of the map from $Pic^0_k(C)$ to A for the divisors whose support lies on the non-singular points of C.

To invert the map we have to nd a divisor which maps under to a given point, P, of A. We now describe how to nd such a divisor by performing the above Proposition in reverse.

We consider d generic points $\{Q_1, \ldots, Q_d\}$ on $C(k)$ (by this we mean that each point is written as $Q_i = (x_i, y_i)$ where x_i and y_i are variables). We map these points to the variety A via the map , to obtain $T_i = (Q_i)$ for $i = 1, \ldots, d$. Since the coordinates of the points are variables, we obtain d equations in $2d$ unknowns. Formally using the group law on A applied to these points T_i we determine the equations for the coordinates of the sum $\sum_{i=1}^{d} T_i$ and then equate this to the given element P. Since A has dimension n this gives us, roughly, another n equations.

Hence in total we have $d+n$ equations in $2d$ unknowns, which de nes a variety V. So as soon as $d > n$ we expect that this de nes a variety of dimension at least $d-n$. For example, a curve when $d = n+1$ and a surface when $d = n+2$. Finding a point on this variety will produce the points Q_i and in general these will be non-singular points on C. Therefore we obtain a divisor $D = \sum_{i=1}^{d}(Q_i) - d(P_0)$ in $Pic^0_k(C)$ which maps under to the given point P on $A(k)$. Note that a di erent point on the variety V will give rise to a di erent divisor D.

Finding points on varieties in high-dimensional spaces is not a computationally trivial matter. There may also be computational issues which arise when constructing this variety.

We repeat the above process for each of the points P_1 and P_2 of the original elliptic curve discrete logarithm problem. Therefore, we obtain divisors D_1 and D_2 in $\mathrm{Pic}_k^0(C)$. Since is a group homomorphism, we know that there exists a rational divisor D_3 in the kernel of such that

$$D_2 = [\ ]D_1 + D_3.$$

Let $h = \#\mathrm{Pic}_k^0(C)$, which can be computed in polynomial time using the algorithm due to Pila [13] (also see [2] and [10]). Suppose that the points P_1 and P_2 on the elliptic curve have prime order p. We shall make the reasonable assumption that p^2 does not divide h. Hence, by restricting to the subgroup of order p, either by using Pohlig-Hellman or adapting our discrete logarithm algorithm accordingly, we can recover if we can nd discrete logarithms in $Pic_k^0(C)$.

There may be other ways to pull the discrete logarithm problem back to the Jacobian of the curve C. This is a topic for future research.

5 An Example

We give an example to illustrate some of the ideas described above. Let $k = \mathbb{F}_{2^m}$ and let K be such that K has a Type-1 Optimal Normal Basis over k. This means that $n + 1$ should be a prime and that 2^m should be primitive in the nite eld $\mathbb{F}_{n+1}$. Then the n roots of

$$(x^{n+1} - 1)/(x - 1) = x^n + x^{n-1} + \quad + x + 1$$

form a basis of K over k.

As an example we take a eld with $n = 4$, for simplicity. Let $\{\ ;\ ^2;\ ^4;\ ^8\}$ denote the basis of K over k, so we have $^4 + \ ^3 + \ ^2 + \ + 1 = 0$ and the element $1 \in K$ is given, in terms of the basis, by $1 = \ + \ ^2 + \ ^4 + \ ^8$. Consider the following elliptic curve de ned over K,

$$Y^2 + XY = X^3 + b \tag{1}$$

where $b \neq 0$ and b is given by

$$b = b_0\ + b_1\ ^2 + b_2\ ^4 + b_3\ ^8.$$

By setting

$$X = x_0\ + x_1\ ^2 + x_2\ ^4 + x_3\ ^8 \text{ and } Y = y_0\ + y_1\ ^2 + y_2\ ^4 + y_3\ ^8;$$

where $x_i, y_i \in k$, substituting into (1) and equating powers of we obtain four equations in the eight unknowns, $\{x_0, \ldots, x_3, y_0, \ldots, y_3\}$. This describes

the abelian variety A as a 4-dimensional variety in 8-dimensional a ne space. Note that if one tries to extend this to a projective equation for A in the obvious manner then there are too many points at in nity. For the application we must remember that there is some projective equation for A which only has one point which does not lie on our a ne equation.

The group law on A can be evaluated by translating a point

$$(x_0; x_1; x_2; x_3; y_0; y_1; y_2; y_3) \in A(k)$$

back to the point

$$(x_0 + x_1^2 + x_2^4 + x_3^8; y_0 + y_1^2 + y_2^4 + y_3^8) \in E(K)$$

and then using the addition formulae on the elliptic curve.

If we intersect A with $(\dim A) - 1$ hyperplanes, in general position, which all pass through the zero element of A, then we should end up with a variety of dimension one. By using elimination theory we can then write down the equation of this variety.

In our example we have $\dim A = 4$, and the obvious hyperplanes to choose, given that we want the degree of the resulting curve to be small, are

$$x_0 = x_1 = x_2 = x_3.$$

Intersecting these with our variety A, we obtain the variety

$$V : \begin{cases} y_3^2 + y_0 x_0 + x_0^3 + b_0 = 0; \\ y_0^2 + y_1 x_0 + x_0^3 + b_1 = 0; \\ y_1^2 + y_2 x_0 + x_0^3 + b_2 = 0; \\ y_2^2 + y_3 x_0 + x_0^3 + b_3 = 0. \end{cases}$$

If we then eliminate y_3 by taking the resultant of the rst and fourth of these equations, and eliminate y_1 by taking the resultant of the second and third, then we obtain the variety:

$$V' : \begin{cases} y_2^4 + x_0^6 + b_3^2 + y_0 x_0^3 + x_0^5 + b_0 x_0^2 = 0; \\ y_0^4 + x_0^6 + b_1^2 + y_2 x_0^3 + x_0^5 + b_2 x_0^2 = 0. \end{cases}$$

Finally by eliminating y_2 from these two equations and setting $x = x_0$ and $y = y_0$ we obtain the a ne curve

$$C : y^{16} + x^{15} y + (x^{24} + x^{20} + x^{18} + x^{17} + b_0 x^{14} + b_3^2 x^{12} + b_2^4 x^8 + b_1^8).$$

The only singular point on this a ne model is the point at $(x; y) = (0; \sqrt[8]{b_1})$. There is also a singularity at the point at in nity, above which there will be several points and one of these will correspond to the unique point at in nity on the variety A. The other points will correspond to points on the a ne part of A.

To get a feel for this curve, take $k = \mathbb{F}_2$, this is far too small for real examples but it allows us to compute some invariants. We computed the genus, g, for this curve for all the possible values of the b_i, using the KANT package [3]. For the following values of the b_i, which represent exactly half of all possible values, we found that the genus was equal to 8

$$(b_0, b_1, b_2, b_3) = \begin{cases} (0,0,0,1), (0,0,1,0), (0,1,0,0), (0,1,1,1), \\ (1,0,0,0), (1,0,1,1), (1,1,0,1), (1,1,1,0). \end{cases}$$

The value of $(0,0,0,0)$ is precluded since the original elliptic curve must be non-singular. The other values of (b_0, b_1, b_2, b_3) produce curves which are reducible. As an example, consider $(b_0, b_1, b_2, b_3) = (0,0,1,1)$, in this case we obtain an irreducible factor given by the curve

$$C_1 : y^8 + x^4y^4 + x^6y^2 + x^7y + x^{12} + x^9 + x^4 = 0.$$

This curve has genus 4 and so the Jacobian of the normalisation of C_1 must be isogenous to A.

6 Solving the Discrete Logarithm Problem in the Divisor Class Group of Certain Curves

There are a variety of techniques one can use to solve the discrete logarithm problem in the $\mathrm{Pic}^0_k(C)$. A natural generalization of [1], as described in [5] for superelliptic curves, could be applied.

However if an e cient and e ective version of the Riemann-Roch theorem is available for the curve C then one can use a natural generalization of the Hafner-McCurley method [8]. The advantage with this latter method is that only needs to test \smoothness" of polynomials of degree g. For more details see the full version of the current paper [6].

One drawback with the above methods is that they are more suited to group structure computation rather than discrete logarithm computation. Gaudry, [7], describes an interesting discrete logarithm algorithm for hyperelliptic curves, which naturally generalizes to our situation when an e cient algorithm to solve the Riemann-Roch problem is available. Gaudry's method appears to be very successful in practice.

The complexity of all three of the above methods can be estimated to be around

$$L_{q^g}(1/2, c)$$

for a curve of genus g over a eld of q elements, for some constant c (which could depend on q), as $g \gg 1$.

7 Open Problems and Conclusion

In this paper we have outlined a strategy for solving the elliptic curve discrete logarithm problem and have given some details about each of the main steps in

this process. We now address the issue of whether such a strategy is a threat to the elliptic curve cryptosystems used in practice.

One important observation is that everything in this paper only applies to the case of elliptic curves over fields of the form $\mathbb{F}_{q^n}$ with $n > 1$. Elliptic curves over prime fields appear to be immune to these ideas.

The method can be broken down into 4 main stages (see the Introduction). Stage 1 (computing an affine equation for the Weil restriction) causes no practical problems.

The actual solution of the discrete logarithm comes from Stage 4, where the index calculus algorithm discussed in Section 6 is applied. The motivating problem is to solve a discrete logarithm problem on an elliptic curve which has approximately q^n points, but we do this by solving a related discrete logarithm problem in a group of size q^g, where g is the genus of the curve C.

If g is very large compared to n then the discrete logarithm problem has been buried in a much larger group and so the method is not useful. For the Weil descent to be a danger it is therefore necessary that the genus, g, not grow too large in relation to the original degree, n. On the other hand, the index calculus method is subexponential only asymptotically (i.e., when the field size is fixed at q and when the value of g is \sufficiently large"). Therefore, for the Weil descent attack to work, the values of n and g must strike a balance between these conflicting forces.

For Stage 2 it is necessary to find a curve lying on the abelian variety A. As we have seen, it is important for Stage 4 that the genus g of C be large, but not too large compared with n. The method used in our example to find such a curve involves eliminating variables. This leads to a curve whose degree is exponential in n (and so we expect the genus to also be exponential as long as C is not too singular). If the genus of C grows exponentially with n then the complexity of the Weil descent attack would be subexponential in q^g but this is exponential in terms of the elliptic curve group size q^n.

It is not known to the authors what values might be expected for the smallest possible genus for a curve C on such an A. This question is equivalent to asking about the expected dimension for a Jacobian with a given abelian variety as a factor. It is therefore an interesting problem to determine if there is a curve C of genus $O(n^d)$ on any such A for some fixed d. If such curves exist then it would be very interesting to have a method for obtaining equations for them in terms of the variables describing the variety A. When n is small there is a higher chance that there will be small genus curves lying on the abelian variety A (this was seen in our example, when half the time A was actually isogenous to a Jacobian). When n is large it seems to be very unlikely that A have curves on it of genus close to n and so it is unlikely that the Weil descent method would give a practical attack.

For Stage 3 it is necessary to find a point on a large dimensional variety over a small finite field. When n is very small then this problem is not too difficult to solve. It is not known to the authors how difficult this is to achieve when n is large. The method we describe appears very practical when $n = 2$ or 3, but when

$n = 4$ the equations needed to describe the variety become unwieldy. Hence this question deserves further study.

In conclusion, there are several problems which require further analysis before the Weil descent method can be fully assessed. We expect that, for a random elliptic curve E over a field $\mathbb{F}_{q^n}$ with n reasonably large, it will be possible to show that it is unlikely that there are relatively small genus curves on the Weil restriction of E over $\mathbb{F}_q$. This means that it is unlikely that the method could ever be used to solve the elliptic curve discrete logarithm problem on the sort of curves used in practice. Nevertheless, it seems that the Weil descent is most likely to succeed for elliptic curves defined over $\mathbb{F}_{2^m}$ where m has a small factor (say of size around 3{15). This may explain why some standards bodies have only recommended the use of elliptic curves over prime fields $\mathbb{F}_p$ and fields of the form $\mathbb{F}_{2^p}$ for prime p.

References

1. L. Adleman, J. De Marrais, and M.-D. Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In *ANTS-1 : Algorithmic Number Theory*, L.M. Adleman and M-D. Huang, editors. Springer-Verlag, LNCS 877, 28{40, 1994.
2. L.M. Adleman, M.-D. Huang. Primality testing and abelian varieties over finite fields. Springer LNM 1512, 1992.
3. M. Daberkow, C. Fieker, J. Klüners, M. Pohst, K. Roegner, M. Schörnig, and K. Wildanger. KANT V4. *J. Symbolic Computation*, 24, 267{283, 1997.
4. G. Frey. Weil descent. Talk at Waterloo workshop on the ECDLP, 1998. http://cacr.math.uwaterloo.ca/conferences/1998/ecc98/slides.html
5. S.D. Galbraith, S. Paulus and N.P. Smart. Arithmetic on super-elliptic curves. Preprint, 1998.
6. S.D. Galbraith and N.P. Smart. A cryptographic application of Weil descent. HP-Labs Technical Report, HPL-1999-70.
7. P. Gaudry. A variant of the Adleman-DeMarrais-Huang algorithm and its application to small genera. Preprint, 1999.
8. J.L. Hafner and K.S. McCurley. A rigorous subexponential algorithm for computation of class groups. *J. AMS*, 2, 837{850, 1989.
9. R. Hartshorne. Algebraic geometry. Springer GTM 52, 1977.
10. M.-D. Huang, D. Ierardi. Counting points on curves over finite fields. J. Symbolic Computation, 25, 1{21, 1998.
11. J.S. Milne. Jacobian Varieties. In *Arithmetic Geometry*, G. Cornell and J.H. Silverman, editors. Springer-Verlag, 167{212, 1986.
12. D. Mumford. Abelian varieties. Oxford, 1970.
13. J. Pila. Frobenius maps of abelian varieties and finding roots of unity in finite fields. Math. Comp., 55, 745{763, 1990.

Edit Probability Correlation Attack on the Bilateral Stop/Go Generator

Renato Menicocci[1] and Jovan Dj. Golic[2]

[1] Fondazione Ugo Bordoni
Via B. Castiglione 59, 00142 Roma, Italy
rmenic@fub.it
[2] School of Electrical Engineering, University of Belgrade
Bulevar Revolucije 73, 11001 Belgrade, Yugoslavia
golic@galeb.etf.bg.ac.yu

Abstract. Given an edit transformation de ned by the stop/go clocking in the bilateral stop/go generator, an edit probability for two binary strings of appropriate lengths is proposed. An e cient recursive algorithm for the edit probability computation is derived. It is pointed out how this edit probability can be used to mount a correlation attack on one of two clock-controlled shift registers. By estimating the underlying false alarm probability, it is shown that the minimum output sequence length required to be known for a successful attack is linear in the length of the shift register. This is illustrated by experimental correlation attacks on relatively short shift registers.

Key words. Stream ciphers, mutual clock control, bilateral stop/go, edit probability, correlation attack.

1 Introduction

Clock-controlled shift registers are an important tool for designing keystream generators for stream cipher applications. Several keystream generators based on clock-controlled shift registers are known to produce sequences with long period, high linear complexity, and good statistical properties (e.g., see [1]). The stop-and-go clocking is particularly appreciated in practice because of its suitability in high-speed applications. At any time, a stop/go shift register is clocked once if the clock-control input bit is equal to 1 (or 0) and is not clocked at all otherwise.

The bilateral stop/go generator (BSG) is a combination of two binary LFSRs, $LFSR_1$ and $LFSR_2$, which mutually clock-control each other (see [3],[4]). More precisely, a clock-control function derives two clock-control bits from the states of the two LFSRs. Each clock-control bit is used to stop/go clock-control one of the LFSRs. The two clock-control bits are never simultaneously equal to zero, so that at each step at least one of the two LFSRs is stepped. The output sequence is formed as the bitwise sum of the two stop/go clocked LFSR sequences.

No attacks on such a structure are reported in the open literature. The objective of this paper is to investigate whether a divide-and-conquer correlation

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 201{212, 1999.

attack on one of the LFSRs is possible. In such a correlation attack, the cryptanalyst would try to reconstruct the initial state of the chosen LFSR from a known segment of the keystream sequence by using an appropriate edit probability as a measure of correlation.

For the stop/go clocking, a speci c edit probability correlation attack on the alternating step generator is proposed in [2]. This generator consists of two stop/go clocked LFSRs and a regularly clocked clock-control LFSR. At each time, the clock-control bit de nes which of the two LFSRs is clocked, and the output sequence is obtained as the bitwise sum of the two stop/go LFSR sequences. The target of this correlation attack are the initial states of the individual stop/go clocked LFSRs.

Problems to be addressed in this paper are how to de ne the edit probability, how to compute it e ciently, and how to estimate the known keystream sequence length required for a successful correlation attack. The fact that the rst binary derivative of the BSG output sequence is bitwise correlated to the rst binary derivative of the output sequence of each of the stop/go clocked $LFSR_1$ and $LFSR_2$ suggests that a divide-and-conquer attack may be possible. The edit probability is based on an edit transformation taking into account the stop/go clocking in the BSG. By introducing a suitable partial edit probability, a recursive algorithm for computing the edit probability is derived.

Accordingly, a correlation attack on $LFSR_1$ based on the edit probability is proposed. More speci cally, this edit probability is de ned for two binary strings of appropriate lengths: a given input string corresponding to the output sequence of $LFSR_1$ when regularly clocked and a given output string corresponding to the rst binary derivative of the output sequence of the BSG. The (random) edit transformation consists of the stop/go clocking as in the BSG of the given input string X and a purely random binary string Y (corresponding to the unknown $LFSR_2$ sequence) according to X and an auxiliary purely random and independent binary clock-control string R, of the bitwise addition of the two stop/go clocked strings, and of taking the rst binary derivative of the combination string. The auxiliary binary clocking string R is introduced in order to enable a recursive computation. The edit probability is then de ned as the probability that a given input string is transformed into the given output string by the described random edit transformation.

In the proposed correlation attack, for every possible $LFSR_1$ initial state, an input string of su cient length is generated and the edit probability for a given output string is computed. The correct $LFSR_1$ initial state is then likely to belong to a set of states with the associated edit probability close to being maximal. This attack can be successful only if there is a su cient statistical distinction between the probability distributions of the edit probability when the input string is guessed correctly and randomly, respectively. By computer simulations, for an appropriate missing event probability, the underlying false alarm probability is approximated by an exponentially decreasing function of the string length. If L denotes the common length of the two LFSRs, the minimum

output sequence length required to be known for a successful attack is then linear in L. The time complexity of the attack is then estimated as $O(2^{L+3\log_2 L})$.

In Section 2, a more detailed description of the BSG is provided. The edit probability for the auxiliary clocking string and the recursive algorithm for its e cient computation are presented in Section 3. The underlying false alarm probability is estimated in Section 4 and the corresponding correlation attack is explained in Section 5. Experimental correlation attacks conducted by computer simulations are reported in Section 6. Conclusions are given in Section 7 and a number of tables displaying the statistics of the edit probability are presented in the Appendix.

2 Description of Bilateral Stop/Go Generator

As shown in Fig. 1, the output of the bilateral stop/go generator (BSG) is obtained by bitwise addition (modulo 2) of the output sequences of two binary linear feedback shift registers, LFSR_1 and LFSR_2, which mutually clock-control each other by stop/go clocking (see [3],[4]). It is assumed that LFSR_1 and LFSR_2 have primitive feedback polynomials of the same degree L. At each step, the output bit is assumed to be produced in the step-then-add manner as follows. Let $s^i_{L;t}$ and $s^i_{L-1;t}$, $i = 1;2$, denote the contents at step $t \quad 0$ of the stages at positions L and $L-1$, respectively, of LFSR_i. From input bits $s^1_{L;t}; s^1_{L-1;t}; s^2_{L;t}$, and $s^2_{L-1;t}$, a clock-control function h determines the clock-control bits c^1_{t+1} and c^2_{t+1}. From four binary inputs $a;b;c$, and d, $h(a;b;c;d)$ outputs $f1g$ if $(a;b) = (0;1)$, $f2g$ if $(c;d) = (0;1)$ ~~6~~ $(a;b)$, and $f1;2g$ otherwise. To get the BSG output bit o_t at step t, $t \quad 1$, we step LFSR_i, $i = 1;2$, or not depending on whether $c^i_t = 1$ or $c^i_t = 0$, respectively, and then we add modulo 2 the output bits ($s^1_{1;t}$ and $s^2_{1;t}$) of the shift registers.

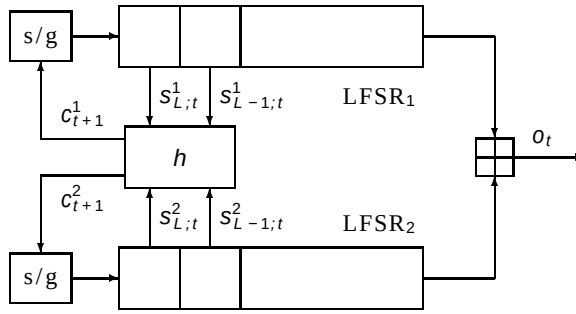


Fig. 1. The bilateral stop/go generator.

Some theory of the bilateral stop/go generator is exposed in [3]. In a few words, the state diagram of the BSG consists of $3 \quad 2^{L-2} - 1$ branched cycles each of length $T = 5 \quad 2^{L-2} - 1$. At any cycle state there is at most one branch. Every

branch has length 1 and starts with a state having no predecessor. By using L such that T is prime, the linear complexity of the sequence produced while the generator covers a cycle has a lower bound of the same order of magnitude as T (see [3],[4]).

We assume that the cryptanalyst knows the feedback polynomials of the two LFSRs and operates in the known plaintext scenario. The cryptanalyst's objective is then to reconstruct the secret-key-controlled LFSR initial states from a known segment of the keystream sequence.

In the sequel, we denote by A a sequence of symbols $a_1; a_2; \ldots$ and by A^n a string $a_1; a_2; \ldots a_n$ constituted by the first n symbols of A. When A is a binary sequence, we denote its first derivative by $\underline{A} = \underline{a}_1; \underline{a}_2; \ldots$, where $\underline{a}_t = a_t \quad a_{t+1}$, standing for modulo 2 addition.

Let $X = x_1; x_2; \ldots$ and $Y = y_1; y_2; \ldots$ denote two binary input sequences and let $C = c_1; c_2; \ldots$ denote a three-valued clock-control sequence, where $c_i \; 2 \; C$, $C = \{1\}; \{2\}; \{1; 2\}$. Let $O = G(X; Y; C) = o_1; o_2; \ldots$ denote the combination sequence produced from X and Y by the step-then-add bilateral stop/go clocking according to C, where X and Y correspond to the regularly clocked LFSR_1 and LFSR_2 sequences, respectively. Note that c_t determines which register is stepped in order to get the BSG output bit at step t.

We initially have $o_1 = x_2 \quad y_1$ if $c_1 = \{1\}$, $o_1 = x_1 \quad y_2$ if $c_1 = \{2\}$, and $o_1 = x_2 \quad y_2$ if $c_1 = \{1; 2\}$. Let $w_i(C^{s+1})$ denote the number of occurrences of the symbol $\{i\}$, $i = 1; 2$, in the string C^{s+1}. For simplicity, let $w_1(C^{s+1}) = l_1$ and $w_2(C^{s+1}) = l_2$. The number of occurrences of the symbol $\{1; 2\}$ in C^{s+1} is then $s+1-l_1-l_2$. Thus, for any $s \quad 0$, $o_{s+1} = x_{s+2-l_2} \quad y_{s+2-l_1}$. As for the generation of C, according to the BSG scheme from Fig 1, we have $c_1 = h(x_L; x_{L-1}; y_L; y_{L-1})$ and, for $s \quad 0$, we can readily write $c_{s+2} = h(x_{L+s+1-l_2}; x_{L+s-l_2}; y_{L+s+1-l_1}; y_{L+s-l_1})$.

Consequently, for input/output strings of finite length, we have $O^{n+1} = G^{n+1}(X^{n+2}; Y^{n+2}; C^{n+1})$, with $C^{n+1} = H^{0n+1}(X^{n+L}; Y^{n+L})$. Alternatively, we can write $O^{n+1} = F^{0n+1}(X^{n+L'}; Y^{n+L'})$, where $L^0 = \max(2; L)$ and F^0 represents the joint action of G and H^0.

3 Edit Probability for Auxiliary Clocking String

In this section, we adopt a simplified model for the BSG which allows us to define and recursively compute a suitable *edit probability*. The edit probability so defined can be used for inferring about the LFSR_1 initial state from a given BSG output segment.

Consider an auxiliary random sequence $R = r_1; r_2; \ldots$ which is used to replace the input sequence Y in the role of generating the input bits for the clock-control function h. The simplified BSG model is as follows. X, Y, and R are independent and purely random binary sequences (a sequence of independent uniformly distributed random variables over any finite set is called purely random). The clock-control string C^{n+1} is generated as follows. Initially, we have $c_1 = h(x_L; x_{L-1}; r_2; r_1)$ and, for $s \quad 0$, $c_{s+2} = h(x_{L+s+1-l_2}; x_{L+s-l_2}; r_{2s+4}; r_{2s+3})$,

where, as above, $I_1 = w_1(C^{s+1})$ and $I_2 = w_2(C^{s+1})$. We represent this by writing $C^{n+1} = H^{n+1}(X^{n+L}; R^{2n+2})$. The output string is generated as $O^{n+1} = G^{n+1}(X^{n+2}; Y^{n+2}; C^{n+1})$. The joint action of G and H is represented by F as $O^{n+1} = F^{n+1}(X^{n+L'}; Y^{n+2}; R^{2n+2})$, where $L' = \max(L; 2)$.

Now we can define a suitable *random edit transformation* and an associated *edit probability*. In the given model, we start by considering the string $O^n = F^n(X^{n+L'}; Y^{n+2}; R^{2n+2})$. The transformation of a given input string $X^{n+L'}$ into a given output string O^n, according to a random input string Y^{n+2} and an auxiliary random clocking string R^{2n+2}, defines a random edit transformation.

Let $Z^n = z_1; z_2; \ldots; z_n$ denote a given output string. The associated edit probability for a given input string $X^{n+L'}$ and a given output string Z^n is the probability that $X^{n+L'}$ is transformed into Z^n by a random edit transformation according to random Y^{n+2} and R^{2n+2}. Formally, we have

$$P(X^{n+L'}; Z^n) = \Pr\{F^n(X^{n+L'}; Y^{n+2}; R^{2n+2}) = Z^n \mid X^{n+L'}\}. \tag{1}$$

The statistically optimal edit probability (minimizing the error probability when deciding on $X^{n+L'}$ given Z^n) is then given as

$$\Pr\{X^{n+L'}; F^n(X^{n+L'}; Y^{n+2}; R^{2n+2}) = Z^n\} = P(X^{n+L'}; Z^n) \Pr\{X^{n+L'}\}. \tag{2}$$

As $\Pr\{X^{n+L'}\} = 2^{-(n+L')}$, the edit probability (1) is also statistically optimal.

Our objective is to examine whether the defined edit probability can be computed efficiently by a recursive algorithm whose computational complexity is significantly smaller than $O(2^{3n+4})$, which corresponds to the computation of (1) by the summation of the elementary probability $2^{-(3n+4)}$ over all Y^{n+2} and R^{2n+2} such that $F^n(X^{n+L'}; Y^{n+2}; R^{2n+2}) = Z^n$. To this end, we define the *partial edit probability* depending on the distribution of symbols in the clock-control string C^{n+1}.

For any $0 \le s \le n$, a pair $(I_1; I_2)$ is said to be *permissible* if $0 \le I_1; I_2 \le s+1$ and $I_1 + I_2 \ge s+1$. For a given s, the set of all the permissible values of $(I_1; I_2)$ is denoted by L_s. For any $1 \le s \le n$ and $(I_1; I_2) \in L_s$, the partial edit probability is defined as the conditional joint probability

$$P(I_1; I_2; s) = \Pr\{O^s = Z^s; w_1(C^{s+1}) = I_1; w_2(C^{s+1}) = I_2 \mid X^{s+L'}\} \tag{3}$$

where $O^s = G^s(X^{s+L'}; Y^{s+2}; C^{s+1})$ and $C^{s+1} = H^{s+1}(X^{s+L'}; R^{2s+2})$. The following theorem shows how to compute the edit probability efficiently, on the basis of a recursive property of the partial edit probability.

Theorem 1. *For any given $X^{n+L'_1}$ and Z^n, we have*

$$P(X^{n+L'}; Z^n) = \sum_{(I_1; I_2) \in L_n} P(I_1; I_2; n) \tag{4}$$

where the partial edit probability $P(I_1; I_2; n)$ is computed recursively by

$$P(l_1;l_2;s) = P(l_1 - 1;l_2;s-1)(1 \quad z_s \quad \underline{x}_{s+1-l_2})(1 \quad x_{L+s-l_2})x_{L+s-l_2-1}$$
$$+\frac{1}{8} \quad P(l_1;l_2-1;s-1)(1 \quad (1 \quad x_{L+s-l_2+1})x_{L+s-l_2})$$
$$+\frac{3}{8} \quad P(l_1;l_2;s-1)(1 \quad (1 \quad x_{L+s-l_2})x_{L+s-l_2-1}) \qquad (5)$$

for $1 \quad s \quad n$ *and all* $(l_1;l_2)$ 2 L_s, *with the initial values* $P(0;0;0) = \frac{3}{4}(1 \quad (1 \quad x_L)x_{L-1})$, $P(0;1;0) = \frac{1}{4}(1 \quad (1 \quad x_L)x_{L-1})$ *and* $P(1;0;0) = (1 \quad x_L)x_{L-1}$. *(For each* $0 \quad s \quad n$, *if* $(l_1;l_2)$ *is not permissible, then it is assumed that* $P(l_1;l_2;s) = 0$, *so that the corresponding terms in* (5) *are not computed.)*

Proof First observe that (4) is a direct consequence of (3) and (1).

Assume that $s \quad 2$. We partition all clock-control strings C^{s+1} into three subsets with respect to the value of the last symbol c_{s+1}. For simplicity of notation, the conditioning on $X^{s+L'}$ is removed from the probability (3) and all the resulting equations. Then (3) can be put into the form

$$\begin{aligned}
P(l_1;l_2;s) = \\
&\Pr\{\underline{o}_s = z_s | O^{s-1} = Z^{s-1}; w_1(C^s) = l_1 - 1; w_2(C^s) = l_2; c_{s+1} = \{1\}\} \\
&\quad \Pr\{c_{s+1} = \{1\} | O^{s-1} = Z^{s-1}; w_1(C^s) = l_1 - 1; w_2(C^s) = l_2\} \\
&\quad \Pr\{O^{s-1} = Z^{s-1}; w_1(C^s) = l_1 - 1; w_2(C^s) = l_2\} \\
&+ \Pr\{\underline{o}_s = z_s | O^{s-1} = Z^{s-1}; w_1(C^s) = l_1; w_2(C^s) = l_2 - 1; c_{s+1} = \{2\}\} \\
&\quad \Pr\{c_{s+1} = \{2\} | O^{s-1} = Z^{s-1}; w_1(C^s) = l_1; w_2(C^s) = l_2 - 1\} \\
&\quad \Pr\{O^{s-1} = Z^{s-1}; w_1(C^s) = l_1; w_2(C^s) = l_2 - 1\} \\
&+ \Pr\{\underline{o}_s = z_s | O^{s-1} = Z^{s-1}; w_1(C^s) = l_1; w_2(C^s) = l_2; c_{s+1} = \{1;2\}\} \\
&\quad \Pr\{c_{s+1} = \{1;2\} | O^{s-1} = Z^{s-1}; w_1(C^s) = l_1; w_2(C^s) = l_2\} \\
&\quad \Pr\{O^{s-1} = Z^{s-1}; w_1(C^s) = l_1; w_2(C^s) = l_2\}: \qquad (6)
\end{aligned}$$

The third factor in each addend of (6) is easily recognized to be the partial edit probability, of argument $s - 1$, appearing in the corresponding addend of (5).

Now, under the condition that $w_1(C^{s+1}) = l_1 - 1$ and $w_2(C^{s+1}) = l_2$, we have $c_{s+1} = h(x_{L+s-l_2}; x_{L+s-l_2-1}; r_{2s+2}; r_{2s+1})$, which produces $\{1\}$ if and only if $(x_{L+s-l_2}; x_{L+s-l_2-1}) = (0;1)$. Moreover, if $c_{s+1} = \{1\}$, then $\underline{o}_s = \underline{x}_{s+1-l_2}$. Similarly, under the condition that $w_1(C^{s+1}) = l_1$ and $w_2(C^{s+1}) = l_2$, we have $c_{s+1} = h(x_{L+s-l_2+1}; x_{L+s-l_2}; r_{2s+2}; r_{2s+1})$, which produces $\{2\}$ if and only if $(x_{L+s-l_2+1}; x_{L+s-l_2}) \in (0;1)$ and $(r_{2s+2}; r_{2s+1}) = (0;1)$. Moreover, if $c_{s+1} = \{2\}$, then $\underline{o}_s = \underline{y}_{s+1-l_1}$. Finally, under the condition that $w_1(C^{s+1}) = l_1$ and $w_2(C^{s+1}) = l_2$, we have $c_{s+1} = h(x_{L+s-l_2+1}; x_{L+s-l_2}; r_{2s+2}; r_{2s+1})$, which produces $\{1;2\}$ if and only if $(x_{L+s-l_2}; x_{L+s-l_2-1}) \in (0;1)$ and $(r_{2s+2}; r_{2s+1}) \in (0;1)$. Moreover, if $c_{s+1} = \{1;2\}$, then $\underline{o}_s = \underline{x}_{s+1-l_2} \quad \underline{y}_{s+1-l_1}$.

Consequently, we have (conditioned on $X^{s+L'}$) that

$$\Pr\{c_{s+1} = \{1\} \mid O^{s-1} = Z^{s-1}; w_1(C^s) = l_1 - 1; w_2(C^s) = l_2\} = (1 - x_{L+s-l_2}) x_{L+s-l_2-1} \quad (7)$$

$$\Pr\{c_{s+1} = \{2\} \mid O^{s-1} = Z^{s-1}; w_1(C^s) = l_1; w_2(C^s) = l_2 - 1\} = (1 - (1 - x_{L+s-l_2+1}) x_{L+s-l_2}) \ 1/4 \quad (8)$$

$$\Pr\{c_{s+1} = \{1,2\} \mid O^{s-1} = Z^{s-1}; w_1(C^s) = l_1; w_2(C^s) = l_2\} = (1 - (1 - x_{L+s-l_2}) x_{L+s-l_2-1}) \ 3/4. \quad (9)$$

Further, we get

$$\Pr\{\underline{o}_s = z_s \mid O^{s-1} = Z^{s-1}; w_1(C^s) = l_1 - 1; w_2(C^s) = l_2; c_{s+1} = \{1\}\} = 1 - z_s - \underline{x}_{s+1-l_2} \quad (10)$$

$$\Pr\{\underline{o}_s = z_s \mid O^{s-1} = Z^{s-1}; w_1(C^s) = l_1; w_2(C^s) = l_2 - 1; c_{s+1} = \{2\}\} = 1/2 \quad (11)$$

$$\Pr\{\underline{o}_s = z_s \mid O^{s-1} = Z^{s-1}; w_1(C^s) = l_1; w_2(C^s) = l_2; c_{s+1} = \{1,2\}\} = 1/2. \quad (12)$$

Equation (11) follows from $\underline{o}_s = \underline{y}_{s+1-l_1}$ by taking into account that y_{s+2-l_1} remains to be independent of y_{s+1-l_1} when conditioned on $O^{s-1} = G^{s-1}(X^{s-1+L'}, Y^{s+1}; C^s)$, $w_1(C^s) = l_1$, and $w_2(C^s) = l_2 - 1$, as this condition involves only Y^{s+1-l_1} (not y_{s+2-l_1}). Equation (12) is proved analogously.

Equation (5) is obtained from (6) by plugging in the determined probabilities.

For $s = 1$, the edit probability values are directly obtained from (3). When these values are expressed in terms of the unknown initial values by the recursion (5), a system of linear equations is obtained. The initial values are then determined by solving this system. □

The time and space complexities of the recursive algorithm corresponding to Theorem 1 are $O(n^3)$ and $O(n^2)$, respectively. Since the edit probability is exponentially small in the string length, the following normalization turns out to be computationally convenient: $P(X^{n+L'}; Z^n) = 2^{n+1} P(X^{n+L'}; Z^n)$. It results from the right-hand side of (5) and the initial values being multiplied by 2.

4 False Alarm Probability

In order to investigate whether a correlation attack based on the proposed edit probability can be successful, we develop a statistical hypothesis testing model similar to the one introduced in [2]. We start by considering the probability distribution of the edit probability $P(X^{n+L'}; Z^n)$ under the following two probabilistic *hypotheses*:

- H_0 (*correlated case*): $X^{n+L'}$ and $Y^{n+L'}$ are purely random and independent and $Z^n = F^{0n}(X^{n+L'}; Y^{n+L'})$ (that is, $Z^n = G^n(X^{n+2}; Y^{n+2}; C^{n+1})$ and $C^{n+1} = H^{0n+1}(X^{n+L}; Y^{n+L})$).
- H_1 (*independent case*): $X^{n+L'}$ and Z^n are purely random and independent.

This means that when generating the correlated samples, the third and the fourth input bits of the clock-control function h are taken from the input sequence Y as in the actual BSG scheme, rather than from an auxiliary random sequence. For the correlation attack to work, it is necessary that the separation between the probability distributions in the correlated and independent cases increases with the string length n, and the faster the increase, the smaller the string length required for successful decision making is. Analyzing the separation of the two probability distributions seems to be a difficult task from a theoretical point of view, but one can measure the separation experimentally.

We conducted systematic experiments for the normalized edit probability and produced histograms of the two distributions for each $n = 100, (10), 800$ on random samples of 1000 pairs $(X^{n+L'}, Z^n)$ generated according to H_0 (for fixed $L = 100$) and H_1, respectively. They show that the separation of interest increases with the string length n. It thus turns out that, for a sufficiently large n, the normalized edit probability is much larger in the correlated than in the independent case. For illustration, Tables 1 and 2 given in the Appendix display the observed minimum, maximum, mean, and median values along with the standard deviation of the normalized edit probability, for each $n = 100, (100), 800$, for the independent and correlated case, respectively.

As we deal with a decision making problem, the separation between the two distributions is measured by the false alarm probability (derived from the distribution under H_1) when the missing event probability (derived from the distribution under H_0) is fixed. Since the number of correct input strings is only one, reasonable values for the missing event probability seem to be $p_m = 0.1$ or $p_m = 0.05$. Therefore, in the statistical hypothesis testing considered, a threshold is set according to p_m and a tested input string is classified under H_0 or H_1 depending on whether the normalized edit probability is bigger or smaller than the fixed threshold. The false alarm probability p_f then becomes a function of n, and if and only if this function is decreasing, the separation between the two distributions increases with n, as desired. The value of n should be chosen so as to make p_f inversely proportional to the number of incorrect input strings. We evaluated thresholds and false alarm probabilities relative to the data collected for the above histograms. For illustration, Table 3 given in the Appendix displays the estimated threshold, P_{th}, and false alarm probability, p_f, for each $n = 100, (100), 800$.

The data collected show that, for each considered p_m, the estimated p_f decreases with n. Moreover, for large n, p_f appears to be following the exponential form $a b^n$, $b < 1$. As a consequence, the minimum string length n required for the expected number of false input string candidates to be reduced to about one is linear in the logarithm (to the base two) of the total number of tested input strings. The corresponding estimates of the parameters a and b were obtained by the least mean square approximation method applied to the logarithms to the base two of the false alarm probability estimates for $n = 100, (10), 800$ and are presented in Table 4 given in the Appendix. The parameters a and b were estimated on the first 10, 20, and 25 points for $p_m = 0.1$ and $p_m = 0.05$. The most

reliable estimates were obtained for the first 10 points. To be on the conservative side, the false alarm probabilities $p_f^{0.1}(n)$ and $p_f^{0.05}(n)$ can be approximated for large n by

$$p_f^{0.1}(n) \approx 0.542\ 0.986^n; \quad p_f^{0.05}(n) \approx 0.520\ 0.990^n. \tag{13}$$

5 Correlation Attack

In this section, we propose a correlation attack on the BSG based on the properties of the introduced edit probability. It is assumed that the LFSR feedback polynomials and a sufficiently long segment of the BSG output sequence are known to the cryptanalyst. The attack consists of two phases. The goal of the first phase is to recover the initial state of LFSR_1 by using the (normalized) edit probability defined in Section 3. Suppose a number of candidates for the LFSR_1 initial state is obtained in this way. Then, in the second phase, the correct initial states of LFSR_1 and LFSR_2 are reconstructed. The solution is very likely to be unique as the equivalent initial states, producing the same output sequence, are unlikely to exist.

We first produce n bits of the first binary derivative, Z^n, of the first $n+1$ successive bits of the known BSG output sequence. The string Z^n comes from the (unknown) output strings of the regularly clocked LFSR_1 and LFSR_2 of maximum possible length $n + L'$ (the actual lengths are random and depend on the unknown LFSR initial states). In the correlation attack on LFSR_1, for any possible LFSR_1 initial state, by using its linear recursion, we first generate $n + L'$ output bits which constitute the input string $X^{n+L'}$. Then, we compute the normalized edit probability $P(X^{n+L'}; Z^n)$ by the recursive algorithm derived in Section 3. This is repeated for the $2^L - 1$ possible initial states of LFSR_1 (the all zero state is excluded). Roughly speaking, the candidates for the correct LFSR_1 initial state are obtained as the ones with the computed normalized edit probability close to being maximal. More precisely, a threshold is set according to the missing event probability (for the correct hypothesis $\mathbf{H_0}$) which is fixed to a value which need not be very small (e.g., $p_m = 0.1$ or $p_m = 0.05$). Then, every possible initial state is classified as a candidate if the corresponding normalized edit probability is not less than the threshold. This threshold can be obtained experimentally, as described in Section 4.

Ideally, for n sufficiently large, there should remain only one candidate for the initial state of LFSR_1. This can happen if and only if the false alarm probability (for the alternative hypothesis $\mathbf{H_1}$) $p_f(n)$ defined in Section 4 is sufficiently small. Namely, since the expected number of false candidates for an average Z^n is $(2^L - 1)p_f(n)$, the correlation attack would be successful if and only if, approximately,

$$2^L p_f(n) \lesssim 1. \tag{14}$$

If $p_f(n)$ has the exponential form $a\,b^n$, where $b < 1$, then (14) reduces to

$$n \gtrsim \frac{L + \log_2 a}{-\log_2 b} \tag{15}$$

which means that the required output segment length is linear in the length of $LFSR_1$. By (13), the required output segment length is then estimated as

$$n \approx 49.2L - 43.4 \quad (16)$$

$$n \approx 69.0L - 65.1 \quad (17)$$

for $p_m = 0.1$ and $p_m = 0.05$, respectively.

Accordingly, we should obtain a relatively small number of candidate initial states for $LFSR_1$ in time $O(2^{L+3\log_2 L})$. These candidate initial states are then ranked in order of decreasing normalized edit probabilities. The candidate states are tested in the second phase of the attack, according to decreasing normalized edit probabilities. Namely, each candidate state is associated with each of the $2^L - 1$ possible initial states for $LFSR_2$. Each resulting LFSR state pair is then used to initialize the state of the BSG under attack. The corresponding output sequence is finally compared with the given BSG output sequence. All the solutions for the LFSR initial states are thus found in time $O(2^L)$.

6 Experimental Results

The attack described in the previous section was tested on short LFSRs by computer simulations, which verified that the attack can work in practice.

Some results of our experiments are shown in Table 5. In each experiment we used primitive LFSRs of the same length L. For $L = 14, 15,$ and 16, the needed BSG output string length, $n + 1$, was first estimated by (16). The experiments were also repeated by halving this string length. The good results (and time reduction) obtained for $L = 14, 15,$ and 16 when moving from n to $n/2$ motivated the choice of using $n = 500(400)$ instead of $n = 800(400)$, for $L = 17$. The thresholds for the normalized edit probability were obtained from the data collected for $n = 100, (10), 800$ (see Table 3 for $n = 100, (100), 800$). For $n = 325, 375$, we used interpolation.

We counted the number of $LFSR_1$ states giving rise to a normalized edit probability not smaller than the given threshold (*candidates*). For every candidate initial state for $LFSR_1$ we searched for a companion initial state for $LFSR_2$ and counted the joint solutions (*solutions*). Table 5 shows that a unique joint solution was always found. Finally, we determined the position of the $LFSR_1$ component of this solution in the list of $LFSR_1$ initial state candidates ranked in order of decreasing normalized edit probabilities (*rank*).

As for the $LFSR_1$ initial state candidates, we found that a candidate is very likely obtainable from the correct $LFSR_1$ initial state by a small positive or negative phase shift.

7 Conclusions

It is pointed out that the stop/go clocking in the bilateral stop/go keystream generator can be viewed as a random edit transformation of an input string into

one output binary string. The input string corresponds to the output sequence of $LFSR_1$ when regularly clocked and the output string corresponds to the first binary derivative of the keystream sequence. The output sequence of $LFSR_2$ and an auxiliary binary clock-control string are assumed to be independent and purely random. The related edit probability is then defined and a recursive algorithm for its computation is derived.

It is shown how the edit probability can be used to mount a correlation attack on $LFSR_1$. For the underlying statistical hypothesis testing problem, the false alarm probability is estimated by computer simulations. According to the experiments conducted, the minimum output sequence length required to be known for a successful attack is linear in the length, L, of the LFSRs. The time complexity of the attack is estimated as $O(2^{L+3\log_2 L})$. Successful experimental correlation attacks performed on relatively short shift registers demonstrate the effectiveness of the developed methodology.

References

1. D. Gollmann and W. G. Chambers, "Clock-controlled shift registers: A review," *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 525-533, May 1989.
2. J. Dj. Golic and R. Menicocci, "Edit probability correlation attack on the alternating step generator," Sequences and Their Applications - SETA '98, *Discrete Mathematics and Theoretical Computer Sciences*, C. Ding, T. Helleseth, and H. Niederreiter eds., Springer-Verlag, pp. 213-227, 1999.
3. K. Zeng, C. H. Yang, and T. R. N. Rao, "Large primes in stream-cipher cryptography," Advances in Cryptology - AUSCRYPT '90, *Lecture Notes in Computer Science*, vol. 453, J. Seberry and J. Pieprzyk eds., Springer-Verlag, pp. 194-205, 1990.
4. K. Zeng, C. H. Yang, D. Y. Wey, and T. R. N. Rao, "Pseudorandom bit generators in stream-cipher cryptography," *IEEE Computer*, vol. 24, no. 2, pp. 8-17, Feb. 1991.

Appendix

Table 1. Statistics of P on 1000 independent pairs $(X^{n+L'}; Z^n)$.

n	Min	Max	Mean	Median	Std Dev
100	6.341E-9	3.339E2	2.314E0	1.43E-1	1.294E1
200	4.307E-15	3.952E2	2.121E0	1.597E-2	1.69E1
300	4.279E-15	2.774E2	1.632E0	2.674E-3	1.347E1
400	2.669E-14	6.531E2	1.2E0	4.36E-4	2.168E1
500	2.246E-14	7.893E1	2.456E-1	1.13E-4	2.82E0
600	6.1E-20	3.019E2	4.643E-1	2.023E-5	9.68E0
700	3.915E-18	1.598E4	1.613E1	3.771E-6	5.055E2
800	5.145E-19	7.301E1	1.669E-1	7.699E-7	2.848E0

Table 2. Statistics of P on 1000 correlated pairs $(X^{n+L'}; Z^n)$.

n	Min	Max	Mean	Median	Std Dev
100	7.637E-2	7.097E3	1.097E2	2.486E1	4.006E2
200	8.051E-2	1.703E6	6.538E3	2.033E2	7.162E4
300	5.31E-2	3.59E7	1.269E5	1.863E3	1.62E6
400	3E-1	2.658E8	1.759E6	1.056E4	1.438E7
500	3.053E0	7.872E10	1.213E8	1.108E5	2.573E9
600	8.349E-1	3.3E12	6.27E9	6.357E5	1.222E11
700	4.942E1	2.123E12	4.933E9	5.135E6	7.462E10
800	3.484E0	1.987E13	7.507E10	2.374E7	8.822E11

Table 3. Estimation of thresholds and false alarm probabilities.

n	$P_{th}^{0.1}$	$P_{th}^{0.05}$	$p_f^{0.1}$	$p_f^{0.05}$
100	3.122E0	1.893E0	1.22E-1	1.67E-1
200	1.214E1	5.655E0	2.7E-2	4.8E-2
300	4.138E1	1.687E1	1.1E-2	1.9E-2
400	1.808E2	5.709E1	1E-3	3E-3
500	7.413E2	2.822E2	0	0
600	7.164E3	1.505E3	0	0
700	2.172E4	3.835E3	0	1E-3
800	7.587E4	1.891E4	0	0

Table 4. Estimation of a and b on 10 , 20 , and 25 points.

p_m	a	a	a	b	b	b
0.1	.542	.585	.551	.986	.986	.986
0.05	.520	.736	.731	.990	.987	.987

Table 5. Experimental results.

L	n	*threshold*	*candidates*	*solutions*	*rank*
14	650(325)	11290(75)	20(51)	1(1)	1(1)
15	700(350)	21720(109)	8(72)	1(1)	1(5)
16	750(375)	46070(145)	30(171)	1(1)	1(1)
17	500(400)	741(180)	35(139)	1(1)	26(29)

Look-Up Table Based Large Finite Field Multiplication in Memory Constrained Cryptosystems

(Extended Abstract)

M.A. Hasan

University of Waterloo, Waterloo, ON N2L 3G1, Canada.
Currently, on sabbatical with Motorola Labs., Schaumburg, IL 60196, USA.

Abstract. In this article, a look-up table based algorithm for GF(2^n) multiplication is presented. In each iteration of the algorithm, a group of bits of one of the input operands are examined and two look-up tables are accessed. The group size determines the table sizes but does not a ect the utilization of the processor resources. It can be used for both software and hardware realizations and is particularly suitable for implementations in memory constrained environment, such as, smart cards and embedded cryptosystems.

1 Introduction

In this article a look-up table based algorithm for GF(2^n) multiplication is presented. For the multiplication of a and b in GF(2^n), this algorithm considers a group of g bits of b at a time and provides the product $a \quad b$ in about $n=g$ iterations. The algorithm is di erent from the other look-up table based multiplication algorithms in two important ways. First, it utilizes the full width of the datapath of the processor in which the algorithm is implemented; secondly, it uses two tables{ one of which is precomputed during the eld initialization process and the other is computed during the run (or, execution) time of the multiplication operation. Because of the run time generation of one table, it directly a ects the multiplication operation and a mechanism is needed to quickly determine the table entries. Towards this end, a complementary algorithm for e ciently generating the table is presented, and this has enabled the multiplication algorithm to be implemented in memory constrained computing systems with a lower computation time.

The organization of this article is as follows. A brief discussion on the representation of the eld elements and the basic eld multiplication operation using the polynomial basis is given in Section 2. A brief description of the conventional bit-level algorithm for GF(2^n) multiplication is also given in this section. Then the new look-up table based multiplication algorithm is presented in Section 3. An e cient way to generate the look-up tables with fewer computations is developed in Section 4. Finally, concluding remarks are made in Section 5.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 213{221, 1999.

2 Preliminaries

2.1 Field Element Representation

The nite eld GF(2^n) has 2^n elements where n is a non-zero positive integer. Depending on the applications, the value of n can vary over a wide range. In cryptosystems, it can be as large as 512 or more. Each of the 2^n elements of GF(2^n) can be uniquely represented with a polynomial of degree up to $n-1$ with coe cients from GF(2). For example, if a is an element in GF(2^n), then one can have

$$a = A(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \quad + a_1x + a_0: \tag{1}$$

This type of representation of the eld elements is referred to as the *polynomial* or *standard basis* representation and has been used in many implementations.

For the polynomial basis representation as shown in (1), the addition of two eld elements of GF(2^n) is simply bit-wise XOR operation of the coe cients of the equal powers of x, that is, if a and b are in GF(2^n), then

$$a + b = A(x) + B(x) = \sum_{i=0}^{n-1} (a_i + b_i)x^i$$

where the addition in the bracket indicates an XOR or modulo 2 addition operation. On the other hand, the multiplication of the eld elements using the polynomial basis representation is much more complicated. It can be performed by rst multiplying $A(x)$ with $B(x)$ and then taking modulo $F(x)$ on $A(x)B(x)$, *i.e.*, if p is the product of a and b then

$$p = P(x) = A(x)B(x) \bmod F(x): \tag{2}$$

In (2), $F(x)$ is a polynomial over GF(2) of degree n which de nes the representation of the eld elements. Such $F(x)$ has to be an irreducible polynomial which has the following form:

$$F(x) = x^n + f_{n-1}x^{n-1} + f_{n-2}x^{n-2} + \quad + f_1x + 1 \tag{3}$$

where f_i's belong to $\{0, 1\}$. The choice of $F(x)$ can play an important role in determining the performance of the implementation of nite eld multipliers. For example, using irreducible *trinomials* which have only three non-zero coe cients, several researchers have proposed multipliers which provide advantages in terms of both speed and space [1,2]. Examples of other special forms of $F(x)$ include *all-one* polynomials and *equally-spaced* polynomials [3,4].

2.2 Bit-Level Multiplication Algorithm & Its Complexity

Let $F(x)$ be the irreducible polynomial de ning the representation of GF(2^n), and a, b, and p be any three elements of GF(2^n) such that $p = a \ b$ as de ned earlier. Then

$$\begin{aligned} P(x) &= A(x)B(x) \bmod F(x) \\ &= A(x)\left(b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \quad + b_1 x + b_0\right) \bmod F(x) \\ &= ((\quad (A(x)b_{n-1}x + A(x)b_{n-2})x + \quad)x + A(x)b_1)x + A(x)b_0 \bmod F(x) \\ &= ((\quad (A(x)b_{n-1}x \bmod F(x) + A(x)b_{n-2})x \bmod F(x) + \\ &\quad)x \bmod F(x) + A(x)b_1)x \bmod F(x) + A(x)b_0. \end{aligned} \tag{4}$$

In (4), the repeated operations involve multiplying $A(x)$ with x depending on the coe cients of $B(x)$ and then taking mod $F(x)$. These operations can be speci ed in terms of an algorithm as follows:

Algorithm 1 (Bit-Level Algorithm for GF(2^n) Multiplication)
Input: $A(x)$, $B(x)$ and $F(x)$
Output: $P(x) = A(x)B(x) \bmod F(x)$

Step 1.1 If $b_{n-1} = 1$, $P(x) := A(x)$
else $P(x) := 0$
Step 1.2 For $i = n - 2$ to 0 f
$P(x) := xP(x) \bmod F(x)$
If $b_i = 1$, $P(x) := P(x) + A(x)$
g □

After the nal iteration, $P(x)$ is the polynomial basis representation of the required product $a \quad b$. In Algorithm 1, a coe cient, *i.e.*, a single bit of $B(x)$ is checked in each iteration. The loop in Step 1.2 is executed $n - 1$ times. Its operation $P(x) := xP(x) \bmod F(x)$ can be realized by left shifting the coe cients of $P(x)$ to obtain $xP(x)$, and then subtracting/adding $F(x)$ from it if the nth coe cient of $xP(x)$ is 1. Assuming that 0 and 1 appear as the coe cients of $P(x)$ with an equal probability, the operation $P(x) := xP(x) \bmod F(x)$ takes $\frac{n-1}{2}$ polynomial additions on average. Similarly, there are $\frac{n-1}{2}$ more polynomial additions on average for the other operation $P(x) := P(x) + A(x)$. Thus, in Algorithm 1, there are approximately $n - 1$ polynomial additions on average. If we assume that the time to add two polynomials is T_{poly_add}, then the average computation time of a GF(2^n) multiplication can be approximated as:

$$T_{multi\_in\_GF(2^n)} \quad (n - 1) \quad T_{poly\_add}. \tag{5}$$

For a cryptosystem which uses a large value of n, the above bit-level algorithm may not be the best solution especially when speed is of major concern.

3 Group-Level Look-Up Table Based Multiplication

Let us divide the n coe cient bits of $B(x)$ into s groups of $g \quad 2$ bits each[1]. If n is not a multiple of g, then the number of bits in the most signi cant group is taken as $n \bmod g$. Thus,

[1] Unlike [5,6,7], the group size g in this work is expected to have a much smaller value. For the convenience of implementation, the value of g should divide w.

$$B(x) = x^{(s-1)g}B_{s-1}(x) + x^{(s-2)g}B_{s-2}(x) + \quad + x^{g}B_1(x) + B_0(x)$$

where

$$B_i(x) = \begin{cases} \sum_{j=0}^{g-1} b_{ig+j}x^j & 0 \le i \le s-2; \\ \sum_{j=0}^{(n \bmod g)-1} b_{ig+j}x^j & i = s-1. \end{cases} \tag{6}$$

Then

$$\begin{aligned} P(x) &= A(x)B(x) \bmod F(x) \bmod F(x) \\ &= A(x)\left[x^{(s-1)g}B_{s-1}(x) + x^{(s-2)g}B_{s-2}(x) + \quad + x^{g}B_1(x) + B_0(x)\right] \\ &\quad \bmod F(x) \\ &= \Big(\quad \big(\big(A(x)B_{s-1}(x) \bmod F(x) \big)x^g \bmod F(x) \\ &\quad + A(x)B_{s-2}(x) \bmod F(x) \big)x^g \bmod F(x) \\ &\quad \vdots \\ &\quad + A(x)B_1(x) \bmod F(x) \big)x^g \bmod F(x) \\ &\quad + A(x)B_0(x) \bmod F(x). \end{aligned} \tag{7}$$

Based on (7), now we have the following group-level intermediate algorithm for GF(2^n) multiplication.

Algorithm 2 (Group-Level GF(2^n) Multiplication)
Input: $A(x)$, $B(x)$ and $F(x)$
Output: $P(x) = A(x)B(x) \bmod F(x)$

Step 2.1 $P(x) := B_{s-1}(x)A(x) \bmod F(x)$
Step 2.2 For $k = s-2$ to 0 f
$P(x) := x^g P(x) \bmod F(x)$
$P(x) := P(x) + B_k(x)A(x) \bmod F(x)$
g $\square$

After the final iteration of the above algorithm, the coefficients of $P(x)$ correspond to the polynomial basis representation of the product $a \cdot b$. The loop in Step 2.2 is executed $s-1$ times. Since $P(x) = \sum_{i=0}^{n-1} p_i x^i$ is a polynomial of degree up to $n-1$ with coefficients from GF(2), the first operation of Step 2.2 can be written as follows:

$$P(x) := \underbrace{x^g \sum_{i=0}^{n-1-g} p_i x^i}_{X_1} + \underbrace{x^g \sum_{i=n-g}^{n-1} p_i x^i \bmod F(x)}_{X_2}. \tag{8}$$

In (8), X_1 is a g-fold left shift of the least significant $n-g$ coefficients of $P(x)$. The middle term X_2 depends on the g most significant bits of $P(x)$ as well as the

coe cients of $F(x)$. In practice, $F(x)$ does not change in a single cryptographic session, and in many cases, it remains unchanged as long as the dimension of the eld does not change. In such circumstances, a table which is hereafter referred to as the M (or, modulo) table can be created to store $x^g \sum_{i=n-g}^{n-1} p_i x^i \bmod F(x)$. The table entries are precomputed as part of the eld initialization process.

Referring to the second operation $P(x) := P(x) + B_k(x)A(x) \bmod F(x)$ in Step 2.2, let

$$X_3 = B_k(x)A(x) \bmod F(x). \tag{9}$$

The product $B_k(x)A(x)$ results in a polynomial of degree $n-g-2$. In order to reduce the degrees of $x^{n-g-2}, x^{n-g-3}, \ldots, x^n$ of $B_k(x)A(x)$, we need polynomial shifts and additions. For reasonable values of g, the term X_3 can however be directly read from a precomputed look-up table (hereafter referred to as T) and thus the above shift and addition operations can be avoided. Since in practice $n >> g$, the table can be more conveniently built to store $B_k(x)A(x) \bmod F(x)$ for all possible $B_k(x)$'s. The size of the table would then be $n2^g$ bits. However, unlike the previous M table, this table needs to be created on the fly each time a new $A(x)$ is chosen, and care must be taken to reduce the task to compute the table entries as it lies in the critical path of the loop in Algorithm 2.

Algorithms for generating the tables are given in Section 4. We wind up this section by incorporating the M and T tables into Algorithm 2. In this regard let $e = \sum_{i=0}^{g-1} e_i 2^i$ be an integer in the range $[0, 2^g - 1]$ and let the contents of the e-th entry of the M and T tables be

$$M[e] = (e_{g-1}x^{g-1} + e_{g-2}x^{g-2} + \cdots + e_0)x^n \bmod F(x), \text{ and}$$
$$T[e] = (e_{g-1}x^{g-1} + e_{g-2}x^{g-2} + \cdots + e_0)A(x) \bmod F(x),$$

respectively, then we have the following algorithm where $P_{s-1}(x) = \sum_{i=0}^{g-1} p_{n-g+i}x^i$.

Algorithm 3 (Look-up Table Based Group-Level Multiplication)
Input: $A(x)$, $B(x)$, $F(x)$, and the M table
Output: $P(x) = A(x)B(x) \bmod F(x)$

Step 3.1 Generate table T
Step 3.2 $P(x) := T[B_{s-1}(x = 2)]$
Step 3.3 For $k = s-2$ to 0 f
$X_1 := x^g \sum_{i=0}^{n-1-g} p_i x^i$
$X_2 := M[P_{s-1}(x = 2)]$
$X_3 := T[B_k(x = 2)]$
$P(x) := X_1 + X_2 + X_3$
g □

In Algorithm 3, the numbers of n-bit word read from the T and M tables are s and $s-1$, respectively. The algorithm requires $2(s-1)$ polynomial additions, or $2(s-1)\lceil \frac{n}{w} \rceil$ XOR instructions. For X_1, one needs $(s-1)\lceil \frac{n}{w} \rceil$ SHIFT instructions. For the evaluation of the indices of the two tables, one can use $2s-1$ SHIFT and $2s-1$ AND instructions. The cost of computing $T[B_i(x = 2)]$, for $i = s-1, \ldots, 1, 0$, is given in the following section.

4 Table Generation Algorithms

In this section, we consider algorithms for generating the T table. This table needs to be created with minimum possible delay to make its use feasible in the GF(2^n) multiplication operation. This algorithm is equally applicable to the generation of the M table where the speed is however not that critical.

There are g bits in $B_k(x)$ and table T has 2^g entries. The e-th entry of T is

$$T[e] = e(x) \quad A(x) \bmod F(x) \tag{10}$$

where $e(x) = \sum_{i=0}^{g-1} e_i x^i$ as assumed earlier. Thus, once the table has been generated, $B_k(x)A(x) \bmod F(x)$ is obtained by reading the table entry at

$$B_k(x = 2) = \sum_{j=0}^{g-1} b_{kg+j} 2^j :$$

In the sequel, the following g entries of the table, namely, $T[1]$, $T[2]$, $T[2^2]$, , and $T[2^{g-1}]$ are referred to as the *base* entries. From (10), the j-th base entry is

$$T[2^j] = x^j A(x) \bmod F(x);$$

from which one can write

$$T[2^{j+1}] = xT[2^j] \bmod F(x):$$

Thus, given the j-th base entry, the computation of the $(j + 1)$-st base entry takes a maximum of one polynomial addition. Thus, if the rst base entry $T[2^0]$ is initialized with $A(x)$, then the maximum number of XOR instructions needed to compute the remaining $g - 1$ base entries is $(g - 1) \lceil \frac{n}{w} \rceil$, and the corresponding average number is $\frac{1}{2}(g - 1) \lceil \frac{n}{w} \rceil$.

Lemma 1. If $A(x) \neq 0$, then all the entries expect $T[0]$ contain non-zero polynomials of degree up to $n - 1$. □

In order to compute the *regular* (i.e., non-zero and non-base) entries of the table, below we present two schemes. Both assume that the g base entries have already been computed and use these entries to compute the regular entries of the table.

4.1 Entry Computation on Demand

When $s < 2^g$, only a part of the table is read when Algorithm 3 is executed. In such cases, instead of computing all the regular entries and storing them, it is advantageous to compute the entries as they are needed. In this e ect, the following algorithm can be used.

Algorithm 4 (Regular Entry Computation on Demand)
Input: Index e, and base entries $T[2^i]$; $i = 0; 1; \quad ; g - 1$
Output: The e-th entry $T[e]$

Step 4.1 If $e_0 = 0$, $tmp := 0$
else $tmp := T[1]$
Step 4.2 For 1 to $g - 1$ f
If $e_i = 1$, $tmp := tmp + T[2^i]$
g
Step 4.3 $T[e] := tmp$ □

The number of polynomial additions needed in Algorithm 4, depends on the Hamming weight of the binary representation of e. On average, the algorithm requires $(g - 1)/2$ polynomial additions. Thus, the cost of computing $T[B_i(x = 2)]$, $0 \quad i \quad s - 1$, in Algorithm 3 is approximated as $s(g - 1)/2$ polynomial additions. The cost of creating all the regular entries of T, which will be used in our forthcoming discussions, is given below.

Corollary 2. The generation of all the regular entries of the T table using Algorithm 4 requires $2^{g-1}(g - 2) + 1$ polynomial additions. □

4.2 Entry Computation in Window Sequence

We now consider the case of $s \quad 2^g$ where each table entry is expected to be accessed at least once. In this case, one can create all the table entries. In this regard, the $2^g - g - 1$ regular entries are partitioned into $g - 1$ windows, namely, $W_1; W_2; \quad ; W_{g-1}$, where window W_i, $1 \quad i \quad g - 1$, consists of the following $2^i - 1$ entries $T[2^i + 1]; T[2^i + 2]; \quad ; T[2^i + 2^i - 1]$.

Lemma 3. For window W_i,

$$T[2^i + j] = T[2^i] + T[j]; \quad 1 \quad j \quad 2^i - 1: \tag{11}$$

□

Given the entries of windows W_j, $1 \quad j \quad i - 1$, the base entries $T[2^j]$, $0 \quad j \quad i$, one can compute an entry of W_i using only one polynomial addition. The entries are computed in the window sequence, *i.e.*, the entries of W_i are computed only after the entries of W_{i-1} have been computed. The entries within W_i can however be computed in any order. A systemic way to obtain all the regular entries using the window-by-window updating scheme is given below.

Algorithm 5 (Computation of Regular Entries in Window Sequence)
Input: Base entries $T[2^i]; i = 0; 1; \quad ; g - 1$
Output: All regular entries of T

For $i = 1$ to $g - 1$ f
For $j = 1$ to $2^{i-1} - 1$ f
$T[2^i + j] := T[2^i] + T[j]$
g
g □

In Algorithm 5, the loop with index i corresponds to the window computing. Since there are $2^i - 1$ entries in window W_i, the cost for generating entries of all the $g - 1$ windows is

$$(2^1 - 1) + (2^2 - 2) + \quad + (2^{g-1} - 1) = 2(2^{g-1} - 1) - g + 1 \qquad (12)$$

polynomial additions.

5 Conclusions

In the proposed GF(2^n) multiplication algorithm, the operand dependent table is accessed to read s n-bit entries. Thus, on average an entry of the table is read $s=2^g$ times. Consequently, to have computational advantages of the usage of the table, we should choose g such that $d\frac{n}{w}e \quad 2^g$. To use a 32 bit processor for implementing an elliptic curve cryptosystem, where n can have a value of a couple hundreds, if g is chosen to be four it appears from our experience that the proposed multiplication algorithm would perform its best.

Acknowledgment

This work was done when the author was with Motorola Labs., Schaumburg, IL, USA on a sabbatical leave from the University of Waterloo, Waterloo, Canada. The author wishes to thank Larry Puhl and Ezzy Dabbish for their encouragement to pursue this work. Thanks are also due to Dean Vogler, Tom Messerges and L. Finkelstein, for their help with the various computing resources of the labs.

References

1. E. D. Mastrovito, *VLSI Architectures for Computations in Galois Fields*. PhD thesis, Dept. Elect. Eng., Linköping University, Linköping, Sweden, 1991.
2. C. Koc and B. Sunar, \Mastrovito Multiplier for All Trinomials," *IEEE Trans. Computers*, 1999.
3. T. Itoh and S. Tsujii, \Structure of Parallel Multipliers for a Class of Fields GF(2^m)," *Inform. and Comp.*, vol. 83, pp. 21{40, 1989.
4. M. A. Hasan, M. Z. Wang, and V. K. Bhargava, \Modular construction of low complexity parallel multipliers for a class of nite elds GF(2^m)," *IEEE Trans. Comput.*, vol. 41, pp. 962{971, Aug. 1992.
5. G. Harper, A. Menezes, and S. Vanstone, \Public-key cryptsystems with very small key lengths," in *Advances in Cryptology- EUROCRYPT '92, Lecture Notes in Computer Science*, pp. 163{173, Springer-Verlag, 1992.
6. E. Win, A. Bosselaers, S. Vandenberghe, P. D. Gersem, and J. Vandewalle, \A Fast Software Implementation for Arithmetic Operations in GF(2^n)," in *Advances in Cryptology- ASIACRYPT '96, Lecture Notes in Computer Science*, pp. 65{76, Springer, 1996.

7. C. Koc and T. Acar, \Montgomery Multiplication in GF(2^k)," *Design, Codes and Cryptography*, vol. 14(1), pp. 57{69, Apr. 1998.
8. J. Guajardo and C. Paar, \E cient Algorithms for Elliptic Curve Cryptosystems," in *Advances in Cryptology- CRYPTO '97, Lecture Notes in Computer Science*, pp. 342{356, Springer-Verlag, 1997.
9. C. Paar, \A New Architecture for a Parallel Finite Field Multiplier with Low Complexity Based on Composite Fields," *IEEE Trans. Computers*, vol. 45(7), pp. 856{ 861, 1996.
10. Certicom Research, \GEC1: Recommended Elliptic Curve Domain Parameters," in *Standards for E cient Cryptography Group*, http://www.secg.org, 1999.

On the Combined Fermat/Lucas Probable Prime Test*

Siguna Müller

University of Klagenfurt, Dept. of Math., A-9020 Klagenfurt, Europe
siguna.mueller@uni-klu.ac.at

Abstract. Based on the well-known Baillie/Wagsta suggestion [2] we introduce a rapid pseudoprimality test with high con dence. The test is extremely fast and only requires evaluation of power polynomials and the Lucas V-sequence. This is in contrast to the original version, where usually the more cumbersome evaluation of the Lucas U-sequence is required as well. We analyze the underlying properties of the proposed test and give a characterization of the pseudoprimes. Software and hardware evaluation methods for both modular exponentiation and evaluation of recursion sequences are widely employed and very e cient. Therefore the test can be run at low cost for varieties of di erent bases/parameters. The number of those that pass the test are of great interest. We exhibit the exact number of these \liars".

1 Motivation and Background

1.1 Pseudoprimality Testing Based on the Power Function

Most of the pseudoprimality tests originate in some sense on Fermat's Little Theorem $a^{n-1} \equiv 1 \bmod n$: The congruence is ful lled for any $a \in \mathbb{Z}_n$ when n is prime. On the other hand, composite numbers that are Carmichael numbers, also ful ll the condition for any base a. Consequently, the most common method for determining whether or not a number is prime is the *Strong Probable Prime Test* (cf. e.g. [18]), which is a re nement of the Fermat test. It is known that any pseudoprime n to this test ($\mathrm{spsp}(a)$) is a Fermat pseudoprime, for which additionally $_2(ord_p(a))$ is a constant value c for all primes p dividing n; where $_2(b)$ denotes the largest power of 2 dividing b. The strong pseudoprime test has been implemented in many di erent Algebra systems. In most of these, the rst prime numbers are being used as bases. This test seems to be a good indicator for the primality of n. In particular, the variation of the bases are expected to guarantee a high con dence of the test. However, examples of composite numbers are known (cf. [1], [3], [7]) which are strong pseudoprimes to various pre-speci ed prime bases. In particular, F. Arnault found a 337−digit number that is a strong pseudoprime to all 46 prime bases up to 200. The existence of such composites provides incentive to create other tests which are similarly fast, but which have fewer, or at least di erent pseudoprimes.

* Research supported by the Österreichischen Fonds zur Förderung der wissenschaftlichen Forschung, FWF-Project no. P 13088-MAT

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 222{235, 1999.

1.2 Pseudoprimality Testing Based on the Lucas Sequences

Let P and Q be integers and $D = P^2 - 4Q$. The Fundamental, respectively Primordial Lucas sequence, associated with the parameters $P; Q;$ is de ned by means of the second-order recursion relation $U_{m+1} = PU_m - QU_{m-1}$; respectively $V_{m+1} = PV_m - QV_{m-1}$; with initial terms $U_0(P;Q) = 0$, $U_1(P;Q) = 1$, $V_0(P;Q) = 2$, and $V_1(P;Q) = P$. The counterpart to the Fermat test involving the Lucas sequences is based on the congruences $V_n(P;Q) \equiv P \bmod n;$ and $U_{n-(\frac{D}{n})}(P;Q) \equiv 0 \bmod n;$ where n is an odd prime and $(D;n) = 1$. If an odd composite number n satis es the former, respectively latter congruence, then it is called a Dickson pseudoprime, respectively Lucas pseudoprime, for P and Q ($\mathrm{Lpsp}(P;Q)$). Conditions are known (cf. [8], [9], [14]) for pseudoprimes for the V-test with respect to all parameters P and Q and examples of such numbers have been found (cf. [6], [15]). It turns out that pseudoprimes to the U-test can be characterized in a similar way to the Fermat test, when the rank of appearance is employed as counterpart of the order of a group element.

De nition 1 *Let $U = U(P;Q)$ be the Lucas sequence of the rst kind and let m be any integer. The rank of appearance $(m;P;Q)$ modulo m (or simply (m)) is the smallest integer l, if it exists, such that $U_l(P;Q) \equiv 0 \bmod m$.*

It can be shown that $(m;P;Q)$ exists if $p \mid Q$ for all prime divisors p of m (cf. [4]). If p is any prime with $(p;QD) = 1$, then it is known that the rank of appearance is a divisor d of $p - (\frac{D}{p})$. In this vein, a Lucas pseudoprime is a composite number for which $(p;P;Q) \mid (n - (\frac{D}{n}))$ for all primes p dividing n. Similarly to the Strong Probable Prime Test, a stronger version of the Lucas test can be obtained by means of the *Strong Lucas Probable Prime Test* $U_d(P;Q) \equiv 0 \bmod n;$ or $V_{2^r d}(P;Q) \equiv 0 \bmod n$ for some $r;\ 0 \le r < s;$ where $n - (\frac{D}{n}) = 2^s d$, $2 \nmid d$. It has been shown in [11] that pseudoprimes to the Strong Lucas Probable Prime Test (in short sLpsp) can be characterized in the following way.

Lemma 1 *Let n be a positive, odd composite integer and $Q \in \mathbb{Z}_n$. Then n is a $\mathrm{sLpsp}(P;Q)$ if and only if it is a $\mathrm{Lpsp}(P;Q)$ and $_2(\ (p;P;Q))$ is a constant value c for all prime factors p of n.*

Nevertheless, sLpsps can be constructed that pass the test for variations of parameters P and/or Q (cf. [11]).

1.3 Combined Tests

In [2] Baillie and Wagsta established a compositeness test that is based on a combination of the Strong Probable Prime Test and the Strong Lucas Probable Prime Test that seems to be very powerful. Indeed, nobody has been able to either calculate a composite number that passes the test, or to prove the non-existence of such pseudoprimes. Their test seems to heavily rely on the choice of the parameters in that they present two speci c algorithms for selecting the

parameters for the Lucas test. In [10] this selection process was replaced by another, faster search routine for the parameter P, such that $\frac{P^2-4Q}{n} = -1$, where $Q \in \{2, -2\}$ and $n \equiv 1 \bmod 24$. By exhaustive search until 10^{13} no composite number was found that passes this combined test according to the parameter choice suggested in [10]. Recently, an extremely reliable primality test that utilizes combinations of different types of tests has been presented in [5]. The novel idea of this test is to replace the modular integer arithmetic by arithmetic in residue rings $\mathbb{Z}[x]/(n, f(x))$ where $f(x)$ is a quadratic polynomial and n an odd positive integer that is to be tested for primality.

1.4 The Main Goal of This Paper

Since the combination of different *types* of primality tests seem to be much more efficient than simply varying the parameters/bases w.r.t. one type of test, we suggest a modification of the combined probable prime test introduced in [2]. As with any pseudoprimality test we want the following properties to be fulfilled.

Efficient and easy to implement algorithms: From a practical viewpoint the rapid evaluation of the underlying testing functions is of essential interest. But efficiency in evaluation is actually one of the most attractive features of recurrence sequences. In fact, it is known that the evaluation of the sequence $V_k(P, Q)$ has complexity $O(ld(k))$ (cf. [17]). It is probably due to the fact that these evaluation algorithms are already frequently used and easy to implement, that the Fermat/Lucas approach [2] has received so much attention.

High confidentiality: Although the Fermat Test is fulfilled for all bases a when n is a Carmichael number, an analysis of the Lucas Test has shown (cf. [12]) that, if n is composite, the Lucas U-test cannot be fulfilled for all P and Q with $(n, DQ) = 1$. Thus, by replacing the Fermat- by the Lucas test we can avoid the existence of Carmichael numbers. This means that there are no Lucas pseudoprimes with respect to all P and Q. Even more, the author has also shown a stronger result [12], namely that there are no composite integers n which are Lucas pseudoprimes for a fixed value of $Q \in \mathbb{Z}_n$ and all varied values of P (or vice versa) with discriminants $D = P^2 - 4Q$ coprime to n. Clearly, since there are no Carmichael numbers w.r.t. variations of P in the Lucas test $U_{n-(D/n)}(P, Q_0)$ when $Q = Q_0$ is fixed, one can expect that the number of parameters P that pass the *strong* Lucas test w.r.t. a fixed parameter $Q = Q_0$ is even much smaller. Moreover, if these Lucas tests are even combined with a Fermat test w.r.t. the base Q_0 then the number of parameters P that pass, when n is composite, will be much smaller still.

An exact formula for the number of liars: Whenever a probable prime tests allows the existence of pseudoprimes, then it certainly is of great interest to establish the number of such \liars" i.e. those parameters for which the test passes. In this paper we establish this number of parameters P in our specific combined Fermat/Lucas test.

The goal of this paper is to capture these requirements. We introduce a speci c modi cation of the test suggested in [2].

1.5 The Proposed Test and the Main Results

In detail, we will investigate the combined test that consists of checking that, (1), n is a spsp(Q_0) for a selected base $Q_0 \in \mathbb{Z}_n$, and, (2), n is a slpsp($P_i; Q_0$) for di erent $P_i \in \mathbb{Z}_n$ such that $\frac{P_i^2 - 4Q_0}{n} = -1$.

We rstly will derive various necessary conditions for composite numbers to pass this test. Based on these conditions we will be able to show that a wide class of composite numbers cannot pass the test described above. Contrary to the more sophisticated arithmetic of [5] the test we propose only requires modular arithmetic of the power functions and of recurrence sequences. A detailed analysis of the test will demonstrate its e ciency. As our main result we will, depending on the parameter Q_0, establish the exact formula for the number of these P_i that pass the proposed combined test. This number turns out to be extremely small, which also will be demonstrated by several numerical examples.

2 Some Preliminaries

2.1 The Rank of Appearance of the Lucas Sequence

Let $U = U(P; Q)$ be the Lucas sequence of the rst kind, let $m; m_i$ be integers, and p any odd prime. Throughout, we will assume that $(mm_i; QD) = 1$ and $(p; QD) = 1$. Then the rank of appearance of (U) is known to have the following properties (cf. [4], [18], [19])

$$m \mid U_k(P; Q) \text{ if and only if } \alpha(m; P; Q) \mid k; \tag{1}$$

$$\alpha(p; P; Q) \mid \alpha(p - \left(\frac{D}{p}\right)); \tag{2}$$

$$\alpha(p; P; Q) \mid \frac{p - (D/p)}{2} \text{ if and only if } \left(\frac{Q}{p}\right) = 1; \tag{3}$$

$$\alpha(\operatorname{lcm}(m_1; \ldots; m_k)) = \operatorname{lcm}(\alpha(m_1); \ldots; \alpha(m_k)). \tag{4}$$

2.2 The Number of Parameters with the Same Rank of Appearance

In what follows, let $Q \in \mathbb{Z}$ be xed and let $P \in \mathbb{Z}$ be arbitrary with D coprime to a xed prime p. Whenever we want to stress that Q is a special xed value, we will write Q_0 instead of Q. Further, we will let $D(P) = P^2 - 4Q_0$.

De nition 2 *Let $Q_0 \in \mathbb{Z}_p$, $\in \{-1; 1\}$ be xed, and $d > 1$ be any divisor of $p - $. The function $(d; Q_0;)$ is de ned to be the number of distinct values of P modulo p for which $\alpha(p; P; Q_0) = d$.*

Remark 1 *Note that* $(d;Q_0;\ )$ *counts the values of* P *with both* $\frac{D(P)}{p} =$ *and* $(p) = dj(p - \ )$ *if* $d > 2$*:*

The following property has been proved in [13].

Proposition 1 *Let* $d \quad 2$ *and suppose* d *is a divisor of* $p - \ ;$ *where* $\ 2\ f1; -1g$*:*

1. *If* $\frac{Q_0}{p} = -1$ *then* $(d;Q_0;\ ) = \begin{cases} (d); & \text{if } {}_2(d) = {}_2(p - \ ); \\ 0; & \text{otherwise.} \end{cases}$
2. *If* $\frac{Q_0}{p} = 1$ *then* $(d;Q_0;\ ) = \begin{cases} (d); & \text{if } {}_2(d) < {}_2(p - \ ); \\ 0; & \text{otherwise.} \end{cases}$

3 The Number of P That Pass the Lucas Test for a Fixed $Q = Q_0$

Proposition 2 *For any odd prime* p *and any positive integers* $k;n;$ *and the following is true.*

$U_k(P;Q) \quad 0 \mod n$ *i* $V_{2^i k}(P;Q)\ 6\ 0 \mod n$ *for every* $i \quad 0;$
if either $U_k(P;Q)$ *or* $V_k(P;Q) \quad 0 \mod n$ *then* $U_{2k}(P;Q) \quad 0 \mod n;$
$U_{2k}(P;Q) \quad 0 \mod p$ *i either* $U_k(P;Q)$ *or* $V_k(P;Q) \quad 0 \mod p$ *:*

Proof. The rst two assertions are obvious. So is the fourth when considering the well known identity $U_{2k}(P;Q) = U_k(P;Q)V_k(P;Q)$ and the fact that p cannot simultaneously divide both $U_k(P;Q)$ and $V_k(P;Q)$. □

In this section we establish the number of zeros $P\ 2\ \mathbb{Z}_p$ of U_k and V_k in dependence of the signature (cf. [19]) $\frac{P^2-4Q}{p}$ *:*

Theorem 1 *Let* p *be an odd prime,* $k;$ *positive integers,* $(k;p) = 1$*, and* $\ 2\ f-1;1g$ *a constant. For a xed value of* Q_0*,* $(Q_0;p) = 1$*, the number of distinct numbers* $P \mod p$ *with* $\frac{D(P)}{p} =$ *and* $U_k(P) \quad 0 \mod p$ *, is given in the following way.*

1. *For* $\frac{Q_0}{p} = -1$ *as* $\begin{cases} \frac{(k;p-\ )}{2}; & \text{when } {}_2(k) \quad {}_2(p - \ ); \\ 0; & \text{otherwise,} \end{cases}$
2. *for* $\frac{Q_0}{p} = 1$ *as* $\begin{cases} \frac{(k;p-\ )}{2} - 1; & \text{when } {}_2(k) \quad {}_2(p - \ ); \\ (k;p - \ ) - 1; & \text{otherwise.} \end{cases}$

Proof. We rst count the values of P modulo p and then modulo p . Now, (1) asserts that $U_k(P;Q_0) \quad 0 \mod p$ i $(p;P;Q_0)jk$. Moreover, $(p;P;Q_0)$ always divides $p - \ ,$ hence we obtain the condition

$$(p;P;Q_0)j(k;p - \ ): \tag{5}$$

The number of integers $P \in \mathbb{Z}_p$ with $(p;P;Q_0) = t|(p-\ )$ was defined to be $(t;Q_0;\ )$. By Theorem 1 the number of the P's depends on the quadratic residue symbol $\frac{Q_0}{p}$. We consider the two possibilities as separate cases.

1. $\frac{Q_0}{p} = -1$. Then the number of distinct P's, for $_2((k;p-\ )) = !$, is

$$\sum_{\substack{d|(k;p-\ ) \\ {}_2(d) = {}_2(p-\ )}} (d) = \sum_{d' \mid \frac{(k;p-\ )}{2^!}} (2^! d') = (2^!) \sum_{d' \mid \frac{(k;p-\ )}{2^!}} (d') = \frac{(k;p-\ )}{2};$$

provided $_2(k) \geq {}_2(p-\ )$. However, for $_2(k) < {}_2(p-\ )$ there are no integers P with the desired properties, since

$$\sum_{\substack{d|(k;p-\ ) \\ {}_2(d) = {}_2(p-\ )}} (d) = 0:$$

2. For $\frac{Q_0}{p} = 1$ the desired number is, if $_2(k) \geq {}_2(p-\ )$,

$$\sum_{\substack{d|(k;p-\ ) \\ {}_2(d) < {}_2(p-\ ) \\ d>1}} (d) = \sum_{\substack{d|(k;p-\ ) \\ d>1}} (d) - \sum_{\substack{d|(k;p-\ ) \\ {}_2(d) = {}_2(p-\ )}} (d) = \frac{(k;p-\ )}{2} - 1;$$

and, if $_2(k) < {}_2(p-\ )$,

$$\sum_{\substack{d|(k;p-\ ) \\ {}_2(d) < {}_2(p-\ ) \\ d>1}} (d) = \sum_{\substack{d|(k;p-\ ) \\ d>1}} (d) = (k;p-\ ) - 1:$$

This completes the part of the proof where the module is a prime p.

For a prime power module p we investigate the derivative $U_k'(P;Q_0)$, which can easily found to be $\frac{kV_k(P;Q_0) - PU_k(P;Q_0)}{P^2-4Q_0}$. However, $U_k'(P;Q_0) \not\equiv 0 \bmod p$, since, as $p|U_k(Q)$ and $(p;P;Q_0)|k$, the contrary would imply $p|V_k$, which cannot occur. It follows that for each zero of $U_k(P;Q_0) \equiv 0 \bmod p$ there is exactly one zero of $U_k(P;Q_0) \equiv 0 \bmod p$ (which is congruent to this zero modulo p). This completes the proof. □

Theorem 2 *Under the hypotheses of Theorem 1, the number of parameters P with $V_k(P;Q_0) \equiv 0 \bmod p$ for a fixed $Q_0 \in \mathbb{Z}_p$ is,*

when $\frac{Q_0}{p} = -1$*; given as* $\begin{cases} (k;\frac{p-}{2}); & \text{for } {}_2(k)+1 = {}_2(p-\ ); \\ 0; & \text{otherwise;} \end{cases}$

when $\frac{Q_0}{p} = 1$*; given as* $\begin{cases} (k;p-\ ); & \text{for } {}_2(k)+1 < {}_2(p-\ ); \\ 0; & \text{otherwise.} \end{cases}$

Proof. By Proposition 2 we need to count those P for which $U_{2k}(P;Q_0) \equiv 0 \bmod p$ and additionally $U_k(P;Q_0) \not\equiv 0 \bmod p$: Consider the case $\left(\frac{Q_0}{p}\right) = -1$. Then by Theorem 1 the number of P^0s, for $_2(k) < _2(p-)$, is $\frac{(2k;p-)}{2} - \frac{(k;p-)}{2} = 0$, for $_2(2k) = _2(k) + 1 = _2(p-)$, the number is $\frac{(2k;p-)}{2} = (k;\frac{p-}{2})$, and for $_2(k) + 1 < _2(p-)$ this number obviously is $0 - 0$. In a similar manner the result follows for $\left(\frac{Q_0}{p}\right) = 1$. □

4 The Proposed Strong Fermat/Strong Lucas Combination

4.1 Some Fundamentals

A more speci c form of the Fermat pseudoprimes w.r.t. the base Q are the Euler pseudoprimes (Epsps) n that, although composite, satisfy $Q^{\frac{n-1}{2}} \equiv \left(\frac{Q}{n}\right) \bmod n$: In terms of the Lucas sequences it is known that for any odd primes n with $(n;Q) = 1$, either $U_{(n-(D=n))=2} \equiv 0 \bmod n$ or $V_{(n-(D=n))=2} \equiv 0 \bmod n$, according as $\left(\frac{Q}{n}\right) = 1$ or -1. An odd composite integer n such that $(n;QD) = 1$ that satis es the corresponding congruence, respectively, is called an Euler-Lucas pseudoprime with parameters (P,Q), abbreviated ELpsp$(P;Q)$.

Any Euler pseudoprimes w.r.t. the base Q where $\left(\frac{Q}{n}\right) = -1$ is already a spsp(Q). Similarly (cf. [18]), if n is an ELpsp$(P;Q)$ and either $\left(\frac{Q}{n}\right) = -1$ or $n - (n) \equiv 2 \bmod 4$, then n is a sLpsp$(P;Q)$.

4.2 Description of the Proposed Test

The test for one P:

1. Choose $Q = Q_0 \in \mathbb{Z}_n$ such that $\left(\frac{Q_0}{n}\right) = -1$:
2. If n is not an Epsp(Q_0) then return \n is composite" , otherwise go to step 3.
3. Select $P \in \mathbb{Z}_n; P \neq 0$, such that for $D(P) = P^2 - 4Q_0$ one has $\left(\frac{D(P)}{n}\right) = -1$.
4. [ELpsp$(P;Q_0)$?] If $V_{\frac{n+1}{2}}(P;Q_0) \equiv 0 \bmod n$ then return \n is composite", otherwise return \n is probably prime".

Obviously, any odd prime n passes the proposed test. If an odd composite number passes the test, then it is both a spsp(Q_0) and a sLpsp$(P;Q_0)$.

The test, of course can even be made more powerful, if the conditions are checked for variations of parameters. *In the following, we will keep $Q = Q_0$ xed and check condition 4 for di erent choices of $P = P_i$.* Clearly, if a composite number n passes the test for the selected choices of P_i then n is a sLpsp$(P_i;Q_0)$ for all these P_i. For the remainder of the paper we will concern ourselves with the question of establishing the number of P_i for which any composite integer can pass the proposed test.

5 Fundamental Properties of the Proposed Test

Let here and in the following (n), respectively (p) denote the Jacobi, respectively Legendre symbol $\left(\frac{D}{n}\right)$, respectively $\left(\frac{D}{p}\right)$, where p is any odd prime. We will assume throughout that $(D;n) = 1$.

Theorem 3 *Let the odd integer n be a* spsp(Q_0) *for* $\left(\frac{Q_0}{n}\right) = -1$*: Then a necessary condition that* $V_{\frac{n-(n)}{2}}(P;Q_0) \quad 0 \mod n$ *is ful lled for at least one integer* $P \not\equiv 0$*, is that for all prime divisors p of n*

$$\begin{cases} (p) = 1; & \text{if } \left(\frac{Q_0}{p}\right) = 1; \\ (p) = (n); & \text{if } \left(\frac{Q_0}{p}\right) = -1: \end{cases}$$

Proof. Let p be an arbitrary prime divisor of n.

Consider rstly the case that $\left(\frac{Q_0}{p}\right) = 1$. Then it follows from Theorem 2 that the number of P with $V_{\frac{n-(n)}{2}}(P;Q_0) \quad 0 \mod n$ is di erent form zero only when $_2(n-(n)) < _2(p-(p))$: Since n is odd and $(D;n) = 1$ by hypothesis, we obtain that necessarily $p \quad (p) \mod 4$: By hypothesis n is also an Epsp to the base Q_0, so that $Q_0^{\frac{n-1}{2}} \quad -1 \mod n$; and consequently, $_2(n-1) = _2(ord_n(Q_0))$: However, since n is a spsp(Q_0), the latter value is equal to $_2(ord_p(Q_0))$ for any prime p dividing n. Further, since $\left(\frac{Q_0}{p}\right) = 1$, $_2(ord_p(Q_0)) < _2(p-1)$. Consequently, $_2(p-1) \quad 2$, because $_2(n-1) \quad 1$. Therefore $p \quad 1 \mod 4$ and thus $(p) = 1$, as claimed.

Now, let $\left(\frac{Q_0}{p}\right) = -1$. Suppose rstly that $n \quad (n) \mod 4$. Then, by Theorem 2, necessarily $p \quad (p) \mod 4$: Again, since n is a spsp(Q_0) we conclude that $_2(n-1) = _2(ord_p(Q_0))$ for all primes p, but since $\left(\frac{Q_0}{p}\right) = -1$, this is equal to $_2(p-1)$: If $(n) = -1$, then, as $_2(n-1) = _2(p-1) = 1$, we obtain $p \quad 3 \mod 4$, and therefore, $(p) = -1$. Similarly, $(n) = 1$ yields $(p) = 1$: Now, consider the case that $n \quad -(n) \mod 4$: By Theorem 2 this implies $_2(p-(p)) = 1$; that is, $p \quad -(p) \mod 4$: Similarly as above, the two separate cases $(n) = 1$ and $(n) = -1$ yield the desired assertion $(p) = (n)$: □

Corollary 1 *If n is a* spsp(Q_0) *for* $\left(\frac{Q_0}{n}\right) = -1$*; then, a necessary condition that there exists an integer* $P \not\equiv 0$ *such that* $\left(\frac{D(P)}{n}\right) = -1$ *and* $V_{\frac{n+1}{2}}(P;Q_0) \quad 0 \mod n$*; is that, for every prime p dividing n,*

$$\text{if } \left(\frac{Q_0}{p}\right) = 1 \text{ then } \begin{cases} (p) = 1; \\ _2(n+1) < _2(p-1); \text{ and } _2(n-1) < _2(p-1); \\ p \quad 1 \mod 8; \end{cases}$$

if $\frac{Q_0}{p} = -1$ *then* $\begin{cases} (p) = -1; \\ {}_2(n+1) = {}_2(p+1) \text{ and } {}_2(n-1) = {}_2(p-1): \end{cases}$

Proof. This follows from Theorems 2, 3, and Lemma 2 below. □

6 The Number of Parameters That Pass the Proposed Test

6.1 Some Technical Prerequisites

Lemma 2 *If n is a* sLpsp$(P;Q)$ *such that* $\frac{Q}{n} = -1$*; then* ${}_2((p;P;Q)) = {}_2\left(n - \frac{D}{n}\right)$ *for all prime divisors p of n.*

Proof. Since n is an ELpsp$(P;Q)$ we have ${}_2(n-(D=n)) = {}_2((n;P;Q))$: As n is a sLpsp the latter value is ${}_2((p;P;Q))$: □

Lemma 3 *A necessary condition for an odd composite number n to simultaneously pass a* psp(Q_0)- *and a* Lpsp$(P;Q_0)$- *test with* $\frac{D(P)}{n} = -1$ *is that* $(ord_n(Q_0);\ (n;P;Q_0)) = 2$:

Proof. The hypotheses yield $Q_0^{n-1} \equiv 1 \bmod n$ and thus $ord_n(Q_0) | (n-1)$ on the one, and $(n;P;Q) | (n+1)$ on the other hand. Hence $(ord_n(Q_0);\ (n;P;Q)) \mid 2$. By the choice of Q_0, both $ord_n(Q_0)$ and $(n;P;Q_0)$ need to be even. □

6.2 The Main Results

We are now in the position to determine the number of parameters P that pass the proposed test. It is sufficient to establish the desired number modulo any prime power dividing n. We first need the following proposition, which is a specification of Theorem 1, in that n is now by hypothesis a psp(Q_0).

Proposition 3 *Let $n = \prod_{p \mid n} p$ be a* psp(Q_0). *Then the number of $P \in \mathbb{Z}_p$ such that* $\frac{P^2-4Q_0}{p} = (p)$ *and* $U_k(P;Q_0) \equiv 0 \bmod p$ *; is given as follows.*

1. *For* $\frac{Q_0}{p} = -1$ *as* $\begin{cases} \sum_{\substack{d \mid k;\, p-(p)) \\ {}_2(d) = {}_2(p-(p)) \\ (d;ord_n(Q_0))=2}} (d); & \text{when } {}_2(k) \geq {}_2(p-(p)); \\ 0; & \text{otherwise;} \end{cases}$
2. *for* $\frac{Q_0}{p} = 1$ *as* $\begin{cases} \sum_{\substack{d \mid k;\, p-(p)) \\ {}_2(d) < {}_2(p-(p)) \\ (d;ord_n(Q_0))=2}} (d); & \text{when } {}_2(k) \geq {}_2(p-(p)); \\ \sum_{\substack{d \mid k;\, p-(p)) \\ (d;ord_n(Q_0))=2}} (d); & \text{otherwise:} \end{cases}$

Proof. The proof runs along the same lines as the one for Theorem 1. Similarly as for Lemma 3 we get the additional condition $(d;ord_n(Q_0)) = 2$. □

Theorem 4 *If n is a* spsp(Q_0) *for* $\frac{Q_0}{n} = -1$*; and $n = \prod_{p \mid n} p$, then the number of $P \in \mathbb{Z}_p$ such that* $\frac{P^2-4Q_0}{n} = -1$ *and* $V_{\frac{n+1}{2}}(P;Q_0) \equiv 0 \bmod p$ *; is*

$$\text{when } \left(\frac{Q_0}{p}\right) = -1\text{; given as } \begin{cases} \frac{(n+1;p+1)}{2}; & \text{for } {}_2(n+1) = {}_2(p+1); \\ 0; & \text{otherwise;} \end{cases}$$

$$\text{when } \left(\frac{Q_0}{p}\right) = 1\text{; given as } \begin{cases} \frac{(n+1;p-1)}{2}; & \text{for } {}_2(n+1) < {}_2(p-1); \\ 0; & \text{otherwise.} \end{cases}$$

Proof. By Lemma 1 and Lemma 2 we need to count those parameters P for which $U_{n+1}(P;Q_0) \equiv 0 \bmod p$, $U_{\frac{n+1}{2}}(P;Q_0) \not\equiv 0 \bmod p$, and additionally ${}_2(\ (p;P;Q_0)) = {}_2(n+1)$:

Consider first the case that $\left(\frac{Q_0}{p}\right) = -1$: According to Corollary 1 we necessarily need to have ${}_2(p+1) = {}_2(n+1)$: Therefore, if ${}_2(n+1) \neq {}_2(p+1)$, the desired number of the P's is zero. Now, let ${}_2(n+1) = {}_2(p+1)$: Then by Proposition 3 the number of P's with $U_{n+1}(P;Q_0) \equiv 0 \bmod p$ is

$$\sum_{\substack{d \mid (n+1;p+1) \\ {}_2(d) = {}_2(p+1) \\ (d;ord_n(Q_0))=2}} \phi(d);$$

while the number of P's with $U_{\frac{n+1}{2}}(P;Q_0) \equiv 0 \bmod p$ is 0. We show that the condition $(d;ord_n(Q_0)) = 2$ is immaterial. For suppose to the contrary that this gcd is some value $g > 2$. Then $g \mid d \mid (n+1)$ and $g \mid ord_n(Q_0) \mid (n-1)$ which is impossible and we thus obtain the number as $\frac{(n+1;p+1)}{2}$.

Now consider the case that $\left(\frac{Q_0}{p}\right) = 1$: We then have $\ = \phi(p) = 1$ and, additionally, ${}_2(\ (p;P;Q_0)) < {}_2(p-1)$: By Corollary 1 we require that necessarily ${}_2(n+1) < {}_2(p-1)$: Thus, if ${}_2(n+1) \not< {}_2(p-1)$ the number of P is zero. If ${}_2(n+1) < {}_2(p-1)$ we obtain the desired number as

$$\sum_{\substack{d \mid (n+1;p-1) \\ (d;ord_n(Q_0))=2}} \phi(d) - \sum_{\substack{d \mid (\frac{n+1}{2};p-1) \\ (d;ord_n(Q_0))=2}} \phi(d):$$

By similar arguments concerning the gcd as above, these sums evaluate to $(n+1;p-1) - 1 - (\frac{n+1}{2};p-1) + 1 = \frac{1}{2}(n+1;p-1)$; as claimed. □

We thus have established the exact number of parameters P that pass the proposed test.

Theorem 5 *Suppose that a composite number n fulfills for all prime divisors p the following conditions*

$$\text{if } \left(\frac{Q_0}{p}\right) = 1 \text{ then } \begin{cases} \phi(p) = 1; \\ {}_2(n+1) < {}_2(p-1); \text{ and } {}_2(n-1) < {}_2(p-1); \\ p \not\equiv 1 \bmod 8; \end{cases}$$

$$\text{if } \left(\frac{Q_0}{p}\right) = -1 \text{ then } \begin{cases} \phi(p) = -1; \\ {}_2(n+1) = {}_2(p+1) \text{ and } {}_2(n-1) = {}_2(p-1): \end{cases}$$

Let $\gamma = 1$; respectively 0, according as $n \equiv 1$ or 3 modulo 4. Then the number of parameters P that pass the proposed test of section 4.2 is given as

$$\prod_{(\frac{Q_0}{p})=1 \atop p \mid n} \frac{1}{2}(n+1; p-1) \qquad \prod_{(\frac{Q_0}{p})=-1 \atop p \mid n} \frac{1}{2}(n+1; p+1) \quad - \gamma.$$

Otherwise n does not pass the proposed test for any of the parameters P.

6.3 Some Special Cases

It follows from above, that the combined test is most effective, when for the spsp test a basis $Q = Q_0$ with suitable large orders are chosen. Alternatively, we can, prior to the Lucas test, run the strong probable prime test for variations of bases Q. Then the number of parameters P that pass the proposed Lucas test, will be minimized.

Lemma 4 *Let $n = \prod_{i=1}^{r} p_i$, $p_i \neq 2; 3; 5$. Suppose that for some i there exists a parameter a_i with $\frac{p_i-1}{2} \; ord_{p_i}(a_i)$ such that n is a $\mathrm{psp}(a_i)$. A necessary condition for n to be a $\mathrm{Lpsp}(P; Q)$ for $P \in \mathbb{Z}_n$ and $\left(\frac{D}{n}\right) = -1$, is that $\left(\frac{D}{p_i}\right) = -1$.*

Proof. Suppose that $\left(\frac{D}{p_i}\right) = 1$. Then $(p_i; P; Q) \mid (p_i - 1)$ and $(p_i; P; Q) > 2$ since otherwise $(P; n) > 1$. However, since $\frac{p_i-1}{2} \; ord_{p_i}(a_i)$, we have $(\frac{p_i-1}{2}; (p_i)) > 2$, which makes it impossible for n to pass the Lucas test. □

Corollary 2 *A necessary condition for a Carmichael number $n = \prod_{i=1}^{r} p_i$ to be a $\mathrm{Lpsp}(P; Q)$ with $P \in \mathbb{Z}_n$ and $\left(\frac{D}{n}\right) = -1$ is that $2 \mid r$ and $\left(\frac{D}{p_i}\right) = -1$ for all p_i.*

Similarly as above we obtain

Lemma 5 *Let $n = \prod_{i=1}^{r} p_i$ be a spsp to a set A of bases. Suppose that for every i there is an $a_i \in A$ with $u_i \nmid ord_{p_i}(a_i)$ where u_i denotes the odd part of $p_i - 1$. Let $D = P^2 - 4Q$ such that $\left(\frac{D}{n}\right) = -1$ and $P \neq 0$. If $2 \mid r$ or $\left(\frac{D}{p_i}\right) = 1$ for some i then n cannot be a $\mathrm{sLpsp}(P; Q)$.*

For such composites the number of "liars" P to the proposed test becomes extremely small.

Corollary 3 *Suppose that for all primes p_i dividing n (at least) one of the following conditions holds:*

$u_i \mid (n-1)$,
$\left(\frac{Q_0}{p_i}\right) = -1$.

Then, n can pass the proposed test of section 4.2, only if, for all p_i

$$\frac{Q_0}{p_i} = \frac{D(P)}{p_i} = -1;$$

$$_2(n+1) = {}_2(p+1) \text{ and } {}_2(n-1) = {}_2(p-1):$$

In this case the number of P that pass the test equals

$$\prod_{p_i \mid n} \frac{(n+1; p_i+1)}{2} - y;$$

where y is de ned in Theorem 5.

Remark 2 *(i) If n has r prime factors, then with probability $\frac{1}{2^r}$ the base Q_0 is a nonresidue for all the p_i. Since the* spsp *test is extremely fast, it can very e ciently be run for variations of Q_0: As a second step, now Corollary 3 asserts that the combination with the Lucas test as described in section 4.2 is highly reliable.*

(ii) It is known (cf. [3]) that if a composite integer n is a spsp *w.r.t. all possible $\frac{(n)}{4}$ bases, then n is either of the form $n = (k+1)(2k+1)$, or n is a Carmichael number with three factors that are $3 \bmod 4$: The rst from can easily be checked for compositeness, since $n = (k+1)(2k+1)$ implies that $8n+1$ is a perfect square. For numbers of the second form, no e cient algorithms are known. However, in this case Corollary 3 asserts that the proposed test is very e ective. In particular, since $(p_i - 1) \mid (n-1)$ the number of liars P obviously will be very small.*

6.4 Some Numerical Examples

1. The spsp test w.r.t. one base $Q = Q_0$.
   For simplicity we investigated the spsp(2) 10^{13} by R. Pinch [16] in relation to our proposed test. As $Q_0 = 2$ we only search through the numbers n 3 mod 8. There a total of 23637 of such pseudoprimes in Pinch's list. The following table illustrates the e ciency of the proposed test.

Distribution of all composites n 10^{13}, n 3 mod 8 that pass the proposed test for $Q_0 = 2$			
no. of n that don't pass the test for any P		percentage	
14867		62:89	
distribution of n that pass the test for t di erent P			
t	percentage	t	percentage
2	29:88	14	0:38
6	2:89	18	0:18
4	1:70	12	0:16
10	0:72	16	0:09
8	0:40	others	0:66

2. The spsp test w.r.t. variations of bases Q.
   When n is known to be a spsp for more than one base Q then the number of liars P for the proposed test becomes very small. We considered composites that are spsps for all bases t. For example when searching through Bleichenbacher's list [3] of such numbers for t 100, we found that in many cases these composites will not pass the proposed test for any Q_0 and any P. For the pseudoprimes that do pass our test, the number of liars P is, in relation to n, extremely small, e.g. when $\frac{Q_0}{n} = -1$ then n = 168790877523676911809192454171451, and n = 194350792386572081824965358396480711416097814050497 1; pass for 8 di erent P, and n = 10887815362956808231598692418937808 51 passes for 56 di erent P.
   As another example consider Arnault's 331 digit number. Although it is a spsp w.r.t. all prime bases 200 [1] it does not pass the proposed test for any Q_0 nor for any P.

Acknowledgement

I am deeply grateful to Professor W. B. Müller for numerous helpful discussions, his many valuable comments and his qualifying advice.

References

1. Arnault, F.: Rabin-Miller primality test: Composite numbers which pass it. Math. Comp. 64 (209), 355-361 (1995)
2. Baillie, R., Wagsta , S., Jr.: Lucas pseudoprimes. Math. Comp. 35, 1391-1417 (1980)
3. Bleichenbacher, D.: E ciency and Security of Cryptosystems based on Number Theory. Dissertation ETH Zürich 1996.
4. Carmichael, R.D.: On Sequences of Integers De ned by Recurrence Relations. Quart. J. Pure Appl. Math. Vol. 48, 343-372 (1920)
5. Grantham, J.: A Probable Prime Test with High Con dence. J. Number Theory 72, 32-47 (1998)
6. Guillaume, D., Morain, F.: Building pseudoprimes with a large number of prime factors. AAECC 7 (4), 263-277 (1996)
7. Jaeschke, G.: On strong pseudoprimes to several bases. Math. Comp. 61, 915-926 (1993)
8. Kowol, G.: On strong Dickson pseudoprimes. AAECC 3, 129-138 (1992)
9. Lidl, R., Müller, W.B., Oswald, A.: Some remarks on strong Fibonacci pseudoprimes. AAECC 1, 59-65 (1990)
10. More, W.: The LD Probable Prime Test. In: Mullin, R.C., Mullen, G. (eds.) Contemporary Mathematics 225, 185-191 (1999)
11. Müller, S.: On Strong Lucas Pseudoprimes. In: Dorninger, D., Eigenthaler, G., Kaiser, H.K., Kautschitsch, H., More, W., Müller, W.B. (eds.) Contribution to General Algebra, 10, 237-249 (1998).
12. Müller, S.: Carmichael Numbers and Lucas Tests. In: Mullin, R.C., Mullen, G. (eds.) Contemporary Mathematics 225, 193-202 (1999)

13. Müller, S: On the rank of appearance of Lucas sequences. To appear in the Proceedings of the 8^{th} International Conference on Fibonacci Numbers and Their Applications, June 22 - 26, 1998, Rochester, New York.
14. Müller, W.B., Oswald, A.: Dickson pseudoprimes and primality testing. In: Davies, D.W. (ed.) Advances in Cryptology - EUROCRYPT'91, 512-516. Lecture Notes in Computer Science, Vol. 547. Berlin Heidelberg New York: Springer 1991
15. Pinch, R.G.E.: The Carmichael numbers up to 10^{15}. Math. Comp. 61, 381-391 (1993)
16. Pinch, R.G.E.: ftp://ftp.dpmms.cam.ac.uk/pub/rgep/Papers/publish.html#41
17. Postl, H.: Fast evaluation of Dickson polynomials. In: Dorninger, D., Eigenthaler, G., Kaiser H., Müller, W.B. (eds.) Contributions to General Algebra 6, 223-225. B. G. Teubner: Stuttgart 1988
18. Ribenboim, P.: The New Book of Prime Number Records. Berlin: Springer 1996
19. Somer, L.: On Lucas d-Pseudoprimes. In: Bergum, G., Philippou, A., Horadam, A. (eds.) Applications of Fibonacci Numbers, Vol. 7, Kluwer, 369-375 (1998)

On the Cryptanalysis of Nonlinear Sequences
[Invited Paper]

Solomon W. Golomb

Communication Sciences Institute
University of Southern California
Los Angeles, CA 90089-2565 U.S.A.
c/o milly@mizar.usc.edu

Abstract. A nonlinear boolean function $f(x_1; x_2; \ldots; x_k)$ of k binary variables may be used in two basically di erent ways to generate a nonlinear binary sequence, *internally* or *externally*. Internally, f may be part of the feedback computation of a nonlinear feedback shift register. Externally, f may be applied to the output bit stream of another sequence generator (e.g. a linear shift register) to introduce nonlinearity, or greater nonlinearity. A third approach is to use f to obtain a nonlinear combination of k linear sequences. The vulnerability of systems using f in any of these ways to cryptanalysis depends on the multidimensional correlations of f with the modulo 2 sums of the subsets of its variables. This principle was published by the present author in [1] in 1959, and included as Chapter 8 in his book [2] in 1967. It was subsequently rediscovered and republished in 1988 in [3], on the basis of which it is sometimes known as the Xiao-Massey algorithm. Some practical aspects of the use of this principle in code construction as well as code breaking, and for other types of signal design, are discussed.

1 Introduction

There are 2^{2^k} *boolean functions* of k binary variables, i.e., mappings from F_2^k into F_2. Of these, only 2^k are *linear homogeneous* functions, i.e., modulo 2 sums of a subset of the k binary variables. The complements of these 2^k functions may be regarded as *linear inhomogeneous* functions, but that still leaves $2^{2^k} - 2^{k+1}$ *nonlinear boolean functions* of k binary variables.

The *binary maximum-length linear shift register sequences* (called *m-sequences* or *PN sequences* in the literature) are particularly easy to generate, but notoriously insecure against an adversarial \attack" which attempts to identify the sequence from a very short stretch of consecutive terms. An m-sequence \of degree n", generated by an n-stage linear shift register, and having period $2^n - 1$, has only \linear span" n. That is, from any n consecutive terms of the sequence, the recurrence relation, and therefore the entire sequence can be deduced. As \key sequences" for secure communications applications, some type of *nonlinearity* must be introduced.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 236{242, 1999.

2 Nonlinear Shift Register Sequences

There are a number of ways to introduce nonlinearity into sequences generated by shift registers. Here are three of them.

2.1 Nonlinear Feedback Shift Registers

In this approach, a subset of k of the n stages of a shift register are used as input variables to a (nonlinear) boolean function of k variables, and the output of this function is fed back to the front stage of the shift register. The *nonlinear shift register* then generates a periodic binary sequence whose period cannot exceed 2^n. One limitation of this \interior logic" approach to generating a nonlinear sequence is that the period is virtually impossible to predict in advance, in the most general case, and is therefore obtained (if at all) by testing every sequence that may be used as a key. Nonlinear shift register sequences of this type are discussed extensively in [2], particularly in Chapters VI and VII.

2.2 Linear Sequences with External Nonlinear Logic

An n-stage linear shift register may be used to generate an m-sequence of period $2^n - 1$. Then, a (nonlinear) boolean function of k variables uses k positions along the sequence as input variables to generate a nonlinear output sequence of period $2^n - 1$. In this approach, the period of the output sequence is known in advance. The k inputs to the nonlinear function need not be from a subset of the n positions along the register, but may extend to a wider stretch of the linear sequence. This merely requires extra memory cells to extend the \delay line" portion of the shift register beyond the region involved in the feedback calculations. In fact, in this approach, k may be allowed to exceed n if so desired.

2.3 Nonlinear Combinations of Several Sequences

The k inputs to a (nonlinear) boolean function of k variables may be taken from k di erent sequences. These \component" sequences may, in general, have respective periods $p_1; p_2; \ldots; p_k$, in which case the \combined" sequence has period $P = lcm(p_1; p_2; \ldots; p_k)$, the least common multiple of the individual periods. The component sequences may all be taken to be linear sequences of the same period, or linear sequences of di erent periods, or other sequences (not necessarily m-sequences) that are relatively easy to generate.

Hybrids of these three basic approaches to obtaining nonlinear binary sequences are easy to describe. A sequence can be obtained using an external nonlinear logic applied to a sequence generated by a nonlinear feedback shift register. One or more of the component sequences in the \nonlinear combination" approach may themselves be nonlinear sequences. Not all k of the inputs in the \nonlinear combination" approach need to come from *di erent* component sequences.

3 Classi cation of Boolean Functions

The 2^{2^k} boolean functions of k binary variables, $f f(x_1; x_2; : : : ; x_k)g$, fall into equivalence classes (also called *families*, or *orbits*), when two such functions differing only by permutation and/or complementation of their variables are considered equivalent. The permutations and complementations of the k variables form a group H_k of $2^k \quad k!$ elements, and the number of orbits relative to this group is readily found by an enumeration formula variously attributed to Polya, to Burnside, to Frobenius, and to Cauchy (as historical research has pushed farther back into the past). One reference to this enumeration applied to boolean functions is Slepian [4].

One practical application of this partitioning into equivalence classes is that a hardware circuit which mechanizes a particular boolean function f of k variables will also mechanize any other function in the same orbit, simply by permuting and complementing the inputs. In a very real sense, the number of truly distinct cases is only the number of orbits, which has a leading term of $(2^{2^k})=(2^k \quad k!)$.

The group of symmetries H_k is enlarged by a factor of 2 to G_k, with $2^{k+1} \quad k!$ operations, if we also allow complementation of the *output* of f. Note that H_k is a normal subgroup of index 2 in G_k for each $k \quad 1$. When no ambiguity results, we will omit the subscript k from H and G.

In what follows, we will generate a set of *invariants* which categorize the orbits with respect to G, or with respect to H. Two boolean functions will be in the same orbit if and only if they have the same complete set of invariants. Even more importantly, these invariants directly relate to the degree of vulnerability or immunity to correlation attack possessed by a nonlinear sequence generated using the boolean function in question in any of the ways listed earlier.

4 Calculating the Invariants

It is quite possible that [1] is the earliest reference to Walsh functions in the engineering literature. Forty years later, it is universally known that the Walsh functions form a complete orthonormal set on a standard interval, a discrete analogue to the Fourier sine and cosine functions.

The truth table of a boolean function of k variables, which contains 2^k binary entries, may be expanded in a \Walsh series", using only the 2^k Walsh functions, as basis vectors, whose \jumps" are limited to the result of dividing the standard interval into 2^k equal parts. (This set of Walsh functions is clearly isomorphic to the codewords in a corresponding rst-order Reed-Muller code.) The truth table is expressed as a binary waveform on the standard interval for purposes of the Walsh series expansion. Everything can be done using either $+1$ and -1 as the two binary values, or 0 and 1 as the two values, for both the truth table and the Walsh functions, provided that the same approach is maintained consistently throughout.

To summarize briefly, the Walsh expansion coe cients *are* the invariants of the equivalence classes of the boolean functions relative to the symmetry group

H, and the invariants relative to the larger group G are readily computed from these. (As the group of symmetries is enlarged, the number of invariants will in general decrease.) The details, which are presented in full in [1] and [2], and illustrated with examples, will be discussed more concisely here.

Theorem 1 *Let* $c_0 = \sum_{\text{all } x_j} f(x_1; x_2; \ldots; x_k)$, *which is the number of 1's in the truth table for* $f(x_1; x_2; \ldots; x_k)$, *and let* $T_0 = \max(c_0; 2^k - c_0)$. *Then* c_0 *is an invariant for the orbit of* f *under* H_k, *and* T_0 *is an invariant for the orbit of* f *under* G_k.

Proof. Permutation and complementation of the variables in $f(x_1; x_2; \ldots; x_k)$ rearranges the entries in the truth table for f, but leaves their sum c_0 unchanged. Complementing f replaces c_0 by $2^k - c_0$. Thus c_0 is invariant under H, and $T_0 = \max(c_0; 2^k - c_0)$ is invariant under G. □

Definition 1 *T_0 and c_0 are the zero order invariants of f with respect to G and H, respectively.*

Theorem 2 *For each i; $1 \le i \le k$, let $R_1^i = \max(d_1^i; 2^k - d_1^i)$; where $d_1^i = \sum_{\text{all } x_j} (f(x_1; x_2; \ldots; x_k) \oplus x_i)$. Then the set of numbers $R_1^1; R_1^2; \ldots; R_1^k$, when rearranged in descending order, forms a collection of k invariants (the "first-order invariants") $T_1^1; T_1^2; \ldots; T_1^k$ for the orbit of f under G. These are also invariants for the orbit of f under H. (For notational consistency, we may denote these same invariants by $c_1^1; c_1^2; \ldots; c_1^k$, when referring to them as the "first-order invariants" with respect to H.) [The symbol $\oplus$ denotes modulo 2 addition, but the summation sign denotes ordinary addition.]*

Proof. The number R_1^i is not changed when x_i is replaced by its complement, since $\max(d_1^i; 2^k - d_1^i) = \max(2^k - d_1^i; d_1^i)$, so that the set $\{R_1^1; R_1^2; \ldots; R_1^k\}$ is invariant under complementation of variables. Since the set (or multiset) $\{R_1^i\}$ is reordered according to decreasing size to obtain $\{T_1^i\}$, the (multi)set $\{T_1^i\}$ is invariant under permutation as well as complementation of the variables of f. Finally, complementing f in $f(x_1; x_2; \ldots; x_k) \oplus x_i$ has the same effect as complementing x_i, so that the invariants $\{T_1^i\}$ under H are also invariants under G. □

Definition 2 *For every pair $(i; j)$ with $1 \le i < j \le k$, define $d_2^{ij} = \sum_{x_1; x_2; \ldots; x_k} (f(x_1; x_2; \ldots; x_k) \oplus x_i \oplus x_j)$. The complement of d_2^{ij} is $2^k - d_2^{ij}$. That permutation and complementation of the variables of f (not necessarily unique) which is consistent with yielding the values $\{R_1^i\}$ in descending order and which makes the sequence of $\{d_2^{ij}\}$ numerically greatest (i.e. larger values occur ahead of smaller values, to the extent permitted by consistency with the ordering of $\{T_1^i\}$) yields the second-order invariants $T_2^{1;2}; T_2^{1;3}; \ldots; T_2^{2;3}; \ldots; T_2^{k-1;k}$ for the orbit of*

f relative to G, which are also the second-order invariants $c_2^{1,2}; c_2^{1,3}; \ldots; c_2^{2,3}; \ldots; c_2^{k-1,k}$ *for the orbit of f relative to H. (The assertions inherent in this definition are proved in the same way as the proof of Theorem 2.)*

More generally, the r^{th}-order invariants are obtained by calculating $d_r^{i_1,i_2,\ldots,i_r} = \sum_{\text{all } x_j} (f(x_1, x_2, \ldots, x_k) \cdot x_{i_1} \cdot x_{i_2} \cdot \ldots \cdot x_{i_r})$ for each of the $\binom{k}{r}$ subsets of the k variables in f taken r at a time. Then $R_r^{i_1,i_2,\ldots,i_r} = 2^k - d_r^{i_1,i_2,\ldots,i_r}$, and these $\binom{k}{r}$ R_r-values are reordered, consistently with the ordering of all the lower-order invariants, in descending order, to obtain the r^{th} order invariants $T_r^{i_1,\ldots,i_r}$ relative to G, and the invariants $c_r^{i_1,\ldots,i_r}$ relative to H.

The calculation of all these invariants is illustrated explicitly for the particular boolean function of four variables $f(v,x,y,z) = v\bar{y}\bar{z} + \bar{v}z + \bar{x}z$ in [1] and [2]. For this function, it is shown that $T_0 = 8$; $(T_1^1, T_1^2, T_1^3, T_1^4) = (12, 10, 10, 8)$; $(T_2^{12}, T_2^{13}, T_2^{14}, T_2^{23}, T_2^{24}, T_2^{34}) = (10, 6, 12, 8, 6, 6)$; $(T_3^{123}, T_3^{124}, T_3^{134}, T_3^{234}) = (8, 6, 10, 8)$, and $T_4^{1234} = 8$.

5 Significance of the Invariants

A "correlation attack" on a sequence generated by the use of a nonlinear boolean function takes the form of correlating the sequence under attack with the modulo 2 sums of the subsets of $\{x_1, x_2, \ldots, x_k\}$, taken r at a time, for $r = 0$; $r = 1$; $r = 2$, etc. The smaller the value of r for which any of these correlations deviates from the "random", or *balanced* value (equally many 0's and 1's being counted), the easier it is to start determining the identity of the boolean function being used. These multi-dimensional correlation values are essentially the "invariants" of the boolean function which we have been describing. Ironically, the only boolean functions of k variables for which *all* these correlation values appear random (i.e. the sequence appears *uncorrelated* with any subset of the input variables) are the modulo 2 sum of $\{x_1, x_2, \ldots, x_k\}$ and the complement of this sum. However, as noted earlier, these two *linear* functions are vulnerable to a different type of attack.

It is shown in [2], Chapter VI, that an n-stage shift register with a (nonlinear) feedback function $f(x_1, x_2, \ldots, x_n)$ produces "pure" cycles, without branches, if and only if we can write $f(x_1, x_2, \ldots, x_n) = g(x_1, x_2, \ldots, x_{n-1}) \cdot x_n$. That is, the "highest order term" x_n (the one farthest back in time) must enter (only) *linearly* in the feedback calculation in order to avoid the undesirable phenomenon of shift register states with more than one predecessor. Suppose that $\{a_j\}$ is the sequence produced by such a shift register generator. The statistics of $\{a_j\}$ will reflect the invariants of the function f. However, the statistics of $\{a_j\} \cdot \{a_{j-n}\}$ will reflect the invariants of g. In particular, if g does not have equally many 0's and 1's in its truth table ($T_0 \neq 2^{k-1}$, where g is a function of k variables), this will be quickly apparent from the statistics of $\{a_j\} \cdot \{a_{j-n}\}$, and the cryptanalyst will be able to determine n very quickly (if it was not already known), and then proceed to learn more about the function g. For the general case of nonlinear

shift register sequences where the nonlinear boolean function is internal to the feedback process, it is the invariants of g (where $f = g \circ x_n$) which are of primary concern. However, since $g \circ h = f \circ x_n \circ h$, the r^{th}-order invariants of g are a subset of the $(r+1)^{st}$-order invariants of f.

A nonlinear sequence produced by a boolean function f of k variables is "relatively immune" to a correlation attack if all the lower order invariants of f are equal to 2^{k-1}, the "uncorrelated" case. The number of variables, k, that f should have, and the value of b such that all invariants of f of order $r \le b$ should have the value 2^{k-1}, will depend on the degree of security that the system is intended to achieve.

6 Historical Notes

When I joined the Jet Propulsion Laboratory (JPL) of the California Institute of Technology in the summer of 1956, JPL was supported by the U.S. Army Ballistics Missile Command, and the application of primary interest for "pseudo-random sequences" was the secure guidance of missiles. In particular, it was desired to be able to generate long binary sequences in a deterministic manner in a way that an "intelligent jammer" would not be able to predict future values of the sequence quickly enough to assist him in jamming the guidance signal. Using such binary sequences to modify an RF carrier prior to adding information-bearing modulation became known somewhat later as "direct sequence spread spectrum" communication, and later still as CDMA (code division multiple access) communication for digital cellular telephony. It was this application to secure missile guidance that motivated my earlier work on "nonlinear sequences", and led to my investigation of the "classi cation of boolean functions", including their invariants, as described in [1] and [2].

After the launch of Sputnik 1 on October 4, 1957, JPL participated in the U.S. Army project that led to the successful launch of Explorer 1, the first non-Soviet satellite, on January 31, 1958. When NASA was created, quite a few months later, JPL's primary source of support shifted to NASA, but missile guidance for the Army Ordnance Command was an ongoing area. When I turned my attention (in 1959) to designing an interplanetary ranging system, I realized that it would be advantageous to combine a number of shorter binary sequences of relatively prime period, using a nonlinear boolean function, into a single binary sequence with a long period, where the objective now was to make it *easy* to extract the component sequence from the combined sequence, the exact opposite of the objective for secure communications. When k is odd, the function $f(x_1, x_2, \ldots, x_k) = \mathrm{maj}(x_1, x_2, \ldots, x_k)$ is positively correlated with each of the x_j's, where "maj" denotes the majority decision function. Moreover, there is a sense in which this is the function which has the greatest "simultaneous correlation" with each of its variables. Speci cally, for this boolean function, $T_0 = 2^{k-1}$ (the "random" value), but $T_1^1 = T_1^2 = \cdots = T_1^k = 2^{k-1} + \binom{k-1}{\frac{1}{2}(k-1)}$. Some aspects of the JPL ranging system are described in [5], believed to be the very first book to have "digital communications" in its title.

In 1959, when [1] appeared, and even in 1967, when [2] was published, applications to secure communications and cryptography were not explicitly mentioned in the open literature. However, when [3] appeared, by Xiao and Massey, in 1988, that restriction was no longer in e ect. The Xiao-Massey paper was apparently motivated by the publication of [6], in which it is shown that using correlation methods is an e ective way to break ciphers which use a nonlinear combination of linear shift register sequences as a keystream generator. The rst reference cited in [6] is [2], but only in connection with *linear* sequences.

Several other people who were at JPL in the 1950's deserve special mention. Lloyd R. Welch was a doctoral student in mathematics at Caltech, working part-time at JPL, when I arrived there, and was a major collaborator in my work on shift register sequences. I hired Harold M. Fredricksen to work part-time at JPL in October, 1957, while he was a student at Pasadena City College. His rst assignment was to classify all $2^{16} = 65{,}536$ boolean functions of four variables into their equivalence classes (*orbits*) relative to G. I thought this would require many months, but he completed the task in two weeks, almost entirely without the aid of computers. Andrew J. Viterbi started at JPL in the summer of 1957, having just received a master's degree in electrical engineering from MIT. He was a contributing author to [5]; and more recently, has played the leading role in developing CDMA as the emerging standard for digital cellular telephony.

Dr. Lloyd Welch has been my faculty colleague and collaborator at USC for over thirty years. Dr. Harold Fredricksen was my doctoral student at USC in the late 1960's. Dr. Andrew Viterbi, the co-founder of Qualcomm, Inc., who received his Ph.D. from USC in 1962, recently endowed a chair in Communications at USC, to which I have been appointed as the rst incumbent.

References

1. Golomb, S.W.: On the Classi cation of Boolean Functions. Transactions of the International Symposium on Circuit and Information Theory: IRE Transactions on Circuit Theory. CT-6 (1959) 176-186; IRE Transactions on Information Theory. IT-5 (1959) 176-186.
2. Golomb, S.W.: Shift Register Sequences. Holden-Day, Inc., San Francisco (1967).
3. Xiao, G.-Z., Massey, J.L.: A spectral characterization of correlation-immune combining functions. IEEE Trans. on Information Theory, IT-34, no. 3 (1988) 569-571.
4. Slepian, D.: On the number of symmetry types of boolean functions of n variables, Can. J. Math. 5, no. 2 (1953) 185-193.
5. Golomb, S.W., ed.: Digital Communications with Space Applications. Prentice-Hall, Englewood Cli s, NJ (1964).
6. Siegenthaler, T., Decrypting a Class of Stream Ciphers Using Ciphertext Only. IEEE Trans. on Computers, C-34 (1985) 81-85.

Securing Aeronautical Telecommunications
[Invited Paper]

Simon Blake-Wilson

Certicom Corp., 200 Matheson Blvd W, Mississauga, Ontario L5R 3L7, Canada

Abstract. The Aeronautical Telecommunications Network or ATN is the next generation network being developed by the international aviation community for providing voice and data communications between ground stations and aircraft. It is currently being trialed and will ultimately be used for applications such as air tra c control.

This talk will discuss recent work to add security provisions to the ATN to prevent threats like disruption of service, misdirection of aircraft, and disclosure of commercially sensitive data relating to aircraft operations. The talk will focus on the challenges faced when designing a security solution in this environment, how these challenges were addressed by the ATN security design team, and what the cryptographic community can do in the future to help designers of other solutions meet these challenges. Examples of the challenges include:

- providing security in a bandwidth-constrained wireless environment;
- balancing the conflicting needs of con dentiality, integrity, and availability in safety-critical applications; and
- minimizing the amount of mutual trust necessary between users with limited trust in each other.

The goal of this talk is to identify which areas will be the focus of future cryptographic research by investigating the problems facing cryptographic designers today.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 243{243, 1999.

Tensor-Based Trapdoors for CVP and Their Application to Public Key Cryptography
(Extended Abstract)

Roger Fischlin and Jean-Pierre Seifert

Fachbereich Mathematik (AG 7.2)
J.W. Goethe-Universität Frankfurt am Main, Postfach 111932
D-60054 Frankfurt/Main, Germany
ffischlin,seifertg@informatik.uni-frankfurt.de
http://www.mi.informatik.uni-frankfurt.de/

Abstract. We propose two trapdoors for the Closest-Vector-Problem in lattices (CVP) related to the lattice tensor product. Using these trapdoors we set up a lattice-based cryptosystem which resembles the McEliece scheme.

1 Introduction

Since the invention of public key cryptography, the security of most cryptosystems has been based on the (assumed) hardness of factoring or computing discrete logarithms. Only a few schemes based on other problems remain unbroken. Among which there is the McEliece scheme [St95] based on the computational difficulty of decoding a random code. It is still a challenge to develop new public key cryptosystem originating from the hardness of non number-theoretic problems.

Using the idea from the McEliece cryptosystem it is straightforward to set up a cryptosystem based on the hardness of the Closest-Vector-Problem (CVP). A message is encoded as a lattice point plus a small error vector. To decipher the encrypted message we look for the closest lattice point, eliminating the small error. The open problem (which we address in this paper) is to find a suitable trapdoor. For the general case the only known trapdoor is an obvious one: a strongly reduced lattice base. But if we apply lattice reduction algorithms to compute a reduced base for a given lattice, then an adversary can do it, too, and in general it is not known how to create strongly reduced bases. At Crypto '97 Goldreich, Goldwasser and Halevi [GGH97] took a practical approach to design a CVP-based cryptosystem (GGH scheme). Based on experiments they restrict themselves to a class of lattices defined by rather simple reduced bases. But simple attacks on the secret key show some weakness in the security of the trapdoor [SF+97]. A detailed cryptanalysis is given by Nguyen [N99].

In this paper we propose two trapdoors for the Closest-Vector-Problem based on the tensor product. We build a kind of strongly reduced lattice base using the tensor product of low dimensional lattices. The construction resembles iterated

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 244–257, 1999.

codes where one e ciently decodes the tensor product given decoding algorithms for the individual codes. The second idea also applies the tensor product of lattices but in a di erent way. Finding the nearby vector in one component lattice enables to solve a restricted closest vector problem for the tensor product which is used to hide this secret structure.

2 Lattices, Reduction, and Closest-Vector-Problem

In this section we recall facts about lattices and lattice reduction. To simplify, we usually restrict ourselves to full dimensional lattices.

De nition 1 (Lattice) *Given an ordered set (matrix) $B := [b_1, \ldots, b_n]$ of n linear independent column vectors in $\mathbb{R}^m$, the set of all integral linear combinations of the vectors*

$$L = L(B) := \left\{ \sum_{i=1}^{n} t_i b_i \quad t_i \in \mathbb{Z} \right\} = \sum_{i=1}^{n} \mathbb{Z} b_i$$

is called a lattice generated by the base B. Its dimension is $\dim L := n$ and if $n = m$ we call it a full dimensional lattice. The vectors L are called lattice points. A lattice $L_{\text{sub}} \subseteq L$ with $\dim L_{\text{sub}} = \dim L$ is a sublattice of L. Sublattices of $\mathbb{Z}^m$ are called integer lattices.

There are several bases for a lattice. Multiplying a base vector with -1 or adding an integral multiple of another base vector does not change the generated lattice. Two bases B, B' generate the same lattice i there is an unimodular matrix $U \in \mathrm{GL}_n(\mathbb{Z}) := \{ U \in \mathbb{Z}^{n \times n} \mid \det U = \pm 1 \}$ with $B' = BU$.

De nition 2 (Reciprocal Base and Lattice) *If B is a base for the lattice $L \subseteq \mathbb{R}^m$, then the $m \times n$ matrix B^* with $B^* (B^*)^T = \mathrm{Id}_n$ is called the reciprocal (or dual) base to B. $L^* := L(B^*)$ is called the reciprocal lattice to L.*

The relation between primal/dual lattice and the lattice determinant are independent of the chosen lattice base for the lattice L:

De nition 3 (Determinant) *The determinant $\det L$ of a lattice $L = L(B)$ with base vectors $b_1, \ldots, b_n$ is the n-dimensional volume of the fundamental parallelepiped $\sum_{i=1}^{n} [0,1) b_i$ which equals $\sqrt{\det(BB^T)}$ and in case of a full dimensional lattice $|\det B|$.*

Obviously $\det \mathbb{Z}^n = 1$. For a full dimensional lattice $L \subseteq \mathbb{Z}^n$ we have $(\det L)\, e_i \in L$ for $i = 1, \ldots, n$ [DKT87]. Thus, given a lattice base $b_1, \ldots, b_n$ one derives a base \reduced modulo the lattice determinant" for the same lattice by iteratively adding integral multiples of $(\det L)\, e_i \in L$ such that the n vectors are linear independent and their entries are in $[0, \det L]$.

With a base $b_1, \ldots, b_n$ of a lattice L we associate the *Gram-Schmidt orthogonalization* $\tilde{b}_1, \ldots, \tilde{b}_n$ which is computed together with the Gram-Schmidt coe cients $\mu_{i,j} := \langle b_i, \tilde{b}_j \rangle / \| \tilde{b}_j \|^2$ for $i > j$ by the recursion $\tilde{b}_1 := b_1$ and

$\tilde{b}_i := b_i - \sum_{j=1}^{i-1} \mu_{i,j}\tilde{b}_j$ for $i = 2,\ldots,n$: $\|\tilde{b}_i\|$ is the height of the i^{th} base vector. Unless stated otherwise, we use the standard scalar product $\langle \cdot,\cdot \rangle$ and the corresponding Eulicidian norm $\|\cdot\|$. Letting $\mu_{i,i} := 1$ and $\mu_{i,j} := 0$ for $i < j$ one gets the Gram-Schmidt decomposition $[b_1,\ldots,b_n] = [\tilde{b}_1,\ldots,\tilde{b}_n]\,[\mu_{i,j}]^T$. Observe that $\det L = \prod_{i=1}^{n} \|\tilde{b}_i\|$ and that in general the vectors $\tilde{b}_1,\ldots,\tilde{b}_n$ do not generate the lattice L. Call π_i the orthogonal projection $\pi_i : \mathrm{span}\{b_1,\ldots,b_n\} \to \mathrm{span}\{b_1,\ldots,b_i\}^{\perp}$. We have $\pi_i(b_i) = \tilde{b}_i$.

De nition 4 (Successive Minima) *The i^{th} successive minima $\lambda_i(L)$ of a lattice $L \subseteq \mathbb{R}^m$ is the smallest $\lambda > 0$ such that there are i linear independent lattice points $v_1,\ldots,v_i \in L \setminus \{0\}$ with $\|v_i\| \leq \lambda$.*

Minkowski derived a general upper bound for the rst successive minima in terms of the lattice determinant [Ca97]:

Proposition 1 (Minkowski 1896) *If $L \subseteq \mathbb{R}^m$ is a lattice, then $\lambda_1(L) \leq \sqrt{m}\,(\det L)^{\frac{1}{\dim L}}$.*

If we scale the lattice by a factor $\alpha > 0$, i.e. multiply each lattice point by α, the successive minima are scaled, too. To normalize the quantity $\lambda_1(L)$ one takes the ratio $\lambda_1(L)/\det L^{\frac{1}{\dim L}}$. The squared maximum of this ratio for full dimensional lattices is called the Hermite constant γ_n.

The closest lattice point is uniquely determined if the minimal distance is less than $\frac{1}{2}\lambda_1(L)$. Given the value for the lattice determinant we look for lattices with large rst successive minima (so called dense lattices).

De nition 5 (Closest-Vector-Problem CVP) *Given a full dimensional lattice $L \subseteq \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$ the Closest-Vector-Problem is to nd a lattice point $b \in L$ with minimal distance $\delta(L,x) := \min_{b \in L} \|x - b\|$.*

CVP is NP-hard [K87] and for large dimension it is conjectured to be \average-case" intractable. On the other hand, Babai [B86] proposed a procedure which e ciently approximates the nearby vector within a factor of $2^{n/2}$. Using a (fairly theoretical) algorithm of Schnorr [S87] one approximates in polynomial time the closest vector up to a factor $(1+\varepsilon)^n$ for any xed $\varepsilon > 0$, but the running time badly depends on ε. If the distance $\delta(L,x)$ is below a threshold and we know a suitable base B for the lattice L, then CVP can be easily solved [B86, FK89]:

Lemma 1 (Nearest Plane) *Suppose L is a full dimensional lattice given by a base $b_1,\ldots,b_n$ with $\|\tilde{b}_i\| > 2d$. For $x \in \mathbb{R}^n$ with $\delta(L,x) \leq d$ one can e ciently compute the uniquely determined closest lattice vector.*

The aim of lattice reduction is to nd a base such that the vectors are rather orthogonal and the heights are large. We de ne lattice reduction in terms of the generalized δ-reduction introduced by Schnorr [S87, S94]:

De nition 6 (Lattice Reduction) *Given a base $B = [b_1,\ldots,b_n]$ and $\delta \in [2,n]$ denote $L_{i,\delta} := L(b_i,\ldots,b_{\min(i+\delta-1,n)})$. We call the base B δ-reduced if*

a) $|\mu_{i,j}| \leq \frac{1}{2}$ *for* $1 \leq j < i \leq n$.
b) $\|b_i^*\| = \lambda_1(\Lambda_{i,\ldots,n})$ *for* $i = 1,\ldots,n$.

There are two special cases: For $\beta = 2$ *it is called LLL base and for* $\beta = n$ *one calls it HKZ base.* B *is a reciprocal* β*-reduced base, if the reduction holds for the reverse ordered reciprocal base* B^* *[LLS90].*

For reduced bases in the sense of Hermite, Korkine and Zolotarev (HKZ) the first base vector is a shortest non-zero vector of the lattice. Like finding a shortest non-zero vector (Shortest Vector Problem, SVP) computing a HKZ base is intractable. To the best of our knowledge it is an open problem if a β-reduced base can be efficiently computed for a given lattice (even if β is fixed). For practical purposes an implementation of Schnorr and Hörner [SH95] quickly computes a β-reduced base for $\beta = 50$, $n \leq 200$. On the other hand a reduced base in the sense of Lenstra, Lenstra and Lovasz (LLL) can be computed in polynomial time [LLL82].

Proposition 2 ([LLL82, LLS90]) *Let* $B = [b_1,\ldots,b_n]$ *be a base of the lattice* $L \subseteq \mathbb{R}^m$. *For* $i = 1,\ldots,n$

a) If B *is LLL reduced, then* $\|b_i\| \leq \frac{\lambda_i(L)}{2^{(i-1)/2}} \leq \frac{\lambda_1(L)}{2^{(n-1)/2}}$.

b) If B *is reciprocal HKZ reduced, then* $\|b_i\| \leq \frac{\lambda_1(L)}{\gamma_i} \leq \frac{\lambda_1(L)}{n}$.

For reciprocal β-reduced bases the lower bound is about $\lambda_1(L)/\gamma_i \geq \gamma_\beta^{n/\beta}$ (combine the proof of [LLS90, Prop. 4.1] and [S94, Theorem 4]).

3 Tensor Product of Lattices

Starting in 1954 when P. Elias introduced so called iterated or product code the tensor product has become a major way to combine two codes [MS77]. The tensor product applies to lattices, too [M96]. Given two full dimensional lattice bases A, B the Kronecker product $A \otimes B$ of the two matrices is a base for the tensor product $L(A) \otimes L(B)$ which is a lattice, too. The Kronecker product of a $k \times l$ matrix $A = [a_{ij}]$ and an $m \times n$ matrix $B = [b_{ij}]$ is the $km \times ln$ matrix obtained from A by replacing each entry a_{ij} by $a_{ij}B$. For example, scaling a lattice L by a factor $\alpha > 0$ can be written as tensor product $\alpha L = L \otimes \mathbb{Z}$ as $[\alpha]$ is a base of $\mathbb{Z}$. Despite the fact that in general $A \otimes B \neq B \otimes A$ the associative and distributive law still hold [L96]. Let $L := L_1 \otimes L_2$ denote the tensor product of two full-dimensional lattices $L_1 \subseteq \mathbb{R}^{n_1}$ and $L_2 \subseteq \mathbb{R}^{n_2}$. The lattice point $b := (b_1,\ldots,b_{n_1 n_2}) \in L$ can be written as a two dimensional array

$$\begin{array}{cccc} b_1 & b_2 & \ldots & b_{n_2} \in L_2 \\ \vdots & \vdots & & \vdots \\ b_{(n_1-1)n_2+1} & b_{(n_1-1)n_2+2} & \ldots & b_{n_1 n_2} \in L_2 \\ \in L_1 & \in L_1 & & \in L_1 \end{array}$$

such that the column vectors belong to the lattice L_1 and the row vectors to L_2.

Finally, let us recall some facts about the tensor product of lattices, for details see [K93, Chapter 7] or [M96, x1.10].

Proposition 3 *Suppose $L_1; L_2$ are two full dimensional lattices. Then we have* $\dim(L_1 \quad L_2) = \dim L_1 \quad \dim L_2$ *and* $\det(L_1 \quad L_2) = (\det L_1)^{\dim L_2} \quad (\det L_2)^{\dim L_1}$.

Using induction we derive for the tensor product of t lattices $L_1; :::; L_t$:

$$\det(L_1 \qquad L_t) = \prod_{i=1}^{\quad} \det L_i^{\prod_{j \neq i} \dim L_j}: \tag{1}$$

Proposition 4 *Suppose $L_1; L_2$ are two full dimensional lattices. Then we have* $_1(L_1 \quad L_2) \quad _1(L_1) \quad _1(L_2)$ *with equality if* $\dim L_1 \quad 43$ *or* $\dim L_2 \quad 43$.

4 CVP-Based Public Key Cryptosystems

Frame Work. We call lattices $L \quad \mathbb{R}^n$ $d(n)$-decodable if using a trapdoor one can easily determine the nearby vector for $x \; 2 \; \mathbb{R}^n$ with $(L; x) \quad d(n)$ (bounded distance decoding). Let B be a base of the $d(n)$-decodable lattice L. The trapdoor (for example B) is the secret key. The public key consists of a base B_{pub} for the public lattice $L_{\text{pub}} := L(B_{\text{pub}})$ which is related to L (or even equals L). By choosing a public base B_{pub} one must regard two aspects:

1. Given the base B_{pub} an adversary may not get \useful information" about the secret trapdoor enabling him to break the system.
2. It should be intractable to solve the closest vector problem by simply applying lattice reduction to B_{pub}.

To avoid native attacks select a random unimodular matrix U transforming the base B. Analogous to the McEliece scheme we use a random orthogonal mapping R (rotation, i.e. $R^{-1} = R^{\mathsf{T}}$) to hide the trapdoor (tensor lattice structure). Set

$$B_{\text{pub}} := R \quad (BU):$$

Note that rotating the base also changes the lattice, i.e. $L_{\text{pub}} = R(L)$, meanwhile this does not change the vector lengths nor the lattice determinant. To encrypt a message $m \; 2 \; \mathbb{Z}^n$ select an error vector $e \; 2 \; \mathbb{R}^n$ with $kek \quad d(n)$ and send

$$y := B_{\text{pub}} m + e = RBUm + e:$$

Using the trapdoor one determines the closest vector to $R^{-1}y = B(Um) + R^{-1}e$ which is $b := Um$ because $\|R^{-1}e\| = kek \quad d(n)$. Now $U^{-1}b$ equals m.

Choosing the unimodular matrix U is done by multiplying several elementary matrices (representing the elementary operations: exchanging two columns, adding a multiple of a column to another, flipping the sign of a column) [GGH97].

Choosing a random rotation R is more difficult. This problem has been addressed by Sloane [Sl82] suggesting orthogonal matrices based on Hadamard matrices. This solves a second problem as, in general, orthogonal matrices have real coefficients. But for the restricted class of orthogonal matrices the elements are rational numbers such that scaling the lattice by $\sqrt{n}$ yields an integer one (details are given in Appendix A). Then we can reduce the public base \modulo $\det L_{\text{pub}}$".

But scaling has a disadvantage: The determinant increases by a factor $(\sqrt{n})^n$ and the bit length of the public key grows by $O(n^3 \log_2 n)$ bits. So depending on the trapdoor and efficiency one may use only a permutation matrix P instead of a rotation R. Or simply apply a unimodular transformation without any rotation. The GGH scheme just applies a permutation as the structure of the secret base is publicly known. It uses as secret key a random lattice base $B \in_R \sqrt{n}\,\mathrm{Id}_n + [-4, +4]^{n \times n}$ which enables one to decrypt a message with probability of order $1 - \frac{1}{n}$ for a random error vector $e \in_R \{\pm g\}^n$ where $g = 2$.

Finding the Tensor Decomposition. To the best of our knowledge no efficient algorithm for finding a tensor lattice decomposition is known. For matrices over a finite field a decomposition can be found in (small) dimensions [OL97]. But for the CVP trapdoor the matrices are over $\mathbb{Z}$, the dimension is rather large and the coordinates are permutated (respectively we rotate the lattice).

One possible weakness is the special form of the lattice determinant (1). Factoring this number may yield the dimension of the component lattices. To counteract this possible weakness simply choose lattices with equal determinant whose dimensions have many prime factors (for example, powers of 2). Even if the adversary cannot deduce the dimensions from the determinant he knows that the dimensions of the component lattices are at most 43.

To undo the permutation P the adversary tries to identify coordinates belonging to the same copy of a component lattice. Let $L_1 = L(A)$ and $L_2 = L(B)$ denote the component lattices and $L := P(L_1 \otimes L_2)$ the public key. From the proof given in Appendix B we derive

$$\det L = \prod_{i=1}^{\dim L_1} \prod_{j=1}^{\dim L_2} \|\mathbf{a}_i\| \, \|\mathbf{b}_j\| = \prod_{i=1}^{\dim L_1} \|\mathbf{a}_i\|^{\dim L_2} \det L_2.$$

If one removes a coordinate then the heights change, the new lattice has no tensor structure and the determinant is no longer of the form (1). But if we delete $\dim L_2$ coordinates belonging to the same i^{th} copy of the component lattice L_2, then the lattice determinant is divided by $\|\mathbf{a}_i\|^{\dim L_2} \det L_2$. But in general this in no integer because the height is a real number (although its square is a rational number). So the adversary cannot verify the correctness of his choice by checking the divisibility. For higher dimension this \attack" is intractable as there are $\binom{\dim L_1 \dim L_2}{\dim L_2}$ possible choices.

Attacks by Lattice Reduction. There is a simple but very powerful heuristic attack on the GGH scheme and all other CVP-based cryptosystems [SF+97].

Given an encrypted message $x = B_{\text{pub}}m + e$ apply a lattice reduction algorithm to the lattice L_{ext} generated by the $(n + 1) \quad (n + 1)$ matrix

$$B_{\text{ext}} := \begin{matrix} x & B_{\text{pub}} \\ 1 & 0 \end{matrix} = \begin{matrix} e & B_{\text{pub}} \\ 1 & 0 \end{matrix} \quad U; \qquad \text{for some } U \ 2 \ \text{GL}_{n+1}(\mathbb{Z}):$$

One assumes that the shortest non-zero vector of L_{ext} is $e_{\text{ext}} := \quad (e; 1)$ (which yields the plaintext) and that $\ _2(L_{\text{ext}}) = \ _1(L_{\text{pub}})$. To retrieve e_{ext} the adversary approximates the shortest vector for L_{ext} within a factor less than $\ _1(L_{\text{pub}})=kek$ (ignoring the additional $\quad 1$ entry).[1] To minimize this ratio take $kek \quad \frac{1}{2} \ _1(L)$ such that the adversary retrieves the message if he computes the shortest non-zero vector up to a factor of $\quad 2$. The LLL algorithm [LLL82] approximates the shortest vector up to a factor of $2^{\frac{n-1}{2}}$ and for $\quad$-reduced bases the factor is about $2^{n=}$ [S94, Theorem 4]. Although in case of the GGH scheme one must approximate the shortest vector within a factor 2, in dimension 150 it takes 15 minutes to retrieve the error vector [SF+97]. As within $1\frac{1}{2}$ hours lattice reduction even yields the secret key for the lattice. Thus, it is questionable if the class of lattice is a good choice. The recent cryptanalysis of Nguyen [N99] strengthens this results and reveals a second weakness: As $e \ 2 \ f \quad g^n$ the adversary retrieves $m \bmod 2 \quad$ by solving the modular system $y + [\ ;:::;\ ]^T = B_{\text{pub}}m \pmod 2 \ )$ which simpli es the closest vector problem where the error vector length is now $\frac{1}{2}\sqrt{n}$ compared to $\sqrt{n}$. To counteract this attack the error vector should not have a pattern like $e \ 2 \ f \quad g^n$. For example, take a random $e \ 2_{\text{R}} [- \ ; + \ ]^n$.

Comparison with McEliece Scheme. The McEliece scheme has not become widely used because rather large key is required. It is commonly believed that CVP-based cryptosystems are weaker than the analogous McEliece schemes due to the powerful lattice reduction algorithms. On the other hand, the strong attacks on the McEliece schemes [CSe98] seem not to be suitable for its lattice based variants. Let G be the generating $n \quad k$ matrix of the secret $[n; k; d]$ error correcting code C, i.e. $C \quad f0; 1g^n$, $\dim C = k$ and the minimal Hamming distance is d. The public matrix for the McEliece scheme is $G_{\text{pub}} := SGP$ over the eld $\mathbb{Z}=2\mathbb{Z}$ where $S \ 2 \ \text{GL}_k(\mathbb{Z}=2\mathbb{Z})$ and P is a permutation matrix. The matrix S ensures that the generating matrix G is not systematic, e.g. $G \ 6= [\text{Id}_k \ A]$, because otherwise an encrypted message $y := mG_{\text{pub}} + r$ reveals some bits of the plain text. The trapdoor is only hidden by applying the permutation which restricts the class of suitable codes.

For codes and lattices multiplying the generating matrix respectively base matrix by an unimodular matrix does not change the code and lattice. Applying a permutation P changes the code into an equivalent code while the Hamming weight of code words remains. Applying a rotation R changes the lattice into an isometric lattice while the Euclidean length of lattice vectors remains. In both cases knowing the transformation one reduces decoding respectively nd

[1] We say an algorithm approximates the shortest vector for the lattice L within a factor $\quad 1$ if it outputs $b \ 2 \ L \ n \ f0g$ with $kbk \quad \ _1(L)$.

the closest vector to the original code (lattice). But without this knowledge it is assumed that an adversary does not get any \useful information" about the underlying structure.

Although there are unique normal forms for lattice bases they do not play the same role as systematic generator matrices for codes where most algorithms are based on this form. The well-known normal forms for lattice bases rely on the matrix structure rather on reduced lattice bases. Nevertheless the known SVP algorithms are far more powerful than algorithms for nding the smallest code word. Thus, it seems that the McEliece scheme is more liable to structural attacks trying to retrieve the secret key meanwhile for CVP systems it is easier to regain the error vector of a single message.

5 HKZ-Tensor-Trapdoor

We construct a kind of strongly reduced lattice bases for a large general class of lattices using the tensor product to iteratively combine low dimensional lattices. If A and B are reciprocal HKZ-reduced then the lower bound for the heights of $A \quad B$ is the product of both lower bounds. This enables us to apply Nearest-Plane like given a reciprocal HKZ base. In Appendix B we show:

Proposition 5 *Suppose A and B are bases of two lattices with $k\mathbf{a}_i k \quad h_A$ and $k\mathbf{b}_i k \quad h_B$. For the base $C := A \quad B$ of $L(A) \quad L(B)$ we have $k\mathbf{c}_i k \quad h_A h_B$.*

Using this result, one sets up the candidate trapdoor by composing the lattice L with dimension n out of t lattices $L_1; : : : ; L_t$ with $\dim L_i \; 2$ [2; 43]. For simplicity we assume all lattices have the same dimension c, e.g. $n = c^t$ and $t = \log_c n$. As we do not require any secret structure for the underlying low dimensional lattices we take \random" lattices (see discussion in Appendix C). For these lattices of xed and low dimension we can e ciently compute (reciprocal) HKZ bases both in a theoretical setting [K87] and in practice [SH95]. Let $B_1; : : : ; B_t$ denote the reciprocal HKZ bases, $B := B_1 \quad B_2 \quad ::: \quad B_t$ their Kronecker product and $L := L(B)$ the tensor product lattice. By induction based on Proposition 3 we derive $\det L = \prod_{i=1}^{t} (\det L_i)^{c^t} = \prod_{i=1}^{t} \det L_i^{\,n}$ and according to Proposition 4 and Proposition 5:

$$(L) = \prod_{i=1}^{\cdot} {}_1(L_i) \qquad k\mathbf{b}_i k \quad \prod_{i=1}^{\cdot} \frac{{}_1(L_1)}{\gamma_c} = \frac{{}_1(L)}{\gamma_c^t}$$

Now applying Nearest-Plane we are in the same situation as given a reciprocal HKZ base for the lattice L. Using the Kronecker product of the reciprocal HKZ-reduced bases we nd the nearby vector if the distance is below $\frac{1}{\gamma_c^t} \; {}_1(L)$. An adversary trying to nd the error vector by embedding it into a SVP problem has to approximate the shortest lattice vector within a factor at least γ_c^t. Using stronger bounds [CS88, Chapter 1] we have $\gamma_c < 1{:}75 + 0{:}12c$ for $c \; 2$ [8; 50]. For $n = 529$ this means: taking two 23-dimensional lattices (i.e. $c = 23$, $t = 2$), the

adversary has to approximate the shortest vector within a factor less than 21. Note, this is a worst case scenario and by building the lattice one may simply re-select a small lattice L_i if the heights of the HKZ base are too close to $\frac{_1(L_i)}{\gamma_c}$.

6 Tensor-Hiding-Trapdoor

Again, we use the tensor product but unlike creating a reduced base for the product we try to hide the component lattices. We solve a restricted Closet-Vector-Problem for the tensor product using an algorithm which nds the nearby vector in a component lattice. To generate a candidate trapdoor for an n-dimensional lattice L select two full dimensional lattices L_{decode} and L_{hide} with the following properties:

- We have $n_{\text{decode}} := \dim L_{\text{decode}}$ 43, $n_{\text{hide}} := \dim L_{\text{hide}} = \frac{n}{n_{\text{decode}}}$.
- L_{decode} is a dense lattice, i.e., for a given lattice determinant the rst successive minima $_1(L_{\text{decode}})$ is large.
- For the lattice L_{decode} we e ciently nd the nearby vector if the distance is at most a large threshold d_{decode}, for example d_{decode} $\frac{1}{2}$ $_1(L_{\text{decode}})$. Let $:= d_{\text{decode}}/\sqrt{n_{\text{decode}}}$ denote the distances in terms of the maximum norm.
- For a small 1: $\frac{\sqrt{2n_{\text{decode}}}\ d_{\text{decode}}}{_1(L_{\text{decode}})}$ $_1(L_{\text{hide}})$ $\frac{\sqrt{2n_{\text{decode}}}\ d_{\text{decode}}}{_1(L_{\text{decode}})}$
  The lower bound ensures that the plaintext still corresponds to the closest vector and the upper bound makes it intractable nding the message by means of simple lattice reduction.

The lattice L_{decode} may be based on a tower of nested error correcting codes. For the lattice L_{hide} use the tensor product and (for example) the lattices $D_n; E_n$ [CS88, Chapter 4] or the method given in Appendix C Let $L := L_{\text{decode}}$ L_{hide} (or visa versa) denote the public lattice. As in Section 4 we set up the cryptosystem but with two restrictions:

1. The error vector is e 2 $[-\ ;+\ ]^n$ (note $\|e\|$ $\frac{1}{2}\sqrt{2n}$ d_{decode}) and
2. Instead of using an orthogonal matrix we just apply a permutation because rotations do not keep the length in terms of the maximum norm xed.

According to the choice of $_1(L_{\text{hide}})$ and n_{decode} $n_{\text{hide}} = n$ Proposition 4 implies $\sqrt{2n}$ d_{decode} $_1(L)$ $\sqrt{2n}$ d_{decode}. Retrieving the plaintext is done by computing the unique nearby point: given a point $y := b + e$ with b 2 L we write y as the two-dimensional array and apply the given algorithm to determine the nearby point in L_{decode} for the $\dim L_{\text{hide}}$ rows. The concatenation of the results is the closest vector in L because otherwise there are two nearby lattice points. On the other hand, an adversary trying to nd the error vector by embedding it into a SVP problem has to approximate the shortest lattice vector within a factor of at least 2 . To avoid lattice reduction attacks the lattice dimension should be at least 500.

7 Conclusions and Open Problems

We have suggested two trapdoors for the Closest-Vector-Problem based on the conjectured computational difficulty of finding tensor decomposition after applying a permutation (respectively a rotation). It is nearly as hard as possible to retrieve the cipher text by means of simply applying lattice reduction. But like the McEliece scheme the public key is very large in a way that CVP-based crypto systems have no practical impact as long as factoring or discrete logarithm remain intractable. Given two lattices $L_1; L_2$ and oracles solving the closest vector problem in $L_1; L_2$, can one efficiently compute the nearby vector for $L_1 \quad L_2$? To the best of our knowledge this is an open problem whereas iterated codes are majority decodable if one of the component codes is majority decodable [R70].

References

[B86] L. Babai: *On Lovasz' Lattice Reduction and the Nearest Lattice Point Problem*, Combinatorica, vol. 6, pp. 1{13, 1986.

[CSe98] A. Canteaut and N. Sendrier: *Cryptanalysis of the Original McEliece Cryptosystem*, Asiacrypt '98, LNCS #1541, pp. 187-199, 1998.

[Ca97] J.W.S. Cassels: *An Introduction to the Geometry of Numbers*, Springer Verlag, 1997.

[Co93] H. Cohen: *A Course in Computational Algebraic Number Theory*, Graduate Texts in Mathematics, vol. 138, Springer Verlag, 1993.

[CS88] J.H. Conway and N.J. Sloane: *Sphere Packings, Lattices and Groups*, Springer Verlag, 1988.

[DKT87] P.D. Domich, R. Kannan and L.E. Trotter: *Hermite Normal Form Computation using modulo Determinant Arithmetic*, Mathematics of Operation Research, vol. 12(1), pp. 50{59, 1987.

[FK89] M.L. Furst and R. Kannan: *Succinct Certificates for Almost all Subset Sum Problems*, SIAM Journal on Computing, vol. 18(3), pp. 550{558, 1989.

[GGH97] O. Goldreich, S. Goldwasser, S. Halevi: *Public-Key Cryptosystems from Lattice Reduction Problems*, Crypto '97, LNCS #1294, pp. 112{131.

[K87] R. Kannan: *Minkowski's Convex Body Theorem and Integer Programming*, Mathematics of Operation Research, vol. 12(3), pp. 415{440, 1987.

[K93] Y. Kitaoka: *Arithmetic of Quadratic Forms*, Cambridge Tracts in Mathematics, vol. 106, Cambridge University Press, 1993.

[LLL82] A.K. Lenstra, H.W. Lenstra and L. Lovasz: *Factoring Polynomials with Rational Coefficients*, Mathematische Annalen, vol. 261, pp. 515{534, 1982.

[LLS90] J.C. Lagarias, H.W. Lenstra and C.P. Schnorr: *Korkin-Zolotarev Bases and successive Minima of a Lattice and its Reciprocal Lattice*, Combinatorica, vol. 10, pp. 333{348, 1990.

[L96] H. Lütkepohl: *Handbook of Matrices*, John Wiley & Son, England, 1996.

[MS77] F.J. MacWilliams and N.J. Sloane: *The Theory of Error Correcting Codes*, Mathematical Library Vol. 16, North-Holland, 1977.

[M96] J. Martinet: *Les Reseaux Parfaits des Espaces Euclidiens*, Masson, 1996.

[MO90] J.E. Mazo and A.M. Odlyzko: *Lattice Points in high-dimensional Spheres*, Monatshefte Mathematik, vol. 110(1), pp. 47{61, 1990.

[N99] P. Nguyen: *Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97*, Crypto '99, LNCS #???.

[OL97] E. O'Brien and C.R. Leedham-Green: *Recognising Tensor Products of Matrix Groups*, Int. Journal Algebra Computing, vol. 7, pp. 541{559, 1997.

[PS87] A. Paz and C.P. Schnorr: *Approximating Integer Lattices by Lattices with cyclic Factor Group*, 14.th ICALP, LNCS #267, pp. 386{393, 1987.

[R70] S.M. Reddy: *On Decoding Iterated Codes*, IEEE Transaction on Information Theory, Vol. 16(5), pp. 624{627, 1970.

[S87] C.P. Schnorr: *A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms*, Theoretical Computer Science, vol. 53, pp. 201{224, 1987.

[S94] C.P. Schnorr: *Block Reduced Lattice Bases and Successive Minima*, Combinatorics, Probability and Computing, vol. 3, pp. 507{522, 1994.

[SH95] C.P. Schnorr and H.H. Hörner: *Attacking the Chor-Rivest Cryptosystem by improved Lattice Reduction*, Eurocrypt '95, LNCS #921, pp. 1{12, 1995.

[SF+97] C.P. Schnorr, M. Fischlin, R. Fischlin, H. Koy and A. May: *Lattice Attacks on the GGH Cryptosystem*, Crypto '97 Rump Session, 1997.

[Sl82] N.J.A. Sloane: *Encryption by Random Rotations*, Workshop on Cryptography Burg Feuerstein, 1982, LNCS #149, pp. 71{128, 1983.

[St95] D.R. Stinson: *Cryptography: Theory and Practice*, CRC Press, 1995.

A Quasi-Random Orthogonal Matrix

In Section 4 we faced the problem how to generate a random orthogonal matrix and how to transform the possible real lattice into an integer one. Sloane [Sl82] has suggested using orthogonal matrices based on Hadamard matrices. He calls the generated rotations quasi-random orthogonal matrices. Let H_n be a Hadamard matrix of order n, in other words H_n is an $n \quad n$ matrix with coe - cients $\quad 1$ and $H_n H_n^{\mathsf{T}} = n\mathrm{Id}_n$. Note that $\frac{1}{\sqrt{n}} H_n$ is an orthogonal matrix. Flipping the sign of any row or column changes a Hadamard matrix into another. There are several known constructions for Hadamard matrices if the order n is a multiple of 4 [MS77, Ch. 2, x3]. To create an orthogonal matrix choose arbitrary permutations matrices $P; P'$ and diagonal matrices $D; D'$ with coe cients $\quad 1$. Let

$$R := \frac{1}{\sqrt{n}} \quad DPH_n P' D'$$

i.e. we permute and flip the sign of the rows and columns of the Hadamard matrix. Restrict n to powers of 2 and scale the lattice to get integral coe cients:

$$B_{\mathrm{pub}} := \sqrt{n} \quad RBU = (DPH_n P' D')BU$$

As we scale the lattice in each direction we have $L(B_{\mathrm{pub}}) = \sqrt{n} \quad L(RB)$. Successive minima and the base heights are also stretched by $\sqrt{n}$ and instead of e with $kek \quad d$ we demand $kek \quad \sqrt{n}d$. Clearly, the complexity of the closest vector problems remains the same. But the determinant increases by $\sqrt{n}^n$. To reduce this disadvantage may combine independent rotations which apply

only to a fixed number c of directions. Select $c \times c$ random orthogonal matrices $R_1, R_2, \ldots, R_{\frac{n}{c}}$ and a $n \times n$ permutation matrix P':

$$R := P' \begin{pmatrix} R_1 & & \\ & \ddots & \\ 0 & & R_{\frac{n}{c}} \end{pmatrix} P'.$$

Using this class of orthogonal matrices it is sufficient to scale the lattice by a factor $\sqrt{c}$ instead of $\sqrt{n}$.

B Proof of Proposition 5

Given two bases A, B of lattices with $\|\mathbf{a}_i\| \geq h_A$ and $\|\mathbf{b}_i\| \geq h_B$ we show $\|\mathbf{b}_i\| \geq h_A h_B$ for the base $C := A \otimes B$ of $L(A) \otimes L(B)$. Recall [L96]:

Proposition 6 *For two orthogonal matrices A, B the Kronecker product $C := A \otimes B$ is orthogonal, too.*

We use the so called Iwasawa decomposition [Co93, Corollary 2.5.6]. The base matrix C can be uniquely written as $C = DOT$ with

- a diagonal matrix D,
- an orthogonal matrix O and
- an upper triangle matrix T with diagonal elements equal to 1.

Compare the Iwasawa decomposition to the Gram-Schmidt decomposition. As $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ are orthogonal we have that T is the transposed Gram-Schmidt matrix and the i^{th} diagonal entry of D equals the height $\|\mathbf{b}_i\|$. Let $A = D_A O_A T_A$ and $B = D_B O_B T_B$ denote the Iwasawa decomposition of A and B. Using the associative and distributive law [L96]

$$U \otimes (X \otimes Y) = (U \otimes X) \otimes Y \qquad (U \otimes V)(X \otimes Y) = (UX) \otimes (VY)$$

we get for the Kronecker product $C = A \otimes B$:

$$\begin{aligned} C &= (D_A O_A T_A) \otimes (D_B O_B T_B) \\ &= [D_A O_A] \otimes [D_B O_B] \cdot T_A \otimes T_B \\ &= (D_A \otimes D_B) \cdot (O_A \otimes O_B) \cdot (T_A \otimes T_B). \end{aligned}$$

$D_A \otimes D_B$ is a diagonal matrix

$$D_A \otimes D_B = \begin{pmatrix} \|\mathbf{a}_1\| \cdot D_B & & 0 \\ & \ddots & \\ 0 & & \|\mathbf{a}_{n_A}\| \cdot D_B \end{pmatrix} \qquad D_B = \begin{pmatrix} \|\mathbf{b}_1\| & & 0 \\ & \ddots & \\ 0 & & \|\mathbf{b}_{n_B}\| \end{pmatrix}$$

where $n_A := \dim L(A)$ and $n_B := \dim L(B)$. Applying Proposition 6 yields that $O_A \otimes O_B$ is an orthogonal matrix, too. Finally, it is straightforward to verify that $T_A \otimes T_B$ is an upper triangle matrix with diagonal elements equal to 1. Hence, the diagonal coefficients of $D_A \otimes D_B$ are the heights of the base C. Using the lower bound for the entries of D_A, D_B we get the desired lower bound for the heights of the base C.

C Choosing a Random Lattice

We used the term \random lattice" in a sloppy way not precisely de ning it. The main reason is that there is no canonical uniform distribution for lattices. One approach is to set up the distribution in terms of lattice bases in a given unique normal form like the *Hermite normal form* [Co93, DKT87]. A matrix $B = [b_{ij}]$ with full row rank is in Hermite normal form, if

1. B is an lower triangular matrix[2] and
2. $0 \quad b_{ij} < b_{ii}$ for $j < i$.

Every lattice has a unique lattice base in Hermite normal form where the product of the diagonal elements equals the lattice determinant. To choose a random lattice base B, rst select random diagonal coe cients from a given interval $b_{ii} \; 2_R \; [1; r]$ and afterwards select the other non-zero coe cients, i.e. $b_{j;i} \; 2_R \; [0; b_{ii})$. This method can be used for generating the underlying lattices for the HKZ-Tensor-Trapdoor introduced in Section 5.

But for the Tensor-Hiding-Trapdoor we require some side information about the rst successive minima, e.g. $_1(L) = (\sqrt{n})$ for some small constants. We relax our de nition of \random lattices" because to the best of our knowledge given a base in normal form one cannot derive a bound for the rst successive minima beside Minkowski's upper bound. A well understood class of integer lattices is given by modular homogeneous linear equations [FK89]:

$$L_{a;m} := \left(x \; 2 \; \mathbb{Z}^n \; \middle| \; \sum_{i=1}^{n} x_i a_i \quad 0 \pmod{m} \right) :$$

Paz and Schnorr [PS87] have shown that any integer lattice can be \approximated" by such a lattice. $L_{a;m}$ is a n-dimensional lattice with determinant equal to $m = \gcd(a_1; : : : ; a_n; m)$. We restrict ourselves to prime moduli turning $\mathbb{Z}=m\mathbb{Z}$ into a nite eld. Fix a prime module p. What is the probability for $a \; 2_R \; [1; p)^n$ that $_1(L_{a;p}) \quad k$ for a given threshold k? Given a non-zero $x \; 2 \; S_n(k) \setminus \mathbb{Z}^n$ (where $S_n(k)$ denotes the sphere around the origin with radius k), let i with $x_i \neq 0$. Then $x \; 2 \; L_{a;p}$ i $x_i a_i \quad - \sum_{j \neq i} a_j x_j \pmod{p}$: As p is a prime, given x and a_j for $j \neq i$, there exists exactly one solution a_i in $[0; p)$. This yields:

$$\Pr_{a \; 2_R [1;p)^n} [\; _1(L_{a;p}) \quad k] < \frac{jS_n(k) \setminus \mathbb{Z}^n j}{p}$$

It is by no means trivial to derive a (sharp) bound for the number of integer points in a sphere [MO90]. But for our purpose it is su cient to approximate the number by the volume of the sphere:

$$\mathrm{vol}_n (S_n(k)) = k^n \quad \mathrm{vol}_n (S_n(1)) = \frac{k^n \quad \frac{n}{2}}{\Gamma \left(1 + \frac{n}{2}\right)}$$

[2] Some authors de ne it in terms of an upper triangular matrix.

If $k = c_0 \sqrt{n}$ for a small constant c_0, then for sufficient large n

$$\Pr_{a \in_R [1;p)^n} \left[\lambda_1(L_{a;p}) \geq c_0 \sqrt{n} \right] \geq \frac{2^{c_1 n}}{p}$$

where $c_1 > 0$ is a moderate constant, too. Now, choose a random $(c_1 + 1)n$ bit prime p and $a \in_R [1;p)^n$, then with probability $1 - 2^n$

$$c_0 \sqrt{n} < \lambda_1(L_{a;p}) < \sqrt{n}\, 2^{c_1 + 1 + \frac{1}{n}}$$

where the upper bound is derived using $\det L_{a;p} = p < 2^{(c_1+1)n+1}$ and Minkowski's Proposition 1.

Delegated Decryption

Yi Mu, Vijay Varadharajan, and Khan Quac Nguyen

School of Computing and IT, University of Western Sydney, Nepean,
P.O.Box 10, Kingswood, NSW 2747, Australia
fyimu,vijay,qnguyeng@cit.nepean.uws.edu.au

Abstract. This paper proposes a new public key based system that enables us to have a single public key with one or more decryption keys and a unique signing key. One straightforward application for our system is in delegated or proxy based decryption. The proxy based decryption requires that the decryption authority can be delegated to another party (proxy) without revealing the signing key information. This suggests that the proxy who has the legitimate right for decryption cannot sign on behalf of the public key owner; only the legitimate signer can be the owner of the public key.

1 Introduction

Public key cryptography is generally considered to be asymmetric, because the value of a public key di ers from the value of its corresponding secret key. One paramount achievement of public key cryptography lies in the fact that it enables both encryption and signature, where a secret key can be used for both decryption and signing, while its corresponding public key can be used for both encryption of a message and veri cation of signature. It is clear that in a public key system a party who can decrypt a message must possess the associated secret key that can also be used to sign.

Let's consider the situation where you want some party to check your encrypted email messages, whereas you do not want the party to sign on your behalf. The common solution for this situation is to disable the signing function of the secret key by explicitly specifying the public key as encryption only and the secret key as decryption only. On the other hand, you need to have another secret-public key pair for yourself, which can be used for signing. This approach is not convenient due to the maintenance of two key pairs and the separation of public key's duties. Moreover, for some popular email software such as PGP and PEM you can not disable the signing capacity of a secret key.

It is quite clear that we want a public key system where several secret keys map onto a single public key. This kind of mapping can actually be found in group signatures[1,2,3,4]. In a group signature system, any one in the group can sign using its secret key on behalf of its group. The signatures can be veri ed with the unique group public key. However, we can not use the method of group signatures in our system, because all existing group signature schemes cannot be used for encryption.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 258{269, 1999.

In this paper, we propose a proxy decryption system that addresses the problem raised above. In our system, there are two di erent types of secret keys associated with a unique public key. They are called *owner secret key* and *proxy secret key(s)* respectively. The owner secret key is to be used for signing, whereas the proxy secret key(s) are to be used for decryption only. Our scheme relies on the di culty in solving discrete logarithm problems associated with polynomial functions. Both the encryption and signing methods are based on ElGamal algorithms[5].

There are a couple of immediate applications of our scheme. First, let us consider a boss-secretary environment. The boss (say, Bob) in an organisation could be too busy to read his emails and may allow his secretary or proxy (say, Alice) to handle his encrypted incoming emails; However Bob does not allow Alice to sign. Second, consider an electronic commerce environment. In a bank, it may be necessary that majority of sta can receive and verify signed and encrypted electronic cash and cheques, whereas only some of them are allowed to sign/issue electronic cash and cheques on behalf of the bank.

The rest of this paper is organised as follows. Section 2 describes the cryptographic polynomial functions required for our scheme. To simplify our presentation we introduce the concept of vectors. The main task of this section is to prove that the designated vectors used in our system are secure. Section 3 outlines the setup of our system and de nes the cryptographic keys to be used. Section 4 is devoted to the proxy decryption scheme and the associated digital signature scheme. Section 5 describes two interesting extensions to the system, where we study how to combine a signature with an encryption. We also study a proxy signature scheme, where a proxy (or proxies) can sign on behalf of the key owner in such a way that the signer can be recognised as the legitimate proxy by the signature veri er. The nal section is the conclusion.

2 Preliminaries

In this section, we discuss the polynomial functions and introduce the concept of vector space.

2.1 Construction of Polynomial Functions

Throughout the paper we use the following notations: p is a large prime, Z_p is a multiplitive group of order $p-1$, and $g \in Z_p$ is a generator.

Given vectors $\mathbf{a} = (a_0; a_1; \ldots; a_n) \in (Z_{p-1})^{n+1}$ and $\mathbf{b} = (b_0; b_1; \ldots; b_n) \in (Z_{p-1})^{n+1}$, where $a_i; b_i \in Z_{p-1}$, the inner product of $\mathbf{a}$ and $\mathbf{b}$ is de ned as $\mathbf{a} \cdot \mathbf{b} = \sum_{i=0}^{n} a_i b_i \pmod{p-1}$. Vectors $\mathbf{a}$ and $\mathbf{b}$ are called orthogonal, i the inner product $\mathbf{a} \cdot \mathbf{b} = 0 \pmod{p-1}$.

Lemma 1. *Given $x_j \in Z_{p-1}$; $j = 1; 2; \ldots; n$, let*

$$a_0 = \prod_{j=1}^{n} (-x_j);$$

$$
\begin{aligned}
a_1 &= \sum_{i=1}\ \prod_{i \neq j} (-x_j); \\
&\vdots \\
a_{n-2} &= \sum_{i \neq j} (-x_i)(-x_j); \\
a_{n-1} &= \sum_{i=1} (-x_j); \\
a_n &= 1.
\end{aligned} \tag{1}
$$

Then the vector $a = (a_0; :::; a_n)$ *is orthogonal to the vector* $x_j^{(n)} = (1; x_j; x_j^2; :::; x_j^n)$, *i.e.* $a \cdot x_j = 0$ *for* $j = 1; :::; n$.

Proof: Given $x_i;\ i = 1; :::; n$, consider the function $f(x) = \prod_{i=1}^{n}(x - x_i)$. We have $f(x) = \prod_{i=1}^{n}(x - x_i) = \prod_{i=1}^{n}(-x_i) + (\sum_{i=1}^{n}\prod_{i \neq j}(-x_j))x + ::: + \sum_{i=1}^{n}(-x_i)x^{n-1} + x^n = a_0 + a_1x + a_2x^2 + ::: + a_nx^n$. Therefore, $\sum_{i=0}^{n} a_i x_j^i = 0$ or $a \cdot x_j^{(n)} = 0$. □

We can see that, if $a_1; :::; a_n$ are given, it could be possible to find x_i, $i = 1; :::; n$, by solving the equations $a \cdot x_j^{(n)} = 0$. The question then is how to hide the information of a such that all vectors $x_j^{(n)}$ that are orthogonal with a become computationally hard to determine. More precisely, we require our system to satisfy the following criteria:

(a) $a = (a_0; :::; a_n)$ is hidden such that it is computationally hard to be determined.
(b) It is computationally hard to determine a vector $x_j^{(n)}$ such that $a \cdot x_j^{(n)} = 0$.
(c) Given some hidden information of $a \in (Z_{p-1})^n$ where $x_j^{(n)} \cdot a = 0 \pmod{p-1}; j = 1; :::; n$, it is computationally hard to determine a $x_{n+1} \in Z_{p-1}$ that is not equal to $x_j;\ j = 1; :::; n$, and $a' = (a'_0; :::; a'_{n+1}) \in (Z_{p-1})^{n+1}$ such that $x_j^{(n+1)} \cdot a' = 0 \pmod{p-1}; j = 1; :::; n+1$.

The criteria (a) and (b) can be realised due to the difficulty in solving discrete log problems:

Lemma 2. *Let* $g(a) = (g^{a_0}; g^{a_1}; :::; g^{a_n})\ (g_0; g_1; :::; g_n) \bmod p$. *Given* $g(a)$ *for* $g \in Z_p$, *it is computationally hard to determine a vector* $x^{(n)}$ *such that* $a \cdot x^{(n)} = 0\ (\bmod p - 1)$ *or* $g^{a \cdot x^{(n)}} = 1\ (\bmod p)$.

Proof: Immediate. Based on discrete logarithms, the difficulty is due to finding $a_i = \log_g g_i$ is hard and finding a x_j such that $\prod_{i=0}^{n} g_i^{x_j^i} = 1 \pmod{p}$ is hard. □

However, Lemma 2 does not cover Criterion (c). According to Lemma 3 below it is easy to find a $x_{n+1} \in Z_{p-1}$ such that the dimension of g can be arbitrarily increased without the need knowing x_j, $j = 0; :::; n$.

Lemma 3. *Assume that* $g = (g_0, g_1, \ldots, g_n)$ *is given as in Lemma 2. Given* $x_{n+1} \in Z_{p-1}$, *a new vector* $g' = (g^{a'_0}, \ldots, g^{a'_{n+1}})$ *can be determined without the knowledge of* x_j, $j = 1, \ldots, n$, *such that* $g^{\alpha' \cdot x^{(n+1)}} = 1 \pmod{p}$, *for* $x = x_j$, $j = 1, \ldots, n+1$, *where* $a' = (a'_0, \ldots, a'_{n+1})$.

Proof: Let $a' = (a'_0, \ldots, a'_{n+1})$ be the coefficients of the polynomial function $f(x) = \prod_{i=1}^{n}(x - x_i)(x - x_{n+1}) = a' \cdot x^{(n+1)}$. Given the information $(g_0, \ldots, g_n)$, $g^{a'_l}$, $l = 0, 1, \ldots, n+1$, can be determined in the following way:

$$\begin{aligned} g^{a'_k} &= g^{\sum_{1 \le i_1 < \cdots < i_{n+1-k} \le n} x_{i_1} \cdots x_{i_{n+1-k}}} \\ &= g^{\sum_{1 \le i_1 < \cdots < i_{n+1-k}; i_j = n} x_{i_1} \cdots x_{i_{n+1-k}}} g^{x_{n+1} \sum_{1 \le i_1 < \cdots < i_{n-k}; i_j \ne n} x_{i_1} \cdots x_{i_{n-k}}} \\ &= g^{a_{k-1}} g^{x_{n+1} a_k} \\ &= g_{k-1} (g_k)^{x_{n+1}}; \quad k = 0, 1, \ldots, n. \end{aligned}$$

Note that $g^{a'_{n+1}} = g$. It is easy to check that $g^{\alpha' \cdot x_j^{(n+1)}} = 1 \pmod{p}$ for all $j = 0, \ldots, n+1$. □

The problem given in Lemma 3 is serious, as we will see later that it is equivalent to allowing any user to legally add proxies to the system. In Corollary 1, we show that this problem cannot be solved by simply adding a secret salt parameter (say, $s \in_R Z_{p-1}$) to g_i in the way of g_i^s where g_i is hidden by s.

Corollary 4. *Assume that* $x_1, x_2, \ldots, x_n$ *are solutions of* $f(x) = \sum_{i=0}^{n} a_i x^i = 0$, *where* a_n *may be not equal to 1 (due to the salt). If* $g^{a_0}, g^{a_1}, \ldots, g^{a_n}$ *are known, given* x_{n+1}, $g^{a'_0}, g^{a'_1}, \ldots, g^{a'_{n+1}}$ *can be determined, without the knowledge of* $x_1, x_2, \ldots, x_n$, *such that function*

$$g^{H(x)} = g^{a'_0 + a'_1 x + \cdots + a'_{n+1} x^{n+1}} = 1,$$

for $x = x_j, j = x_1, \ldots, x_{n+1}$.

The proof is straightforward and is omitted.

From Lemma 3 and Corollary 1, we conclude that one way to avoid someone illegally adding x_{n+1} to the set $x_1, x_2, \ldots, x_n$ is to hide one of the elements of $g_1, \ldots, g_{n-1}$ which are defined in Eq. (1). To achieve this, the following method is recommended.

For the given $(a_0, a_1, a_2, \ldots, a_n)$, we define a new vector $a' = (a_0, a'_1, a'_2, \ldots, a'_{n-1}, a_n)$, where $a'_1 = \cdots = a'_{n-1} = \sum_{i=1}^{n-1} a_i$. Let $g^{\alpha'} = (g^{a_0}, g^{a'_1}, g^{a'_2}, \ldots, g^{a'_{n-1}}, g^{a_n})$ and $A_j = \sum_{i=1; l=1; i \ne l}^{n-1} a_l x_j^i$, then $g^{-A_j} g^{\alpha' \cdot x_j} = 1$, for all x_j; $j = 1, \ldots, n$. Since g^{a_i} are not given in a clear form, i.e. they are hidden in $g^{a'_i}$; $i = 1, \ldots, n-1$, the issues raised in Lemma 3 are solved. For the sake of convenience, we rewrite $g^{\alpha'}$ as g, then $g = f(g_i, g_{i=0}^n)$.

3 System Setup

The system of proxy decryption consists of several senders, a boss and several decryption proxies.

- Encryption. Any party can encrypt a message using the unique public key of the boss.
- Decryption. Either the boss or any legitimate proxy can decrypt the message encrypted with the owner or boss public key.
- Signature. Only the boss can sign a message that can be veri ed by any recipient using the boss' public key.

Let Bob be the boss and Alice be his secretary, a proxy. Bob is the principal who owns the public key and the associated secret or decryption keys. In order to construct the secret-public key pair, Bob needs to choose a vector $x_j^{(n)}$ and the associated secret vector a such that $a \quad x_j^{(n)} = 0$ for $j = 0;:::;n$. The properties of a and $x_j^{(n)}$ are as described in the previous section.

To construct his secret-public key pair for a single-proxy system (without losing generality, let's set $n = 3$), Bob chooses three random numbers, $x_1; x_2; x_3 \; 2_R \; Z_{p-1}$ and computes $a_i; i = 0;:::;3$. a_i must be su ciently large such that nding discrete log $a_i = \log g^{a_i}$ is hard. Bob then picks a number $\quad 2_R \; Z_{p-1}$ at random and computes its inverse $\quad ^{-1}$ and proxy parameters $_j \qquad A_j$, for $j = 1; 2; 3$. Consider the set $f\not{g}_i g \qquad (g^{a_0}; g^{a_1+a_2}; g^{a_1+a_2}; g^{a_3}; g \quad ^{-1})$, for $i = 0; 1;:::; 4$. Namely, $\not{g}_0 = g_0$, $\not{g}_1 = \not{g}_2$, $\not{g}_3 = g$ and $\not{g}_4 = g \quad ^{-1}$. The public key of Bob is then $PK \qquad fp; f\not{g}_i g g$. Bob keeps x_1, x_2, $_1$, $_2$ and all a_i's and can use either x_1 or x_2 as his secret decryption key and a_i's as his signing key. Bob gives x_3 and $_3$ to Alice who uses x_3 as her secret proxy key and $_3$ as her proxy parameter de ned as follows:

De nition 5. *A proxy parameter is an indicator that proves to another party the legitimacy of the proxy. Proxy parameters are public. That is,*

- *they are not secrets to the public,*
- *they can be distributed along with signatures, and*
- *unlike public keys, they do not need to be signed by a trusted authority.*

Proxy parameters have to be computed by Bob. Each proxy needs to have a proxy parameter to be used in proxy decryption and proxy signing. We will see later that proxy parameters may actually be kept private when only encryption/decryption is involved in actual applications. They become public only for proxy signatures.

4 Proxy Decryption

Conceptually, we describe proxy decryption as follows:

De nition 6. *Given public key PK and its corresponding owner decryption key D, owner signing key S and the proxy key P, the ciphertext $PK(m)$ of message m can be decrypted using either D or P, i.e. $m = D(PK(m)) = P(PK(m))$. Only S but not P can be used to sign, i.e. $PK(S(m)) = m$ and $PK(P(m)) \not= m$.*

This de nition has omitted proxy parameters $_j$. Proxy parameters are only needed during a decryption process, whereas encryption involves only the public key of Bob. Bob and Alice can keep their proxy parameters secret if they wish.

4.1 Encryption Scheme

Consider a simple situation where three players are involved in the protocol, a sender, Bob and Alice. The public key in the protocol is $PK\ 2\ (p; g_i)$. Let H be a one way hash function and m be the message to be sent to Bob by the sender. The idea of this protocol is for the sender to encrypt m using the public key PK and produce the corresponding ciphertext that can be decrypted by either Bob or his proxy, Alice.

To encrypt a message m, the sender computes $k = H(m)$ and encrypts m using Bob's public key to obtain the ciphertext c $(c_1; c_2)$, where c_1 $(g^k g_0^k; g_i^k)$; $i = 1; :::; 4$, and $c_2 = mg^k \pmod p$. c is sent to Bob whose computer then automatically forwards the message to Alice.

Since Bob possesses the secret key, $D = x_1$ or x_2, and Alice possesses the proxy key $P = x_3$, either Bob or Alice can obtain m by decrypting the ciphertext c_2. The decryption key used to decrypt c_2 can be obtained by using either Bob's secret key or Alice's proxy key:

$$\begin{aligned} d \quad & g^k g_0^k \prod_{i=1}^{3} g_i^{kx_j^i} g_4^{-x_j} \\ &= g^k \prod_{i=0}^{3} g_i^{kx_j^i} g_4^{-x_j} \\ &= g^k \prod_{i=0}^{3} g^{a_i^0 k x_j^i} g_4^{-x_j} \\ &= g^k g^{k \sum_{i=0}^{3} a_i x_j^i} \\ &= g^k \pmod p. \end{aligned}$$

The last equality holds because $\sum_{i=0}^{3} a_i x_j^i = 0$. The message can be recovered by computing $m = c_2 / d \pmod p$. Once m is obtained, Bob or Alice can verify the correctness of the encryption by checking whether c_1 $fg^k g_0^k; g_i^k g$ with $k = H(m) \pmod{p-1}$.

Theorem 7. *Encryption protocol.*

> Completeness. *For a given ciphertext c, if the sender follows the correct procedures, both Bob and Alice will receive the same message.*
> Soundness. *(1) The sender cannot cheat by producing the ciphertext that can only be decrypted by Alice. (2) The recipient can verify the correctness of the encryption process even if he does not have the group public key.*

Proof:
Completeness. The proof is straightforward. As shown above, any one with a valid x_j can decrypt the ciphertext c and hence obtain the message m.
Soundness. (1) Notice that a is orthogonal to all x_j and based on the discussion given in Section 2, a cannot be modi ed. Therefore if x_j can decrypt the ciphertext, any x_i, $i \neq j$, can also decrypt it. (2) The proof is as follows: Once the

message m is decrypted, Bob or Alice can verify the correctness of ciphertext. It is done as follows:

- { Compute $k = H(m) \pmod{p-1}$.
- { Compute $\hat{g}_i = (\hat{g}_i^k)^{1=k}$ for $i = 1;:::;4$ and $\hat{g}_0 = (g^k \hat{g}_0^k)^{1=k} = g \pmod{p}$.
- { Reconstruct the group public key $Y = (\hat{g}_i^0)$, $i = 0;1;:::;4$.
- { Verify the correctness of the newly constructed public key by checking $g_4{}^j g^{a'} {}^{x_j} = 1$ with his private key x_j.

Since k is computed using $H(m)$, all the veri cations are successful if and only if the sender follows the correct procedures. □

4.2 Signing Scheme

The signature scheme is based on the ElGamal signature scheme [5]. In our system, Bob can sign a message and his digital signature can be veri ed using his public key; whereas Alice cannot sign on behalf of her boss.

We assume that the signer is Bob. The system is set up by Bob in the following manner: Bob chooses a large prime $p \; 2 \; Z_p$, selects a primitive element $g \; 2 \; Z_p$, computes his secret or signing key such that $d = \sum_{i=0}^{4} a_i^0$ where $a_4^0 = g^{-1}$, and sets the public key parameters to the tuple $(p;e)$ with $e = \sum_{i=0}^{4} \hat{g}_i$. Note that e can be computed by the veri er; so Bob does not need to inform the veri er of e. Actually, the public key is still in the form of $PK = (p;\hat{g}_i)$.

To sign a message m, Bob carries out a regular ElGamal signing operation, i.e. he rst picks a random number $r \; 2_R \; Z_{p-1}$ such that any r is relatively prime to $p-1$ and then computes $z = g^r \pmod{p}$ and $= r^{-1}(m - zd) \pmod{p-1}$ to form the digital signature $S_m \quad fm;z; \; g$, where $z \; 2 \; Z_p$ and $\; 2 \; Z_{p-1}$. The signature S_m is then sent to the veri er. To verify the signed message, V checks $e^z z \stackrel{?}{=} g^m$. If the equality holds, the signature is accepted.

5 Extension

In this section, we describe two interesting extensions to our scheme: sign-encryption and proxy signature.

5.1 Proxy Decryption with Sign-Encryption

In a sign-encrypting system encryption is combined with a sender's signature, sometimes called signcryption[6]. The advantage of such a scheme lies in the reduction of computational complexity. In other words, it is computationally more e cient than signing and then encrypting a message.

We propose such a sign-encryption by combining ElGamal signature scheme with our encryption scheme. Assume that the secret-public key pair of the sender consists of $(x_s;y_s)$, where $y_s = g^{x_s} \pmod{p}$ and $x_s \; 2 \; Z_p$.

To send a message m to Bob, the sender carries out the following steps:

1. Chooses a secret encryption key $2_R Z_{p-1}$, two random numbers $r_1; r_2 \; 2_R \; Z_{p-1}$ that are relatively prime to $p-1$, and $g \; 2 \; Z_p$ of order $p-1$ a primitive element.
2. Computes:

$$\begin{aligned} z_1 &= g^{r_1} \pmod p; \\ z_2 &= g^{r_2} \pmod p; \\ z &= z_1 z_2 \pmod p; \\ c_1 &\quad (z_2 g_0^r; g_1^r; g_2^r; g_3^r; g_4^r) \pmod p; \quad \text{where } r = r_1 + r_2 \\ c_2 &= mg \pmod p; \\ &= r^{-1}(\; - z_1 x_s) \pmod{p-1}: \end{aligned}$$

   z_2 and z are known only to the sender. The sign-encrypted message is then $c \quad (c_1; c_2; z_1; \quad)$.
3. Sends c to Bob whose computer may forward it to Alice.

Either Bob or Alice can open and verify the message by computing the decryption key and recovering the message m:

$$\begin{aligned} d &= y_s^{z_1}[z_1(z_2 g_0^r \prod_{i=1}^{3} g_i^{r x_j^i}) g_4^{r \; j}] \\ &= y_s^{z_1} z \; g^{r \sum_{i=0}^{3} a_i x_j} \\ &= g^{x_s z_1} z \\ &= g \pmod p; \\ m &= c_2 = d: \end{aligned}$$

Theorem 8. *Sign-encryption Protocol.*

Completeness. *If the sender follows the correct process, Bob or Alice will always accept the sign-encrypted message.*
Soundness. *A cheating signer cannot convince the veri er to accept the sign-encrypted message. A cheating veri er cannot recover the sign-encrypted message.*

Proof:
Completeness. The proof is straightforward by inspecting the protocol. Note that the completeness of the protocol follows from the properties of ElGamal algorithm.
Soundness. The message recovery is the combination of the ElGamal encryption and signature schemes. We examine the soundness by inspecting both the signature veri cation part and the encryption part. The encryption part is the ciphertext of $z_2 g_0^r$ and is based the same scheme used in Protocol 1. According

to the lemmas given in Section 2, only the recipient who holds a valid secret key or proxy key can recover the message. The signature part can be written in a more obvious form:

$$\begin{aligned} \text{Signature: } z_1 &= g^{r_1} \pmod p \\ &= r^{-1}(\ - z_1 x_s) \pmod{p-1}; \\ \text{Message recovery: } d &= g^{x_s z_1}(z_1 z_2) \ ; \\ m &= c_2 / d: \end{aligned}$$

We nd that it is a variant of the ElGamal signature scheme and is equivalent to the signature message recovery [7]. Actually, we can refer to the ElGamal signature as a special form of our scheme by omitting parameters r_2^{-1} and z_2 from the expressions above. The security in such systems is similar to discrete log problem in the ElGamal signatures. For example, assume that Eve tries to forge a signature for a given message m, without knowing x_s. Eve chooses a value z and then tries to nd the corresponding ; for this to be successful she must compute the discrete logarithm $\log_z y_s^{z_1} d$. □

5.2 Proxy Signature

In some cases, Bob may allow Alice to sign on his behalf, when say he is not available. This is the so-called proxy signature. Any veri er can use the boss' public key to verify her signature. It is important that proxy signatures should be distinguishable from Bob's signatures.

The concept of proxy signature was rst presented in [8]. However, our proxy signature scheme is fundamentally di erent. We de ne our proxy signature as follows:

De nition 9. *Given public key PK of Bob, proxy secret key P_j and the associated proxy parameter $_j$ for proxy j, the proxy signature on m is de ned as: $S_m = P_j(m) j_j$. The proxy signature can be veri ed using PK, i.e. $m = PK(S_m)$.*

The Scheme There is no additional setup requirement for enabling proxy signatures. Assume that the system is being set up by Bob. The secret key for Bob and Alice are x_1 (or x_2) and x_3 respectively. The public key is still $PK = (p; f g_i g)$. To sign a message m, Alice rst picks a set of random numbers $f r_i g \ 2_R \ Z_{p-1}; \ i = 0; :::; 3$, such that any r_i is relatively prime to $p - 1$ and then computes

$$z_i = g_i^{r_i} \pmod p; \ i = 1; 2; 3$$

and

$$_i = r_i^{-1}(m x_j^i - z_i) \pmod{p-1}; \ i = 1; 2; 3$$

to form the digital signature $S_m = (m; z_i; \ _i; \ _j)$. To verify the signed message, the veri er checks

$$\prod_{i=1}^{3} g_i^{z_i \ _i} g_4^{m \ _j} \stackrel{?}{=} g_0^{-m};$$

Theorem 10. *Proxy signature protocol.*

Completeness. *The receiver of the signature will always receive a correct signature from the corresponding proxy, if the proxy and veri er are honest.*
Soundness. *A cheating proxy cannot convince the veri er to accept the signature.*

Proof:
Completeness. Immediate.
Soundness. Assume that the proxy signature can be forged by an adversary, Eve. Then given $g\ 2\ Z_p$, Eve should be able to nd some $\sim_i$ and $\bar{z}_i$ such that

$$\prod_{i=1}^{3} g_i^{\bar{z}_i} \bar{z}_i^{\sim_i} = g^X; \text{ where } X = g_4^{-m\ j} g_0^{-m}:$$

Assume that the above equality holds. Then given $\bar{z}_2$, $\bar{z}_3$, $\sim_2$ and $\sim_3$, Eve should be able to determine $\bar{z}_1$ and $\sim_1$ and the above equation can be rewritten as

$$g^Y g_1^{\bar{z}_1} \bar{z}_1^{\sim_1} = g^X \quad \text{or} \quad g_1^{\bar{z}_1} \bar{z}_1^{\sim_1} = g^{X'}; \text{ where } X' = X - Y:$$

This, however, violates the ElGamal's properties. □

Untraceable Proxy Signature As mentioned in Section 3, each proxy parameter is unique to each proxy. Proxy parameters are not secret, so proxies in the above scheme are traceable. Since proxy parameters are not linked to the identities of proxies, it is easy to make proxies anonymous. In fact, we can slightly modify the protocol so that the proxies also become untraceable:

In the setup step, the proxy also initialises a random number $y\ 2_R\ Z_{p-1}$ such that has the new form:

$$_i = r_i^{-1}(ymx_j^i - z_i) \pmod{p-1};\ i = 1;2;3$$

to form the digital signature $S_m = (m; z_i;\ _i; y\ _j)$. To verify the signed message, the following con rmation protocol is needed:

1. The veri er computes $\prod_{i=1}^{3} g_i^{z_i\ i} g_4^{my\ j}$. Please note that if this step was implemented correctly, we have $= g_0^{-ym}$. However, because y is unknown to the veri er, three additional steps are needed to complete the veri cation.
2. The veri er selects a random number $2_R\ Z_{p-1}$ and computes its inverse $^{-1}$. The veri er then computes (g_0^{-ym}) and sends to the signer.
3. The signer computes $^{y^{-1}}$ and then sends to the veri er.
4. The veri er checks $^{-1} \stackrel{?}{=} g_0^{-m}$.

Obviously, the proxy's signature is untraceable, since veri er cannot link proxy parameters in di erent signatures to a particular proxy. Nevertheless, the untraceability applies only to users excluding Bob. This is because Bob has full information about $_j$, and therefore he can obtain y very easily by checking $\stackrel{?}{=} g_0^{-ym}$.

Theorem 11. *The untraceable protocol:*

Completeness. *If the prover and veri er follow the protocol then the veri er accepts the signature as a valid signature of* m.
Soundness. *A cheating prover cannot convince the veri er to accept the signature.*

Proof:
Completeness. There are two facts: First, the secret number and its inverse $^{-1}$ are known to the veri er only. To remove in , the prover or signer is faced with a discrete log problem. Second, only the prover knows the secret number y and its inverse y^{-1} and hence can remove y in . Therefore, the veri er accepts the signature if $^{-1} = g_0^{-m}$.
Soundness. Assume that a cheating signer who computes $_i$ without using one of proxy or owner secret keys and thus generates $S^0_{m^0} = (m^0; z^0_i; {}^0_i; y_j)$. This will result in a value other than $g_0^{-ym^0}$ in the veri cation: $\prod_{i=1}^{3} g_i^{z^0_i z^0_i {}^0_i} g_4^{m^0 y_j} \stackrel{\text{def}}{=} X$. The veri er then selects (; $^{-1}$) and computes X (mod p) that is then sent to the cheating signer. One way that he can convince the veri er is to nd and compute $(g_0^{-m^0})$. However, he needs to solve the discrete log problem, $\log_X X$, which is computationally hard to him. □

It is noted that the proof of knowledge on y by the signer is equivalent to proving her ability of removing y from . Therefore, we can make the veri cation process non-interactive by adapting non-interactive equality proof of discrete log proposed in [3]. In our case, the prover is the signer, who should prove that she knows y without revealing y to the veri er. The common knowledge is the primitive h g_0^m. The prover will prove that s/he knows the secret y from h^y (mod p) without revealing y.

The prover:

- Chooses $2_R Z_{p-1}$ at random and computes h (mod p).
- Computes $! = H(hk \ k)$ (mod $p-1$) and $= - !y$ (mod $p-1$).
- Sends ($!$; ;) with other signature data to the veri er who can verify the knowledge of equality proof, by checking $h = {}^{!}h$ (mod p), where $= \prod_{i=1}^{3} g_i^{z_i} z_i {}^{i} g_4^{my_j}$.

Readers are referred to Ref. [3] for details of non-interactive discrete log proof. Please note that the security assurance relies on the fact $!$ can only be computed after h has been computed or after has been chosen.

6 Conclusion

We have proposed a public key based system, where the tasks of decryption and signature have been separated by introducing proxy keys. Our system is based on a special polynomial function whose security properties have been investigated and found to be suitable for our system. Furthermore, we have also extended our scheme to sign-encryption as well as proxy signatures that allow proxies to sign

on behalf of the owner with the condition that their signature can be identi ed only by their owner. We believe that our system has its potential applicability in electronic commerce.

References

1. D. Chaum and E. van Heijst, \Group signatures," in *Advances in Cryptology | EUROCRYPT '91*, (New York), pp. 257{265, Springer-Verlag, 1991.
2. L. Chen and T. P. Pedersen, \New group signature schemes," in *Adances in cryptology - EUROCRYPT'94, Lecture Notes in Computer Science 950*, pp. 171{181, Springer-Verlag, Berlin, 1994.
3. J. Camenisch, \E cient and generalized group signatures," in *Adances in cryptology - EUROCRYPT'97, Lecture Notes in Computer Science 1233*, pp. 465{479, Springer-Verlag, Berlin, 1997.
4. J. Camenisch and M. Stadler, \E cient group signature schemes for large groups," in *Advances in Cryptology, Proc. CRYPTO 97,* LNCS 1296, pp. 410{424, Springer-Verlag, Berlin, 1997.
5. T. ElGamal, \A public-key cryptosystem and a signature scheme based on discrete logarithms," in *Advances in Cryptology, Proc. CRYPTO 84,* LNCS 196, pp. 10{18, Springer-Verlag, Berlin, 1985.
6. Y. Zheng, \Digital signcryption or how to achieve cost(signature & encryption) cost(signature) + cost(encryption)," in *Advances in Cryptology − CRYPTO '97 Proceedings*, Springer Verlag, 1997.
7. K. Nyberg and R. A. Rueppel, \Message recovery for signature schemes based on the discrete logarithm problem," in *Advances in Cryptology, Proc. EUROCRYPT 94,* LNCS 950, (Berlin), pp. 182{193, Springer-Verlag, 1994.
8. M. Mambo, K. Usuda, and E. Okamoto, \Proxy signatures for delegating signing operation," in *Proc. of the Third ACM Conf. on Computer and Communications Security,* pp. 48{57, 1996.

Fast and Space-E cient Adaptive Arithmetic Coding*

Boris Ryabko[1] and Andrei Fionov[1]

Siberian State University of Telecommunications and Information Sciences
Kirov St. 86, Novosibirsk 630102 Russia
fryabko, fionovg@neic.nsk.su

Abstract. We consider the problem of constructing an adaptive arithmetic code in the case when the source alphabet is large. A method is suggested whose coding time is less in order of magnitude than that for known methods. We also suggest an implementation of the method by using a data structure called \imaginary sliding window", which allows to signi cantly reduce the memory size of the encoder and decoder.

1 Introduction

Arithmetic coding is now one of the most popular methods of source coding. Its basic idea was formulated by Elias in the early 1960s (see [1]). However, the rst step toward practical implementation was made by Rissanen [2] and Pasco [3] in 1976. Soon after that in [4,5,6] the modern concept of arithmetic coding was suggested. The method was further developed in a number of works, see, e.g., [7,8].

The advantage of arithmetic coding over other coding techniques is that it allows to attain arbitrarily small coding redundancy per one source symbol at less computational e ort than any other method (the redundancy is de ned as the di erence between the mean codeword length and the source entropy). Furthermore, arithmetic coding may easily be applied to sources with unknown statistics, when being combined with adaptive models in an adaptive coding scheme (see, e.g., [9,10,11,12]).

The main problem in linking arithmetic coding with adaptive models is the following. All known adaptive models provide some estimate of the actual probability $p(a_i)$ for each letter a_i over a source alphabet A. But arithmetic coding is based on cumulative probabilities $q(a_i) = \sum_{j<i} p(a_j)$. Since probability estimates change after each coded symbol, cumulative probabilities must always be re-computed. This requires about $|A|/2$ operations per symbol, where $|A|$ denotes the size of the alphabet, which a ects the speed of coding. The reduction of the coding speed becomes the more noticeable as the alphabet size increases. From this point of view, the alphabet of $2^8 = 256$ symbols, which is commonly used in

* Supported by the Russian Foundation of Basic Research under the Grant no. 99-01-00586

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 270{279, 1999.

compressing computer les, may already be treated as a large one. With adoption of the Unicode the alphabet size will grow up to $2^{16} = 65536$ and become su ciently large to prevent any direct computation of cumulative probabilities. In this paper, we suggest an algorithm that computes cumulative probabilities in $O(\log jAj)$ time, which is exponentially less than for known methods.

To obtain a speci ed (arbitrarily small) model redundancy, denoted henceforth by r, we use an adaptive scheme with sliding window. The sliding window scheme possesses many useful properties and the only disadvantage, that has so far prevented its wide usage, is the necessity to store the entire window in the encoder (decoder) memory. In obtaining arbitrarily small redundancy, the memory allocated to the window becomes a dominating part in space complexity of the encoder and decoder, which thus grows as $O(1=r)$. To remedy this disadvantage we use a type of the scheme called \imaginary sliding window" (the concept of which was presented in [13]). This approach allows to preserve all the properties of sliding window without storing the window itself.

The space complexity (S) and time complexity (T) of the proposed method in case of a memoryless source, seen as functions of two variables, the alphabet size jAj and redundancy r, are upper bounded by the following estimates:

$$S < \text{const} \quad jAj \log \frac{jAj}{r} \quad \text{bits;}$$

$$T < \text{const} \quad \log \frac{jAj}{r} \quad \log jAj + \log\log \frac{jAj}{r} \log\log\log \frac{jAj}{r} \qquad \text{bit operations}$$

(here and below $\log x = \log_2 x$ and const denotes some (di erent) constants greater than 1). Generalizations toward Markov or tree sources are straightforward and can be done by known techniques.

The paper is organized as follows. In Sect. 2 we consider the problem of combining a conventional sliding window scheme with arithmetic coding. We describe a method for fast operation with cumulative probabilities and investigate its complexity. In Sect. 3 we present an imaginary sliding window scheme which dispenses with the need for storing the window.

2 Fast Coding Using Sliding Window

Let there be given a memoryless source generating letters over the alphabet $A = \{a_1; a_2; \ldots; a_n\}$ with unknown probabilities $p(a_1); p(a_2); \ldots; p(a_n)$. Let the size of the alphabet be a power of two, $n = 2^m$ (it is not a restrictive assumption, because, if $n < 2^m$, the alphabet may be expanded with $2^m - n$ dummy symbols having zero probability with trivial modi cations of the algorithms). Let the source generate a message $x_1 \ldots x_{l-1} x_l \ldots$, $x_i \in A$ for all i. The window is de ned as a sequence of the last w symbols generated by the source, where w denotes the size of the window. At the moment l the window contains the symbols $x_{l-w} \ldots x_{l-2} x_{l-1}$. During encoding (or decoding), the window \slides" along the message: as a novel symbol is introduced in the window, the oldest one

is removed. Each letter $a_i \in A$ is assigned a counter c_i of size $t = \lceil \log(w+n) \rceil$ bits that contains a number of occurences of a_i in the current window plus 1 (to prevent zero probability). In these settings the sum of all counters is equal to the window size plus the alphabet size, $\sum_{i=1}^{n} c_i = w + n$ and the estimates of the symbol probabilities $p(a_1), p(a_2), \ldots, p(a_n)$ may be obtained as $p(a_i) = c_i/(w+n)$ for all i.

Denote the novel symbol to be encoded by u and the oldest symbol stored in the window, to be removed, by v (at the moment l, $u = x_l$ and $v = x_{l-w}$). The adaptive encoding of the symbol is performed as follows: encode u according to the current estimated probability distribution ($p(u) = c(u)/(w+n)$); decrement counter $c(v)$ corresponding to the letter v; remove v out of the window; increment counter $c(u)$ corresponding to the letter u; introduce u in the window. The decoder, provided that it starts with the same counter contents as the encoder, decodes the symbol according to the current probability estimates and updates the counters in the same way as the encoder.

Encoding of a symbol given an estimated probability distribution may e - ciently be carried out by means of arithmetic coding (see [7,8] for more details). This technique, however, requires that cumulative range $[Q_i; Q_{i+1})$ be speci ed for the symbol $u = a_i$, where Q are de ned as follows:

$$Q_1 = 0; \quad Q_i = \sum_{j<i} c_j; \quad i = 2,3,\ldots,n+1. \tag{1}$$

The direct calculation of Q using (1) requires $O(tn)$ bit operations. Below we describe a method that allows to reduce complexity down to $O(t \log n)$.

Let us store not only the counters (denote this vector by C^1), but also the sums of successive pairs of counters (C^2), the sums of successive quadruples (C^3), and so on. For example, if $n = 8$, we need to store the following vectors

$$C^1 = (c_1; c_2; c_3; c_4; c_5; c_6; c_7; c_8);$$
$$C^2 = ((c_1 + c_2); (c_3 + c_4); (c_5 + c_6); (c_7 + c_8));$$
$$C^3 = ((c_1 + c_2 + c_3 + c_4); (c_5 + c_6 + c_7 + c_8)).$$

By using C the values of, e.g., Q_4 and Q_8 can be computed as

$$Q_4 = c_1 + c_2 + c_3 = C_1^2 + C_3^1;$$
$$Q_8 = c_1 + c_2 + \ldots + c_7 = C_1^3 + C_3^2 + C_7^1.$$

It is easily seen that computation of any Q_i, $i = 2,3,\ldots,8$, requires to sum up at most $3 = \log 8$ t-bit numbers.

In general case, we store vectors C^1, C^2, ..., C^m, $m = \log n$, which requires $(2n-2)t$ bits of memory. Let us show how to compute Q_{k+1}, $k = 1,2,\ldots,n$ (recall that $Q_1 = 0$). Let k have a binary expansion $k_m k_{m-1} \ldots k_1$, where each $k_i = 0$ or 1. Then

$$Q_{k+1} = k_m C^m_{b2k=nc} + k_{m-1} C^{m-1}_{b4k=nc} + k_{m-2} C^{m-2}_{b8k=nc} + \ldots + k_1 C^1_k$$

(note that only those C_j^i for which $k_i = 1$ are included in the computation). This sum has at most $m = \log n$ t-bit addents, therefore the complexity of computing cumulative counts is $O(t \log n)$ bit operations.

When incrementing or decrementing counters in the adaptive scheme, all the elements of the vectors which depend on those counters should also be incremented or decremented. More exactly, incrementing (decrementing) $c_k = C_k^1$ must be accompanied by incrementing (decrementing) $C_{dk=2e}^2$, $C_{dk=4e}^3$, ..., $C_{dk=(n=2)e}^m$, which requires also $O(t \log n)$ bit operations. To increment, e.g., c_3 ($n = 8$) we must increment not only C_3^1, but also C_2^2 and C_1^3.

Finally, the last operation which relies on cumulative counts is nding an interval $I(z) = [Q_i; Q_{i+1})$ that contains a given number z (and the letter $u = a_i$ to whom this interval corresponds). This problem arises in arithmetic decoding. Let us show how one can solve this problem using the vectors C in $O(t \log n)$ time. First compare z with C_1^m. If $z < C_1^m$ then the target interval lies within $[0; C_1^m)$, otherwise it belongs to $[C_1^m; (w+n))$. Go on comparing z with C_1^{m-1} in the former case, and with $(C_1^m + C_3^{m-1})$ in the latter, which determines one of the intervals $[0; C_1^{m-1})$, $[C_1^{m-1}; C_1^m)$, $[C_1^m; C_1^m + C_3^{m-1})$, or $[C_1^m + C_3^{m-1}; (w+n))$. After $m = \log n$ steps we obtain an interval corresponding to one letter.

The following theorem establishes the computational complexity of the proposed adaptive scheme as function of two arguments, the alphabet size n and redundancy r.

Theorem 1 *Let there be given a memoryless source generating letters over an alphabet of size n. Let the adaptive scheme based on sliding window and arithmetic coding be applied to the source providing a speci ed redundancy r. Then the memory size of the encoder (decoder) S and encoding (decoding) time T are given by the estimates*

$$S < \text{const} \quad \frac{n}{r} \log n + n \log \frac{n}{r} \quad ; \tag{2}$$

$$T < \text{const} \quad \log \frac{n}{r} \quad \log n + \log\log \frac{n}{r} \log\log\log \frac{n}{r} \qquad : \tag{3}$$

Proof. The redundancy r consists of two parts: the one caused by replacing unknown symbol probabilities by their estimates, denote it by r_{m} (model redundancy), and the other caused by the encoder, denote it by r_{c} (coding redundancy), $r \quad r_{\mathrm{m}} + r_{\mathrm{c}}$:

It is known (see, e.g., [14]) that for the sliding window scheme $r_{\mathrm{m}} < \text{const} n=w$. For arithmetic coding $r_{\mathrm{c}} < \text{const} n \ 2^{-} \ $ [8], where is an internal register size which must be $O(t)$, say, $= t + \log t + \text{const}$. For t we have $\log(w+n) \quad t < \log(w+n) + 1$. So we easily obtain that $r_{\mathrm{c}} < \text{const} n=w$. Hence, $r < \text{const} n=w$ and $w < \text{const} n=r$.

The memory size of the encoder and decoder is caused by the necessity to store the window, which requires $w \log n < \text{const}(n=r) \log n$ bits, the vectors C , which requires $(2n-2)t < \text{const} n \log(w+n) < \text{const} n \log(n=r)$ bits, and internal registers for arithmetic coding, which requires $O(\ ) < \text{const} t < \text{const} \log(n=r)$ bits. Summing up dominating complexities gives (2).

The coding time per symbol is determined by the computations over the vectors C , which requires $O(t \log n) < \text{const} \log(n/r) \log n$ bit operations, and the computation of ranges in arithmetic coding, which uses -bit multiplication and division, which, in turn, requires $O(\ \log\ \log\log\ )$ bit operations [8]. Taking into account $< \text{const} \log(n/r)$ and summing up gives (3). □

Corollary 1 *If n is increasing and r is to remain constant then*

$$S = O(n \log n) \ ;$$
$$T = O(\log^2 n) \ :$$

The time estimate is better than that for known methods (where $T = O(n \log n)$).

Corollary 2 *If n is xed and $r \ ! \ 0$ then*

$$S = O(1/r) \ ;$$
$$T = O(\log(1/r) \log\log(1/r) \log\log\log(1/r)) \ :$$

In the next section we show how to improve the memory size estimate to $O(\log(1/r))$ implementing the concept of imaginary sliding window.

3 Imaginary Sliding Window

Let there still be given a memoryless source de ned in Sect. 2. We also use the same notions of the window, counters and probability estimation. Denote, additionally, by $_i$ the number of occurences of the letter $a_i \ 2 \ A$ in the current window, $_i = c_i - 1$ for all i. De ne to be a random variable taking on values $1; 2; : : : ; n$ with probabilities

$$\Pr\{ \ = i\} = \frac{i}{w}; \quad i = 1; 2; : : : ; n \tag{4}$$

(the problem of generating such a variable will be considered further below).

The encoding of the novel symbol u is performed as follows: encode u according to the current estimated probability distribution ($p(u) = c(u)/(w + n)$); generate a random variable ; decrement counter c , i.e., the counter corresponding to the th letter of the alphabet A; increment counter $c(u)$ corresponding to the letter u.

A distinctive feature of this scheme is that a *randomly chosen* counter is decremented rather than a counter corresponding to the oldest symbol in the window. In this scheme, the context of the window is not used, therefore storing the window is not needed, which saves $w \log n$ bits of memory. But a question arises whether this scheme is able to represent the source statistics with the precision su cient to guarantee a speci ed redundancy? So the next our task is to show that it is, more exactly, we show that the estimated distribution provided by imaginary sliding window converges with exponential speed to the distribution provided by real sliding window.

Denote by the set of all vectors $= (\ _1;\ :\ :\ :;\ _n)$ such that all $_i$ are non-negative integers and $\sum_{i=1}^{n}\ _i = w$. It is clear that in case of real sliding window $= (\ _1;\ :\ :\ :;\ _n)$ is a random vector that obeys the multinomial distribution

$$\Pr\{\ _1 = \ _1;\ _2 = \ _2;\ :\ :\ :;\ _n = \ _n\} = \binom{w}{\ _1;\ _2;\ :\ :\ :;\ _n} \prod_{i=1}^{n} (p(a_i))^{\ _i}; \quad (5)$$

where

$$\binom{w}{\ _1;\ _2;\ :\ :\ :;\ _n} = \frac{w!}{\ _1!\ _2!\ :\ :\ :\ _n!}$$

(see, e.g., [15]).

In case of imaginary sliding window, $= (c_1 - 1;\ :\ :\ :;\ c_n - 1)$ is also a random vector whose distribution is a question to be answered. We shall indicate by superscript l the state of vector after the lth encoded symbol, i.e., in the process of coding vector assumes the states $^1;\ ^2;\ :\ :\ :;\ ^l;\ :\ :\ :$ The next theorem decides on the distribution for and shows that imaginary sliding window is a su ciently precise model of real sliding window.

Theorem 2 *Let there be given a memoryless source generating letters over the alphabet* $A = \{a_1;\ :\ :\ :;\ a_n\}$ *with probabilities* $p(a_1);\ :\ :\ :;\ p(a_n)$*; the imaginary sliding window scheme is used with the window size* w*. Then*

$$\lim_{l\ !\ 1} \Pr\{\ ^l_1 = \ _1;\ ^l_2 = \ _2;\ :\ :\ :;\ ^l_n = \ _n\} = \binom{w}{\ _1;\ _2;\ :\ :\ :;\ _n} \prod_{i=1}^{n} (p(a_i))^{\ _i}; \quad (6)$$

the limit in (6) exists for any initial distribution $^0 = (\ ^0_1;\ :\ :\ :;\ ^0_n)$*, and there exists a constant* $C < 1$*, independent of* $(\ _1;\ :\ :\ :;\ _n)$ *and* $(\ ^0_1;\ :\ :\ :;\ ^0_n)$*, such that*

$$\Pr\{\ ^l_1 = \ _1;\ ^l_2 = \ _2;\ :\ :\ :;\ ^l_n = \ _n\} - \binom{w}{\ _1;\ _2;\ :\ :\ :;\ _n} \prod_{i=1}^{n} (p(a_i))^{\ _i} < C^l: \quad (7)$$

Proof. De ne a Markov chain M with states, each state being correspondent to a vector from (informally, each vector $= (\ _1;\ :\ :\ :;\ _n)$ of imaginary sliding window has a corresponding state in M). A transition matrix for M is de ned as follows:

$$P_{\lambda^1\ !\ \lambda^2} = \begin{cases} \dfrac{\ _j}{w} p(a_i); & \text{if } \ ^2_i = \ ^1_i + 1;\ \ ^2_j = \ ^1_j - 1;\ i \neq j, \\ & \ ^2_k = \ ^1_k;\ k \neq i;\ k \neq j\ ; \\ \displaystyle\sum_{k=1}^{n} \frac{\ _k}{w} p(a_k); & \text{if } \ ^1 = \ ^2\ ; \\ 0; & \text{otherwise}\ . \end{cases} \quad (8)$$

This probability matrix corresponds to transition probabilities of imaginary sliding window. The rst line in (8) corresponds to the case when the jth coordinate of the vector ^{l} is decremented by 1, and the ith coordinate is incremented by

1, which means introducing the letter a_i in the window, which, in turn, occurs with probability $p(a_i)$. The second line in (8) corresponds to the case when the vector ^{l} remains intact. This happens if a coordinate $^{l}_{k}$ is rst decremented (with probability $^{l}_{k}=w$) and then incremented (with probability $p(a_k)$).

The chain M has a nite number of states and is plainly seen to be non-periodical (see, e.g., [15]). Consequently, there exist limiting probabilities for M, i.e., such λ that for any $;$ 2 there exists the limit

$$\lim_{l\,!\,1} P_{\mu\,!\,\lambda}(l) = \lambda \tag{9}$$

independent of (here $P_{\mu\,!\,\lambda}(l)$ denotes the probability of transition from to in l steps, l 1).

It is known (see [15]) that limiting probabilities satisfy the system of equations

$$\begin{cases} \lambda = \sum_{\mu 2} \mu P_{\mu\,!\,\lambda}; \quad 2 \quad ; \\ \sum_{\lambda 2} \lambda = 1 : \end{cases} \tag{10}$$

Next we show that for any $= (\ _1;:::;\ _n)$ 2

$$\lambda = \frac{w}{\ _1;\ _2;:::;\ _n} \prod_{i=1}^{\ } (p(a_i))^{\ i} : \tag{11}$$

To do this, put (11) and (8) in (10). As a result,

$$\begin{aligned} & \frac{w}{\ _1;\ _2;:::;\ _n} \prod_{i=1}^{\ } (p(a_i))^{\ i} \\ = & \sum_{i=1}^{\ } \sum_{j=1;j\in i}^{\ } \frac{w}{\ _1;:::;\ _i - 1;:::;\ _j + 1:::\ _n} \\ & \left(\prod_{i=1}^{\ } (p(a_i))^{\ i} \ \frac{p(a_j)}{p(a_i)} \ \frac{\ _j + 1}{w} p(a_i) \right) \\ & + \sum_{k=1}^{\ } \frac{w}{\ _1;\ _2;:::;\ _n} \prod_{i=1}^{\ } (p(a_i))^{\ i} \ \frac{k}{w} p(a_k) \quad : \end{aligned} \tag{12}$$

Put the equality

$$\frac{w}{\ _1;:::;\ _i - 1;:::;\ _j + 1:::\ _n} = \frac{w}{\ _1;\ _2;:::;\ _n} \ \frac{\ _i}{\ _j + 1}$$

in (12) and after simple transpositions obtain

$$\sum_{i=1}^{\ } \sum_{j=1;j\in i}^{\ } \frac{\ _i}{\ _j + 1} \frac{p(a_j)}{p(a_i)} \ \frac{\ _j + 1}{w} p(a_i) + \sum_{k=1}^{\ } \frac{k}{w} p(a_k) = 1 :$$

Consequently, (11) is the solution of (10). Taking into account the de nition (9) obtain (6).

The second statement of the theorem may be easily obtained from the general proposition on exponential speed convergence to a limiting distribution (see [15]) (this requires that a Markov chain should satisfy certain conditions which is easily checked in our case). □

From the rst proposition of the theorem it follows that imaginary sliding window behaves asymptotically just like as real sliding window or, more exactly, the distribution of $(\ _1^{l};\ldots;\ _n^{l})$ approaches, as l increases, the distribution (5) of the numbers of occurences of the letters $a_1;\ldots;a_n$ in the real window. From the second proposition it follows that in case of change in the source statistics at the moment l, the distribution of $(\ _1^{l+};\ldots;\ _n^{l+})$ converges to the new distribution as increases $((\ _1^{l};\ldots;\ _n^{l})$ being thus considered as an initial distribution). Therefore, applied to memoryless sources, imaginary sliding window has the same properties as real sliding window, namely, it allows to estimate the source statistics quite precisely and adapts fast if the statistics change.

Let us now consider the problem of generating random variable . Assume that the only source of randomness is a symmetric binary source generating a sequence $z_1z_2:::z_k:::$, where each $z_k\ 2\ f0;1g$, $\Pr fz_k = 0g = \Pr fz_k = 1g = 1=2$, and all symbols are independent. We do not consider here the complexity of obtaining random bits. Our aim is to convert a random bit sequence into a random variable with probability distribution (4). Denote by z a random number built from $s = d\log we$ random bits, i.e., the binary expansion of z is $z_1z_2:::z_s$. It is plain that z may take on any value from $f0;1;:::;2^s - 1g$ with probability $1=2^s$. It is also plain that, for any $w < 2^s$, the process of generating z until $z < w$ produces a random variable z that takes on any value from $f0;1;:::;w-1g$ with probability $1=w$. Since $w > 2^{s-1}$, less than 2 iterations are required on average.

De ne

$$_1 = 0; \qquad _i = \sum_{j<i} \ _j; \quad i = 2;3;:::;n+1; \quad (\ _{n+1} = w):$$

Consider a random variable over $f0;1;:::;w-1g$ such that $= i$ i $_i \quad z < \ _{i+1}$. It can be easily found that

$$\Pr f \ = ig = \Pr f \ _i \quad z \ < \ _{i+1}g = \frac{_{i+1} - \ _i}{w} = \frac{_i}{w}:$$

This is our intended distribution (4).

To implement this procedure notice that nding $_i$ and $_{i+1}$ given z is the same process as nding an interval and a correspondent symbol in arithmetic decoding (which has been described in Sect. 2). Hence, this procedure may be e ciently performed by using the data structure C de ned in Sect. 2. The only di erence is that the values $_i$ are less by one than counters c_i involved in the construction of C . This may be easily checked by substracting 2^{k-1} from C_j^k

in the process of nding a relevant interval. So, generating does not increase the the order of complexity of the method.

The results obtained in this section are summed up in the following

Theorem 3 *Let there be given a memoryless source generating letters over the alphabet* $A = \{a_1; a_2; \ldots; a_n\}$. *Let the adaptive scheme based on imaginary sliding window and arithmetic coding be applied for encoding symbols generated by the source. Then the memory size of the encoder (decoder)* S *and encoding (decoding) time* T *are given by the estimates*

$$S < \text{const} \quad n \log \frac{n}{r} \quad ; \tag{13}$$

$$T < \text{const} \quad \log \frac{n}{r} \quad \log n + \log \log \frac{n}{r} \log \log \log \frac{n}{r} \quad : \tag{14}$$

So the time complexity of adaptive coding using imaginary sliding window is essentially the same as in case of real sliding window, while the space complexity is less in order of magnitude because the window is not stored (we eliminate the term $(n/r) \log n$ in (2)).

Let us give some remarks, without any strict considerations, about the ways of obtaining random bits $z_1 z_2 \ldots z_k \ldots$. To maintain the imaginary sliding window these bits are to be known to both the encoder and decoder. A usual way to solve the problem is to use synchronized generators of pseudorandom numbers. We suggest an additional way, namely, using the bits of the code sequence (maybe, with some simple randomizing permutations). The ground for this proposal is that the compressed data are almost random, more exactly, the code sequence approaches a sequence of uniformly distributed code letters as the coding redundancy decreases. The encoder and decoder may use a part of already encoded message for producing random numbers. It is important that the code sequence is known to both the encoder and decoder and they need to store only a small current part of it. This way of obtaining random numbers may occur to be easier than using separate generators.

References

1. Jelinek, F.: Probabilistic Information Theory. New York: McGraw-Hill (1968) 476{ 489
2. Rissanen, J. J.: Generalized Kraft inequality and arithmetic coding. IBM J. Res. Dev. 20 (1976) 198{203
3. Pasco, R.: Source coding algorithm for fast data compression. Ph. D. thesis, Dept. Elect. Eng., Stanford Univ., Stanford, CA (1976)
4. Rubin, F.: Arithmetic stream coding using xed precision registers. IEEE Trans. Inform. Theory 25, 6 (1979) 672{675
5. Rissanen, J. J., Langdon, G. G.: Arithmetic coding. IBM J. Res. Dev. 23, 2 (1979) 149{162
6. Guazzo, M.: A general minimum-redundancy source-coding algorithm. IEEE Trans. Inform. Theory 26, 1 (1980) 15{25

7. Witten, I. H., Neal, R., Cleary, J. G.: Arithmetic coding for data compression. Comm. ACM 30, 6 (1987) 520{540
8. Ryabko, B. Y., Fionov, A. N.: Homophonic coding with logarithmic memory size. Algorithms and Computation. Berlin: Springer (1997) 253{262 (Lecture notes in comput. sci.: Vol. 1350)
9. Rissanen J., Langdon G. G.: Universal modeling and coding. IEEE Trans. Inform. Theory 27, 1 (1981) 12{23
10. Cleary, J. G., Witten, I. H.: Data compression using adaptive coding and partial string matching. IEEE Trans. Commun. 32, 4 (1984) 396{402
11. Mo at, A.: A note on the PPM data compression algorithm. Res. Rep. 88/7, Dep. Comput. Sci., Univ. of Melbourne, Australia, 1988.
12. Willems, F. M. J., Shtarkov, Y. M., Tjalkens, T. J.: The context-tree weighting method: Basic properties. IEEE Trans. Inform. Theory 41, 3 (1995) 653{664
13. Ryabko, B. Y.: The imaginary sliding window. IEEE Int. Symp. on Information Theory. Ulm (1997) 63
14. Krichevsky, R.: Universal Compression and Retrieval. Dordrecht: Kluwer Academic Publishers (1994)
15. Feller, W.: An Introduction to Probability Theory and Its Applications. New York: Wiley & Sons (1970)

Robust Protocol for Generating Shared RSA Parameters

Ari Moesriami Barmawi, Shingo Takada, and Norihisa Doi

Department of Computer Science,
Graduate School of Science and Technology, Keio University,
3-14-1, Hiyoshi, Yokohama 223, Japan
{ari, michigan, doi}@doi.cs.keio.ac.jp

Abstract. This paper describes how n parties can jointly generate the parameters for the RSA encryption system while being robust to prevent attacks from cheaters and malicious parties. The proposed protocol generates a public modulus number, without the parties knowing the factorization of that number. Our proposed protocol is similar to that of Boneh-Franklin's protocol. However, when there are two communicating parties our proposed protocol does not need the help of a third party. By using our proposed protocol, we can detect the presence of malicious parties and cheaters among the authorized user. An analysis shows that our proposed protocol has less computational complexity than the protocol of Frankel-MacKenzie-Yung.

1 Introduction

There are several cryptographic protocols that require an RSA modulus number for which none of the communicating parties know the factorization (such as [6],[7],[8],[9]). Therefore, it becomes necessary for the parties to jointly generate the RSA parameters (i.e. modulus number N, public key and secret key). Boneh and Franklin have proposed a protocol for generating shared RSA parameters [1]. However, their protocol is weak against malicious parties (i.e., attackers who can view the servers' memory at any moment, hear the messages which are broadcast and inject his own messages [2]) and cheaters (i.e., authorized parties who cheat during the protocol, e.g., cheating which cause a non-RSA modulus to be incorrectly accepted and resulting in the factor of the modulus number being found [4].)

To overcome this problem, many robust protocols for generating RSA parameters have been proposed such as those of Frankel-MacKenzie-Yung [2] and Malkin-Wu-Boneh [4]. These protocols have a quite high computational and communication complexity.

We propose a protocol for generating shared RSA parameters among n parties. Our protocol is similar to that of Boneh-Franklin's but when there are two communicating parties, it does not require any help of a third party. Our protocol also takes into account robustness against malicious parties and cheaters, with computational complexity lower than Frankel-MacKenzie-Yung's.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 280{289, 1999.

Section 2 describes, our proposed protocol. Section 3 then gives an analysis in terms of security. Section 4 compares our approach with previous ones and section 5 makes concluding remarks.

2 The Proposed Protocol

This section gives an overview of our protocol and the details of each procedure, i.e., the generation of a modulus number, the generation of the shared keys and the recovery of the encrypted message.

2.1 Overview

For simplicity, we will call party i as U_i. Suppose that all parties U_i (for $i = 1; :::; n$) wish to generate shared RSA parameters. After the execution of our proposed protocol, the RSA modulus number N and the encryption key e are publicly known, while the decryption key d will be shared among the parties in a way which enables threshold decryption. All parties should be convinced that N is indeed a product of two prime numbers, but neither party knows the factorization of N.

Our protocol is based on the procedure proposed by Boneh and Franklin. Besides, it is also similar to Cocks' [5], but

we have extended Cock's algorithm to improve computational e ciency and generalized it to handle more than two parties without weakening the protocol's security.

Our proposed protocol consists of the following three procedures:

1. The generation of RSA modulus number N (where N is a product of two prime numbers p and q). No party knows the factorization of N.
2. The generation of shared decryption key d for a given encryption key e.
3. The recovery of an encrypted message.

2.2 The Generation of Modulus Number N

Our proposed protocol determines N based on a set of secret numbers that all parties have. Each party U_i has two secret numbers p_i and q_i and the two prime numbers are de ned as $p = (p_1 + p_2 + ::: + p_n)$ and $q = (q_1 + q_2 + ::: + q_n)$.

The procedure for generating the modulus number consists of two sub-procedures:

1. Generation of N.
2. Primality testing of N.

The Generation of N is performed based on Protocol IA (Figure 1) as follows:

1. All parties have to agree on the length of modulus number N in advance.
2. Each party Ui chooses three large prime numbers X_i, Y_i and $_i$ (where the size of X_iY_i is a few digits greater than the size of N and the size of $_i$ is greater than a half of the size of N), a number n_i which is greater than 2, an odd number A_i where $gcd(A_i; X_iY_i) = 1$, and U_i's secret numbers p_i and q_i. We assume that p_i and q_i must be congruent to 3 mod 4. Furthermore, each party U_i :

- { First, calculates $a_{i(1)} = (A_i)(p_i)^{1-n_i}(q_i)^{-n_i} \bmod X_iY_i$ and then $F_{i(1)}$ ($_i$)$a_{i(1)} \bmod X_iY_i$ $_i$
- { Calculates $a_{i(2)} = (A_i)(q_i)^{1-n_i}(p_i)^{-n_i} \bmod X_iY_i$ and then $F_{i(2)}$ ($_i$)$a_{i(2)} \bmod X_iY_i$ $_i$
- { Calculates $a_{i(3)} = (A_i)(p_iq_i)^{-n_i} \bmod X_iY_i$ and then $F_{i(3)}$ ($_i a_{i(3)} \bmod X_iY_i$ $_i$

Then, U_i calculates $gcd(F_{i(1)}; X_iY_i$ $_i)$, $gcd(F_{i(2)}; X_iY_i$ $_i)$ and $gcd(F_{i(3)}; X_iY_i$ $_i)$. If those values have at least two large prime factors then U_i broadcasts $F_{i(1)}; F_{i(2)}$ and $F_{i(3)}$ along with X_iY_i $_i$. Otherwise U_i has to choose another A_i and repeat step 2.

3. Each party U_{i+1} sends U_i

$$Z_{i(1)} \quad (F_{i(1)}q_{i+1} + F_{i(2)}p_{i+1} + F_{i(3)}(p_{i+1}q_{i+1})) \bmod X_iY_i \; _i$$

4. Each party U_j broadcasts $Z_{i;j(2)} = F_{i(1)}q_j + F_{i(2)}p_j \bmod X_iY_i$ $_i$ for $i = 1; 2; :::; n$ and $i \neq j - 1; j$ (e.g. suppose $j = 5$, $n = 5$, then U_5 has to broadcast $Z_{1;5(2)}, Z_{2;5(2)}, Z_{3;5(2)}$).
5. Each party U_j sends U_i $Z_{i;j(3)}$ where

$$Z_{i;j(3)} = F_{i(3)}[(p_jq_{j+1} + p_{j+1}q_j + p_{j+1}q_{j+1}) + \sum_{\substack{k=1;\\ k\neq i;i+1;:::;j;j+1}}^{k\neq n} (p_jq_k + p_kq_j)]$$

$$\bmod X_iY_i \; _i$$

for $i = 1; 2; :::; n; i \neq j; j + 1$ (e.g. if $j = 2$, $n = 5$, then $i = 1; 4; 5$. This means that U_2 has to send $Z_{1;2(3)}$ to U_1, $Z_{4;2(3)}$ to U_4, $Z_{5;2(3)}$ to U_5).

6. Each party U_i can calculate the RSA modulus number N using the following equation:

$$N \quad (A_i)^{-1}(p_i)^{n_i}q_i^{n_i} \bmod X_iY_i[[Z_{i(1)} + \sum_{\substack{j=1;\\ j\neq i;i+1}}^{j\neq n} Z_{i;j(2)} + \sum_{\substack{j=1;\\ j\neq i-1;i}}^{j\neq n} Z_{i;j(3)}] = \; _i]$$

$$\bmod ((X_iY_i \; _i) = \; _i) + p_iq_i \bmod X_iY_i \qquad (1)$$

The example for calculating N will be described in Appendix A.

7. Finally each party U_i calculates $H(N)$ (where H is any hash function that has been agreed upon by all parties in advance) and then broadcasts it.
8. Each party then compares the $H(N)$ sent by other parties and the $H(N)$ that he just calculated. If all values are equivalent, then all parties will agree on N as their RSA modulus number. Otherwise, they will repeat the procedure.

After generating the modulus number N, all parties have to test whether N is a product of two primes p and q and to prove that these primes are those used in the primality test to detect the presence of a cheater and/or a malicious party. We use the primality test proposed by Boneh and Franklin [1] with additional steps for detecting cheaters and malicious parties. The procedure is as follows:

1. All parties agree on a number g, where the Jacobi symbol of g over N or $g=N$ must be 1, i.e., $(g=N) = 1$.

Message 1. $U_i \rightarrow *$: $X_i Y_i \; _i; F_{i(1)}; F_{i(2)}; F_{i(3)}$

Message 2. $U_{i+1} \rightarrow U_i$: $Z_{i(1)}$

Message 3. $U_j \rightarrow *$: $Z_{i;j(2)}$

Message 4. $U_j \rightarrow U_i$: $Z_{i;j(3)}$

Message 5. $U_i \rightarrow *$: $H(N)$

Fig. 1. Protocol IA

2. Each party U_i calculates
   { $t_{i(1)} \quad g^{(N-p_i-q_i+1)/4} \bmod N$,
   { $t_{i(2)} \quad g^{(p_i+q_i)/4} \bmod N$,
   and exchanges these results.
   Each party U_i can verify whether a party is malicious using the following procedure:
   { Each party U_i chooses any integer $\;_i$ and then calculates

$$_{i;j} = g^{(F_{j(1)}+F_{j(2)}) \; _i} \bmod X_i Y_i \; _i \bmod N \tag{2}$$

   where the value of $F_{j(1)}$ and $F_{j(2)}$ is shown in step 2 of Protocol IA. Then, he broadcasts $_{i;j}$ for $j = 1; 2; :::; n; j \neq i$.
   { Each user U_j calculates

$$_{i;j} = (\;_{i;j})^{(A_j)^{-1}(p_j)^{n_j}(q_j)^{n_j}} \bmod X_i Y_i \; _i \bmod N \tag{3}$$

   and

$$_{i;j} = (t_{j(2)})^{4 \; _j} \bmod X_i y_i \; _i \bmod N \tag{4}$$

   and broadcasts $_{i;j}$ for $i = 1; 2; :::; n; i \neq j$, along with $_{i;j}$.
   { Finally, each user U_i can verify whether U_j is malicious or not using the following expression:

$$_{i;j} = (\;_{i;j}) \; _i \bmod X_i Y_i \; _i \bmod N \tag{5}$$

   If this expression is NOT TRUE then U_j is a malicious party or a cheater. If there is a malicious party or a cheater among the parties, then the protocol for modulus number generation should be repeated without including the malicious party or the cheater.
3. Furthermore, they verify that N is a product of two primes using the following equation:

$$t_{i(1)} = [\prod_{\substack{j=1;\\ j\neq i}}^{j\forall n} t_{j(2)}] \bmod N \qquad (6)$$

Each party may execute equation (6) for $i = 1; 2; :::; n$.

4. If we consider $N = pq$ where $p = (r_1)^{b_1}$, $q = (r_2)^{b_2}$ and $q \quad 1 \bmod (r_1)^{(b_1-1)}$, then the above two steps will pass *incorrectly*. We thus have to check whether N satis es $gcd(N; p + q - 1) > 1$.

For calculating $gcd(N; p + q - 1)$, we will use protocol IB[1] (see gure 2). Each party U_i rst chooses his own number f_i. Then they execute Protocol IB.

Message 1. $U_i \rightarrow * : G_{i(1)}; G_{i(2)}; G_{i(3)}$

Message 2. $U_{i+1} \rightarrow U_i : V_{i(1)}$

Message 3. $U_j \rightarrow * : V_{i;j(2)}$

Message 4. $U_j \rightarrow U_i : V_{i;j(3)}$

Message 5. $U_i \rightarrow * : H(w)$

Fig. 2. Protocol IB

The de nition of all variables used in Protocol IB is given below:

- $c_i = p_i + q_i$ (except for $i = 1$, $c_i = p_i + q_i - 1$)
- $G_{i(1)} = (\ _i)(A_i)(f_i)^{1-n_i}(c_i)^{-n_i} \bmod X_i Y_{i\ i}$
- $G_{i(2)} = (\ _i)(A_i)(p_i + q_i)^{1-n_i}(f_i)^{-n_i} \bmod X_i Y_{i\ i}$
- $G_{i(3)} = (\ _i)(A_i)(f_i c_i)^{-n_i} \bmod X_i Y_{i\ i}$
- $V_{i(1)} = (G_{i(1)} c_{i+1} + G_{i(2)} f_{i+1} + G_{i(3)}(f_{i+1} c_{i+1})) \bmod X_i Y_{i\ i}$
- $V_{i;j(2)} = G_{i(1)} c_{i+1} + G_{i(2)} f_{i+1} \bmod X_i Y_{i\ i}$
- $V_{i;j(3)} = G_{i(3)}[(f_j c_{j+1} + p_{j+1} c_j + f_{j+1} c_{j+1}) + \sum_{\substack{k=1;\\ k\neq i;i+1;:::;j+1}}^{k=i-1} (f_j c_k + f_k c_j)\ ] \bmod (X_i Y_{i\ i})$
- for A_i, Y_i, X_i, n_i and $_i$ all parties can use the values used in Protocol IA with $H(w)$ is the hash of w.

The execution of Protocol IB results in all parties jointly calculating

$$w = (A_i)^{-1}(f_i)^{n_i} c_i^{n_i} \bmod X_i Y_i [[V_{i(1)} + \prod_{\substack{j=1;\\ j\neq i;i+1}}^{j\forall n} V_{i;j(2)} + \prod_{\substack{j=1;\\ j\neq i-1;i}}^{j\forall n} V_{i;j(3)}] = _i]$$

$$\bmod (X_i Y_{i\ i}) = _i + c_i f_i \bmod X_i Y_i$$

[1] Details of Protocol IB is similar to Protocol IA, and will be omitted due to space.

Then calculating $s = w \bmod N$. According to Boneh-Franklin [1], $gcd(w; N) = gcd(s; N)$. Thus, if all parties can verify that $gcd(s; N) > 1$, they will reject this number.

Unfortunately, this test will also eliminate a few valid numbers i.e. moduli $N = pq$, for $q = 1 \bmod p$. We do not describe the protocol for testing whether N is a valid number , but it is similar with the test for two parties which was described in [12].

2.3 The Generation of Shared Public/Secret Keys

We now describe the procedure for generating public/secret keys. Suppose all parties have successfully calculated N. Then, they will jointly generate shared decryption key d, where $d = \sum_{i=1}^{i=n} d_i$ and $d = e^{-1} \bmod \varphi(N)$ for some agreed upon value of e. Each party U_i will have its own decryption exponent d_i. For generating the keys, we will use the method proposed by Boneh and Franklin [1] but without the help of a third party.

The procedure for generating shared keys are as follows:

- Each party U_i broadcasts $(p_i + q_i) \bmod e$.
- Each party calculates $-\varphi(N) \bmod e$ which is congruent to $[(\sum_{i=1}^{i=n} p_i) + (\sum_{i=1}^{i=n} q_i) - N - 1] \bmod e$.
- Since $\varphi(N)$ and e are relatively prime, then all parties can calculate $\gamma = -(\varphi(N))^{-1} \bmod e$.
- Thus, $de = 1 + (\varphi(N))(\gamma)$ and

$$d = \frac{1 + (N + 1 - (\sum_{i=1}^{i=n}(p_i + (q_i))))(\gamma)}{e} \tag{7}$$

Each party's decryption key d_i, is calculated according to the following procedure:

- Let $\gamma(N + 1) - n\gamma(p_i + q_i) \bmod ne \equiv M_i \bmod ne$. Each party U_i (except for $i \neq 1$) broadcasts M_i. U_1 broadcasts $M_1 = n + \gamma(N + 1) - n\gamma(p_1 + q_1) \bmod ne$
- Each party U_i calculates

$$B = n + \sum_{i=1}^{i \neq n} (\gamma(N + 1) - n\gamma(p_i + q_i)) \bmod ne = \sum_{i=1}^{i \neq n} M_i \bmod ne$$

- If $B \neq 0$ then execute the following procedure 1:
  1. U_1 calculates $d_1 = b\frac{n + (N+1)\gamma - n\gamma(p_1+q_1)}{ne}c$.
  2. For $i \neq n - B$, U_i calculates $d_i = b\frac{(N+1)\gamma - n\gamma(p_i+q_i)}{ne}c$
  3. For $i > n - B$, U_i calculates $d_i = d\frac{(N+1)\gamma - n\gamma(p_i+q_i)}{ne}e$
- If $B = 0$ then
  1. U_1 executes step 1 of procedure 1.
  2. For $i \neq bn/2c$, execute step 2 of procedure 1.
  3. For $i > dn/2e$, execute step 3 of procedure 1.

For verifying that the key generation is done successfully, all parties have to agree on a message MSG in advance, then each party has to sign this message using his own decryption key and broadcast the signed message. Furthermore, each party multiplies all signed messages and decrypts it with the public key. th

3 Security Analysis

In this section we will analyze the ability to calculate the factors of modulus number N, and also summarize the security requirements.

3.1 Ability for Finding Factors of N

Since the strength of our proposed protocol is on breaking the modulus number N, we will analyze how far the messages which a party obtained will leak information for obtaining other party's secret number p_i and q_i.

First, we will analyze each message of the protocol for generating N (Protocol IA). Assume that each party only saves $X_i Y_{i\ i}$, while X_i, Y_i, $_i$, A_i and n_i have to be destroyed or changed after a round of executing Protocol IA. Message 1 contains numbers which are functions of n_i, p_i, q_i, A_i, X_i, Y_i and $_i$. There are three ways to obtain the values of p_i and q_i by other parties:

- { Other parties can obtain p_i and q_i i they can nd the inverse of the function $F_{i(3)}$ since the product of $((F_{i(3)})^{-1} \bmod X_i Y_{i\ i})$ and $(F_{i(1)} \bmod X_i Y_{i\ i})$ is $(p_i \bmod X_i Y_{i\ i})$ and the product of $((F_{i(3)})^{-1} \bmod X_i Y_{i\ i})$ and $(F_{i(2)} \bmod X_i Y_{i\ i})$ is $q_i \bmod X_i Y_{i\ i}$. There is no possibility of calculating the multiplicative inverse of $F_{i(3)}$ since $X_i Y_{i\ i}$ and $F_{i(3)}$ are not relatively prime.
- { Other parties can obtain p_i and q_i by nding the factors of $X_i Y_{i\ i}$ because if they knew $_i$, then they can nd $(F_{i(1)} = {}_i \bmod X_i Y_i)$, $(F_{i(2)} = {}_i \bmod X_i Y_i)$ and $(F_{i(3)} = {}_i \bmod X_i Y_i)$. This means that they can nd the multiplicative inverse of $(F_{i(3)} = {}_i \bmod X_i Y_i)$. Thus, they can obtain p_i and q_i by multiplying $(F_{i(1)} = {}_i \bmod X_i Y_i)$ and $(F_{i(2)} = {}_i \bmod X_i Y_i)$ with the multiplicative inverse of $(F_{i(3)} = {}_i \bmod X_i Y_i)$. As described above, only $X_i Y_{i\ i}$ is saved by U_i. Thus, the possibility of obtaining p_i and q_i is equal to the possibility of factoring $X_i Y_{i\ i}$.
- { p_i and q_i can also be obtained by calculating $gcd(F_{i(1)}; X_i Y_{i\ i})$, $gcd(F_{i(2)}; X_i Y_{i\ i})$ and $gcd(F_{i(3)}; X_i Y_{i\ i})$. From these processes they can obtain $_i$ and furthermore obtain p_i, q_i. But, since $F_{i(1)}$,$F_{i(2)}$ and $F_{i(3)}$ have at least two large prime factors, it is still hard to obtain $_i$.

Since messages 2, 3 and 4 contain functions which have more than three unknown variables, then there are no possibility of obtaining p_i and q_i from these messages.

Even if a malicious party or a cheater who can corrupt U_i is still di cult for them to pass our proposed protocol correctly, since to pass the protocol they have to nd the factors of $X_i Y_{i\ i}$ which is not saved in the memory of U_i.

3.2 Security Requirement

Based on the above discussion, there are a few conditions that need to be satis ed to keep the factors of modulus number N secret. These conditions can be summarized as follows:

- All parties have to determine the length of modulus number N in advance.
- Each party U_i has to choose the length of $X_i Y_i$ to be a few digits longer than the length of N.
- Each party U_i has to choose the length of p_i and q_i such that the length of $p_i q_i$ is about the length of N.

4 Comparison with Other Protocol

Protocols have been proposed for jointly generating RSA parameters, such as Boneh-Franklin and Frankel-MacKenzie-Yung for n parties and Cocks and Poupard-Stern [3] for two parties.

Since our proposed protocol is for n parties, then we will compare it with the protocol of Boneh-Franklin [1] and Frankel-MacKenzie-Yung [2].

The bene t of our proposed protocol compared with Boneh-Franklin is that when there are two communicating parties, our protocol does not need any third party for calculating the modulus number N as well as the keys.

The probability of generating an RSA modulus number from two random primes of m bits each is about $(m)^{-2}$, which means that we will have about m^2 rounds. Each round will have computational complexity about $12n$ modular exponentiations and communication complexity is about $O(10n - 3)$. Thus, the computational complexity is about less than a third compared with the protocol of Frankel-MacKenzie-Yung (which is $24n(t + 1)$). Another bene t of our proposed protocol is that the communication complexity does not depend on the size of N.

5 Conclusion

We have proposed a protocol for jointly generating parameters in RSA encryption. When there are two communicating parties, our protocol does not need the help of a third party, and it has less computational complexity compared with previous protocols. The advantage of our proposed protocol is that the communication complexity does not depend on the size of N.

References

1. Boneh, D. and Franklin, M.: E cient Generation of Shared RSA Keys. Advances in Cryptology-Crypto '97. Lecture Notes in Computer Science. Springer Verlag (1997), 423-439.
2. Frankel, Y., MacKenzie,P. D.,Yung, M.: Robust E cient Distributed RSA-Key Generation. Proceedings STOC 98. ACM (1998).
3. Poupard, G. and Stern, J.: Generation of Shared RSA Keys by Two Parties. Proceedings of ASIACRYPT 98. Lecture Notes in Computer Science. Springer Verlag (1998), 11-24.
4. Malkin, M., Wu, T. and Boneh, D.: Experimenting with Shared Generation of RSA Keys. Proceedings of Internet Society's 1999 Symposium on Network and Distributed Network Security, (1999).

5. Cocks, C.: Split Knowledge Generation of RSA Parameters. Proceedings of 6th International Conference of IMA on Cryptography and Coding. Lecture Notes in Computer Science. Springer Verlag (1997), 89-95.
6. Feige, U., Fiat, A. and Shamir, A.: Zero-knowledge Proofs of Identity. Journal of Cryptology, 1, (1988), 77-94.
7. Fiat, A. and Shamir, A.: How to Prove Yourself: Practical Solution to Identi cation Problems. Crypto '86. Lecture Notes in Computer Science (1986), 186-194.
8. Ohta, K. and Okamoto, T.: A modi cation of Fiat-Shamir scheme. Crypto '88. Lecture Notes in Computer Science. Springer Verlag (1988), 232-243.
9. Ong, H. and Schnorr, C.: Fast Signature Generation with a Fiat-Shamir-like Scheme. Eurocrypt '90. Lecture Notes of Computer Science. Springer Verlag (1990), 432-440.
10. Rivest, R. L., Shamir, A. and Adleman, L.: Method for Obtaining Signatures and Public-Key Cryptosystems. Communication of the ACM (1978).
11. Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley and Sons (1994).
12. Barmawi, A. M., Takada, S., Doi, N.: A Proposal for Generating Shared RSA Parameters. Proceeding of IWSEC 99, September, 1999.

A Calculating N by Each User

Here we will simulate how a party U_i can calculate the modulus number N using all information that it has obtained. Suppose $i = 1$, $n = 5$, then U_1 can calculate N using equation (1) as follows:

$$N \quad (A_1)^{-1}(p_1)^{n_1}q_1^{n_1} \bmod X_1Y_1[[Z_{1(1)} + \sum_{\substack{j=1;\\ j\neq 1;2}}^{j\neq 5} Z_{1;j(2)} + \sum_{\substack{j=1;\\ j\neq 1;5}}^{j\neq 5} Z_{1;j(3)}] = {}_1 \bmod ((X_1Y_{1\ 1}) = {}_1) + p_1q_1 \bmod X_1Y_1 \tag{8}$$

Let $N_1 = (A_1)^{-1}(p_1)^{n_1}q_1^{n_1}Z_{1(1)} = {}_1$, $N_2 = (A_1)^{-1}(p_1)^{n_1}q_1^{n_1}(\sum_{j=1+2}^{j=5} Z_{1;j(2)}) = {}_i$, and $N_3 = (A_1)^{-1}(p_1)^{n_1}q_1^{n_1}(\sum_{j=2}^{j=4} Z_{1;j(3)}) = {}_1$.

Then, equation 8 can be written as:

$$N = N_1 + N_2 + N_3 \bmod X_1Y_1 \tag{9}$$

Next, we will calculate N_1, N_2 and N_3 in detail.

$$\begin{aligned} N_1 &= (A_1)^{-1}(p_1)^{n_1}(q_1)^{n_1}Z_{1(1)} = {}_i \\ &= (A_1)^{-1}(p_1)^{n_1}(q_1)^{n_1}[A_1(p_1)^{1-n_1}(q_1)^{-n_1}q_2 + A_1(p_1)^{-n_1}(q_1)^{1-n_1}p_2 + \\ &\quad A_1(p_1)^{-n_1}(q_1)^{-n_1}p_2q_2] \\ &= p_1q_2 + p_2q_1 + p_2q_2 \end{aligned} \tag{10}$$

$$N_2 = (A_1)^{-1}(p_1)^{n_1}(q_1)^{n_1}[\sum_{\substack{j=1;\\ j\neq 1;2}}^{j\neq 5} Z_{1;j(2)}] \bmod i$$
$$= (A_1)^{-1}(p_1)^{n_1}(q_1)^{n_1}[\sum_{j=3}^{j\neq 5} Z_{1;j(2)}] \bmod i$$
$$= (A_1)^{-1}(p_1)^{n_1}(q_1)^{n_1}[Z_{1;3(2)} + Z_{1;4(2)} + Z_{1;5(2)}] \bmod i$$
$$= p_1q_3 + p_3q_1 + p_1q_4 + q_1p_4 + p_1q_5 + p_5q_1 \quad (11)$$

$$N_3 = (A_1)^{-1}(p_1)^{n_1}(q_1)^{n_1}[\sum_{\substack{j=1;\\ j\neq 1;5}}^{j\neq 5} Z_{1;j(3)}] \bmod i$$
$$= (A_1)^{-1}(p_1)^{n_1}(q_1)^{n_1}[\sum_{j=2}^{j\neq 4} Z_{1;j(3)}] \bmod i$$
$$= (A_1)^{-1}(p_1)^{n_1}(q_1)^{n_1}[A_1p_1^{-n_1}q_1^{-n_1}][fp_2q_3 + p_3q_2 + p_3q_3 + \sum_{\substack{k=1;\\ k\neq 1;2;3}}^{k\neq 5} p_2q_k + p_kq_2$$
$$+ fp_3q_4 + p_4q_3 + p_4q_4 + \sum_{\substack{k=1;\\ k\neq 1;2;3;4}}^{k\neq 5} p_3q_k + p_kq_3g$$
$$+ fp_4q_5 + p_5q_4 + p_5q_5 + \sum_{\substack{k=1;\\ k\neq 1;2;3;4;5}}^{k\neq 5} p_4q_k + p_kq_4g$$
$$= p_2q_3 + p_3q_2 + p_3q_3 + p_2q_4 + p_4q_2 + p_2q_5 + p_5q_2 + p_3q_4 + p_4q_3 + p_4q_4$$
$$+ p_3q_5 + p_5q_3 + p_4q_5 + p_5q_4 + p_5q_5 \quad (12)$$

Thus, by using equations (10), (11), and (12), we can obtain:

$$N = N_1 + N_2 + N_3 \bmod X_1Y_1$$
$$= [p_1q_2 + p_2q_1 + p_2q_2 + p_1q_3 + p_3q_1 + p_1q_4 + q_1p_4 + p_1q_5 + p_5q_1 + p_2q_3$$
$$+ p_3q_2 + p_3q_3 + p_2q_4 + p_4q_2 + p_2q_5 + p_5q_2 + p_3q_4 + p_4q_3 + p_4q_4 + p_3q_5$$
$$+ p_5q_3 + p_4q_5 + p_5q_4 + p_5q_5 + p_1q_1] \bmod X_1Y_1$$
$$= (p_1 + p_2 + p_3 + p_4 + p_5)(q_1 + q_2 + q_3 + q_4 + q_5) \bmod X_1Y_1 \quad (13)$$

Some Soft-Decision Decoding Algorithms for Reed-Solomon Codes

Stephan Wesemeyer*, Peter Sweeney, and David R.B. Burgess

Centre for Comm. Systems Research, University of Surrey, Guildford GU2 5XH, U.K.

Abstract. In this paper we introduce three soft-decision decoding algorithms for Reed-Solomon (RS) codes. We compare them in terms of performance over both the AWGN and Rayleigh Fading Channels and in terms of complexity with a special emphasis on RS codes over $\mathbb{F}_{16}$. The algorithms discussed are variants of well known algorithms for binary codes adapted to the multilevel nature of RS codes. All involve a re-ordering of the received symbols according to some reliability measure. The choice of reliability measure for our simulations is based on a comparison of three in terms of how they affect the codes' performances.

1 Introduction

It is well known that one way of facilitating soft-decision decoding for linear block codes is to represent them by a trellis and apply the Viterbi algorithm (VA) to decode them. However, the complexity of the VA makes its use infeasible for all but a small number of linear codes. Because of the widespread use of RS codes, it would be highly desirable to find efficient soft-decision algorithms for them. Various approaches have been proposed (see [1] for a recent example). This paper introduces a further three. Our simulations were based around an AWGN and a Rayleigh fading channel with BPSK (binary-phase-shift-keyed) modulation and 8-level uniform quantisation. Except in a very few cases with extremely long simulation runs, we based the results on 100 error events (word errors, not bit errors). Throughout the paper we denote by $\mathbb{F}_q$, a finite field of $q = 2^l$ elements and assume an $[n; k]$ linear code over $\mathbb{F}_q$ which can correct t errors.

2 The Algorithms

The Dorsch algorithm was proposed in [2] for binary codes and has more recently been applied by Fossorier and Lin [3]. Given a code of length n and dimension k the idea is to find k most reliable symbols whose positions are such that they can be used as an information set of the code. Various error patterns are added to this information set and each result is re-encoded. In each case, the distance of the obtained codeword from the received word is computed. Decoding stops as soon as we have a maximum-likelihood solution or the number of permitted decoding

* The research was supported by an EPSRC grant.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 290–299, 1999.

tries has been exhausted (in which case the best solution up to that point is output). Our first two algorithms ($A1$ and $A2$) are based on this technique.

The codeword closest to the received word in terms of the following metric is the maximum-likelihood solution we want to find.

Definition 1. *Let $s'_i = (s'_{i1}, \ldots, s'_{il}) \in \mathbb{F}_q$ be the symbol obtained by using hard decision ($s'_{ij} \in \{0, 7\}$) on $r_i = (r_{i1}, \ldots, r_{il})$, the ith received symbol after quantisation. The distance between a received word $r = (r_1, r_2, \ldots, r_n)$ and a word $c = (c_1, \ldots, c_n) \in \mathbb{F}_q^n$, with $c_i = (c_{i1}, \ldots, c_{il}) \in \mathbb{F}_q$, is defined as*

$$\mathrm{dist}(r, c) = \sum_{i=1}^{n} (\mathrm{dist}_{sym}(r_i, c_i)) \text{ where } \mathrm{dist}_{sym}(r_i, c_i) = \sum_{j=1}^{l} |r_{ij} - c_{ij}| - \sum_{j=1}^{l} |r_{ij} - s'_{ij}|.$$

Furthermore, all algorithms produce a continuous stream of possible solutions which are subjected to a stopping criterion that, if satisfied, is sufficient (though not necessary) to guarantee a maximum-likelihood solution [4], in which case the decoding stops. Since we are concerned here with RS codes, any k symbols may be used as an information set. Hence we simply sort the symbols according to reliability (see Chapter 3) and, in algorithm $A1$, we use the k most reliable as the information set. $A2$ repeats $A1$ using the k least reliable symbols unless a maximum likelihood solution has already been found by $A1$.

Fossorier and Lin's implementation of the Dorsch algorithm checks error patterns corresponding to i errors in the information set. This has been termed order-i reprocessing [3]. In our version, we take a slightly different approach which is closer to the original Dorsch algorithm. Our order for testing the error patterns to be added to the chosen information set is the proximity of the resulting sequence to the corresponding part of the received word. The index used is the generalisation of 'dist' to different length sequences which takes the sum of all the 'dist_{sym}'s over the symbols of the sequence. This is achieved by using a stack-type algorithm, whereby stacks of sequences of different lengths are kept in storage, ordered according to the index. A sequence from the stack of lowest index is extended in q different ways by appending a symbol, the indices of the resulting sequences are calculated and they are each put in the appropriate stack. The memory requirement of this implementation is determined by the maximum number of decoding tries. Let MDT be this maximum and DT be the number of decoding tries so far. Then we only need to keep $MDT - DT$ information sets of smallest index in our array as none of the others will be used.

Our third algorithm ($A3$) simply applies $A1$ and, if that algorithm does not produce a maximum-likelihood solution, then a Chase-style algorithm is applied to the sorted word, i.e. we apply a fixed number of error patterns of least distance to the least reliable symbols and then use an algebraic decoder to decode. This approach has already been applied successfully to binary codes by Fossorier and Lin [5].

3 Sorting

All the algorithms depend on sorting the received symbols according to some reliability measure. In the case of binary codes, Fossorier and Lin showed that on an AWGN and on a Rayleigh fading channel with BPSK modulation, the absolute value of the received symbol is the most appropriate choice [3]. The higher that value is, the more reliable hard decision on the received symbol will be. With RS or indeed any code whose symbols come from a non-binary nite eld we need to nd a slightly di erent approach. In such a case, each 'received symbol' will, in fact, be a string of symbols which, between them, indicate the binary representation of the 'received symbol'.

De nition 2. *Let* $r = (r_1, \ldots, r_l)$ *with* $0 \quad r_i \quad 7$ *be a received symbol after quantisation and de ne*

$$\mathrm{Rel}_1(r) = \sum_{i=1}^{} |3.5 - r_i|; \quad \mathrm{Rel}_3(r) = \min\{|3.5 - r_i| \; 1 \quad i \quad l\}$$

$$\textit{and } \mathrm{Rel}_2(r) = \prod_{i=1}^{} P(hd(r_i)|r_i); \text{ where } hd(j) = \begin{array}{ccccc} 0 & : & 0 & j & 3 \\ 1 & : & 4 & j & 7 \end{array}$$

The natural generalisation of the reliability measure of the binary case is to add the absolute values of the symbols in the string, thus obtaining an overall reliability of the 'received symbol'. As we use 8-level uniform quantisation this translates into Rel_1 above. Another approach is to use Bayes' rule. One can easily determine the probability $P(j|0)$ (resp. $P(j|1)$) of a received bit being quantised to level j given that a 0 (resp. a 1) was transmitted. From that we can work out the probability $P(0|j)$ (resp. $P(1|j)$) that a 0 (resp. 1) was transmitted given that we are in level j. Hence we arrive at Rel_2. Lastly, the most basic approach simply takes the least reliable bit in a symbol and uses its value as the overall reliability of the symbol (Rel_3). These three reliability measures were felt to be the most natural ones. It can easily be seen that the higher the computed reliability of a symbol is, the more likely it is to be correct.

Figure 1 (respectively Figure 2) contains the results for a $[16, 8, 9]$ ($[16, 12, 5]$) extended RS code over the AWGN channel (see Section 6.2 for the Rayleigh channel results), decoded using algorithm $A1$ with a maximum number of decoding tries corresponding to the number of order-2 (order-2 and order-1) reprocessing attempts with and without sorting. Note that, to compute the probabilities accurately for Rel_2, we need to know at which signal-to-noise ratio (SNR) the bits were transmitted. As this information is not always available in practice, we computed the probabilities for a SNR of 1dB adjusted by the code rate, R say, i.e. SNR$= R \quad 10^{0.1}$ and used these values throughout. (The simulations showed that - if anything - this approach proved slightly better than using the exact values for the di erent SNRs.)

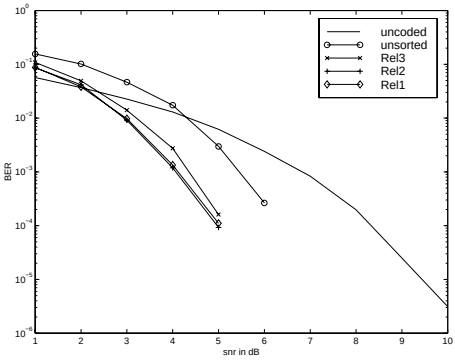


Fig. 1. Sorted vs unsorted [16; 8; 9] extended RS code - 529 decoding tries

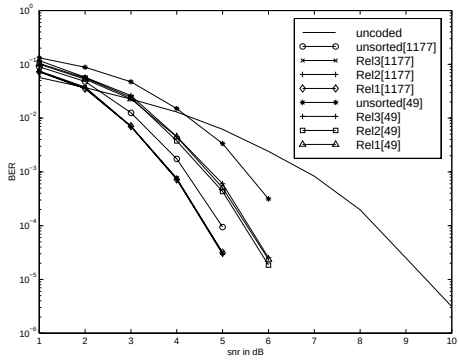


Fig. 2. Sorted vs unsorted [16; 12; 5] extended RS code - 1177 and 49 decoding tries

As can be seen all reliability measures result in a marked improvement over the unsorted case. The reason why Rel_3 is slightly worse than the other two (except in the case of 1177 decoding tries for the higher rate code) can be explained by the observation that the number of different reliability levels attached to each symbol in that method is rather low (4 compared to 26 for Rel_1 and 35 for Rel_2). At 1177 decoding tries, the algorithm performs close to maximum-likelihood decoding in any case - it is not important whether or not the least distorted symbols are used as an information set.

Because there was no significant difference between sorting the symbols of the received words according to Rel_1 or Rel_2 we used sorting by Rel_1 in all the remaining simulations.

4 Number of Decoding Tries

The most crucial feature of the proposed algorithms is the number of decoding tries they entail. The more decoding tries the more likely it is that we find the maximum-likelihood solution. However, as Fossorier and Lin [3] demonstrated the actual gain obtained from further decoding tries has to be measured against the extra computation involved.

Figure 3 is an example of how the maximum number of decoding tries (using algorithm $A1$) after sorting (with respect to Rel_1) can affect the performance of a code and how this performance compares to the unsorted case. Note that 41449, 5489, and 529 correspond to the number of decoding tries given by order-4, order-3, and order-2 reprocessing respectively. There is a marked improvement of about 1dB going from 529 decoding tries to 5489 but only a slight improvement of roughly 0.25dB when 41449 attempts are used instead of 5489 which does not justify the almost 8-fold increase in number of decoding tries. However, even then the complexity of the proposed algorithm is several orders of magnitude lower

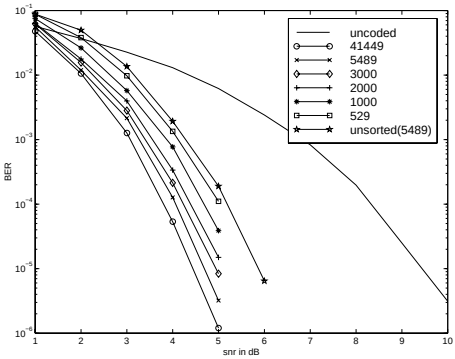


Fig. 3. [16; 8; 9] extended RS code - different numbers of permitted decoding tries

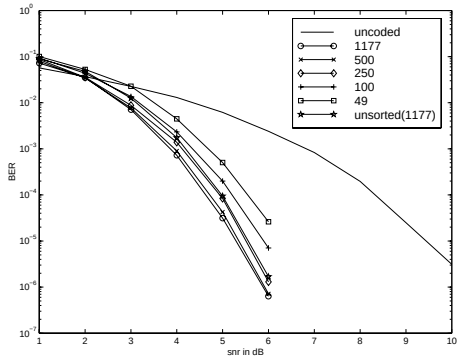


Fig. 4. [16; 12; 5] extended RS code - di erent numbers of permitted decoding tries

than that of the Viterbi algorithm which would have to deal with $16^8 \quad 4.3 \ 10^9$ states for this code.

Figure 4 shows the e ect of di erent numbers of permitted decoding tries (using algorithm $A1$) for a [16; 12; 5] extended RS code. This time we restricted the number of decoding tries to lie in between 49 and 1177(= number of decoding tries for order-1 and order-2 reprocessing respectively). Note that decoding after sorting with a maximum of 250 decoding tries slightly outperforms unsorted decoding with maximum 1177 decoding tries and there is only a very slight improvement going from 500 to 1177 decoding tries.

5 Measures of Complexity

The complexity of each algorithm is expressed in terms of additions, multiplications and comparisons which, for simplicity, are considered equivalent operations. All the estimates we give are based on our implementation; the idea is to give a rough idea of how much computational e ort has to be expended on decoding. To enable us to compare the results with other algorithms and to eliminate the code rate as a factor, for each code considered we measure the complexity in operations per information bit. We compare our results throughout with the Viterbi algorithm applied to a convolutional code of rate $R = 0.5$ and memory $k = 7$ even though the rate of the RS codes vary. Higher rate convolutional codes are usually obtained by puncturing which does not greatly a ect the number of operations which can be estimated at 128 comparisons (= number of states) plus 256 additions (= number of branches).

Our implementation of the $A1$ and $A2$ algorithms require, before the re-encoding starts, computing the metric and some values for the stopping criterion ($n(q-1)^2$ comparisons and n/q additions), sorting the symbols according to reliability (approximately $n \log_2(n)$ comparisons) and reducing a $(k; n)$ matrix to

reduced echelon form (REF) (nk^2 multiplications and $nk(k-1)$ additions). This latter is performed twice in $A2$ (two directions of decoding), so the preliminary operations for algorithm Ar ($r=1,2$) total:

$$PopAr = n(\log_2(n) + lq + (q-1)^2 + r(k^2 + k(k-1))): \quad (1)$$

For each decoding try (both algorithms), there are the following approximations: re-encoding ($(n-k)k$ multiplications and $(n-k)(k-1)$ additions), determining the distance from the received word ($(n-1)$ additions), determining whether the stopping criterion is satis ed ($(n+2+n\log_2(n)$ comparisons and $n-k+1$ additions), determining the best solution (1 comparison per decoding try after the rst). Thus altogether the algorithm Ar ($r=1,2$) requires the following total operations (where DT is the number of decoding tries).

$$TopAr = PopAr + DT(2(n-k)k + 2n + n\log_2(n) + 2) + (DT-1) \quad (2)$$

The estimates for our implementation of the Chase part of $A3$ are based on a very general algorithm presented in Stichtenoth [6] and due to A.N.Skorobogatov and S.G.Vladut. The following are the operations per decoding try: Computing the syndrome ($(n-k)n$ multiplications and $(n-k)(n-1)$ additions), checking whether the syndrome is 0 ($n-k$ comparisons), reducing the $(t; t+1)$ syndrome matrix to REF ($(t+1)t^2$ multiplications and $(t+1)t(t-1)$ additions), nding the error locator polynomial ($t(t+1)=2$ multiplications and the same number of additions), determining the roots of that polynomial (a maximum of qt multiplications, qt additions and q comparisons), nding the error values ($(t+1)(n-k)^2$ multiplications and $(t+1)(n-k)(n-k+1)$ additions), obtaining the codeword (t additions) and computing the distance from the received word and applying the stopping criterion ($(n-1)+(n-k+1)$ additions and $(n+2+n\log_2(n))$ comparisons). Thus, denoting by DTC the number of decoding tries involved in the error-only decoder, the total number of operations required for the $A3$ algorithm is given by

$$TopA3 = TopA1 + DTC\ (\ 2(n-k)n + (t+1)(2t^2 + 2q + (n-k)(2(n-k)+1)) + t + 3n - k + 2 + q + n\log_2(n)) \quad (3)$$

As all our algorithms apply a stopping criterion it is easy to see that the higher the SNR, the fewer the decoding attempts needed on average. In our simulation we computed the average number of decoding tries per received word which is then used to compute the total number of operations as given by the above formulae. It is worth noting that the complexity of all three algorithms is dominated by the number of decoding tries. Only for high SNRs, when the average number of decoding tries becomes very small, do the preliminary operations contribute signi cantly to the average number of operations per information bit.

6 Comparing the Three Decoding Algorithms in Terms of Performance and Complexity

6.1 AWGN Channel

Figures 5, 7 and 9 show the performance of the algorithms when applied to respectively a [16; 8; 9], a [16; 10; 7] and a [16; 12; 5] extended RS code. The numbers next to the various algorithms indicate the number of permitted decoding tries, e.g. for the [16; 8; 9] code, the $A2$ algorithm was run with maximum 1500 decoding tries for each side, and the $A3$ algorithm was run with 529 (rst number) Dorsch-style decoding tries permitted and the same maximum number of Chase-style decoding tries. We have included the performance of Forney's GMD [7] and an error-only decoder to enable the reader to compare the new algorithms with two standard ones. Tables 1, 2 and 3 show how many decoding tries were needed for each algorithm at various SNRs. Figures 6, 8 and 10 show the complexity of the algorithms based on the gures in the tables.

Table 1. Ave. num. of decoding tries ([16; 8; 9] extended RS code)

Algorithm	1dB	2dB	3dB	4dB	5dB
$A1$[41449]	40018	36407	27781	14732	4353
$A1$[3000]	2912	2636	1985	1076	316
$A2$[1500; 1500]	2902	2666	2006	1063	310
$A3$[529; 529]	[519; 519]	[462; 462]	[356; 355]	[189; 187]	[57; 55]

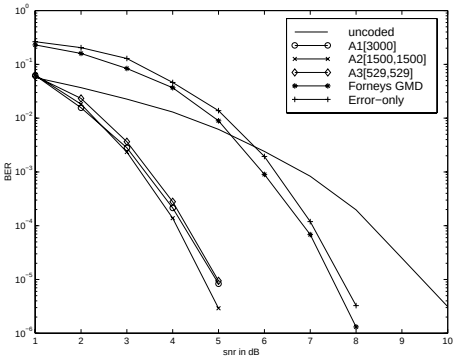


Fig. 5. [16; 8; 9] extended RS code decoded using $A1$, $A2$, and $A3$

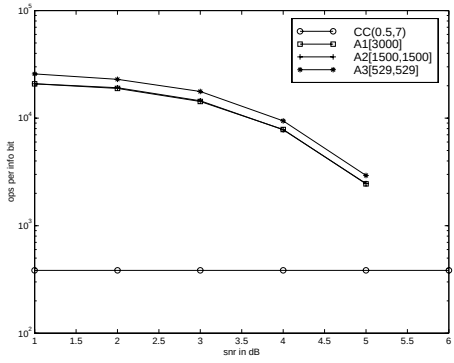


Fig. 6. Complexity of the algorithms ([16; 8; 9] extended RS code)

Comparing Figure 5 with Figure 7, in terms of the BER at various SNR there is hardly any di erence between the [16; 8; 9] and the [16; 10; 7] codes, probably

Table 2. Ave. num. of decoding tries ([16; 10; 7] extended RS code)

Algorithm	1dB	2dB	3dB	4dB	5dB
$A1$[3000]	2865	2513	1805	882	229
$A2$[1500; 1500]	2908	2460	1811	883	229
$A3$[821; 301]	[795; 291]	[693; 254]	[497; 181]	[241; 87]	[64; 22]

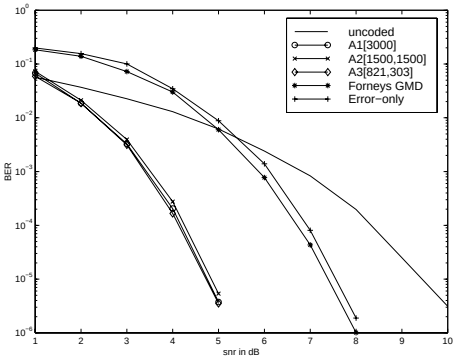


Fig. 7. [16; 10; 7] extended RS code decoded using $A1$, $A2$, and $A3$

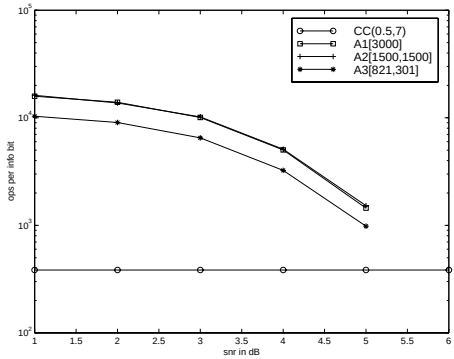


Fig. 8. Complexity of the algorithms ([16; 10; 7] extended RS code)

Table 3. Ave. num. of decoding tries ([16; 12; 5] extended RS code)

Algorithm	1dB	2dB	3dB	4dB	5dB	6dB
$A1$[500]	482	401	305	145	44	7
$A2$[250; 250]	477	403	304	145	44	7
$A3$[49; 17]	[46; 15]	[41; 14]	[29; 10]	[16; 5]	[6; 2]	[2; 1]

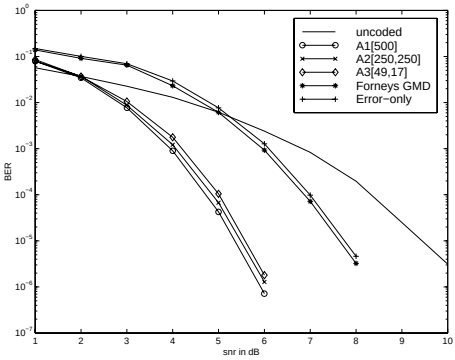


Fig. 9. [16; 12; 5] extended RS code decoded using $A1$, $A2$ and $A3$

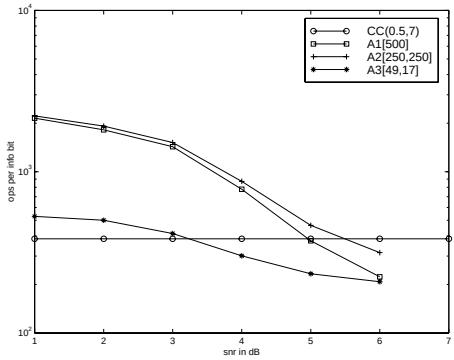


Fig. 10. Complexity of the algorithms ([16; 12; 5] extended RS code)

due to the fact that these decoding algorithms are suboptimal and hence do not achieve the full potential of the lower rate code. In addition, in both Figure 6 and Figure 8, the pair of curves \$A1[3000]" and \$A2[1500; 1500]" overlap. However, whereas the $A3$ algorithm appears to be the best choice for the [16; 10; 7] code in terms of both performance and complexity, for the rate 1=2 code, $A2$ slightly outperforms the other two algorithms and (like $A3$) allows a straightforward parallel implementation, so it is the preferable choice for this code.

In the case of the [16; 12; 5] extended RS code the $A1$ algorithm performs slightly better than the other two. However, it is worth noting that the $A3$ algorithm achieves good results with a very low maximum number of decoding tries and that by slightly increasing the number of decoding tries for the $A3$ algorithm, from [49; 17] to [100; 50], say, one gets a similar performance to the $A1$ algorithm while still having a lower complexity and the advantage of being able to implement it in parallel. This time even for low SNRs the complexity of $A3$ is only slightly worse than that of the Viterbi algorithm. At higher SNRs all algorithms achieve good results with few decoding tries resulting in very few operations per information bit.

6.2 Rayleigh Fading Channel ([16, 8, 9] Extended RS Code Only)

In our simulations we have assumed a perfectly interleaved Rayleigh fading channel, i.e. the fading amplitudes for each bit were completely independent and no channel side information was used in the decoding. We have used the same metric as for the AWGN channel.

Table 4. Ave. num. of decoding tries ([16; 8; 9] extended RS code (Rayleigh channel))

Algorithm	2dB	3dB	4dB	5dB	6dB	7dB	8dB
$A1$[3000]	3000	2929	2865	2670	2383	1941	1397
$A2$[1500; 1500]	3000	2964	2879	2726	2359	1944	1400
$A3$[529; 529]	[529; 529]	[518; 518]	[506; 506]	[475; 475]	[426; 425]	[346; 345]	[250; 248]

Figure 11 shows that the results for the Rayleigh fading channel do not di er very much from the ones we obtained for the AWGN channel. We see, as for AWGN in Section 3, that sorting with respect to the reliability measure Rel_1 or Rel_2 (the two curves overlap) is better than Rel_3. Furthermore, as in the AWGN channel, sorting with 529 decoding tries yields a better performance than unsorted decoding with 5489 decoding tries. For the Rayleigh fading channel, sorting yields a coding gain of about 2dB when compared to the unsorted case with the same number of decoding tries. This time it seems that the algorithms $A1$ and $A2$ perform identically. However, looking at the computed BER values, there is an indication that A2 might outperform A1 slightly for SNRs higher than these. In terms of complexity - see Figure 12 and Table 4 - the only di erence

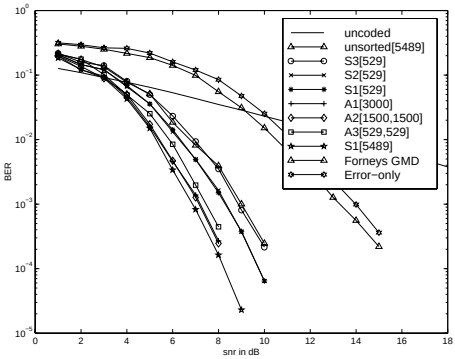


Fig. 11. [16; 8; 9] extended RS code decoded using $A1$,$A2$,$A3$ (Rayleigh)

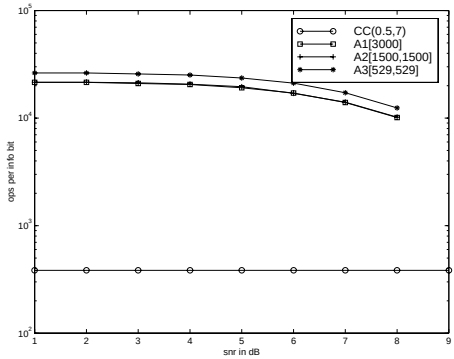


Fig. 12. Complexity of algorithms ([16; 8; 9] extended RS code (Rayleigh))

from the AWGN channel is that the average number of decoding tries decreases more slowly which is obviously due to the nature of the Rayleigh fading channel. Note that, again, the curves for \A1[3000]" and \A2[1500,1500]" overlap.

7 Conclusion

In this paper we have introduced three suboptimal decoding algorithms for RS codes all of which achieve a reduction in complexity of several orders of magnitude over the Viterbi algorithm for these codes whilst keeping the loss in coding gain very small. These algorithms are not restricted to RS codes and could be applied to any linear block code. They achieve their full potential with high rate codes where a small number of decoding tries yields almost maximum-likelihood decoding performance with low decoding complexity.

References

1. Wesemeyer S. and Sweeney P.: Suboptimal soft-decision decoding for some RS-codes. IEE Electronics Letters **34(10)** (1998) 983{984
2. Dorsch B.G.: A decoding algorithm for binary block codes and J-ary output channels. IEEE Trans. Inform. Theory **IT-20(3)** (1974) 391{394
3. Fossorier M.P.C. and Lin S.: Soft-decision decoding of linear block codes based on ordered statistics. IEEE Trans. Inform. Theory **41(5)** (1995) 1379{1396
4. Taipale D.J. and Pursley M.B.: An improvement to generalized-minimum-distance decoding. IEEE Trans. Inform. Theory **37(1)** (1991) 167{172
5. Fossorier M.P.C. and Lin S.: Complementary reliability-based decodings of binary linear block codes. IEEE Trans. Inform. Theory **43(5)** (1997) 1667{1672
6. Stichtenoth, H.: Algebraic function elds and codes. Springer-Verlag, 1993
7. Forney Jr, G.D.: Generalized minimum distance decoding. IEEE Trans. Inform. Theory **IT-12** (1966) 125{131

Weaknesses in Shared RSA Key Generation Protocols

Simon R. Blackburn[?], Simon Blake-Wilson[??], Mike Burmester, and Steven D. Galbraith[???]

Department of Mathematics,
Royal Holloway, University of London,
Egham, Surrey TW20 0EX, United Kingdom.
{S.Blackburn,M.Burmester,S.Galbraith}@rhbnc.ac.uk,
sblakewi@certicom.com

Abstract. Cocks proposed a protocol for two parties to jointly generate a shared RSA key. His protocol was designed under the assumption that both parties follow the protocol. Cocks proposed a modification to the protocol to prevent certain attacks by an active adversary. The paper presents attacks that show that the Cocks protocols are not secure when one party deviates from the protocol.

1 Introduction

Certain applications require two parties to be able to decrypt an RSA ciphertext by working together, but for neither party to be able to decrypt alone. In this situation it would often be useful for the parties to jointly generate their shared RSA key. The need for practical shared RSA key generation protocols has been noted by Bellare and Goldwasser [1] and Gennaro, Jarecki, Krawczyk and Rabin [9]. Such protocols could be used, for example, as part of key escrow protocols (see Denning and Branstad [6]), or for Fiat–Shamir signature parameter generation [7].

The protocols that have been proposed for shared RSA key generation are of two types. Those of the first type are designed to be secure against passive adversaries, i.e. those that adhere to the protocol. Protocols of the second type are designed to be secure against active adversaries, i.e. those that are free to deviate from the protocol.

Cocks [4] was the first to propose a two party shared RSA key generation method. Cocks [5] has extended his protocols to three or more parties. Gilboa [10] has modified the multi-party protocol due to Boneh and Franklin [3] to permit two-party shared RSA key generation. All these protocols are designed only against passive adversaries.

* The author is supported by an EPSRC Advanced Fellowship.
** The author is an EPSRC CASE student sponsored by Racal Airtech.
*** The author thanks the EPSRC for support.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 300–306, 1999.

For many applications of shared RSA key generation, a more realistic security model must include active adversaries. Cocks [4] suggested a modi ed protocol to prevent certain active attacks; we will refer to this as the 'symmetric Cocks protocol'. Poupard and Stern [11] have described a protocol to achieve full security in the presence of an active adversary. Frankel, MacKenzie and Yung [8] have modi ed the Boneh{Franklin protocol to obtain a protocol which is secure in the presence of a minority of active adversaries, in the case of three or more parties.

In this paper we present four types of attack on the Cocks protocols which can be carried out by an active adversary:

1. An attack on the symmetric Cocks protocol which allows one party to substitute a modulus of their choice and to convince the other party that the protocol has been run in a proper manner;
2. Manipulating the primality test so that an RSA key is accepted which is not a product of two primes;
3. Attacks on the asymmetric protocol which force the resulting RSA modulus to have a special form;
4. A high risk strategy by one party which reveals the factorisation of both the resulting shared RSA modulus and their own public key.

Taken together these attacks show that the Cocks protocol should not be used in an active adversary environment without further modi cation (see Blackburn, Blake-Wilson, Burmester and Galbraith [2] for further discussion of this issue).

2 The Cocks Protocols

In this section the two-party shared RSA key generation protocols introduced by Cocks in [4] are described. Cocks proposed two protocols. The rst, asymmetric, protocol is designed to provide security against eavesdropping adversaries. The second, symmetric, protocol prevents certain attacks by active adversaries. In Section 3 we will explain why neither of these protocols can be used in the presence of an active adversary.

2.1 The Asymmetric Cocks Protocol

To generate a shared key pair, Alice and Bob repeat the following process.

Alice chooses two integers p_a and q_a (not necessarily prime) and Bob chooses two integers p_b and q_b (again not necessarily prime). Alice and Bob jointly compute the modulus $N = (p_a + p_b)(q_a + q_b)$ as follows.

1. Alice chooses her own RSA modulus M_a and public exponent e_a, which she makes known to Bob. The exponent e_a should be large to prevent an attack due to Coppersmith; see Cocks [5]. Alice's private decryption key is d_a. The modulus M_a should be larger than the maximum size that N could possibly be. She sends the quantities $p_a^{e_a} \bmod M_a$ and $q_a^{e_a} \bmod M_a$ to Bob.

2. Bob calculates the three elements

$$a_{1;a} = p_a^{e_a} q_b^{e_a} = (p_a q_b)^{e_a} \bmod M_a$$
$$a_{2;a} = p_b^{e_a} q_a^{e_a} = (p_b q_a)^{e_a} \bmod M_a$$
$$a_{3;a} = (p_b q_b)^{e_a} \bmod M_a :$$

3. Bob generates a set of $3K$ numbers $\{b_{i;j;a} : 1 \quad i \quad 3;\ 1 \quad j \quad K\}$, chosen to be random modulo M_a subject to the condition that

$$\sum_{j=1}^{K} b_{i;j;a} = 1 \bmod M_a \quad \text{for all } i \in \{1;2;3\} :$$

Here K is a 'su ciently large' integer. The choice of an appropriate K is discussed below.

4. Bob calculates the $3K$ numbers $x_{i;j;a} = a_{i;a} b_{i;j;a}^{e_a} \bmod M_a$, and sends them to Alice in a new order (for example a random order, or in a sorted order). This is done so that it is impossible for Alice to recover the correspondence between the elements $x_{i;j;a}$ and the pairs $(i;j)$.
5. Alice calculates the elements $y_{i;j;a} = x_{i;j;a}^{d_a} \bmod M_a$, so

$$y_{i;j;a} = b_{i;j;a} a_{i;a}^{d_a} \bmod M_a :$$

Hence Alice can determine

$$p_a q_a + \sum_{i=1}^{3} \sum_{j=1}^{K} y_{i;j;a} = p_a q_a + p_a q_a + p_b q_a + p_b q_b = N \bmod M_a :$$

Since $0 < N < M_a$, Alice has determined N uniquely.

6. Alice sends N to Bob.

Once N has been calculated, Alice and Bob determine whether N is the product of two primes by using, for example, the test due to Boneh and Franklin [3]. The above process is repeated until a candidate N which is the product of two primes has been generated.

Finally, Alice and Bob agree on a small value for e and respectively compute shares d_a and d_b of the corresponding d by exchanging the values of $p_a + q_a$ and $p_b + q_b$ modulo e as described by Boneh and Franklin [3].

Security. We consider the security of this protocol in the case that Alice and Bob do not deviate from the protocol's description.

The security of the protocol certainly relies on choosing the integer K to be su ciently large. The integer K should be chosen so that Alice receives virtually no information about the decomposition of the the sum $\sum_{i=1}^{3} \sum_{j=1}^{K} y_{i;j;a}$ into the three summands $p_a q_b$, $p_b q_a$ and $p_b q_b$. Cocks recommends that K should be chosen so that

$$\frac{(3K)!}{(K!)^3} > M_a{}^2 ;$$

in which case, for most guesses by Alice as to the values of $p_a q_b$, $p_b q_a$, and $p_b q_b$, there will be a partition of the $3K$ fragments into 3 sets which produce these values. If K is chosen to be of this size, it seems that Alice receives no useful information from the protocol, beyond the value of N and her secret information p_a and q_a.

At the end of the protocol, Bob knows p_b, q_b, the modulus N and the encryptions $p_a^{e_a}$, $q_a^{e_a}$ of Alice's secret information. It seems plausible that the extra information that Bob possesses does not help to break the resulting RSA system (i.e. with modulus N) under the assumption that Alice is implementing RSA securely (i.e. with modulus M_a).

2.2 The Symmetric Protocol

Cocks [4] remarks that a dishonest Alice could, of course, cheat in the asymmetric protocol, simply by transmitting a di erent value of N at stage 6. He observes that, to avoid these kinds of active attacks, the process of computing N could be made symmetrical. More explicitly, let $1'$, $2'$, $3'$, $4'$, $5'$ and $6'$ be the steps of the asymmetric protocol with the roles of Alice and Bob reversed. (We attach a subscript of b to all the new variables that arise, to indicate that they are associated with Bob's RSA modulus M_b.) Then a symmetric version of the protocol replaces the steps 1{6 with the steps 1{5, the steps $1'${$5'$, an exchange of hashes of N, and nally the publication and veri cation of N.

3 Attacks on the Cocks Protocols

We now consider various attacks on the symmetric and asymmetric versions of the Cocks protocol. These attacks show that neither protocol prevents attacks by an active adversary.

3.1 Dishonest Alice in the Symmetric Protocol

Our rst attack works on the symmetric version of the protocol. Suppose that Alice is dishonest. Before the protocol begins, she generates an RSA modulus N'. She would like to manipulate Bob into believing that N' is the modulus the pair are trying to generate.

Alice follows the protocol correctly during stages 1 to 5, and during stage $1'$. At this point, she has calculated the modulus N and has received the values $p_b^{e_b} \bmod M_b$ and $q_b^{e_b} \bmod M_b$ from Bob. Suppose that Alice can factor N. (This situation is likely to occur.)

Alice is now able to compute Bob's secret information p_b and q_b as follows. She rst chooses a factorisation $N = p'q'$. Provided that N is not too smooth (which is unlikely to happen), there are not very many choices for p' and q'. For each such choice, she computes $p_b' = p' - p_a$ and $q_b' = q' - q_a$. She then checks whether the following equalities hold:

$$(p_b')^{e_b} = p_b^{e_b} \bmod M_b$$
$$(q_b')^{e_b} = q_b^{e_b} \bmod M_b \ :$$

If these equalities do hold then $p'_b = p_b$ and $q'_b = q_b$, so she has found Bob's secret information. If these equalities do not hold, she tries another choice of the factorisation $N = p'q'$. One choice of the factorisation $N = p'q'$ will always recover p_b and q_b, since $N = (p_a + p_b)(q_a + q_b)$.

Alice now chooses a set of $3K$ elements

$$\{y'_{i;j;b} \in \mathbb{Z}/M_b\mathbb{Z} : 1 \leq i \leq 3;\ 1 \leq j \leq K\}$$

at random subject to the condition that

$$\sum_{i=1}^{3}\sum_{j=1}^{K} y'_{i;j;b} = N' - p_b q_b \bmod M_b\ .$$

She sends elements $x'_{i;j;b}$ in a sorted or random order to Bob, where

$$x'_{i;j;b} = (y'_{i;j;b})^{e_b} \bmod M_b\ .$$

Bob decrypts these elements to recover the elements $y'_{i;j;b}$, and then calculates the sum mod M_b of these elements and $p_b q_b$; this is equal to N' rather than N as Bob hopes. Finally, Alice and Bob exchange a hash of N' rather than N.

Because Alice knows p_b and q_b, she can arrange that N' passes the Boneh-Franklin test. At the end of the protocol, Alice and Bob have agreed on a common modulus N' that is the product of two primes. However, Alice can factor N'.

One way to prevent this attack is to have Alice and Bob exchange messages simultaneously, so that Alice sends flow 1 to Bob at the same time that Bob sends flow $1'$ to Alice, and so on. This can be achieved by the parties exchanging commitments to these flows using a hash function.

3.2 Cheating during the Boneh-Franklin Test

Suppose that Alice and Bob have calculated a candidate N by following the Cocks protocol honestly. Suppose that Alice is dishonest. She could try to factor N. If she succeeds, she is able to recover Bob's secret information p_b and q_b using the method outlined in Subsection 3.1. Now when Alice and Bob execute the Boneh-Franklin test to determine whether N is the product of two primes, Alice can convince Bob that N is the product of two primes, even when this is not the case (since she knows what replies Bob expects to any query). Hence, at the end of the process of generating and testing N, Bob is convinced that N is a valid RSA modulus, but Alice is able to factor N.

This attack is particularly pertinent in the case when Alice has far more computational power than Bob. For then Bob, even after trying (unsuccessfully) to factor N himself, cannot be sure that Alice has not factored N.

Note that the attack could equally well have been launched by Bob.

3.3 Dishonest Bob in the Asymmetric Protocol

In the asymmetric protocol Bob can send elements $x^0_{i;j;a}$ to Alice such that Bob knows the sum of the elements $y^0_{i;j;a} = (x^0_{i;j;a})^{d_a}$ modulo M_a. (For example, Bob chooses random elements $y^0_{i;j;a} \; 2 \; \mathbb{Z}/M_a\mathbb{Z}$ and sends the elements $x^0_{i;j;a} = (y^0_{i;j;a})^{e_a}$ mod M_a to Alice.) Then Alice reveals the integer

$$N^0 = p_a q_a + \sum_{i=1}^{3} \sum_{j=1}^{4} (x^0_{i;j;a})^{d_a} \bmod M_a$$

that she regards as the correct modulus, while Bob is able to compute $p_a q_a$, and hence (assuming that Bob is able to factor this number) he is able to derive p_a and q_a. Since Bob now knows Alice's secret information, he can calculate the answers Alice expects in the Boneh-Franklin test and hence he is able to convince Alice that N^0 is the product of two primes.

There is a second attack that Bob is able to mount. After receiving $p_a^{e_a}$ mod M_a and $q_a^{e_a}$ mod M_a from Alice at stage 1 of the protocol, Bob is able to commit to $p_b = (p_a)^{c_1}(q_a)^{c_2}c_3$, where c_1, c_2 and c_3 are constants of his choice, similarly for q_b. Although he does not know p_a and q_a, he is able to calculate $p_b^{e_a}$ and $q_b^{e_a}$ modulo M_a from the information that Alice has given him, and so Bob can calculate the elements $x_{i;j;a}$. This allows Bob to force N to have a certain form. For example, by taking $p_b = q_a$ and $q_b = p_a$, the resulting N is a square. It is possible that Bob could bene t from the special form of the resulting N and gain an advantage in factorising N.

3.4 Cheating by a Choice of p_a, q_a

In either the symmetric or asymmetric protocol, Alice could cheat by choosing either p_a or q_a to have a factor in common with her public RSA modulus M_a. If she does this, she is able to derive information about the partition of the elements $x_{i;j;a}$ into the three classes $\{x_{1;j;a}\}$, $\{x_{2;j;a}\}$ and $\{x_{3;j;a}\}$ by computing the greatest common divisor of each $x_{i;j;a}$ and M_a. This allows her to obtain Bob's secret information, and so Alice is able to factor N.

A dishonest Bob can carry out a similar attack in the symmetric protocol. In either case, however, the attack is easily foiled by checking that the encrypted material received is coprime to the relevant modulus M_a or M_b.

4 Acknowledgements

We are grateful to Cli ord Cocks for his interest during the preparation of this paper, and Dan Boneh and Alfred Menezes for their comments. Finally, we would like to thank the members of the PostCRYPT group at Royal Holloway.

References

1. M. Bellare and S. Goldwasser, *Lecture Notes in Cryptography*. 1996. Available at http://www-cse.ucsd.edu/users/mihir/
2. S.R. Blackburn, S. Blake-Wilson, M. Burmester and S.D. Galbraith, 'Shared generation of shared RSA keys' Technical report CORR 98-19, University of Waterloo.
   Available from http://www.cacr.math.uwaterloo.ca/
3. D. Boneh and M. Franklin, 'E cient generation of shared RSA keys', in B.S. Kaliski Jr., editor, *Advances in Cryptology { CRYPTO '97, Lecture Notes in Computer Science Vol. 1294*, Springer-Verlag, 1997, pp. 425{439.
4. C. Cocks, 'Split knowledge generation of RSA parameters', in M. Darnell, editor, *Cryptography and Coding: 6th IMA Conference, Lecture Notes in Computer Science Volume 1355*, Springer-Verlag, 1997, pp. 89{95.
5. C. Cocks, 'Split generation of RSA parameters with multiple participants', 1998. Available at http://www.cesg.gov.uk
6. D.E. Denning and D.K. Branstad, 'A taxonomy of key escrow encryption schemes', *Communications of the A.C.M.*, Vol. 39, No. 1 (1996), pp. 24-40.
7. A. Fiat and A. Shamir, 'How to prove yourself: Practical solutions to identi cation and signature problems', in A.M. Odlyzko, editor, *Advances in Cryptology { CRYPTO '86, Lecture Notes in Computer Science Vol. 263*, Springer-Verlag, 1987, pp. 186{194.
8. Y. Frankel, P.D. MacKenzie, M. Yung, 'Robust e cient distributed RSA key generation', In *Proc. of 30th STOC*, 1998, pp. 663{672.
9. R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, 'Robust and e cient sharing of RSA functions', in N. Koblitz, editor, *Advances in Cryptology { CRYPTO '96, Lecture Notes in Computer Science 1109*, Springer-Verlag, 1996, pp. 157{172.
10. N. Gilboa, 'Two party RSA key generation', in M. Weiner, editor, *Advances in Cryptology { CRYPTO '99, Lecture Notes in Computer Science 1666*, Springer-Verlag 1999, pp. 116{129.
11. G. Poupard, J. Stern, 'Generation of shared RSA keys by two parties', In *ASIACRYPT '98*, 1998, pp. 357{371.

Digital Signature with Message Recovery and Authenticated Encryption (Signcryption) - A Comparison

Chan Yeob Yeun*

Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
c.yeun@rhbnc.ac.uk

Abstract. Mitchell and Yeun [8] showed that Chen's scheme [2] is not a digital signature scheme with message recovery, whereas it should be called an authenticated encryption scheme. Also note that similar remarks have been made in [10] regarding schemes recently proposed by Zheng. Thus we will show that there are major differences between a digital signature scheme with message recovery and authenticated encryption scheme by proposing a digital signature with message recovery scheme and signcryption scheme as an example for comparison. The security of the schemes is based on intractability of solving the Diffie Hellman problem as well as finding a collision on one-way hash-function.

1 Introduction

In 1976, Diffie and Hellman [3] introduced the public-key cryptosystem which is based on the discrete logarithm problem (DLP). The intractability of the DLP is equivalent to the security of the ElGamal public-key scheme [4] and its digital signature scheme.

Among the current known signature schemes, RSA [11] is unique in the sense that the signature and encryption functions are inverse to each other. For this reason, an RSA signature can be used with message recovery. On the other hand, a discrete logarithm based signature, such as ElGamal [4] and DSS [1], cannot provide message recovery. The benefits of the message recovery are applications without a hash function and smaller bandwidth for signatures. In 1993, Nyberg and Rueppel [9] proposed the first digital signature with message recovery based on the discrete logarithm problem.

In many applications it is necessary to provide both confidentiality and integrity/origin protection for a transmitted message. This can be achieved using a combination of encryption and a digital signature. However, this doubles the cost of protection, and motivates the work of Horster, Michels and Petersen

* The author is supported by a Research Studentship and Maintenance Award from RHBNC.

M. Walker (Ed.): IMA - Crypto & Coding'99, LNCS 1746, pp. 307{312, 1999.

[5] who introduced an authenticated encryption scheme, designed to provide a combination of services at reduced cost.

Subsequently Lee and Chang [6] modi ed the HMP scheme to remove the need for a one-way function, whilst keeping communication costs the same. This scheme may be advantageous in environments where implementing a one-way function is di cult, e.g. in a smart card with limited memory and/or computational capability.

More recently, Chen [2] introduced a variant of the Lee and Chang scheme, which is claimed to provide the same security level with a simpler speci cation. However, some of the claims made by Chen are incorrect as Mitchell and Yeun [8] pointed out that Chen's scheme [2] is not a digital signature scheme with message recovery, whereas it should be called an authenticated encryption scheme. Also note that similar remarks have been made in [10] regarding schemes recently proposed by Zheng.

2 Comparison for a Digital Signature with Message Recovery and a Signcryption

We observe that there are major di erences between a digital signature scheme with message recovery (see [9]) and authenticated encryption schemes (see [5,6]) as follows:

2.1 A Digital Signature with Message Recovery

Basically, a digital signature with message recovery scheme should satis es the following properties.

- Data integrity/origin protection: This is property whereby data has not been altered in an unauthorised manner since the time it was created, transmitted, or stored by an authorised source as well as protecting one's origin.
- Nonrepudiation: It is computationally feasible for the TTP to settle a dispute between the signer and the recipient in an event where the signer denies the fact that he/she is the sender of the signed text to the recipient. To compare with an authenticated encryption (signcryption), the signer do not reveal any his/her private keys to the TTP.

Thus, in a digital signature with message recovery scheme, the trusted third party (TTP) can always verify the signatures which are sent by the receiver B without B having to divulge any long term secret information to the TTP.

2.2 An Authenticated Encryption (Signcryption)

Basically, an authenticated encryption (signcryption) scheme should satis es the following properties.

- Confidentiality: It is computationally infeasible for an adaptive attacker to find out any secret information from signcrypted text.
- Data integrity/origin protection: It is computationally infeasible for an adaptive attacker to masquerade as the signcrypter in creating a signcrypted text as well as protecting one's origin.
- Nonrepudiation: It is computationally feasible for the TTP to settle a dispute between the signcrypter and the recipient in an event where the signcrypter denies the fact that he/she is the sender of the signcrypted text to the recipient. To compare with a digital signature with message recovery, the signcrypter reveal any his/her private keys to the TTP.

Thus, in a authenticated encryption schemes, only the sender A and the receiver B can only verify a protected message sent from A to B. This is because B can only verify such a message with the aid of his private decryption key. Therefore, one can deduces that this is an unacceptable property for a signature scheme as discussed in section 2.1, where one would normally expect signature verification to be possible without compromise of any private keys.

In the following, we will propose a digital signature with message recovery scheme and signcryption scheme as an example for comparison. The security of the systems is related to the security of Diffie-Hellman [3] and that of randomly chosen one-way collision resistance hash-function. Assume that Diffie-Hellman and one-way collision resistance hash-function are easy to break, then so is the proposed schemes.

3 System Generation

The key centre selects and publishes the system parameters for public usage. Let p be a prime with $2^{511} < p < 2^{512}$, q a prime divisor of $p-1$ with $2^{159} < q < 2^{160}$, g_1 and g_2 $(1 < g_1, g_2 < p)$ integers of order q and R a redundancy function (see Section 11.2.3 of [7]), and its inverse R^{-1}, and h is one-way collision resistant hash-function (see Section 9.2.2 of [7]). $p, q, g_1, g_2, R, R^{-1}$ and h are publicly known.

Suppose Alice has two private keys X_{A_1}, X_{A_2} $(1 < X_{A_1}, X_{A_2} < q)$, and two public keys:

$$P_{A_1} = g_1^{X_{A_1}} \bmod p,\ P_{A_2} = g_2^{X_{A_2}} \bmod p.$$

Similarly, suppose Bob has two private keys X_{B_1}, X_{B_2}, $(1 < X_{B_1}, X_{B_2} < q)$, and two public keys:

$$P_{B_1} = g_1^{X_{B_1}} \bmod p,\ P_{B_2} = g_2^{X_{B_2}} \bmod p.$$

In addition, every participant must have a means of obtaining a verified copy of every other participant's public signature verification keys. This could, for example, be provided by having the key centre certify every participant's public keys, and having every participant distribute their certificate with every signed message they send.

4 A Digital Signature with Message Recovery Scheme

To sign a message $m \in \mathbb{Z}_p$, Alice randomly chooses two integers k_1 and k_2, $1 < k_1; k_2 < q$ and computes the following:

$$m' = R(m);$$

$$r = m' g_1^{-k_1} g_2^{-k_2} \bmod p;$$

$$s_1 = k_1 - h(r) X_{A_1} \bmod q;$$

and

$$s_2 = k_2 - h(r) X_{A_2} \bmod q:$$

Then Alice sends $\mathrm{Sig}(m) = (r; s_1; s_2)$ to Bob. After receiving $\mathrm{Sig}(m)$, the message can be recovered by Bob as follows:

$$m' = r g_1^{s_1} g_2^{s_2} P_{A_1}^{h(r)} P_{A_2}^{h(r)} \bmod p:$$

After checking the validity of m', the message can be recovered by computing

$$m = R^{-1}(m'):$$

This digital signature is secure if one selects a secure redundancy function R and a randomly chosen one-way collision-resistant hash-function h are used and providing that solving two discrete logarithms problems are computationally infeasible.

Observe that this scheme is a digital signature with message recovery as discussed in section 2.1, i.e. it satis es the data integrity/ origin protection and nonrepudiation. The trusted third party (TTP) can always verify the signatures which are sent by the receiver Bob without Bob having to divulge two long term private keys to the TTP.

5 An Authenticated Encryption (Signcryption) Scheme

Suppose that Alice wants to send a message m to Bob. Then she rst chooses two random integers k_1 and k_2, $1 < k_1; k_2 < q$ and computes the following:

$$K_1 = (P_{B_1}^{k_1} \bmod p) \bmod q;$$

$$K_2 = (P_{B_2}^{k_2} \bmod p) \bmod q;$$

$$m' = R(m);$$

$$r = m' K_1 - K_2 \bmod p;$$

$$s_1 = k_1 - h(r) X_{A_1} \bmod q$$

and

$$s_2 = k_2 - h(r) X_{A_2} \bmod q:$$

Then Alice sends $(r; s_1; s_2)$ to Bob. After receiving $(r; s_1; s_2)$, Bob computes the following:

$$P_{A_1B_1} = g_1^{X_{A_1} X_{B_1}} \bmod p;$$

$$P_{A_2B_2} = g_2^{X_{A_2} X_{B_2}} \bmod p;$$

$$K_1 = (P_{B_1}^{s_1} P_{A_1B_1}^{h(r)} \bmod p) \bmod q$$

and

$$K_2 = (P_{B_2}^{s_2} P_{A_2B_2}^{h(r)} \bmod p) \bmod q:$$

Thus, he computes

$$K_1^{-1}(r + K_2) \bmod p = m' \bmod p:$$

After checking the validity of m', the message can be recovered by computing

$$m = R^{-1}(m'):$$

This authenticated encryption (signcryption) scheme is secure if one chooses a secure redundancy function R and a randomly chosen one-way collision- resistant hash-function is used and providing that solving the discrete logarithms are computationally infeasible.

Observe that this scheme is a authenticated encryption (signcryption) scheme as discussed in section 2.2, i.e. it satis es con dentiality, data integrity/origin protection and nonrepudiation. Only the sender Alice and receiver Bob can verify an authenticated encryption message sent from Alice to Bob. This is because Bob requires his private keys for veri cation.

6 Conclusion

We have shown that there are major di erences between a digital signature scheme with message recovery and authenticated encryption scheme. We also have proposed a new digital signature with message recovery scheme which satis es the properties of data integrity/origin protection and nonrepudiation, and a new signcryption scheme which satis es the properties of con dentiality, data integrity/origin protection and nonrepudiation are an example for comparison. The security of these schemes is based on intractability of solving the Di e Hellman problem as well as nding a collision on one-way hash-function.

7 Acknowledgements

The author is grateful to Fred Piper for his support, and to Chris Mitchell and Mike Burmester for comments on an early draft of the paper.

References

1. The digital signature standard proposed by NIST. *Communications of the ACM*, 35(7):36{40, 1992.
2. K. Chen. Signatrue with message recovery. *Electronics Letters*, 34(20):1934, 1998.
3. W. Di e and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644{654, 1976.
4. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469{472, 1976.
5. P. Horster, M. Michels, and H. Petersen. Authenticated encryption schemes with low communication costs. *Electronics Letters*, 30(15):1212-1213, 1994
6. W. Lee and C. Chang. Authenticated encryption scheme without using a one-way function. *Electronics Letters*, 31(19):1656-1657, 1995.
7. A.J. Menezes, P.C. van Oorschot and S.A. Vanstone. Handbook of Applied Cryptography, CRC Press, 1997
8. C.J. Mitchell and C.Y. Yeun. Comment{Signature scheme with message recovery. *Electronics Letters*, 35(3):217, 1999.
9. K. Nyberg and R.A. Rueppel Message recovery for signature schemes based on the discrete logarithm problem. In *Advances in Cryptography { Proceedings of EUROCRYPT '94*, pages 175{190, Springer-Verlag, 1995.
10. H. Petersen and M. Michels. Cryptanalysis and improvement of signcryption schemes. *IEE Proceedings on Computers and Digital Techniques*, 145:149{151, 1998.
11. R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21:120{126, 1978.

Index