Lecture Notes in Computer Science 4043

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Andrea S. Atzeni   Antonio Lioy (Eds.)

Public Key Infrastructure

Third European PKI Workshop:
Theory and Practice, EuroPKI 2006
Turin, Italy, June 19-20, 2006
Proceedings

Springer

Volume Editors

Andrea S. Atzeni
Antonio Lioy
Politecnico di Torino
Dip. di Automatica ed Informatica
Corso Duca degli Abruzzi, 24, 10129 Torino, Italy
E-mail: {shocked,lioy}@polito.it

Library of Congress Control Number: 2006927052

CR Subject Classification (1998): E.3, D.4.6, C.2.0, F.2.1, H.3, H.4, K.4.4, K.6.5

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-540-35151-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-35151-1 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11774716 06/3142 5 4 3 2 1 0

Preface

Today, PKIs have come of age and they support the security of several large networked systems, such as company-wide document management systems, e-government applications and secure VPN. However, despite this success, the field has not yet reached its full scientific maturity and there is still room for research in this area. For example, open issues exist in the efficient management of large PKI (especially with respect to certificate validation), better performance could be attained by improved cryptographic techniques and innovative applications are continuously proposed.

To discuss progress in the PKI field, the European PKI workshop series was established in 2004, following similar initiatives in Asia and the USA. The first two events of this series took place on the Island of Samos, Greece (EuroPKI 2004), and in Canterbury, UK (EuroPKI 2005).

This book contains the proceedings of the Third European PKI Workshop (EuroPKI 2006), held at the Politecnico di Torino, Italy, on June 19-20, 2006. In response to the Call for Papers, about 50 submissions were received. All submissions were reviewed by at least two reviewers (external or members of the Program Committee) and most of them got three reviews. At the end of this process, 22 papers were selected, 18 in their full form and 4 as short papers. These papers led to a lively workshop, with a good mixture between theory and application, continuing the success of the previous workshops in the series.

I would like to thank the authors for their papers, the Program Committee and external reviewers for their efforts during the review process, and finally all the workshop participants, without whom the workshop would have not been successful.

June 2006 Antonio Lioy

Organization

EuroPKI 2006 was organized by the TORSEC Computer and Network Security Group (http://security.polito.it) at the Dipartimento di Automatica ed Informatica of the Politecnico di Torino, in cooperation with the Istituto Superiore Mario Boella.

Program Chairman

Antonio Lioy

Organizing Chairman

Andrea S. Atzeni

Program Committee

A. Buldas, University of Tartu (Estonia)
D. Chadwick, University of Kent (UK)
S. Farrell, Trinity College Dublin (Ireland)
S. Furnell, University of Plymouth (UK)
D. Gollmann, Hamburg University of Technology (Germany)
S. Gritzalis, University of the Aegean (Greece)
Y. Karabulut, SAP AG (Germany)
S. Katsikas, University of the Aegean (Greece)
S. Kent, BBN (USA)
K. Kim, Information and Communications Univ. (Korea)
A. Lioy, Politecnico di Torino (Italy)
J. Lopez, Universidad de Malaga (Spain)
F. Martinelli, IIT-CNR (Italy)
F. Maino, Cisco (USA)
D. Mazzocchi, ISMB (Italy)
C. Mitchell, Royal Holloway (UK)
W. Schneider, Fraunhofer SIT (Germany)
A. Vaccarelli, IIT-CNR (Italy)

External Reviewers

A. Atzeni, Politecnico di Torino (Italy)
M. Aime, Politecnico di Torino (Italy)

D. Berbecaru, Politecnico di Torino (Italy)
A. Dent, Royal Holloway (UK)
J. Haller, SAP AG (Germany)
F. Kerschbaum, SAP AG (Germany)
J. Iliadis, University of the Aegean (Greece)
D. Lekkas, University of the Aegean (Greece)
C. Lambrinoudakis, University of the Aegean (Greece)
G. Morgari, Telsy (Italy)
M. Pala, Politecnico di Torino (Italy)
K. Paterson, Royal Holloway (UK)
G. Ramunno, Politecnico di Torino (Italy)
G. Sburlati, IIT-CNR (Italy)

Sponsor

Istituto Superiore Mario Boella, Torino, Italy

Table of Contents

PKI Management

Authentication I

Cryptography

Applications

Authentication II

Short Contributions

Use of a Validation Authority to Provide Risk Management for the PKI Relying Party

Jon Ølnes[1] and Leif Buene[2]

[1] DNV Research, Veritasveien 1, N-1322 Høvik, Norway
[2] DNV Certification, Veritasveien 1, N-1322 Høvik, Norway
{jon.olnes, leif.buene}@dnv.com

Abstract. Interoperability between PKIs (Public Key Infrastructure) is a major issue in several electronic commerce scenarios. A Relying Party (RP), in particular in an international setting, should not unduly put restrictions on selection of Certificate Authorities (CA) by its counterparts. Rather, the RP should be able to accept certificates issued by any relevant CA. Such acceptance implies not only the ability to validate certificates, but also an assessment of the risk related to acceptance of a certificate for the purpose at hand. We analyse common PKI trust models with respect to risk management, and argue that an independent, trusted Validation Authority (VA) may be a better approach for this task. A VA as suggested by this paper will also remove the need for complicated certificate path processing.

1 Introduction

Public key cryptography used with a PKI (Public Key Infrastructure) carries the promise of authentication, electronic signatures and encryption based on sharing of only non-secret information (public keys, names and other information in certificates[1]). The same information (the certificate) may be shared with all counterparts, to replace separate, shared secrets.

The counterpart (RP for Relying Party – relying on certificates) must be able to validate the certificate (with respect to validity period, revocation status, authenticity, and integrity) and interpret its content. In addition, the RP must decide if the quality of the certificate is sufficient for the purpose at hand, and whether or not to accept the issuer of the certificate (the CA – Certification Authority). The latter decisions should be based on evaluation of the risk to the RP.

While the quality of a certificate (chiefly determined by the CA's certificate policy) in most cases is the primary risk element, other aspects of the CA itself, such as

[1] Another term is "electronic ID". A PKI-based electronic ID usually consists of two or three certificates and corresponding key pairs, separating out the encryption (key negotiation) function and possibly also the electronic signature (non-repudiation) function to separate key pairs/certificates. To a user, this separation is normally not visible. This paper uses the term "certificate", to be interpreted as covering the electronic ID term where appropriate.

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 1 – 15, 2006.

nationality, financial status, and reputation may be important. Note also that the policy represents a claimed quality level, and assessment of compliance may be important. An RP will typically also be very interested in the liability taken on by the CA in case of errors, and the possibility for claiming liability if needed.

It is clear that, in particular in an international setting, an RP may need to accept certificates from a large number of CAs. Present approaches to interoperability are trust lists (trusted CAs and their public keys) and formation of trust structures (hierarchy, cross-certification, and bridge-CA) among CAs. We argue that all these approaches have shortcomings with respect to aiding the RP's risk management decisions. Trust structures imply the need to discover and validate potentially complex trust paths through the structures, a major concern in present PKI implementations.

This paper recommends a different approach, where interoperability is offered by means of a trusted Validation Authority (VA), serving as an independent trust anchor for the RP. The VA serves as a clearinghouse between CAs and RPs, and by trusting the VA the RP is able to trust all CAs that the VA answers for.

The model is based on policies and explicit, signed agreements. An overall validation policy for the VA's services is defined, and additionally RPs may define individual policies to tailor services to their needs. The RP has one agreement with the VA, and the VA on the other hand has agreements with the CAs, preferably in a model where one VA-CA agreement covers all RPs that the VA handles. Thus, all actors (including the CAs) obtain a clear risk picture. The VA handles all CAs individually, and as an added value the need for cumbersome certificate path discovery and validation procedures is removed. The RP obtains a one-stop shopping service for acceptance of certificates – one point of trust, one agreement, one bill, one liable actor.

In this trust model, it is important that the VA is neutral with respect to CAs, i.e. the VA service should be offered by an independent actor. In particular, this applies to judgments about quality and other aspects of CAs and their services.

In the following, we clarify DNV's position in 2, describe requirements in 3, take a critical look at existing approaches in 4, describe the independent VA in 5, present elements for certificate validation policies in 6, and conclude in 7.

2 DNV's Position and Role

DNV (Det Norske Veritas, http://www.dnv.com) is an independent foundation offering classification and certification services from offices in more than 100 countries. The maritime sector and the oil and gas industry are the main markets. DNV is also among the world's leading certification bodies for management systems (ISO 9000, ISO 14000, BS 7799 and others), delivering services to all market sectors.

DNV seeks to extend its existing position as a supplier of trusted third party services to digital communication and service provisioning. The first version of a VA service along the lines described in this paper will be offered to pilot customers 3Q 2006. This paper does not describe this pilot service but rather the research leading to the decision to launch the pilot service.

3 The PKI Interoperability Challenge and Scaling

In general, the certificate holder and the RP can be independent entities, who may independently select the CAs to obtain certificates from; then:

- A certificate holder should be able to use the same certificate towards all relevant RPs, regardless of the CA(s) used by the RP itself.
- An RP (e.g. a service provider in an international market) should be able to use and validate certificates from all relevant certificate holders, regardless of the CA of the certificate holder.
- When a digitally signed document is created, the parties involved may be able to identify the relevant CAs. However, the document may need to be verified later by another actor, who may not have any relationship to any of these CAs.

The set of relevant counterparts, and thus the set of relevant CAs, may be limited by criteria such as nationality or business/application area. However, unlimited interoperability may be viewed as the ultimate goal, likened to the ability to make phone calls internationally.

The challenge is primarily on the RP, which is the actor that faces the complexity of a large number of CAs. An RP must not only validate certificates but also assess the risk related to accepting a certificate for a given purpose. This paper suggests using the risk elements: quality of certificate (mainly derived from certificate policy), assessment of quality (e.g. compliance with policy), liability and possibilities to claim liability, and other aspects of the CA and its services (such as nationality). This is further discussed in 6. An uncertain risk situation may be unacceptable to the RP.

PKIs as society infrastructures are being deployed in probably most developed countries for national electronic IDs. Deployment is either based on CAs run by public authorities or on services obtained from the commercial market. Society infrastructures are almost exclusively national, although some international co-ordination takes place. Notably, the EU Directive on electronic signatures [12] defines the concepts of qualified signature/certificate as means to achieve legal harmonisation across the EU in this area. Even in countries with (plans for) public authority CAs, the usual situation is several (2–15 is typical for European countries) public, commercial CAs competing in a national market. PKI interoperability thus may be a challenge even at a national level, and interoperability at an international level is a severe challenge. Some commercial CAs, e.g. Verisign, compete in an international market.

Other PKI deployments add to the scale of the interoperability challenge. Some corporate (business internal) PKIs aim at acceptance of certificates even outside of the corporation. Community infrastructures are under establishment, some even internationally like the SAFE initiative [27] for the pharmaceutical industry. The banking and aerospace industries may be mentioned as other particularly active arenas. The educational sector is very active in the PKI area, and initiatives like the EuroPKI [22] expand the scope outside of the academic sector and internationally.

Thus, the interoperability challenge is necessarily on the agenda. One example is the IDABC (Interoperable Delivery of European E-government Services to Public Administrations, Businesses and Citizens) programme's statement on electronic public procurement [7], related to creation of an internal market[2] in the EU: "The interoperability problems detected [for qualified electronic signatures] despite the existence of standards, and the absence of a mature European market for this type of signatures pose a real and possibly persistent obstacle to cross-border e-procurement." Other examples can be found, notably also from internationally oriented businesses.

4 Present Approaches to PKI Interoperability, Risk Management

4.1 Trust Models and Certificate Paths

Present methods for PKI interoperability are lists of trusted CAs (see 4.5) and creation of trust structures among CAs by issuance of certificates to the CAs themselves; by a peer-CA, a bridge-CA, or a CA at a higher level of a hierarchy. The idea is that an RP should be able to discover and validate a certificate path from a directly trusted CA (typically the root-CA of a hierarchy) to any CA (may be previously "unknown") that is a member of the same trust structure. In this, trust is regarded as a transitive property. The number of CAs directly trusted by an RP can be reduced; however the trust decision must always be derived from a CA accepted as a "trust anchor".

In general, certificate path discovery may be a very difficult task [29], and sufficient support is lacking in many PKI implementations. Certificate path validation may be very resource demanding due to the need for repeated certificate processing. Caching of previously validated trust paths can mitigate this problem. Certificate path validation, possibly also path discovery, may be performed by a validation service (delegated path validation/discovery [30]). Note that the trust model suggested by this paper (see 5.1) eliminates certificate path processing.

In the context of this paper, "trust" not only means the ability to find a trusted copy of a CA's public key but also support for risk management by quality, assessment of quality, and liability issues (the "other aspects" element left out). Below, we examine different trust models with respect to these properties. Note that one may argue that certificate chains increase risk since there is always a >0 probability of failure for each link in the chain.

4.2 Peer-CA Cross-Certification

Peer-CA cross-certification is a mechanism where two CAs mutually (the usual situation, although one-way cross-certification is also possible) issue certificates to one another. With respect to quality, cross-certification with policy mapping means that the two CAs' services are regarded as equal. The complexity involved in the policy mapping depends on the differences in the policies. There are a few common frameworks [6] [8] [9] for structuring of policies. Mapping between the frameworks is not too complicated, and most CAs adhere to one of the frameworks. Still, the real content of policies may differ quite a lot. Without policy mapping, cross-certification may give no indication on quality of the other CA.

Cross-certification is typically carried out in a carefully scrutinized process involving assessment of the peer-CA's claimed quality. As a result, a CA may be willing to

[2] Coined as "the SEEM" (Single European Electronic Market) in EU terms.

take on liability for certificates issued by the other CA. The situation depends on the agreement entered by the two CAs and on the certificate policies applied.

The conclusion is that peer-CA cross-certification may provide risk management since the CA selected by an RP can ensure policy equivalence and take on liability with respect to other CAs.

Achieving cross-certification between CAs that are competitors may be very hard, even if demanded e.g. by public authorities. Peer-CA cross-certification as a scalable solution to interoperability must be regarded as unfeasible. The main use may be where CAs are non-commercial, e.g. corporate PKIs of co-operating businesses.

4.3 Hierarchy

In a hierarchy, CAs are assembled under a common root-CA, which issues certificates to subordinate CAs. Although a hierarchy may in theory have an arbitrary number of levels, practical systems usually have two levels: root-CA and certificate issuing CAs.

With respect to quality, the usual situation is that all CAs are required to have comparable quality. This can be enforced by a common base policy defined by the root-CA. An example may be found in Germany, where all CAs (for qualified certificates) approved for the German government are placed under a root-CA run by the Regulatory Authority for Telecommunications and Post [4].

However, the root-CA policy need not put restrictions on the CAs, as shown by the EuroPKI initiative [13]. In this case, an RP can draw few conclusions (except that the CA's public key is authentic) from the fact that the CA is a member of the hierarchy.

A hierarchy may imply assessment of quality, e.g. requirements that some evaluation must be passed in order to be allowed in the hierarchy. Thus, an RP may consult the root-CA's policy to check if this is covered.

With respect to liability, the weak point is the root-CA. Liability must be balanced by income or funding. A root-CA may run on governmental or international funding, or e.g. by a limited company jointly owned (cost and risk sharing) by the CAs beneath the root-CA. CAs may be willing to pay some amount to join a hierarchy, dependent on how much they gain from joining the hierarchy, but in practice it is not possible to gain much income from operating a root-CA. Without an income, the owner of a root-CA, even if it is a governmental agency, will be reluctant to take on much liability, and liability will remain an issue between the RP and the individual CAs in the hierarchy. Also, the root-CA will typically not take on much liability for its own actions (such as for adding a rogue CA to the hierarchy), but admittedly the chances of events related to the root-CA are very low for well-managed hierarchies.

A hierarchy provides only interoperability between the CAs in the hierarchy. For more general interoperability, one may cross-certify between root-CAs, or use a bridge-CA to connect hierarchies. The main issue here is that cross-certification involving actors that do not take on liability (the root-CAs) may be a questionable approach. To the RP, the liability situation may become unmanageable.

4.4 Bridge-CA

A bridge-CA is a central hub, with which CAs cross-certify. The bridge-CA should be run by some neutral actor, and it shall itself only issue cross-certificates. An RP may start a certificate path at a CA chosen as a trust anchor, and then proceed to a cross-certificate issued to the bridge-CA, and on to the CA that is the endpoint of the trust path. For hierarchies, the usual situation is cross-certification between the bridge-CA and the root-CA.

Indication of quality may be done by requiring a CA to cross-certify with the bridge-CA with policy mapping to the appropriate bridge-CA policy. E.g. the US Federal Bridge CA (FBCA) defines five policy levels [15]. Cross-certification between a CA and a bridge-CA is considerably simpler than peer-CA cross-certification, as the bridge-CA has no (competing) role in issuing of certificates to end entities. But even in this case cross-certification is typically a thorough process where the quality of the CA is assessed by the bridge-CA. This is determined by the bridge-CA's policy.

With respect to liability, a bridge-CA suffers from the same problems as the root-CA of a hierarchy: It may be difficult to get an income from issuance of cross-certificates, and liability must be balanced by income or funding. Thus, the owner of a bridge-CA will be reluctant to take on liability. (For about the same reasons, a bridge-CA usually does not accept the role of a trust anchor but only acts as mediator between CAs/hierarchies.) As an example, the FBCA is not liable to any party unless an "express written contract" exists ([15] section 9.8). Similar limitations also exist for commercial bridge-CAs such as the SAFE Bridge-CA [27] for the pharmaceutical industry. Liability remains an issue between the RP and the individual CA.

Bridge-CAs have so far either a regional scope (as the FBCA) or a defined business scope (may be international, e.g. the SAFE Bridge-CA [27]), which means that there is a need to link bridge-CAs in order to achieve general, global interoperability. The FBCA has defined guidelines for such cross-certification (part 3 of [14]). A preliminary conclusion of ongoing work [1] is that policy mapping must be dropped since policy frameworks for different bridge-CAs are too different. Thus, quality information is not provided. As argued for hierarchies, cross-certification between actors that do not take on liability (the bridge-CAs) may be a questionable approach.

The conclusion is that use of bridge-CAs may provide indication and assessment of quality but not management of liability. In addition, complex certificate paths typically occur. To aid in processing of certificate paths, a bridge-CA may provide directory services and VA services [25] similar to those described in this paper. We argue that with such VA services, the bridge-CA functionality is not needed and the VA functionality is sufficient.

4.5 Trust List Distribution

A trust list consists of named CAs and their public keys. All CAs on the list are trusted. An example is the list of more than 100 CAs included in distributions of Microsoft OSs. CAs may easily be added to or removed from the list, e.g. to introduce national CAs. An RP may manage a trust list entirely on its own.

Trust list management may also be done by a third party, regularly distributing lists to its subscribers. Interoperability is achieved by installation of compatible trust lists at all actors. In Europe, the IDABC Bridge/Gateway CA (EBGCA) actually is a trust list distribution service [5] based on the study in [17][3] and ongoing work in ETSI [10]. The primary purpose of the EBGCA is to list nationally approved or registered issuers of qualified certificates but other CAs may be added. The status of the CA (such as qualified certificates) is indicated as extra quality parameters of the trust list. Quality information is a fairly straightforward extension for any trust list.

The EBGCA is particular in that it defines itself as a trust anchor for the RP and takes on some liability with respect to the RP. In other cases, liability remains an issue between the RP and the individual CA. As for quality information, liability information may in principle be distributed with the trust list; however the distribution service is unlikely to help in claiming liability.

An example of a simpler service is TACAR [23] for the academic sector in Europe. This is a repository of CA keys and policies, available for download to organisations.

5 The Independent, Trusted Validation Authority

5.1 Revising the Trust Model for the RP

In our view, a fundamental flaw in present PKI practice (the EBGCA is an exception) is that a CA is the only actor that can serve as a trust anchor. This requirement leads to the necessity for trust structures and certificate paths in order to navigate from a trusted CA to an “arbitrary” CA.

The CA as the trust anchor is the right model for the certificate holder role, selecting the CA(s) to obtain certificate(s) from. However, the RP role should aim at acceptance of “any” CA’s certificates, regardless of relationships between CAs.

This paper instead suggests a trust model where an independent validation authority (VA) is the trust anchor for the RP. Upon trusting the VA, the RP is able to trust any CA that the VA handles. The VA handles each CA individually, regardless of any trust structure that the CA may participate in. Certificate path discovery and validation are irrelevant (although the VA may use such processing internally to aid in classification and other tasks) since there is no need to prove a path to a “trusted CA”.

This trust model resembles a two-level hierarchy or use of a bridge-CA, but the VA does not issue certificates. It is an on-line service answering requests from RPs. The idea is that the RP is provided with one-stop shopping for validation of certificates: One point of trust, one agreement, one point of billing, one liable actor.

Given this trust model, the state of the art in validation services may be considerably advanced. The RP may outsource all certificate processing to the VA, regardless of the CA that has issued the certificate. The VA checks validity with the appropriate CA, but returns its own answer, not an answer originating from the CA. The answer may include not only the validity of the certificate, but also information guiding the RP’s risk management decision, as defined by validation policies.

Thus, the VA acts as a clearinghouse for information about CAs and their certificates. The VA does not remove the complexity of interoperability, but it handles the

[3] This study disapproves of a VA solution to interoperability. However, in this case the VA is an OCSP [28] service with few similarities to the VA concept presented in this paper.

complexity in one place, for all RPs who have outsourced certificate processing to the VA. Internally, the VA operates a trust list of the CAs it is able to answer for.

Further discussion on the VA concept may be found in [32]. Here, it is suggested to provide VA services as Web Services, extending the XKISS part of XKMS [16].

5.2 Validation Policies, Trust, Quality and Other Characteristics

Accept of a certificate or not is a trust decision for which the RP is responsible. The final trust decision is always a binary yes/no. A VA's reply to a validation request can either be treated as a recommendation with the RP responsible for the final decision, or the VA may be instructed by the RP to give an authoritative answer according to a predefined validation policy (see 6.4). The VA defines a base validation policy, which is deliberately neutral with respect to the CAs. An RP may refine this policy by one or more RP specific validation policies, where non-neutral characteristics such as nationality of CAs may also be included. The validation policies, together with the agreement between the RP and the VA (see 5.3), implements the RP's risk management decisions for acceptance of certificates.

CAs must be allowed to influence or even dictate validation policies. Notably, one must expect that some CAs will deny use of its certificates to RPs in certain countries, and any indication of allowed application in the CA's policy must be obeyed.

There is ample literature suggesting logics and metrics for reasoning about trust in distributed systems in general and PKIs in particular [20] [26] [31]. Our conclusion is that it is difficult to effectively implement these theories in validation policies for a VA. E.g. the VA approach eliminates processing of certificate chains, and many of the metrics are partly based on computations on such chains. Requiring the RPs to define validation policies according to such schemes may break the trust model suggested for the VA. Suggested elements of validation policies are described in 6.

5.3 Agreements, Legal Environment, and VA Liability

Even if a validation policy can be regarded as an "implicit agreement" between VA and RP, a VA's business model is preferably completely based on explicit agreements. It is unlikely that a VA service will accept the risk of running without agreements, and additionally payment for use of the VA service must be ensured.

The VA will typically demand an agreement with the RP according to the VA's jurisdiction. (The alternative is an agreement according to the RP's jurisdiction.) The VA-RP agreement provides one-stop shopping for certificate validation with defined liability and quality of service. The VA's liability must be clearly stated and accepted in the VA's agreement with the RP, and the cost to an RP may depend on the level of risk that the VA takes. Thus, the RP faces a clear risk picture and a defined risk reduction. An important advantage is that the RP is guaranteed to be able to claim liability based on one agreement and under one jurisdiction no matter the CA.

A VA will definitely limit its liability. An RP cannot expect coverage for all losses if a business transaction of high value turns out to be signed using a certificate that should have been revoked. As a rule, the VA may take on the same liability as the CA, and the VA should then be able to transfer liability to the CA if an erroneous answer from the VA is caused by erroneous information from the CA. A VA may decide to provide a common liability level for all CAs of common quality, and the VA may take on more liability (more risk) than CAs if compensated by more income.

A VA will on the other hand need agreements with the CAs. Relying on general statements in a CA's certificate policy will in most cases be too risky. An agreement will normally be according to the CA's jurisdiction since the agreement resembles a relying party agreement with respect to the CA. The CA-VA agreement may be seen as a CA-RP agreement with the VA taking the RP role for all RPs covered by the VA.

Note that an agreement additionally provides risk management for the CA. As an example, the EU Directive on electronic signatures [12] imposes in principle unlimited liability for a CA issuing qualified certificates. Today, the only way for such a CA to control liability is to require agreements with all RPs. With a VA, the chain of agreements from a CA to a VA and on to the RPs may be used to limit liability.

A VA is an on-line service, and there is a clear risk that it will constitute a single point of failure for the RP. Unavailability of the VA will disable use of certificates for all RPs affected by the situation. This situation must be covered by service level agreements between the RPs and the VA. Additionally, the VA actor must ensure a service with very high availability.

An RP must also evaluate the risk related to continuation of the VA's service offering, such as bankruptcy of the actor behind the VA. The agreement between an RP and a VA should ensure that logs and other material of potential evidential value can be transferred to the RP if the agreement is terminated.

Operation of a VA as described in this paper may depend on changes in national legislation. As one example, the German legislation [4] requires a foreign CA to cross-certify with a German CA in order to have its qualified certificates accepted in Germany. The Regulatory Authority for Telecommunications and Post must approve the cross-certification. This is an unfortunate implementation of the paradigm that only a CA may be a trusted actor in PKI. However, an interpretation where a VA may take the CA's role, and the requirement for a cross-certificate as mechanism is relaxed, will solve the situation.

5.4 Customers, Payment, Competition

The liability that the VA takes on, and the operational costs of a VA, must be balanced by an income if the VA shall be able to make a profit out of the service. A VA provides on-line services. The RP will pay for the VA services according to the business model agreed (transaction based, volume based or fixed), and the VA in turn may pay CAs and other information providers according to agreements.

PKI interoperability problems are faced by service providers (government and business), requiring PKI-based authentication and signatures from the customers, and by businesses for (signed) B2B communication. However, VA services to the general public, e.g. to verify signed email no matter the CA of the sender, is also interesting.

An RP should need to trust and have a contract with only one VA. A competitive market exists for certificates (CA services), and correspondingly a competitive market should exist for VA services. Competition should be based on cost and quality of service (QoS). In addition to customary QoS parameters like response time and availability, QoS elements for a VA may be e.g. the number of CAs handled, liability taken on by the VA, the classification scheme used, the interface(s) offered, and additional, value-added services. Open specifications, which eventually may be turned into standards, should be used to enable an RP to switch from one VA to another.

6 Validation Policy Elements

6.1 Identification of Risk Elements

Based on practical experience in use of PKI, we suggest the following elements as the most crucial ones for an RP's risk management as implemented in validation policies:

- Quality of the certificate with respect to the operation at hand.
- Assessment of compliance with claimed quality level.
- Liability and possibility for claiming liability – in the VA model described, this is covered by agreements (see 5.3) and not discussed further below. Some RPs may not be particularly interested in quality, provided that liability is guaranteed.
- Other aspects of the CA and its services.

This ignores obvious requirements such as verified validity, authenticity and integrity of the certificate and a check against allowed use (key usage settings). The elements are discussed further in the rest of this chapter.

6.2 Quality Classification Scheme for Certificates

The quality of a CA's certificates is mainly derived from its certificate policy (CP) [6] [8] [9]. All public CAs are expected to publish their CPs, while intra-organisational CAs and the like today may be run without defined CPs. A classification scheme for CAs without a CP is a difficult task but may be achievable with a thorough examination of the operation of the CA. Other documentation may also be of relevance, such as certification practice statements and agreements with certificate holders and other actors (including membership in hierarchies and cross-certification regimes). Issues related to compliance are discussed in 6.3.

As stated in [24], there is no unified or standardised classification scheme for PKI certificates. However, quality classification of certificates is not a new topic, and a classification scheme for a VA should build on existing work. Examples of existing schemes are the five policy levels of the FBCA [15] in the USA and the quality definitions specified by the EBGCA [5] in Europe. ETSI's document on policy requirements for issuers of non-qualified certificates [9] identifies parameters that influence the quality of certificates and suggests requirements for meeting these parameters at different quality levels.

As stated, the main source of quality parameters is the CP, and important parameters according to RFC3647 [6] are:

- Registration procedure and protection against fake requests for certificates.
- Protection against misuse, i.e. protection of the private key, requirements for use of hardware tokens etc.
- Liabilities and responsibilities of different actors (notably the CA, the customers, the RPs, and RAs), legal environment and jurisdiction, privacy statements, possibly also economical conditions.

- Internal control mechanisms or (preferably) third party audit procedures to assess compliance with CP, CPS and internal procedure descriptions (see 6.3).
- Procedures and rules for certificate revocation.
- Operational requirements for certificate issuing, revocation, logging procedures, archival, and key management.
- Physical, logical, and administrative security related to operation of the service, including requirements for equipment and operators.
- Publishing (directory services etc.) of certificates, CRLs, and other information.
- Disaster recovery plans for the CA, and procedures for termination of the service or transfer of the service to another operator.

Some quality aspects cannot be derived from the CP alone, such as legal and commercial issues. Legal compliance, i.e. the CA and its certificates fulfil requirements of its legal environment, may be necessary to ensure legal value of a signed document.

Important commercial aspects for a CA actor are financial status, insurance coverage and other parameters that go in an ordinary corporate rating system, e.g. for credit worthiness. The CA's (and its owners') track record with respect to conflicts and resolutions may be brought in, as well as the customer base and market penetration of the CA as parameters that influence the chances that the CA will survive over time.

A particular criterion for a CA may be its degree of independence with respect to actors involved in the business scenario at hand. While such an evaluation cannot be done for all business cases, an "independence index" may be derived, and it may be possible to list actors for whom independence may be questionable.

Based on this picture, a generalised classification scheme can be defined and represented as a data structure, providing scores for a CA for all or selected elements. Suggestions for such a scheme are not discussed further in this paper. Most criteria will yield (fairly) static information, but the situation must of course be monitored over time to accommodate e.g. changes in CP. The scheme must be neutral with respect to CAs, i.e. aspects such as nationality should not be considered.

Quality information for a CA may be indicated to the RP by disclosing the contents of this data structure. However, it is expected that many RPs will prefer a simpler approach, where a limited number of quality levels are defined. Thus, an algorithm to get from the structure to a discrete level (e.g. 1-10 with increasing quality) should also be devised, where some criteria may be mandatory for certain levels and other criteria may be weighted against their importance.

Such a classification scheme resembles policy mapping for cross-certification, but the system is more flexible. The classification scheme rates certain characteristics of a CA and its services to obtain either an overall score or a descriptive structure, whereas a policy mapping needs to determine compliance between two policies. A classification scheme with just a few discrete classes may be closer to a policy mapping scheme (e.g. the five levels of the FBCA), while a more fine grained classification allows CAs to differ in policies but still fit in the classification scheme.

Existing initiatives at quality classification have a regional scope. There is no defined link between a qualified certificate in Europe and a certificate at a high quality level for the FBCA (thus the issue of policy mapping for bridge-bridge cross-certification, as discussed in 4.4). Thus, either the number of classes must be high enough to accommodate all the schemes, or correspondence between schemes must be defined. In the latter case, a problem may occur if a scheme is changed, although this must be expected to be a rare event.

Note that qualified certificate according to the EU directive on electronic signatures [12] only sometimes can be used as a binary quality indicator. However, even certificates marked as “qualified” can differ in quality. A CP for qualified certificates usually requires use of a smart card or similar token, but examples can be found where qualified certificates are issued for software key stores.

The classification scheme may be standardised or be left as a competitive element for a VA. We would suggest a standardised scheme since this can be used for compliance assessment to eventually obtain a certification system for CAs (see below).

6.3 Assessment of Compliance with Claimed Quality

In itself the CP (and other documentation) represents only a quality level claimed by the CA. A statement of compliance will frequently be required. Such a statement may in rough order of assurance be:

- Study of documentation without compliance checking.
- Statement of compliance made by the CA itself.
- Reputation, i.e. statements from actors that have chosen to accept the CA.
- The CA’s membership in trust structures (hierarchies and cross-certification regimes). Many such regimes have requirements for compliance assessment, e.g. a rigid assessment is necessary for cross-certification with the FBCA.
- Accreditation or surveillance regimes accepted by the CA. In Europe, surveillance by a national body is required for a CA issuing qualified certificates and in some countries (e.g. Germany [4]) accreditation is required.
- Third party audit reports (many CPs demand third party audits).
- Compliance certificates and certificates such as BS7799 [3], ISO9000, and ISO15408 [18] for organisation, equipment or systems.

The assurance level will contribute parameters for a classification data structure as described in 6.2. Assurance level may be incorporated in a discrete level scheme such that higher assurance level implies higher quality classification. Assurance level may also be mediated as a separate indicator additional to quality.

Today, no international standards exist for compliance assessment in the PKI area. Parts of the compliance work can rely on BS7799 and ISO15408 (to the degree that protection profiles exist for equipment). Standardised assessment and accreditation may be an important aspect of international PKI interoperability [2].

When a classification scheme for CAs has been developed as suggested in 6.2, the scheme may be used to develop a certification scheme for CAs. By use of an accredited auditor, a CA can be certified against a particular profile or a particular level of the classification scheme.

Such assessment systems exist today on national, regional or business sector levels, such as PAG (PKI Assessment Guidelines) of the ABA (American Bar Association), and the American WebTrust Program. As stated in [2]: “With no common agreed method for rating the quality of digital certificates, trust in extra-domain certificates can only be slowly and expensively constructed through a small number of given models, such as cross-certification, bridge-CAs and cross-recognition, each of which has its own shortcomings." The VA approach suggested by this paper aims at improvements in this area since quality may be the most important risk parameter.

6.4 Other Aspects – Validation Policies for Individual RPs

A VA as described in this paper shall be neutral with respect to CAs, which means that both the classification scheme and the base validation policy must be based on only objective criteria. However, trust is always a subjective decision, which means that an RP must be allowed to add subjective criteria to the VA's general scheme. This can be done by allowing RP specific validation policies to be defined.

An RP's validation policy will consist of requirements related to the classification scheme, e.g. "at or above level 5", plus possibly other parameters not covered by the neutral base policy and classification scheme. An RP may want to block (or explicitly allow) certificates from certain CAs based on criteria like nationality, competitive situation (e.g. owned by competitors) etc. Several response models may be used to accommodate RP specific policies:

- The VA answers according to the base validation policy, and the RP applies its own, local validation policy to the answer.
- The RP registers pre-specified validation policies with the VA and selects the policy to refer to when calling the VA. The VA compares requirements to the profile of the certificate in question and returns a yes/no answer or a report on deviations from the validation policy (requirements may be mandatory or desired, and deviations on desired quality properties need not necessarily block a transaction).
- The RP passes the desired validation policy as a parameter in the call to the VA, and the VA answers as for pre-registered policies.

This model enables an RP to define its requirements in general terms, and thus be able to accept certificates from CAs that the RP has no knowledge of, but which fulfils the requirements. Quality classification given as a discrete number is easily integrated with the RP's systems. Models where the RP must interpret a data structure require more software on the RP side.

The RP should in principle treat the answer from the VA as a recommendation. The final decision to accept the certificate or not lies with the RP, but the VA can be instructed to give a "no" answer on the validity of the certificate if the RP's policy is registered with the VA, and the policy requirements are not met.

7 Conclusions

International interoperability of PKI-based certificates and digital signatures is difficult today, and one of the main problems is how a Relying Party (RP) can assess the risk related to acceptance of a certificate from an "arbitrary" CA for a given purpose (such as a signed document). We suggest quality of certificate, assessment of quality, and control of liability as the main risk elements; together with RP specific elements such as nationality of CA. Existing trust models for interoperability are examined and found to have deficiencies with respect to risk management.

An independent, trusted Validation Authority (VA) is proposed as a better solution to interoperability, and the paper shows how guidance in risk management can be provided by use of the VA as a one-stop shopping service for certificate validation. As an important added value, the need for certificate path processing is removed, as the VA serves as an independent trust anchor for the RP.

References

1. Alterman P., Blanchard D., Chokani S., Rea S.: Bridge-to-Bridge Interoperability. Panel presentation at the 5th Annual PKI R&D Workshop (2006)
2. Backhouse J., Hsu C., Tseng J., Baptista J.: A Question of Trust – An Economic Perspective on Quality Standards in the Certification Services Market. Communications of the ACM, Vol. 48 No 9 (2005)
3. British Standards Institute: Specification for Information Security Management Systems. British Standard BS 7799-2:2002 (2002)
4. Bundesnetzagentur: Ordinance on Electronic Signatures. (2001)
5. Certipost: Certification Practices Statement, European IDABC Bridge/Gateway CA for Public Administrations v2.0. EBGCA-DEL-015 (2005)
6. Chokani S., Ford W., Sabett R., Merrill C., Wu S.: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework. RFC3647 (2003)
7. Commission of the European Communities: Action Plan for the Implementation of the Legal Framework for Electronic Public Procurement. Communication from the Commission to the Council, the European Parliament, the European Economic and Social Committee and the Committee of the Regions (2004)
8. ETSI: Electronic Signatures and Infrastructures; Policy Requirements for Certification Authorities Issuing Qualified Certificates. ETSI TS 101 456 v1.4.1 (2006)
9. ETSI: Electronic Signatures and Infrastructures; Policy Requirements for Certification Authorities Issuing Public Key Certificates. ETSI TS 102 042 v1.2.2 (2005)
10. ETSI: Electronic Signatures and Infrastructures; Provision of Harmonized Trust Service Provider Information. Draft ETSI TS 102 231 v1.2.1 (2005)
11. ETSI: Electronic Signatures and Infrastructures; International Harmonization of Policy Requirements for CAs Issuing Certificates. ETSI TR 102 040 v1.3.1 (2005)
12. EU: Community Framework for Electronic Signatures. Directive 1999/93/EC of the European Parliament and of the Council (1999)
13. EuroPKI Top Level Certification Authority: EuroPKI Certificate Policy, Version 1.1. (2004)
14. Federal PKI Policy Authority (FPKIPA): US Government Public Key Infrastructure: Cross-Certification Criteria and Methodology Version 1.3. (2006)
15. Federal PKI Policy Authority (FPKIPA): X.509 Certificate Policy for the Federal Bridge Certification Authority (FBCA) Version 2.1. (2006)
16. Hallam-Baker P., Mysore S.H. (eds.): XML Key Management Specification (XKMS 2.0). W3C Recommendation. (2005)
17. IDA: A Bridge CA for Europe's Public Administrations – Feasibility Study. European Commission – Enterprise DG, PKICUG project final report (2002)
18. ISO: Evaluation Criteria for IT Security. ISO 15408 Parts 1-3 (1999)
19. ITU-T | ISO/IEC: OSI – the Directory: Authentication Framework. ITU-T X.509 | ISO/IEC 9594-8 (2001)

20. Jøsang A., Knapskog S.J.: A metric for trusted systems. NSA1998 – 21st National Security Conference (1998)
21. Kent S.: Privacy enhancement for Internet electronic mail: Part II: Certificate-Based Key Management. RFC1422 (1993)
22. Lioy A., Marian M., Moltchanova N., Massimiliano P.: The EuroPKI Experience. EuroPKI 2004 – First European PKI Workshop (2004)
23. Lopez D.L., Malagon C., Florio L.: TACAR: A Simple and Fast Way for Building Trust Among PKIs. EuroPKI 2004 – First European PKI Workshop (2004)
24. Lopez J., Oppliger R., Pernul G.: Classifying Public Key Certificates. EuroPKI 2005 – Second European PKI Workshop (2005)
25. Malpani A.: Bridge Validation Authority. ValiCert White Paper. (2001)
26. Maurer U.: Modeling a public-key infrastructure. ESORICS '96 – European Symposium on Research in Computer Security (1996)
27. McBee F., Ingle M.: Meeting the Need for a Global Identity Management System in the Life Sciences Industry – White Paper. SAFE BioPharma Association. (2005)
28. Myers M., Ankney R., Malpani A., Galperin S., Adams C.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP. RFC2560 (1999)
29. OASIS: Understanding Certification Path Construction. White Paper from PKI Forum Technical Group (2002)
30. Pinkas D., Housley R.: Delegated Path Validation and Delegated Path Discovery Protocol Requirements. RFC3379 (2002)
31. Reiter M.K., Stubblebine S.K.: Authentication metric analysis and design. ACM Transactions on Information and System Security, Vol. 2, No. 2, pp 138-158 (1999)
32. Ølnes J.: PKI Interoperability by an Independent, Trusted Validation Authority. 5th Annual PKI R&D Workshop (2006)

Modeling and Evaluation of Certification Path Discovery in the Emerging Global PKI

Meiyuan Zhao[1] and Sean W. Smith[2]

[1] Communications Technology Lab
Intel Corporation
Hillsboro, OR 97124
meiyuan.zhao@intel.com
[2] Department of Computer Science
Dartmouth College
Hanover, NH 03755
sws@cs.dartmouth.edu

Abstract. Establishing trust on certificates across multiple domains requires an efficient certification path discovery algorithm. Previously, small exmaples are used to analyze the performance of certification path discovery. In this work, we propose and implement a simulation framework and a probability search tree model for systematic performance evaluation. Built from measurement data collected from current PKI systems in development and deployment over more than 10 countries, our model is (to the best of our knowledge) the largest simulated PKI architecture to-date.

1 Introduction

Public key infrastructure (PKI) is a powerful tool for protecting information. Current development and deployment of PKI systems shows a trend toward an emerging global PKI, where individual PKI domains by governments, institutions, and enterprise establish trust relationships via cross-certification technology. However, as a PKI becomes more complicated, so does the work required for validating an individual certificate. The first step is *certification path discovery*: constructing a "chain of certificates" that connects the certificate in question to a trust anchor. It is challenging to locate appropriate resources to establish a candidate path and to maximize its chance of being valid.

The global PKI spans many countries and consists of many domains, CAs, repositories, and users. PKI protocols need to be robust in such a complex network environment. By establishing trust relationships between domains, cross-certification confronts us with a complex "certificate topology". Moreover, users in different PKI domains may display completely different behaviors that may impact the effectiveness of PKI protocols.

Previous analyses of certification path discovery focused mostly on using small examples to understand algorithm options. In this study, we evaluate its performance in the context of the emerging global PKI. The power of *simulation* allows us to model such complex certificate topologies and to simulate realistic situations. It also enables us to explore a wide range of algorithm options and different network environments, and to examine the effect of user activities as well. We make the following contributions:

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 16–30, 2006.

- We design and implement a PKI simulation framework for general-purpose PKI performance study. This framework implements classical X.509 PKI services and is flexible to allow new types of models and performance studies.
- We design and implement a *PathBuilder* module for this framework. This module uses novel probability search tree models to simulate a variety of algorithm behaviors for certification path discovery.
- We model a global PKI architecture using measurement data collected from current PKI system deployment over more than 10 countries. To the best of our knowledge, this is the largest simulated PKI architecture to-date.
- Using these tools, we evaluate performance of certification path discovery using a range algorithm options. We show that the performance is sensitive to algorithm options, PKI architectures, and user activities.

We hope to make our tools publicly available, as open source.

In the rest of this paper, Sect, 2 discusses the background of PKI system and certification path discovery. Sect, 3 presents previous research. Sect, 4 discusses our simulation framework for general purpose PKI systems. Sect, 5 discusses details of our work on modeling certification path discovery and performance analysis. Finally, we conclude this work with discussions in Sect, 6 and 7.

2 PKI and Certification Path Discovery

PKI was first proposed [14] for securely distributing public keys. It has now evolved to architectures providing comprehensive services for public key *certificates*; these services include storing and retrieving certificates, maintaining and updating certificate status, and validating certificates. In a traditional X.509 [10] PKI system, the certificate storage service is provided by a repository that supports protocols for users to store and retrieve directory information; the protocol used most commonly here is the *Lightweight Directory Access Protocol (LDAP)* [23]. The *certificate status information (CSI)* service communicates the validity status of certificates. A certificate is typically considered as “valid”, “revoked”, or “unknown”. Classical approaches to CSI includes periodically updated data structures such as a *certificate revocation list (CRL)* [10], and online protocols such as *online certificate status protocol (OCSP)* [17].

2.1 Certification Path Discovery

The user who tries to validate a certificate is referred to as *relying party*. A certificate validation service handles *certification paths*, sequences of certificates representing a trust path to the certificate of interest. In such a sequence, the issuer of the first certificate is called a *trust anchor*; a trust anchor is an entity the relying party trusts by default. The last certificate in the sequence is called the *target*; the target certificate is the one that the relying party is trying to validate. In a path, consecutive certificates are linked together by having the *subject* of the previous certificate match the *issuer* of the next certificate.

A certificate validation service is composed of two stages: certification path *discovery* and certification path *validation.* The latter stage is well-established. RFC3280 defines an algorithm to validate a certification path. Basically, the algorithm examines each certificate in the path to decide if they satisfy all required conditions. Unfortunately, the algorithm for actual construction of candidate certification paths is not well defined. Several issues affect the practibility and efficiency of the certification path discovery process; we now consider some.

PKI Architecture. One critical issue is the increasing complexity of *PKI architectures*, a term we use to describe the organization of CAs and their trust relationships. A typical *PKI domain* defines a set of certification policies to manage certificates for its local users. There could be several *certification authorities(CAs)* in the system issuing certificates. These CAs may form a hierarchy having a root CA issuing *CA certificates* for subordinate CAs who in turn issue *end entity certificates* for normal users. The root CA is the common trust node for all subordinate CAs and users in this domain.

The introduction of *cross-certification* enables isolated PKI domains to efficiently establish trust with each other. In cross-certification, CAs from different PKI domains certify to each other, so that relying parties are able to establish trust paths for certificates in remote PKI domains without changing their trust anchor configuration. Furthermore, *bridge CAs* are introduced to bring structure and efficiency to cross-certification. Bridges ease the job for ordinary CAs by handling PKI policies and other constraints of cross-certification. Bridge CAs also help reduce the number of required certificates. Without a bridge CA, N domains need up to $N(N-1)/2$ cross-certificate pairs to establish trust with each other. A bridge CA reduces this number to N, where every CA cross-certifies only with the bridge CA.

Currently, there are several bridge CAs in operation or in development. In the US, the *Federal Bridge CA (FBCA)* [8] cross-certifies with more than eight Federal agency PKIs. The *Higher Education Bridge CA (HEBCA)* [9] facilitates electronic communications within and between educational institutions and Federal and state governments. The *SAFE* bridge [20] sets up trust between members of the BioPharma Association and other enterprise and government PKIs. *CertiPath* [3] is a commercially-managed bridge CA connecting to enterprise PKIs of several aerospace companies.

The trends toward bridging and cross-certification hasten the emergence of a global PKI architecture. However, this architecture creates new challenges for certification path discovery; algorithms must construct a path by traversing different PKI domains, dealing with different PKI policies and handling different protocols.

An algorithm to build certification paths within a PKI architecture can choose one of two directions: the*forward direction* (from the target to trust anchor) and the *reverse direction* (from the trust anchor to the target). The field has seen some debate on which direction is the best for certification path discovery. It appears that the forward direction is mostly appropriate for hierarchical PKIs. We assert that the choice not only depends on the topology of the PKI architecture, but also on other issues, such as the availability of resources that allow the algorithm to locate the appropriate certificates.

Directories store certificates using tuples of the form `(name, attribute)`, where `name` refers to the identity and `attribute` describes the type of object related to this identity. There are several types of attributes useful for certificate retrieval. The directory uses *cACertificate* attribute to store all certificates issued to the CA by the CAs in the same domain and *userCertificate* attribute to store all certificates issued to the end entity. The *crossCertificatePair* attribute has two elements. Its *issuedToThisCA* element stores all certificates issued to this CA including the ones by the CAs in remote domains. Its *issuedByThisCA* element may contain a subset of certificates issued by this CA to other CAs. All objects in the directory are indexed by the name and the attribute. The response to the retrieval request will return a list of objects that satisfy the criteria.

Several private certificate extensions can be used to indicate how to access services related to the certificate. The *Authority Information Access (AIA)* indicates how to access services by the issuer of the certificate. We can use AIA to specify the address of the directory where users can retrieve directory entries for the issuer. The AIA can also specify a list of CAs that have issued certificates to this issuer. Similarly, *Subject Information Access (SIA)* extension indicates how to access services by the subject of the certificate. Although properly defined, these directory attributes and certificate extensions are not fully populated in practice. This makes it difficult for the discovery algorithm to locate appropriate certificates for the path building procedure.

Optimizations. Often, the discovery algorithm faces choice of branches when building a candidate path in the certificate topology. Several optimization techniques have been proposed to help reduce wrong choices in order to speed up the process. For instance, checking signatures and revocation status early can help eliminate bad certificates early, rather than after we have used them to build a candidate path. However, trade-off exists, since the algorithm spends extra time and resource for these operations. Another approach is to prioritize branches to maximize the chance of sucessful discovery. For instance, the *Certificate Path Library (CPL)* [4] used by the *Certificate Arbitrator Module (CAM)* [21] defines a list of criteria to set priorities for branches.

We realize that many of the optimizations deserve more careful evaluation. Recall that in X.509 certificates, the issuer and subject are uniquely identified by their *distinguished names (DNs)*. DNs are an ordered list of naming attributes. Each attribute is called a *Relative Distinguished Name (RDN)*. The usage of RDNs tend to be meaningful to the local PKI system. One may declare that certificates that match more RDNs between the subject DN and the issuer DN should have priority. In other words, the algorithm expects that the issuer and the subject of a certificate in the local PKI domain have similar distinguished names, and the algorithm prefers to stay in the local PKI domain. It is unclear how effective this optimization is in practice. This is yet another reason why we need a systematic way to evaluate it as well as other proposed optimizations.

3 Related Work

Prior research has analyzed certification path discovery using small examples. Elley et al. [6] stated that optimizations in path construction are valuable. They presented a comparison of two directions for path building (forward vs. reverse), analyzed the advantages and disadvantages of each approach, and concluded that building in the reverse direction is often more effective than building in the forwarding direction. Lloyd published a white paper [15] that discussed options for effective and efficient certification path construction algorithm. He specifically pointed out that the forward direction is best suited for hierarchical trust models and the reverse direction is best suited for distributed trust models; he also suggested that building in both directions and meeting in the middle might be a good approach. Russell et al. analyzed the performance issues for constructing and validating long certification paths in cross-domain PKI systems, and proposed the concept of virtual certificates and synthetic certificates to avoid re-constructing and re-verifying certification paths [19]. Unlike these studies, we quantify the performance of the algorithm and evaluate different building options using simulation. (Our work also has the side-effect of producing a simulation tool that can be used for subsequent analyses as well.)

Some researchers have tried systematic approaches to evaluate PKI systems. Iliadis et al. presented a mechanism-neutral framework for the evaluation of CSI mechanisms [11, 12]. The authors proposed a complete evaluation framework that consists of management, performance, and security criteria. This general purpose framework can be used to evaluate many different types of CSI systems. Unfortunately, this system fails to provide quantitative analysis.

Simulation was used for CSI system evaluation too. Årnes implemented a simulation to evaluate certificate revocation performance [1]. His simulation model contains a set of simulation input and output variables, and the models used these variables to compute intermediate variables. However, the simulation models are strictly controlled by formulas. The network environment and user activities are not included.

Muñoz et al. implemented CERVANTES, a testbed for certificate validation [16]. This is a Java platform that allows researchers to develop and test their own "real" revocation systems and to analyze the temporal behaviors. The model makes a few assumptions about configurations, including population size, latency, and connectivity. The testbed is configured with a CERVANTES server and a few clients generating status checking requests. This testbed approach is more realistic than the simulation model by Årnes in that it has real implementations and it takes into account the network environment. However, it is limited by the scale of experiments.

4 PKI Simulation Framework

We design and implement a simulation framework that is capable of modeling PKI protocols and services in network environments. We focus on realizing several important features for this simulation framework—power, flexibility, and scalability.

The framework should be **powerful** to model various PKI protocols such as certificate issuance, revocation, and validation. It should handle different types of network topologies and environments. It should also include different user activities, since PKI systems involve both computer systems and users.

The framework needs to be **flexible** to allow users to add new simulation models of protocols and configurations easily. For this purpose, we design the simulation framework using modules to provide flexible interface for model users to model their own protocols. Several basic modules serve as the building blocks. These modules provide flexibility to improve the functionality easily.

We also need the simulation framework to be **scalable** to handle large-scale network environments, large relying party populations, large number of certificates, and complicated certificate topologies. For this purpose, we design the simulation framework to allow modeling using different levels of abstratctions.

Given these modeling requirements, we use *SSFNet* [18], the Java-based network simulator, to build the framework. SSFNet is good at modeling large-scale networks. It is able to model protocols at packet level as well as at the higher levels. The *Domain Modeling Language (DML)* [5] provided by SSFNet is a powerful tool to configure a variety of protocol behaviors. Furthermore, the modular design of SSFNet allows us to add more modules for PKI protocols.

The simulation framework models major services by a general-purpose X.509 PKI system. There are five primary components: certificates, storage and retrieval services, CSI services, PKI architectures, and certificate validation services. Each of these components is implemented as an independent module with a flexible interface. The simulation implements basic functionalities. At the high level, the PKI simulation framework consists of four major components—PKI Data, PKI Entities, PKI Protocols, and Network Topology. Each component is a module that provides a set of configurations that allow users to specify the behaviors and parameters.

The ***PKI Data*** module specifies data forms used in a PKI system. We implement certificates and CRLs. The PKI Data module provides a flexible level of abstraction for modeling. It can be as detailed as the certificate/CRL fields defined by X.509 profile. At the other extreme (when model users do not care about the contents in a certificate/CRL at all), the "length" parameter can be used to model the entire data structure.

The ***PKI Entities*** module manipulates PKI data. The module has built-in support for three basic PKI entities—relying parties, CAs, directories.

The *Relying Party* submodule fulfills any task by end entities in a PKI system, including requesting certificate issuance, requesting certificate revocation, retrieving data from a directory, and validating certificates. Relying parties may have a local cache to store the retrieved certificates and CRLs. Furthermore, one relying party submodule can model common features as well as differences of entire relying party population in a PKI domain. Thus one relying party module may represent many relying parties at a time.

The *Certification Authority* submodule models basic functionality by a CA, such as issuing certificates, manipulating data in directories, and validating certificates.

The *Directory* submodule models the database of certificates and CRLs; it supports data access protocols such as LDAP. The database grants read-only privilege to relying parties and full privilege to CAs on their own data. The directory model supports several popular directory attributes: *cACertificate*, *userCertificate*, *crossCertificatePair*, and *CertificateRevocationList*.

The activities or behaviors of PKI entities are configured and controlled by the ***PKI Protocols*** module. We have identified four categories of protocols—issuing certificates, revoking certificates, storing and retrieving certificates, and validating certificates. We have implemented their basic functions In SSFNet, each host contains a *protocol graph* representing the network protocols that are supported by the host. Fig. 1 illustrates typical types of hosts in this framework and typical supported protocols. Advanced protoocls for issuing, revoking, and validating certificates rely on the LDAP client protocol.

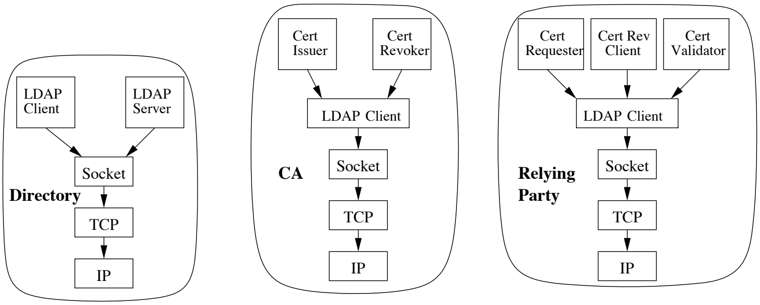


Fig. 1. The demonstration of example protocol graphs for each type of PKI entity. The basic communication protocol is LDAP.

The PKI protocol module models protocol behaviors and produces the resulting performance overhead. In our implementation of LDAP, for instance, the LDAP client can send LDAP requests to corresponding LDAP server to request data. This procedure also produces related performance data such as network latency and the amount of transmitted data.

Finally, all PKI protocols operate with the help of the ***Network Topology*** module. Model users can use DML to configure any type of network topology. We suggest a star-shaped network topology that can be easily scaled to a large number of PKI domains. The network is centered around a number fully connected routers running inter-domain routing protocol, BGP. They establish the “routing core”. The routing core connects all the PKI domains in. Each PKI domain forms a subnetwork with its own administration policies.

Within one PKI domain, users may configure any type of network topology with the choice of PKI-related entities and protocols. For demonstration purpose, we use a simple configuration. In each PKI domain, one directory serves the entire PKI domain. Multiple CAs share this directory. One relying party represents the relying party population in the PKI domain. All PKI entities are directly connected with the border router.

Monitoring and Measurement. In order to measure performance of PKI protocols and activities in the simulation framework, we design a set of monitoring options for monitoring a simulation run. Model users can turn on a subset of the options to observe the desired types of behavior. Current implemented options support five types of events: (1) LDAP states; (2) LDAP data sending and receiving; (3) timer setting and expiration; (4) directory data changes; and (5) message sending and receiving.

Limited by space, we omit[1] the detailed list of monitoring operations in this report. Basically, model users can print output in ASCII form or store it as a binary record. We design the records to cover as much information as possible. Model users can use such measurement data to produce meaningful results, such as the number of requests, the timing of requests, the data size for each request, and the network delay for each request.

[1] Full details can be found in [24].

5 Evaluating Certification Path Discovery

PathBuilder is a special model for evaluating certification path discovery. PathBuilder models the behavior of the algorithm and relies on the PKI simulation framework to perform network activities. This section discusses the design of PathBuilder and presents the performance results.

5.1 PathBuilder Model

In designing the PathBuilder model, we need to take into account several important issues. PathBuilder should be able to model the trials and errors that occur during certification path discovery. Furthermore, PathBuilder should handle large-scale models, a variety of building optimizations, and user activities.

The PathBuilder module is part of the `Cert Validator` protocol, a new protocol model that handles the certificate validation process. In the protocol graph of a host, it resides on top of the LDAP Client. There are four primary modules in the PathBuilder: the Certificate Topology module, the Search Tree module, the Build Options module, and the Monitoring module. The Search Tree module is the central component. The Certificate Topology and Building Options modules configure the behavior of the Search Tree module. The Monitoring module handles the experimental output produced by the Search Tree module.

Certificate Topology. The *Certificate Topology* module is shared by all PathBuilder instances. It configures the complete certificate topology. A PathBuilder instance may configure its own partial view of the certificate topology, which is decided by the local certificate cache of the host.

Search Tree. The *Search Tree* module is the central focus in our design. As we have discussed in Sect, 2, the certification path discovery process is similar to exploring a graph. In fact, we can use a search tree to represent all choices that the algorithm has when traversing the certificate topology. The root is the start point of path building. Each branch in the tree represents a certificate. A candidate certification path (if it exists) is a path in the tree that connects the root with a leaf. The certification path building is the procedure that the algorithm walks in the tree to find this path. On reaching a node in the search tree, the algorithm retrieves certificate information either from the local cache or from the remote directories. The latter case involves LDAP requests and responses, which thus introduce network latency and data transmission overhead.

Following this logic, we model the procedure of building a certification path in four phases: constructing a search tree, assigning probabilities on branches, tree walking with probability, and generating LDAP requests.

In phase one, the model generates a search tree based on the configuration parameters: trust anchors, target, and the building direction. The algorithm may construct a search using a *forward search tree* rooted at the issuer of the target or a *reverse search tree* rooted at the relying party's trust anchors.

In phase two, the model assigns probabilities to each branch in the tree; the probability on a branch represents the likelihood that that certificate is chosen as the next step in the tree walk. Unless we are considering prioritizing the branches, each child branch from an internal node has equal probability to be chosen.

The third phase is the actual tree walking process. This process is the depth-first-search that chooses branches according to their probabilities. At each step, the model randomly chooses a branch based on the assigned probabilities. Available branches have positive probability assignments. Once one branch is chosen, its probability is set to zero so that it won't be considered in the future; consequently, the model needs to adjust the probabilities of the remaining branches to maintain their priority relation. This process ends when a candidate certification path is found or when the entire tree is explored. In the case where multiple candidate certification paths exist, any one of them satisfies the termination condition.

In the last phase, the log of tree walking is sent to the Monitoring module. The model also translates this log into a sequence of LDAP requests, either for CA certificates or for cross-certificate pairs. For the certificates that do not exist in local cache of the relying party, the model passes a request list to the LDAP client protocol, which then executes these requests and produce corresponding performance measurement.

Build Options. Our *Build Options* module handles build options: criteria to distinguish branches and change the way the probabilities get assigned. There are a variety of build options, each of which has its own properties and features. Our analysis indicates that we need to model them case by case. For one example, the CAM implementation requires that certificates matching more RDNs within the issuer DN and the subject DN have priority. We denote this option as the *RDN matching option*. Using the RDN matching option, the model assigns positive probabilities to the branches with the highest matching number. The rest of the branches all have zero probability.

Monitoring. The *Monitoring* module outputs any types of events related to PathBuilder. There are mainly three types of events: (1) search tree statistics, such as tree size, tree height, etc.; (2) LDAP retrieval activities; and)3) performance, such as network latency and amount of data transmission.

5.2 Experiment Configurations

In this section, we present simulation experiments that use the simulation framework and PathBuilder module to evaluate the certification path discovery algorithm. The experiment settings contain a set of configurations of the simulation model and a new protocol module that invokes the certificate path building processes.

Certificate Topology. We design a certificate topology based on both current state and future directions of PKI deployment. To the best of our knowledge, this certificate topology is the first systematic attempt to model the emerging global PKI architecture. It is also the largest simulated PKI architecture model, and expresses the current major efforts in building a bridge-to-bridge environment for PKI systems. The configured PKI architecture models 5 bridge CAs, 51 PKI domains with 103 ordinary CAs, and 30 million certificate users over 13 countries.

The certificate topology for our experiment is illustrated in Fig. 2. We use the four principal bridge CAs (FBCA, HEBCA, SAFE, and CertiPath) as the central piece in our experimental certificate topology. We configure the implemented or prototyped cross-certification relationships between them. We also added the *U.S. Higher Education Root*

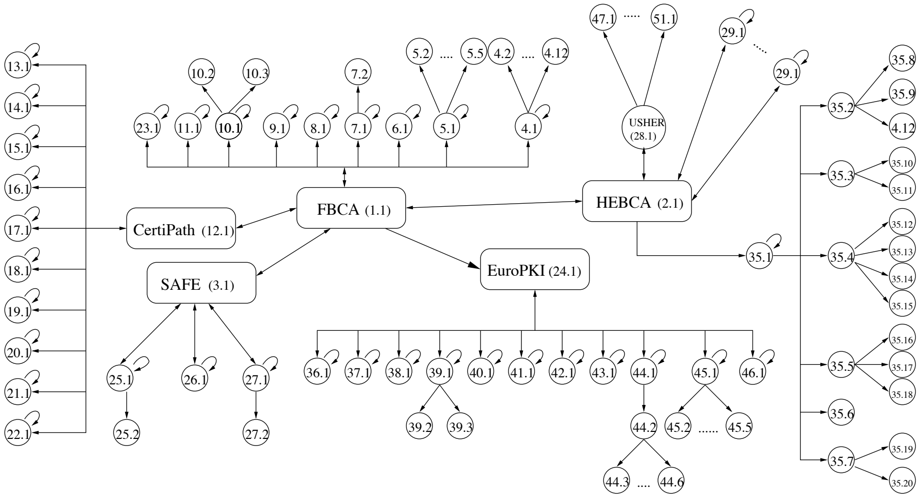


Fig. 2. The configured certificate topology for experiments. The topology is a combination of current deployment and future plans. Each CA with a self-issued certificate can be treated as a trust anchor. The unique ID for each CA is the tuple: (domainID, caID). We assign an index number to each CA in the model for simple implementation.

(USHER) [22], a large sector CA in development. These bridge CAs have cross-certified with many PKI domains for government agencies, institutions, and enterprises. Our configuration is mostly based on the current deployment situation. We have obtained complete data about FBCA and government agency PKI systems that cross-certify with FBCA. For systems that we could not get information for, we approximate each as a simple hierarchy that has one root CA. We use the same strategy to configure architectures of other PKI domains.

Besides PKI systems in the United States, we also try to model the connections to the PKI systems in other countries. EuroPKI [7] is currently a root CA in Europe that connects many PKI systems from several countries. We model it as a bridge CA in our certificate topology to further expand the scale and to predict that PKI systems in Europe may cross-certify with FBCA in the future. We also expand the topology to cover PKI development in South America. The PKI domain number 35 shown in Fig. 2 is the current Brazilian PKI system for all government agencies and enterprises [2]. This PKI system may cross-certify with HEBCA in the future. Finally, the certificate topology is configured with DNs of CAs. They are partially configured using the collected data. The configuration of user population size is based on the combination of measurement data and random assignment.

Configuring PKI Simulation Framework. We use the simplest network configuration to minimize the impact of network protocols on the certification path discovering process. Each PKI domain has one router, one directory, and one relying party sending out certificate validation requests. For these path-building experiments, all certificates are configured statically. As one Relying Party module models the entire relying party population in a PKI domain, we use the configured local preference rate to generate random target certificates for the experiments. Each relying party has one trust anchor—its root CA in local PKI domain.

5.3 Performance Results

In this section, we evaluate performance of certification path discovery by comparing building directions and building options. We conduct the simulation experiments with 10 runs. The standard deviation of experiment results is less than 5%. Thus, the mean value is sufficient for presentation.

Forward vs. Reverse. Table 1 compares the performance by building directions. In terms of search tree properties, the reverse search trees are significantly larger. Experiments show that the average tree size is doubled. And the reverse search trees are flatter according to the path length measurements. Overall, forward search trees are more efficient than reverse search trees. This result is reasonable given that the experimental certificate topology is mostly hierarchical except in the center where bridge CAs are cross-certified with each other. The forward direction encounters only one choice when exploring a hierarchy from a leaf to the root. On the other hand, the reverse direction needs to handle many branches going from the root to a leaf.

Both directions generate similar number of LDAP requests for one target certificate. In some cases, the forward direction fails to retrieve certificates from the cACertificate attribute, then tries to search for issuedToThisCA element of a cross-certificate pair. Thus, one tree walk step may need two LDAP requests. Nonetheless, the forward direction still out-performs the reverse direction. The network latency and the amount of data transmission is smaller for the forward direction.

Table 1. Properties and network performance of the forward search tree vs. reverse search tree

Property	Forward	Reverse
`avg_tree_size`	31.3	69.1
`avg_num_leaf`	26.9	55.9
`max_path_len`	3.9	4.9
`min_path_len`	2.8	2.3
`avg_path_len`	3.6	3.7

Property	Forward	Reverse
# LDAP requests	36.2	40.0
# retrieved CA certs	18.2	0
# retrieved x-cert pairs	81.5	152.8
building delay	7.7 s	9.1 s
data size	89.8KB	122.19KB

Local Preference. In this set of experiments, we vary the local preference rate in the range of 0.2 to 0.9. We found that for both building directions, the performance overheads decrease linearly as the local preference rate increases. This makes sense. Local targets require shorter certification paths. If there is only one CA in the PKI domain, the issuer of the target is the same as the relying party's trust anchor.

Fig. 3 illustrates the performance results for network operations. We notice that the reverse direction leads to slightly more data transmission and longer network latency, although the resulting number of LDAP requests is similar to the forward direction. On average, the reverse direction requires about 16% to 24% more data transmission. The reverse direction relies on retrieving cross-certificate pairs. LDAP server will respond

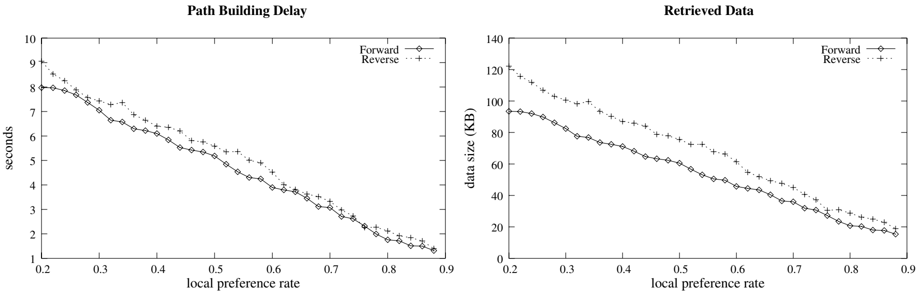


Fig. 3. Network performance by each certification path building process

with all certificates issued to and issued by a CA. In general, the returned amount is larger than retrieving data using only cACertificate attribute.

RDN Matching Option. Next, we examine how the RDN matching option helps to improve performance, especially for the reverse direction. Limited by our collected data, the certificate topology does not have a configured distinguished name for every entity. We thus assume that the RDN match value is zero if there is no DN for either the issuer or the subject.

Fig. 4 shows the impact of the RDN matching option. The RDN matching option helps in reducing the number of retrieved cross-certificate pairs for the reverse direction. The average 11% improvement suggests that the RDN matching option helps the reverse direction by avoiding some CAs with a large number of branches. Thus, the algorithm spends less time in exploring hierarchies from the root to the target. However, the RDN matching option does not reduce the amount of data transmission significantly. The amount of data in each cross-certificate pair retrieval response is still a leading factor. On the other hand, the RDN matching option has no noticeable impact on path building in the forward direction. The forward direction only encounter branch choices when dealing with bridge CAs. These CAs typically have completely different DNs and RDN elements. The RDN matching option cannot reduce the number of choices.

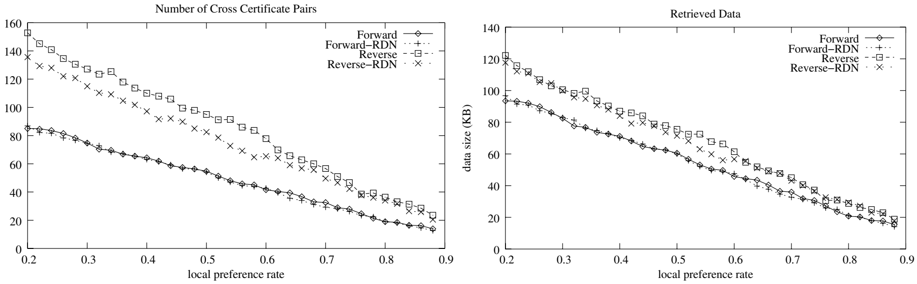


Fig. 4. Performance with RDN matching optimization. "Forward" and "Reverse" denote tree walks without optimization. "*-RDN" denotes the performance by RDN matching optimization.

Overall, simulation experiments with the RDN matching option have shown that it can help speed up the certification path building process in the reverse direction. The improvement is limited, however; the forward direction is still more efficient.

6 Discussions

Using our simulation models, we have just scratched the surface in understanding the performance of certification path discovery. Yet, we have made some important observations. In this section, we discuss these observations and further make suggestions on efficient certification path discovery.

First, the performance difference from building direction heavily depends on the architecture of certificate topology. The emerging global PKI contains a few bridge CAs and a number of hierarchical PKI systems. This architecture favors the forward direction. In practice, we suggest that the algorithm should use the forward direction as much as possible. To further make the tree walk process more effective, we suggest that relying parties set their trust anchors close to the edge of their local domains.

The hierarchical structure of local PKI domains favors any approach if it explores the local PKI domains bottom to top. We suggest that building the certification path in both directions and meeting in the middle may be the best choice. This approach not only maximally takes advantage of the hierarchies, but also significantly reduces the number of branches to explore when the algorithm is working in the center area of the certificate topology where multiple bridge CAs cross-certify with each other. Starting from both the target and the trust anchor, the algorithm quickly reaches the center area from both directions. At this point, the algorithm has dicovered two neighbor sets that may possibly contain several bridge CAs. By comparing these two neighbor sets, the algorithm may be able to discover the common node or the direct link between them quickly.

How does the algorithm decide when to pause for meeting? There are several approaches. One approach is to examine the DNs. The sudden change of similarity in DNs indicates that the algorithm may have just crossed the boundary of a PKI domain. Or, the algorithm looks for self-issued certificates, typically issued by the root CA of a hierarhical PKI domain. In general, the algorithm has a fairly good sense on when it crosses the boundary.

Second, we observe that a building optimization as simple as the RDN matching option can help improve performance if buiding in the reverse direction. The savings come from the reduced number of cross-certificate pairs retrieved from directories. Besides the RDN matching option, there are many other possible optimizations. In general, if the reverse direction is necessary, any build option that helps reduce the number of choices when exploring the certificate topology can significantly improve the performance. For instance, Elley et al. [6] suggested that name constraints and policy processing are two important optimizations. We expect these optimizations may reduce the network latency as well as the amount of transmitted data.

Lastly, the relying parties' certificate usage patterns significantly affect the performance. The simple criterion of local preference rate shows this difference. We suggest that deployer of the algorithm obtain a good understanding of the certificate usage pattern. If relying parties make frequent requests regarding validating certificates in remote domains, the deployer may need to explore approaches to minimize the performance impacts. For instance, one can choose to carefully deploy certificate caches to store certificates and revocation information as much as possible. We should also try to maintain the maximal availability of these caches to relying parties. Smart organization of the information in the cache can help too. For instance, CoreStreet implemented an online certificate validator that is able to return a sequence of certificates that may lead to the most efficient certification path in Federal PKI systems [13].

7 Conclusions

In conclusion, we use simulation to evaluate performance of certification path discovery. We have implemented a simulation framework suitable for performance studies of general-purpose PKI systems. It provides facilities to model data structures, entities, protocols, and large-scale network environments. Classical X.509 PKI services are implemented in the framework. The flexible interface of this framework enables researchers to evaluate new protocols or services in the simulated environment. We design a novel search tree model to simulate certification path discovery. Probabilistic tree walking is an effective technique to model a variety of algorithm options.

In our performance study, we examined several example algorithm options and their impact on performance. Given the current situation of PKI deployment and our experimental results, we suggest that building certification path in both the forward and reverse directions is the best choice. We also have shown that choosing certificates smartly can help improve algorithm efficiency significantly.

We are just getting started on understanding the performance of PKI services in complicated systems. In the future, we plan to extend the experiments to quantify the performance of the “meet-in-the-middle” approach and to explore more algorithm options, such as name constraints and policy mappings. We will also examine the algorithm in more realistic environments, e.g., varying number of LDAP servers for each domain, allowing the certificates to be issued or revoked dynamically, configuring more trust anchors for each relying party, or allowing relying parties cache some certificates and certificate status information. In the long run, we can use the simulation framework not only for performance evaluation, but also for other purposes, such as risk analysis. The framework can be used to model attacking scenarios and risk management system to help us understand the security of current PKI design.

References

1. André Årnes, Mike Just, Steve Lloyd, and Henk Meijer. Certificate Revocation Performance Simulations. project paper, June 2000.
2. Brazilian Government PKI System. http://www.icpbrasil.gov.br/.
3. CertiPath: Enabling Trusted Communication. www.certipath.com.
4. Certification Path Library (CPL). Cygnacom Solutions. http://www.cygnacom.com/products/index.html#cpl.
5. Domain Modeling Language (DML) Reference Manual. http://www.ssfnet.org/SSFdocs/dmlReference.html

6. Yassir Elley, Anne Anderson, Steve Hanna, Sean Mullan, Radia Perlman, and Seth Proctor. Building Certification Paths: Forward vs. Reverse. In *The 10th Annual Network and Distributed Systems Security Symposium (NDSS'01)*, February 2001.
7. EuroPKI Top Level Certification Authority. `http://www.europki.org/ca/root/en_index.html`.
8. Federal Bridge Certification Authority. `http://csrc.nist.gov/pki/fbca`.
9. Higher Education Bridge Certification Authority (HEBCA)—Transforming Education Through Information Technologies. `http://www.educause.edu/hebca/`.
10. R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC3280, `http://www.ietf.org/rfc3280.txt`, April 2002.
11. John Iliadis, Stefanos Gritzalis, Diomidis Spinellis, Danny De Cock, Bart Preneel, and Dimitris Gritzalis. Towards a Framework for Evaluating Certificate Status Information Mechanisms. *Computer Communications*, 26(16):1839–1850, October 2003.
12. John Iliadis, Diomidis Spinellis, Dimitris Gritzalis, Bart Preneel, and Kokratis Katsikas. Evaluating Certificate Status Information Mechanisms. In *Proceedings of the 7th ACM conference on Computer and Communications Security (CCS'00)*, pages 1–8. ACM Press, 2000.
13. CoreStreet Inc. Distributed Path Validation—Massive Scalability for Federated PKIs. Presentation st FBCA Path Discovery & Validation Working Group, August 2004.
14. Loren M. Kohnfelder. Toward a Practical Public-Key Cryptosystem. bachelor's thesis, Dept. Electrical Engineering, MIT, Cambridge, Mass., 1978.
15. Steve Lloyd. Understanding Certification Path Construction. PKI Forum White Paper, September 2002.
16. Jose L. Muñoz, Jordi Forné, Oscar Esparza, and Miguel Soriano. CERVANTES—A Certificate Validation Test-Bed. In *First European PKI Workshop: Research and Applications (EuroPKI 2004)*, volume 3093 of *LNCS*, pages 28–42, Samos Island, Greece, June 2004. Springer-Verlag.
17. M. Myers, R Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol. RFC2560, `http://www.ietf.org/rfc/rfc2560.txt`, June 1999.
18. Andy T. Ogielski and James H. Cowie. SSFNet: Scalable Simulation Framework - Network Models. `http://www.ssfnet.org`. See `http://www.ssfnet.org/publications.html` for links to related publications.
19. Selwyn Russell, Ed Dawson, Eiji Okamoto, and Javier Lopez. Virtual Certificates and Synthetic Certificates: New Paradigms for Improving Public Key Validation. *Elsevier Computer Communications*, 26:1826–1838, 2003.
20. SAFE Bridge Certification Authority TEST Environment. SAFE-BioPharma Association, `http://www.safe-biopharma.org/`.
21. MitreTek Systems. Certificate Arbitrator Module. `http://cam.mitretek.org/cam/`.
22. USHER: The Root Certificate Authority for Trust in Higher Education Research and Education. `http://usher.internet2.edu`.
23. M. Wahl, T. Howes, and S. Kille. Lightweight Directory Access Protocol (v3). RFC2551, `http://www.ietf.org/rfc/rfc2551.txt`, March 1997.
24. Meiyuan Zhao. *Performance Evaluation of Distributed Security Protocols Using Discrete Event Simulation*. PhD thesis, Dartmouth College, Hanover, NH, October 2005. TR2005-559.

Distributing Security-Mediated PKI Revisited *

Jong-Phil Yang[1], Kouichi Sakurai[1], and Kyung Hyune Rhee[2]

[1] Graduate School of Information Science and Electrical Engineering,
Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-0053, Japan
{bogus, sakurai}@itslab.csce.kyushu-u.ac.jp
[2] Division of Electronic, Computer and Telecommunication Engineering,
Pukyong National University, 599-1, Daeyeon3-Dong, Nam-Gu,
Pusan 608-737, Republic of Korea
khrhee@pknu.ac.kr

Abstract. The SEM approach to PKI offers several advantages, such as immediate revocation of users' signing ability without CRLs and compatibility with the standard RSA. However, it has a weakness against denial of service attack caused by breaking down or being compromised. G. Vanrenen et al. proposed a distributed SEM approach to overcome the weakness. However, it does not provide the desirable properties such as instant availability and immunity against denial of service attack, due to inadequate usage of threshold cryptography and proactive secret sharing. In this paper, we point out its structural shortcomings and propose a modified version.

Keywords: Certificate Status Information, Reliability, Fault-tolerance.

1 Introduction

Without doubt, the promise of public key infrastructure (PKI) technology has attracted a significant amount of attention in these days. The IETF PKIX Working Group is developing the Internet standards to support an X.509-based PKI. A certificate is a digitally signed object binding a set of attributes to a public key. The correctness of the trust decisions a relying party makes depends on the assumption that the entity knowing the matching private key possesses those properties. When this binding ceases to hold, this certificate needs to be revoked, and this revocation information needs to propagate to relying parties, lest they make incorrect trust judgments regarding that public key. There are well-known standard mechanisms to solve the revocation of the certificate such as Certificate Revocation List (CRL) and Online Certificate Status Protocol (OCSP), and non-standard mechanisms such as delta CRL, indirect CRL, Certificate Revocation Tree (CRT) and Certificate Revocation System (CRS) [4][8].

* This work was partially supported by IT Scholarship Program supervised by Institute for Information Technology Advancement (IITA) & Ministry of Information and Communication (MIC) in Republic of Korea, Grant No. R01-2006-000-10260-0 from the Basic Research Program of KOSEF, and Strategic International Cooperative Program, Japan Science and Technology Agency (JST).

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 31–44, 2006.

In [3], Boneh et al. proposed a mechanism to fast certificate revocation centered around the concept of an on-line semi-trusted mediator (SEM). The basic idea of SEM is as follows. To sign or decrypt a message, a client must first obtain a message-specific token from its SEM. Without this token, the user cannot accomplish the intended task on the message. To revoke the user's ability to sign, SEM just stop issuing tokens for that user's future request. The SEM approach to PKI offers several advantages like immediate revocation of users' signing ability without Certificate Revocation Lists (CRLs) and compatibility with the standard RSA. However, it has a weakness against denial of service attack caused by breaking down or being compromised. To overcome the weakness, G. Vanrenen et al. proposed a distributed SEM approach based on threshold cryptography and proactive secret sharing [5]. However, it does not provide the desirable properties such as instant availability and immunity for denial of service attack, because of inadequate usage of threshold cryptography and proactive secret sharing.

Our Contributions
This paper introduces firstly the structural shortcomings of G. Vanrenen et al's proposal according to the following topics.

- Efficiency and meaning of performing a proactive secret sharing.
- Immediacy of the distributed SEM approach.
- Specifying the number of servers in the distributed SEM approach.

Then, we derive new requirements to address the shortcomings of G. Vanrenen et al.'s proposal and to design a modification of the distributed SEM approach. We introduce additionally two new cryptographic tools to satisfy new requirements. Finally, we design a modification of the distributed SEM with respect to the new requirements. Our modification has the following benefits: *removal of both insecurity and ambiguity, efficient and timely signing or decrypting, strong against denial of service attack and meaningful proactive secret sharing with the simplified procedure.*

Organizations
The remainder of this paper is organized as follows. Section 2 reviews the original SEM approach and the distributed SEM approach. We discuss notable problems of the distributed SEM and present requirements for designing a modified version in section 3. We present two cryptographic tools and design a modification of the distributed SEM in section 4. Section 5 discusses the security and the desirable properties of our modification. We conclude in section 6.

2 Related Work

2.1 SEM: Semi-trusted Mediator

In [3][17], the SEM system is based on a variant of RSA called as mediated RSA (mRSA). As in RSA, each user has a public key (e, N) and the corresponding private key (d, N), where the modulus N is the product of two large prime p and q, $gcd(e, \phi(N)) = 1$ and $d \times e = 1 \ mod \ \phi(N)$. The public key of a user is the same as in the standard RSA. However, the two parts of a user's private key are d_{sem} and d_{user}, where $d = d_{sem} + d_{user} \ mod \ \phi(N)$. d_{user} is the part held by the user and d_{sem} is the part held by the SEM. Since SEM must not know d_{user} and the user must not know d_{sem}, it is necessary to change RSA key setup procedure. That is, a Certification Authority (CA) generates the private key d instead of the user, chooses a random integer d_{sem} in $[1, N]$, and computes the last value as $d_{user} = d - d_{sem} \ mod \ \phi(N)$. Because the private key d is split into two halves, private key operations require the participation of both the user and SEM; each party raises the message to its half-exponent and the results are then multiplied. The SEM approach provides several advantages such as *compatibility with the standard RSA*, *immediate revocation of users' signing ability* and *no need for CRLs.*

2.2 Distributing Security-Mediated PKI

In [5], G. Vanrenen et al. introduced disadvantages concerned with scalability. Since a user's d_{sem} lives on exactly one SEM, the following problems are inevitable; *temporary denial of service* if the network is partitioned, *permanent denial of service* if SEM suffers a serious failure and inability to revoke the key pair if an adversary compromises SEM and learn its secrets. To address the problems mentionded, G. Vanrenen et al. proposed a distributed SEM (DSEM) network which acts as SEM. DSEM consists of trustworthy *islands* distributed in Peer-to-Peer (P2P) network. An individual island may still become compromised and reveal its data to the adversary. It may also become unavailable, due to crash or partition. To handle these scenarios, they built *migration* scheme based on threshold cryptography and proactive secret sharing.

Key Setup

Fig. 1 shows key setup procedure in DSEM. Each island acts as a SEM. A CA generates a key pair for a user and splits d into two halves. It transmits d_{user} to the user and d_{sem} to an island L. Then, it shares additionally d_{sem} to k islands in the network using threshold cryptography. After those steps are completed, d_{sem} is stored both on the primary island L and on k other islands, so an attacker must either compromise L or compromise t of the k islands in order to get d_{sem}.

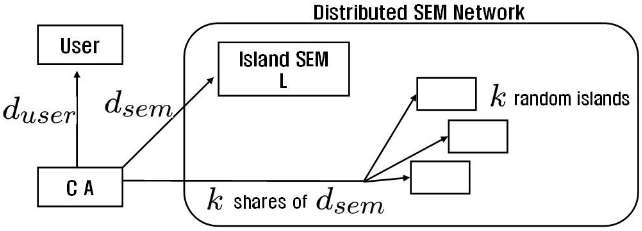


Fig. 1. Key setup in DSEM

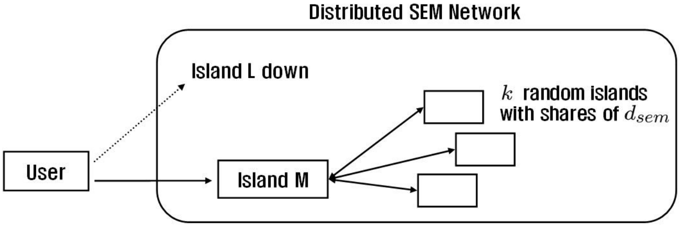


Fig. 2. Migration in DSEM

Additionally, they mentioned that the shares can be proactively updated using proactive secret sharing schemes in [1][2][19][20].

Migration
If a user issues a request but the island L holding d_{sem} is not available, then the user selects another island M and requests *migration*. Fig. 2 shows migration procedure.

(Step 1). The user connects to another island M instead.
(Step 2). To guarantee strong forward security, the island M generates a new δ in a range $[-r, r]$, and changing d_{sem} to $d_{sem} - \delta$, where r is big enough to keep the key halves changing unpredictably, but small enough to be smaller than d_{sem} and d_{user} for a practically indefinite number of rounds.
(Step 3). M sends δ to the user. Then, the user replaces d_{user} with $d_{user}+\delta$. M splits δ into k shares and sends each to the corresponding d_{sem} shareholder island. Each shareholder island uses its piece to update its share.
(Step 4). Finally, migration is completed and M can then fulfill the user's request.

For M to reconstruct d_{sem} in (Step 2), it must know $\phi(N)$ to interpolate a polynomial which was used to perform (k,t)-secret sharing for d_{sem} in key setup. Moreover, they did not specify the value r.

3 Notable Problems

In this section, we question several inadequate system operations in DSEM and introduce five requirements to address the shortcomings of G. Vanrenen et al.'s proposal.

Question 1. *How can we make k islands perform efficiently a proactive secret sharing ?*
After key setup procedure, k islands periodically participate in a proactive secret sharing to renew periodically their shares for d_{sem} by using the schemes in [1][2][19][20]. However, the schemes in [1][2] cannot be adopted to DSEM, because they are based on discrete logarithm (i.e., the modulus operators different from $\phi(N)$ in mRSA are public). Moreover, the scheme in [20] must use

$\lambda(N) = lcm(p-1, q-1)$ instead of $\phi(N)$ to make the system *proactive*. Then, the scheme in [19] can be used in DSEM from the viewpoint of modulus operator. However, DSEM cannot avoid lots of consumption of system resources because it must perform subsharings as many times as the number of shares; each instance of subsharing requires lots of consumption of system resources. We hope to perform an instance of proactive secret sharing without subsharings if possible.

Question 2. *Is DSEM always performed as efficient as SEM ?*
In case that the scheme in [15] or [20] is used for threshold protection, we can image the following bad situation. A user's d_{sem} may be shared among k islands by using (k,k)-additive secret sharing. Let a user A's primary island be L_A and a user B's primary island be L_B. Then, k shareholder islands of A's d_{sem} consist of $L_{A1}, L_{A2}, \ldots, L_{Ak}$. We assume that L_B is L_{A4} and both L_A and L_B are eventually compromised at the same time. Then, the following procedure is performed for A to migrate from L_A to M_A successfully:

1. At least t out of $k-1$ island, $L_{A1}, L_{A2}, L_{A3}, L_{A5}, \ldots, L_{Ak}$, collaboratively recover the share of L_{A4} by performing an instance of the polynomial interpolation of (k,t)-polynomial secret sharing. To do so, the system must consume lots of system resources to perform a verifiable recovery protocol similar to the schemes in [9][16][19][20].
2. After that, M_A must perform (Step 2) and (Step 3) of migration procedure in section 2.2.
3. Of course, B must migrate from L_B to M_B. However, if L_A is L_{B5} (i.e., one of the shareholders of B's d_{sem}), the migration procedure should be more complex.

The main objective of DSEM is to make SEM instantly applicable and scalable. Nevertheless, DSEM cannot present cryptographic operation services such as signing or decrypting before finishing the complex procedure mentioned above.

Question 3. *Is the execution of the proactive secret sharing meaningful ?*
In DSEM, a user's d_{sem} is stored in the primary island L and shared among k islands. To make shares in k islands robust against adversaries, DSEM performs a proactive secret sharing among k islands. Since a long-term secret d_{sem} is stored in L, the target of adversaries is not one of k islands but L. That is, since the long-term secret d_{sem} is kept in the networking island and the proactive secret sharing does not change it, a proactive secret sharing cannot contribute to the security of d_{sem}.

Question 4. *How many peers are necessary to serve a threshold protection in DSEM ?*
Let us consider (k,t)-secret sharing. In the synchronous communication model, the system allows at most $t-1$ servers to be compromised by an adversary, and needs at least t servers to be correct. That is, k must be greater than or equal to $2t-1$, and at least t server must be available. Since DSEM assumes P2P network, we must consider an inherent property of P2P network such that correct peers

in P2P are not always connected to the network. Moreover, an island which acts as a primary island for specific users is also a peer in P2P. So, we must precisely specify the number of all islands to prevent a user from performing frequent migration because of simple power down of the primary island without being compromised.

To sum up, we devise five requirements to design our modified DSEM as follows:

R.1 The system must accomplish efficiently a RSA signing/encypting and a proactive secret sharing without releasing $\phi(N)$. There must be no ambiguity of the value of r.

R.2 To reduce the overhead caused by subsharing, the system must perform a proactive secret sharing without subsharing.

R.3 DSEM must be modified to make a RSA signing/encrypting immediate. That is, the cryptographic service must be independent of migration.

R.4 Through all of d_{user}, d_{sem} and k shares for d_{sem} are periodically renewed at the same time, we can make the execution of the proactive secret sharing meaningful in DSEM.

R.5 Let Δ be the maximum number of correct peers which are not currently connected to the network. We precisely define the number of servers as $k + \Delta$, where $k = 2t - 1$. So, we must perform $(k + \Delta, t)$-secret sharing to serve a successful threshold protection in the stateless model such like P2P.

4 Our Modified DSEM

4.1 Cryptographic Tools

In this section, we introduce three cryptographic tools used in our modified DSEM.

N-mRSA

To remove the ambiguity of the value of r in migration procedure and the insecurity of releasing $\phi(N)$, for the future use such as a polynomial interpolation (i.e., to satisfy **R.1** in section 3), we propose a modified version of mRSA based on threshold RSA signature scheme in [6].

(Key setup). A CA generates a private key d based on the standard RSA. Then, the CA splits the private key into two halves by using $d = d_u^N + d_s^N \ mod\ N$. It securely transmits d_u^N to the user, and d_s^N to the server.

(Signing). To sign a message m, the user sends m to the server. Then, the server computes $PS_s = m^{d_s^N} \ mod\ N$ and returns it to the user. The user computes $PS_u = m^{d_u^N} \ mod\ N$ concurrently. On receiving PS_s, the user computes a candidate signature CS as follows:

$$CS = PS_s \cdot PS_u = m^{d_s^N} \cdot m^{d_u^N} = m^{t \cdot N + d} \ mod\ N,$$

where $0 \leq t < 2$. Finally, the user applies CS to *2-bounded coalition offsetting algorithm* in [6], and computes a valid signature on m.

Combinatorial Secret Sharing

In [7], L. Zhou et al. proposed *combinatorial secret sharing*; we use a function $CSS(k,t,x)$ for describing (k,t)-combinatorial secret sharing, where x is a secret. To avoid confusion, they used *share sets* to denote shares of a secret x by using a combinatorial secret sharing and used *shares* of x only for the values comprising a standard secret sharing. We can construct share sets, one for each server, through the following steps. To simplify the description, we use abstract modulus operator.

(Step 1). Create $l = \binom{k}{t-1}$ different sets $P_1, \ldots, P_l$ of servers. These sets of servers represent the worst-case *failure scenarios*: sets of servers that could all fail under the assumption that at most $t-1$ servers are compromised.

(Step 2). Create a sharing $\{s_1, \ldots, s_l\}$ using (l,l) additive secret sharing scheme. Associate share s_i with failure scenario P_i.

(Step 3). Include secret share s_i in S_p, the share set for a server p, if and only if p is not in corresponding failure scenario P_i. That is, for any server p, share set S_p equals $\{s_i | 1 \leq i \leq l \ \wedge \ p \notin P_i\}$. Note that, by not assigning s_i to any server in a failure scenario P_i, they ensure that servers in P_i do not together have all l shares to reconstruct the secret x. For any set P of servers, the constructed share sets satisfy the following conditions:

- **Condition 1** : $\bigcup_{p \in P} S_p \ = \ \{s_1, s_2, \ldots, s_l\}$, where $|P| \geq t$.
- **Condition 2** : $\bigcup_{p \in P} S_p \ \subset \ \{s_1, s_2, \ldots, s_l\}$, where $|P| \leq t-1$.

Server-Assisted Threshold Signature

In [14], S. Xu et al. proposed a formal method to construct server-assisted threshold signature schemes. It is based on the hybrid of threshold signature schemes and two-party signature schemes such as [3][10]. In this paper, we propose a practical instance based on N-mRSA and threshold RSA signature scheme in [6].

(Assumption). There are k servers which store securely shares of secret information in the system.

(Key setup). A CA generates a private key d based on the standard RSA. Then, the CA splits the private key into two halves by using $d = d_u^N + d_s^N \ mod \ N$. The CA performs $CSS(k,t,d_s^N)$ and generates k share sets. Then, the CA transmits securely d_u^N to the user, and each share set to the corresponding server, respectively.

(Signing). To sign a message, the user broadcasts m to the servers. Then, at least t servers compute $PS_s \ = \ m^{t_1 \cdot N + d_s^N} \ mod \ N$ collaboratively, where $l = \binom{k}{t-1}$ and $0 \leq t_1 < l$. The user computes $PS_u \ = \ m^{d_u^N} \ mod \ N$ concurrently. On receiving PS_s, the user computes a candidate signature CS as follows:

$$CS = PS_s \cdot PS_u = m^{t_1 \cdot N + d_s^N} \cdot m^{d_u^N} = m^{t \cdot N + d} \ mod \ N,$$

where $0 \leq t < l+1$. Finally, the user applies CS to *(l+1)-bounded coalition offsetting algorithm* in [6], and computes a valid signature on m.

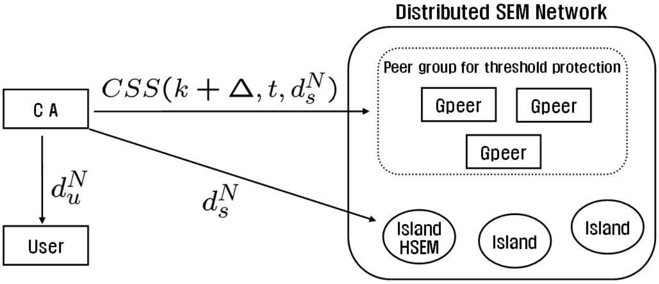


Fig. 3. Architecture and Key setup in our modified DSEM

Under two proposed signature schemes above, we can compute RSA signatures and perform proactive secret sharing without insecurity of releasing $\phi(N)$. By using combinatorial secret sharing, we can design our modified DSEM without computing a polynomial.

4.2 Architecture

Fig. 3 shows our modified DSEM. There is a peer group (PG) which consists of $(k+\Delta)$ peers, where $k = 2t-1$ and each peer in PG is called as Gpeer. Each Gpeer has *share set*s for all users' d_s^Ns. Since PG must consist of trustworthy peers, we can depend on *reputation techniques* in P2P for pre-selection of such peers. Each user registers at a single island. The island becomes home SEM (shortly, HSEM) for the user and possesses d_s^N for the user. In this paper, we do not restrict the number of HSEMs in P2P network. The five protocols for our modified DSEM are proposed:

- Key setup : an initial setup of a user' private key in our modified DSEM.
- Signing : the user signs a message via N-mRSA.
- Periodical renewal : all secret values for the user are periodically renewed.
- Recovery of compromised islands : a compromised HSEM is recovered.
- Recovery of compromised Gpeer : a compromised Gpeer is recovered.

4.3 Protocol Details

This section shows the detailed procedures of five protocols mentioned in the previous section.

Key Setup

A CA generates a private key d based on the standard RSA. Then, the CA splits the private key into two halves as N-mRSA. It transmits d_u^N to the user and d_s^N to HSEM for the user, respectively. Then, it shares d_s^N among $(k+\Delta)$ Gpeers by $CSS(k+\Delta, t, d_s^N)$. Our modified DSEM depends on **R.5** in section 3, and the value of Δ is system-wide constants.

Signing

To sign a message m, a user sends m to HSEM. Then, both the user and HSEM perform signing procedure in N-mRSA. Finally, the user can obtain a valid signature on m.

Periodic Renewal

To renew periodically d_u^N, d_s^N and all *share set*s for a specific user, the system performs the following steps.

(Step 1). A user generates a new δ in the range $[-N, N]$ and performs $CSS(k+\Delta, t, \delta)$ to compute $k+\Delta$ share sets. The user transmits securely each share set to the corresponding Gpeer, respectively. After that, the user computes a renewed half key as $(d_u^N)^{new} = d_u^N - \delta$.

(Step 2). When each Gpeer receives a share set for δ, it adds shares in the received share set to shares in the current share set, respectively. The share sets newly generated can be used to generate the renewed half key as $(d_s^N)^{new} = d_s^N + \delta$. Then, at least t out of $k+\Delta$ Gpeers send securely their renewed share sets for $(d_s^N)^{new}$ to HSEM for the user.

(Step 3). On receiving at least t share sets, HSEM can combine them and compute the renewed half key $(d_s^N)^{new}$. After that, HSEM generates a random string rs and computes $challenge = rs^{(d_s^N)^{new}}\ mod\ N$. Then, HSEM sends both rs and *challenge* to the user.

(Step 4). The user computes $response = rs^{(d_u^N)^{new}}\ mod\ N$ and combines it with *challenge* for generating a RSA signature for rs. After that, the user checks the validity of the RSA signature. If the result is successful, the user sends *success notification* and *response* to HSEM and replaces d_u^N with $(d_u^N)^{new}$. Otherwise, the user sends *error notification* to HSEM.

(Step 5). If HSEM receives *success notification* from the user, it also checks the validity of *response* through the same way that the user performed. If the result is successful, HSEM replaces d_s^N with $(d_s^N)^{new}$ and finishes the procedure. Otherwise, it accuses Gpeer, who sent an erroneous share set, to PG by broadcasting an *accusation message*. Then, HSEM tries to perform (Step 3) again by using another shares in the received share set.

By using a simple challenge/response, the periodic renewal can be verifiably performed. During the run of periodic renewal, the user can perform signing operation with old keys (i.e., d_u^N and d_s^N). Our modified DSEM satisfies therefore **R.3** in section 3. As you have seen, the periodic renewal depends on both **R.2** and **R.4**.

Recovery of Compromised SEM

For successful recovery of a compromised SEM from adversaries, the system performs the following steps.

(Step 1). To sign a message m, a user sends it to HSEM and requests a partial signature PS_s.

(Step 2). During the specific time bound, if the user does not receive PS_s from HSEM, the user broadcasts m and an *accusation message* to PG. Then, the user can compute a valid signature on m by using the *server-assisted threshold signature* between the user and PG.

(Step 3). If the number of accusation messages from users exceeds a specific limit based on the system policy, HSEM is rebooted and initialized by clean copy.

(Step 4). At least t Gpeers send securely their share sets to HSEM. Then, HSEM can obtain l shares to reconstruct d_s^N.

(Step 5). After that, the user performs *periodic renewal* mentioned before.

During the recovery of the compromised SEM, the user also performs cryptographic operation like signing or decrypting via server-assisted threshold signature scheme in Section 4.1, although the performance is lower than N-mRSA. Therefore, our modified DSEM also satisfies **R.3** in spite that HSEM is compromised.

Recovery of Compromised Gpeer

In (Step 5) of periodic renewal, whenever every Gpeer in PG receives accusation messages exceeding a specific limit from island SEMs, the system performs the recovery procedure for the compromised Gpeer.

(Step 1). The accused Gpeer is rebooted and initialized by clean copy.

(Step 2). Each Gpeer except the accused Gpeer sends shares, which are owned by both each Gpeer and the accused Gpeer, to the accused Gpeer, respectively.

(Step 3). Then, the accused Gpeer can reconstruct share set for itself. If the accused Gpeer wants to verify the validity of the reconstructed share set, given for a challenge message m, both HSEM and the accused Gpeer check the equivalence of a generated partial signature on m by collaborating with another $t-1$ Gpeers.

(Step 4). The user is recommended to perform *periodic renewal*. However, the user will perform *periodic renewal* in the near future without the recommendation.

During the proactive activities such as periodic renewal and two recovery protocols in our modified DSEM, the system does not perform subsharing for each share. That is, our modified DSEM satisfies **R.2**.

5 Discussion

In this section, we discuss the security of our proactive scheme (i.e., periodic renewal and two recovery protocols) in section 4.3 and the notable features of our modified DSEM.

5.1 Security of Proactive Scheme

Now, we discuss the security of our proactive scheme: periodic renewal and two recovery protocols. For the security of both N-mRSA and an instance of server-assisted threshold signatures in section 4.1, please refer to [6][14]. We assume *short-term constrained adversary* introduced in [6] to characterize adversary; given that time is divided into periods, the adversary cannot break t or more servers during any time period, where the number of server is $k + \Delta$. Now, we show simply that our proactive scheme satisfies the following properties.

- **Independency:** *New shares for the secret cannot be combined with old shares to reconstruct the secret.*

  After a user generates a random value, δ, in (Step 1) in Periodic renewal, the random value can be used to renew shares for both d_u^N and d_s^N. In contrast that the existing proactive secret sharings used in the original DSEM do not change d_s^N but shares for d_s^N, our periodic renewal changes/renews the secret itself (i.e., d_s^N). So, the shares in PG during a time period can be only used to reconstruct d_s^N in the time period. Therefore, an adversary who even knows d_s^N in a time period without keeping corruption of at least t servers (i.e., who succeeds in corrupting HSEM of the user) cannot know the newly renewed d_s^N in the next time period. Our proactive scheme satisfies therefore *Independency.*
- **Secrecy:** *The secret remains unknown to adversaries.*

  To show Secrecy, it suffices to show that an adversary cannot obtain all l shares by corrupting at most $t-1$ Gpeers in a time period. Let P be the set of Gpeers corrupted in a time period, $|P| \leq t-1$ holds. Due to **Condition 2** in the section 4.1, there is at least one share which the adversary cannot obtain. Our proactive scheme satisfies therefore *Secrecy.*
- **Availability:** *Correct servers together have sufficient shares of the secret to reconstruct it.*

  To show Availability, it suffices to show that correct servers can reconstruct d_s^N. In a time period, there are at least t correct Gpeers in PG connected in the network because of **R.5**. Let P be the set of correct Gpeers connected in the network, $|P| \geq t$ holds. Due to **Condition 1** in the section 4.1, correct Gpeers can collect l shares for reconstructing d_s^N. So, the correct Gpeers can perform recovery of compromised SEM or Gpeer. That is, our proactive scheme satisfies *Availability.*

In [12], S. Jarecki et al. introduced the weakness of proactive scheme of threshold RSA in [6]. The scheme in [6] is a basis of cryptographic tools in section 4.1. However, since our proactive scheme does not depend on subsharing unlike the scheme in [6], an adversary in [12] cannot succeed in learning the private exponent d; i.e., the adversary can learn at most $lg(k + \Delta)$ most significant bits(MSBs) of d during the entire life-time. So, we do not need to consider the weakness introduced in [12].

5.2 Notable Features

Our modified DSEM has the same features as the original SEM, because it succeeds to the advantages of the original SEM. Moreover, our modified DSEM solves the problems mentioned in section 3 with the following desirable properties.

- **Removal of both insecurity of releasing $\phi(N)$ and ambiguity of the value of r**
  Our modified DSEM adopted three cryptographic tools which are based on N for modular operator of RSA exponent.
- **Efficient and timely signing or decrypting**
  In the DSEM, the user cannot perform singing or decrypting until the migration is finished. On the other hand, the user can still perform signing or decrypting via server-assisted threshold signature, in spite that either periodic renewal or recovery is under way in our modified DSEM. That means the capability for signing and decrypting is independent of the executions of periodic renewal and recovery.
- **Strong against denial of service attack**
  Our modified DSEM is strong against denial of service attack by using an alternative operation (i.e., server-assisted threshold signature), although the performance is lower than N-mRSA. That is, the user can still perform signing or decrypting in spite that the user's HSEM is compromised.
- **Meaningful proactive secret sharing**
  In contrast to DSEM, our modified DSEM can appropriately renew a user's half, the corresponding half of SEM, and shares for the half of SEM per time period for renewal.
- **Simplified renewal and recovery**
  In DSEM, they used well-known proactive secret sharing schemes such as [19][20] to perform periodic renewal, recovery of a compromised island or migration. The schemes referred must require lots of system resources to perform subsharing and verifiable secret sharing. However, since our modified DSEM does not require any subsharing and verifiable secret sharing, it consumes the minimized system resources. Such the simplified renewal and recovery can be achieved by adopting combinatorial secret sharing, user intervention and simple challenge/response.

R. Sandhu et al. introduced a password based two-party signature scheme called as virtual smartcard to build Password-Enabled PKI [11]. Unlike mRSA, × operator is adopted to split the private d; i.e., $d = d_{sem} * d_{user}\ mod\ \phi(N)$, where d_{user} is derived from the user defined password. In this way, the signing procedure between the user and SEM must be sequential. However, the signing procedure in mRSA can be performed in parallel. In [18], X. Wang pointed out the vulnerability to the dictionary attack in the virtual smartcard, and proposed an intrusion-tolerant virtual smartcard. To initialize the system, the private key is divided into two parts as mRSA; the part held by the user is derived from the user's password, and the other part is secretly shared among k servers by using (k,t)-polynomial secret sharing scheme. Therefore, the signing of the intrusion-tolerant virtual smartcard can be regarded as an instance of server-assisted threshold signature scheme. However, X. Wang did not present the proactive activities (i.e., periodic renewal and periodic recovery) in [18].

In [13], S. Koga et al' proposed a solution to prevent denial of service attack by picking out malicious users' requests though one-time ID. Since their solution did not consider the possibility of the corruption of SEM, it did not present a solution for recovering the compromised SEM. Nevertheless, we believe that S. Koga et al.'s proposal can be used for supporting authentication of users' requests in our modified DSEM.

6 Conclusion

In this paper, we reviewed G. Vanrenen et al.'s distributed SEM approach and proposed a modified model. Additionally, we presented two new cryptographic tools to design our model. Our modified DSEM succeeds to the advantages of the original SEM and also provides desirable features.

References

1. A. Herzberg, M. Jakobsson, S. Jarechi, H. Krawczyk and M. Yung, "Proactive public key and signature systems," ACM Conference on Computer and Communications Security, pp.100-110, 1997.
2. A. Herzberg, S. Jarecki, H. Krawczyk and M. Yung, "Proactive secret sharing or: How to cope with perpetual leakage," Advanced in Cryptology-CRYPTO 95, pp.339-352, 1995.
3. D. Boneh, X. Ding, G. Tsudik and C.M. Wong, "A method for fast revocation of public key certificates and security capabilities," 10th USENIX Security Symposium, pp.297-308, 2001.
4. C. Adams and S. Lloyd, "Understanding public-key infrastructure: concepts, standard, and deployment considerations," Indianapolis: Macmillan Technical Publishing, 1999.
5. G. Vanrenen and S.W. Smith, "Distributing Security-Mediated PKI," 1st European PKI Workshop Research and Applications, pp.213-231, 2004.
6. H. Luo and Songwu Lu, "Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks," UCLA Computer Science Technical Report 200030, 2000.
7. L. Zhou, "Towards Fault-Tolerant and Secure On-line Services," PhD Dissertation, Department of Computer Science, Cornell University, Ithaca, NY USA, 2001.
8. M. Naor and K. Nissim, "Certificate revocation and certificate update," 7th USENIX Security Symposium, pp.217-228, 1998.
9. P. Felman, "A Pracitcal Scheme for Non-Interactive Verifiable Secret Sharing," 28th Annual IEEE Symposium on Foundations of Computer Science, 1987.
10. P. MacKenzie and M. Reiter, "Networked Cryptographic Devices Resilient to Capture," IEEE Security and Privacy 01, pp.21-25, 2001.
11. R. Sandhu, M. Bellare and R. Ganesan, "Password-Enabled PKI: Virtual Smartcards versus Virtual Soft Tokens," 1st Annual PKI Research Workshop, pp.89-96, 2002.
12. S. Jarecki, N. Saxena and J. H. Yi, "An Attack on the Proactive RSA Signature Scheme in the URSA Ad-Hoc Network Access Control Protocol," ACM Workshop on Security of Ad Hoc and Sensor Networks, pp.1-9, 2004.
13. S. Koga, K. Imamoto and K. Sakurai, "Enhancing Security of Security-Mediated PKI by One-time ID," 4th Annual PKI R&D Workshop, pp.176-189, 2005.
14. S. Xu and R. Sandhu, "Two Efficient and Provably Secure Schemes for Server-Assisted Threshold Signatures," CT-RSA, pp.355-372, 2003.

15. T. Rabin, "A Simplified Approach to Threshold and Proactive RSA," Advanced in Cryptology-CRYPTO 98, pp.89-104, 1998.
16. T. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," Advanced in Cryptology-CRYPTO 91, pp.129-140, 1991.
17. X. Ding, D. Mazzocchi and G. Tsudik, "Experimenting with server-aided signtures," Network and Distributed Systems Security Symposium, 2002.
18. X. Wang, "Intrusion-Tolerant Password-Enabled PKI," 2nd Annual PKI Research Workshop Proceedings, pp.44-53, 2003.
19. Y. Frankel, P. Gemmell, P.D. MacKenzie and M. Yung, "Optimal resilience proactive public key cryptosystems," IEEE Symposium on Foundations of Computer Science, pp.440-454, 1997.
20. Y. Frankel, P. Gemmell, P.D. MacKenzie and M. Yung, "Proactive RSA," Advances in Cryptology-CRYPTO 97, pp.440-454, 1997.

An Improved Lu-Cao's Remote User Authentication Scheme Using Smart Card

Eun-Jun Yoon and Kee-Young Yoo[⋆]

Department of Computer Engineering, Kyungpook National University,
Daegu 702-701, South Korea
Tel.: +82-53-950-5553; Fax: +82-53-957-4846
ejyoon@infosec.knu.ac.kr, yook@knu.ac.kr

Abstract. In 2005, Lu-Cao proposed an improvement on Hwang-Li's remote user authentication scheme using a smart card that could withstand an impersonation attack, but also it required fewer computational costs. However, the current paper demonstrates that Lu-Cao's scheme has some drawbacks. We present an improved authentication scheme in order to isolate such problems.

Keywords: Authentication, Password, Network security, Smart card.

1 Introduction

A remote password authentication scheme is used to authenticate legitimacy of the remote users over an insecure channel. ISO 10202 standards have been established for the security of financial transaction systems that use integrated circuit cards (IC cards or smart cards). The main characteristics of a smart card are its small size and low-power consumption. In general, a smart card contains a microprocessor which can quickly manipulate logical and mathematical operations, RAM, which is used as a data or instruction buffer; and ROM, which stores the user's secret key and the necessary public parameters and algorithmic descriptions of the executing programs. The merits of a smart card regarding password authentication are its simplicity and its efficiency in terms of the log-in and authentication processes. The main merits of a smart card-based authentication scheme include: (1) there is no password or verification table kept in the remote server; (2) users can freely choose and change their passwords; and (3) lower communication and computation costs. In 1981, Lamport [1] proposed a remote password authentication scheme using a password table to achieve user authentication. In 2000, Hwang-Li [2] pointed out that Lamport's scheme suffers from the risk of a modified password table. Moreover, there is the cost of protecting and maintaining the password table. Therefore, they proposed a new user authentication scheme using smart cards to eliminate risks and costs. Hwang-Li's scheme can withstand replay attacks and can also authenticate users without maintaining a password table. However, there is a weakness in the scheme, as

[⋆] Corresponding author.

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 45–53, 2006.

previously noted [3][4], in that an attacker can easily impersonate other user to log in the system. To overcome such a weakness, Shen et al. [5] proposed a modified version that they claimed it is secure against such attacks. However, Leung et al. [6] showed the weakness still exists in Shen et al.'s scheme.

In 2005, Lu-Cao [7] proposed an improvement on Hwang-Li's scheme that not only could it withstand an impersonation attack, but that required fewer computational costs. Furthermore, it does not require modular exponentiation computations. However, the current paper demonstrates that Lu-Cao's scheme has some drawbacks; that is, the password of a user has to be computed by the system and the scheme has unnecessary computation costs. In general, this cannot satisfy a user's and an authentication scheme's requirements, respectively. To achieve the aim of user friendliness as well as low communication and low computation, we present an improved authentication scheme to the scheme in order to isolate such problems which still achieves the same advantages as Lu-Cao's scheme. The proposed scheme has the following two advantages. First, it is user friendly since a variable-length password can be chosen and changed freely by the user without the help of a remote server, while also providing mutual authentication. Secondly, it is more secure and efficient than Lu-Cao's scheme.

The remainder of this paper is organized as follows: Section 2 briefly reviews Lu-Cao's scheme. Some drawbacks of Lu-Cao's scheme are demonstrated in Section 3. The proposed authentication scheme is presented in Section 4, while Sections 5 and 6 discuss the security and efficiency of the proposed scheme. Our conclusions are given in Section 7.

2 Review of Lu-Cao's Authentication Scheme

This section briefly reviews the Lu-Cao's authentication scheme [7]. Lu-Cao's scheme consists of four phases: an initialization, registration, login, and authentication phase. Figure 1 shows Lu-Cao's authentication scheme. The scheme works as follows:

Initialization Phase: To set up a remote system, the remote server first must prepare the following parameters:

- p, q: two distinct security large primes, satisfying $p \equiv q \equiv 3 \bmod 4$;
- $n : n = p \cdot q$;
- a: a random number in Z_n^*, satisfying $(\frac{a}{n}) = -1$;
- $H(\cdot) : \{0,1\}^* \rightarrow Z_n^*$ is one-way hash function;
- $H_1(\cdot) : \{0,1\}^* \times Z_n^* \rightarrow Z_n^*$ is another one-way hash function.

Then, the remote server can accept the user registration request operation.

Registration Phase: When a new user U_i submits his or her identity ID_i to the remote server for registration. The server does the following:

(1) check the validity of ID_i. If it is valid, the operation will continue, otherwise stop;

(2) use the improved Rabin signature scheme [10] to compute (s^*, c_1, c_2), where $s^{*2} \equiv (-1)^{c_2} \cdot a^{c_1} \cdot H(ID_i)(\mathrm{mod} n)$ and $(\frac{s^*}{p}) = (\frac{s^*}{q}) = 1$;
(3) set the user password $PW_i = s^*$ and store the public parameters $(n, H_1(\cdot))$ to a smart card;
(4) issue the smart card and PW_i to the user via a secure channel.

Login Phase: User U_i attaches his or her smart card to the login device and keys in his or her ID_i and PW_i. Then, the smart card performs as follows:

(1) select a random number $r \in_r Z_n^*$;
(2) compute $c \equiv r \cdot PW_i(\mathrm{mod} n)$;
(3) pick up the current date and time T of the login device;
(4) compute $h = H_1(T, r)$;
(5) send a login message $C = (ID_i, T, c, h)$ to the remote server.

Authentication Phase: Suppose that the remote server receives the message C at T', where T' is the current date and time of the system; then the remote server performs as follows:

(1) check the time interval between T and T'; if $(T' - T) \geqslant \Delta T$, where ΔT is the expected legal time interval for transmission delay, the server will reject the login request;
(2) check the validity of identity ID_i; if the format of ID_i is incorrect, the login request will be rejected;
(3) use the same way in the Step (2) of registration phase to compute the user password PW_i;
(4) compute $r \equiv c \cdot PW_i^{-1} \bmod n$;
(5) check $h \overset{?}{=} H_1(T, r)$. If it holds, the server will accept the login request. Otherwise, the request will be rejected;

3 Drawbacks of Lu-Cao's Authentication Scheme

This section demonstrates the drawbacks of the Lu-Cao's scheme. Based on an improved Rabin signature scheme [10], Lu-Cao proposed an efficient remote authentication scheme using smart cards. Lu-Cao's scheme has the advantages of no password table, low communication and low computation costs. However, the scheme has a disadvantage that the password PW_i of the user must be computed and is assigned by a remote server. The lengthy assigned password PW_i does not satisfy the user's requirement and is also against the habit of the user. If PW_i is very long for the user (e.g. 1024 bit), it is very hard to remember the password PW_i. Thus, it is not user friendly. If the length of PW_i satisfies the user's requirement (e.g. 32 or 64 bits), then Lu-Cao's scheme can be vulnerable to the off-line password guessing attacks [8], where an attacker can easily guess a legal users' password and impersonate the legal user. We consider the off-line password guessing attacks on Lu-Cao's scheme as below. In their scheme, an attacker can record the login message $C = (ID_i, T, c, h)$ as a verifiable-text.

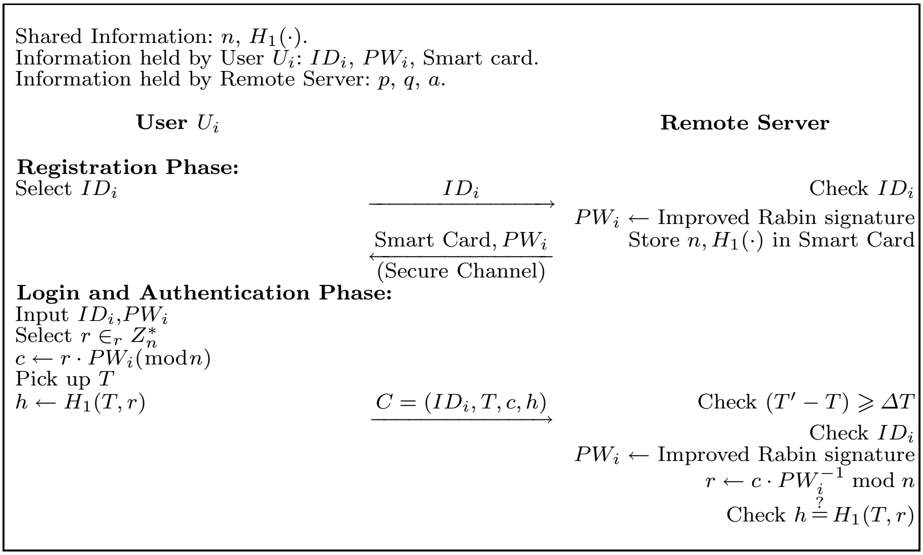

Shared Information: n, $H_1(\cdot)$.
Information held by User U_i: ID_i, PW_i, Smart card.
Information held by Remote Server: p, q, a.
User U_i
Remote Server
Registration Phase:
Select ID_i
$\xrightarrow{ID_i}$
Check ID_i
$PW_i \leftarrow$ Improved Rabin signature
Store $n, H_1(\cdot)$ in Smart Card
$\xleftarrow[\text{(Secure Channel)}]{\text{Smart Card}, PW_i}$
Login and Authentication Phase:
Input ID_i,PW_i
Select $r \in_r Z_n^*$
$c \leftarrow r \cdot PW_i(\text{mod} n)$
Pick up T
$h \leftarrow H_1(T, r)$
$\xrightarrow{C = (ID_i, T, c, h)}$
Check $(T' - T) \geqslant \Delta T$
Check ID_i
$PW_i \leftarrow$ Improved Rabin signature
$r \leftarrow c \cdot PW_i^{-1} \bmod n$
Check $h \overset{?}{=} H_1(T, r)$

Fig. 1. Lu-Cao's authentication scheme

With the knowledge of T, c and h, the attacker can randomly guess a password PW_i^* and then check if $h = H_1(T, c \cdot PW_i^*(\text{mod} n))$. If it holds, it means that PW_i^* is U_i's password. Then, the attacker can successfully impersonate user U_i by using the guessed password PW_i^* to login the remote server.

In order to provide freshness and prevent replay attack, Lu-Cao's scheme uses both a large random number r and timestamp T. However, the r is a needless value for both authentication and efficiency. That is, in the login phase, if U_i computes $h = H_1(T, PW_i)$ instead of $H_1(T, r)$, and sends a login message $C = (ID_i, T, h)$ to the remote server, then the remote server can simply check whether $h \overset{?}{=} H_1(T, PW_i)$ holds by using the computed user password PW_i in the Step (3) of the authentication phase. It is more efficient as well as more secure. Obviously, the r is a needless value for both security and efficiency.

4 Proposed Authentication Scheme

This section proposes an improvement of Lu-Cao's authentication scheme. The proposed scheme also consists of four phases: an initialization, registration, login, and authentication phase. Figure 2 shows the proposed authentication scheme. The initialization phase is the same as in Lu-Cao's scheme. The proposed remote user authentication scheme works as follows:

Registration Phase: When a new user U_i wants to register, U_i freely chooses his or her identity ID_i and password PW_i, and then submits them to the remote

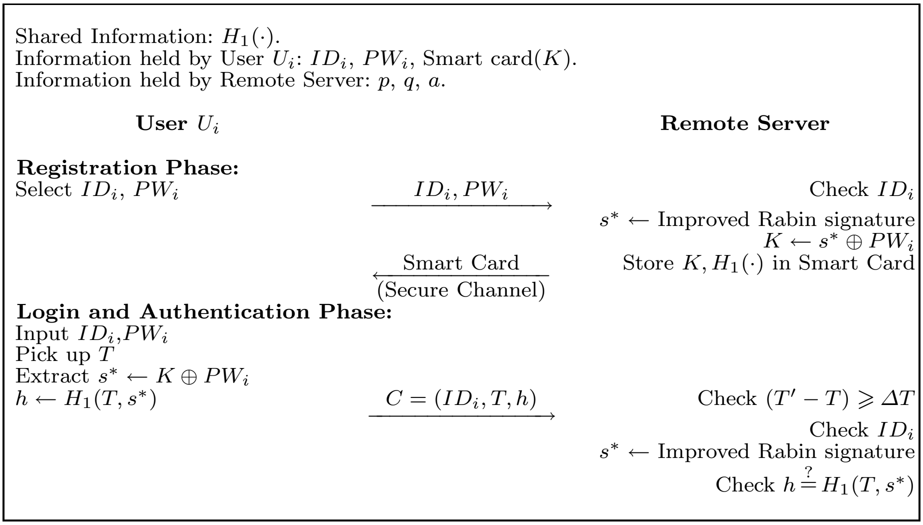

Shared Information: $H_1(\cdot)$.
Information held by User U_i: ID_i, PW_i, Smart card(K).
Information held by Remote Server: p, q, a.
User U_i
Remote Server
Registration Phase:
Select ID_i, PW_i
$\xrightarrow{ID_i, PW_i}$
Check ID_i
$s^* \leftarrow$ Improved Rabin signature
$K \leftarrow s^* \oplus PW_i$
$\xleftarrow[\text{(Secure Channel)}]{\text{Smart Card}}$
Store $K, H_1(\cdot)$ in Smart Card
Login and Authentication Phase:
Input ID_i,PW_i
Pick up T
Extract $s^* \leftarrow K \oplus PW_i$
$h \leftarrow H_1(T, s^*)$
$\xrightarrow{C = (ID_i, T, h)}$
Check $(T' - T) \geqslant \Delta T$
Check ID_i
$s^* \leftarrow$ Improved Rabin signature
Check $h \overset{?}{=} H_1(T, s^*)$

Fig. 2. Proposed authentication scheme

server for registration. These private data must be sent in person or over a secure channel. Upon receiving the registration request, the remote server performs the following steps:

(1) check the validity of ID_i. If it is valid, the operation will continue, otherwise stop;
(2) use the improved Rabin signature scheme [10] to compute (s^*, c_1, c_2), where $s^{*2} \equiv (-1)^{c_2} \cdot a^{c_1} \cdot H(ID_i)(\text{mod} n)$, $(\frac{s^*}{p}) = (\frac{s^*}{q}) = 1$. We assume that s^* is a k-bit master key for U_i. The security parameter k is sufficiently large such that s^* cannot be compromised by the off-line password guessing attack.
(3) compute the secret value $K = s^* \oplus PW_i$, where $\oplus$ is the bit-wise exclusive-or operation, and store the secret value K and public parameter $H_1(\cdot)$ to a smart card.
(4) issue the smart card to the user via a secure channel.

Login Phase: When U_i wants to login to the remote server, U_i attaches his or her smart card to the login device and keys in his ID_i and PW_i. Then, the smart card performs as follows:

(1) pick up the current date and time T of the login device;
(2) extract master key $s^* = K \oplus PW_i$;
(3) compute $h = H_1(T, s^*)$;
(4) send a login message $C = (ID_i, T, h)$ to the remote server.

Authentication Phase: Suppose that the remote server receives the message C at T', where T' is the current date and time of the system.

(1) check the time interval between T and T'; if $(T' - T) \geqslant \Delta T$, where ΔT is the expected legal time interval for transmission delay, the server will reject the login request;
(2) check the validity of identity ID_i; if the format of ID_i is incorrect, the login request will be rejected;
(3) use the same way in Step (2) of the registration phase to compute U_i's secret key s^*;
(4) check $h \overset{?}{=} H_1(T, s^*)$. If it holds, the server will accept the login request. Otherwise, the request will be rejected;

5 Security Analysis

This section provides the proof of correctness of the proposed authentication scheme. First, the security terms [12] needed for the analysis of the proposed scheme are defined as follows:

Definition 1. *A weak secret key* (PW_i) *is the value of low entropy, which can be guessed in polynomial time.*

Definition 2. *A strong secret key* (s^*) *is the value of high entropy, which cannot be guessed in polynomial time.*

Definition 3. *A secure one-way hash function* $y = H_1(x)$ *is where given* x *to compute* y *is easy and given* y *to compute* x *is hard.*

Given the above definitions, the following analyzes the security of the proposed authentication scheme:

(1) Even a valid login message $C = (ID_i, T, h)$ can be eavesdropped, due to the fact that a one-way hash function is computationally difficult to invert. It is extremely hard for any attacker to derive the master key s^* from $h = H_1(T, s^*)$. Even if the smart card of user U_i is picked up by an attacker, it is still difficult for the attacker to derive s^* from K, where $K = s^* \oplus PW_i$, because the attacker cannot know U_i's password PW_i.
(2) Since U_i's master key s^* has k-bit length which is sufficiently large that s^* cannot be compromised by the off-line password guessing attack, unlike Lu-Cao's scheme, the proposed scheme can resist password guessing attacks.
(3) For replay attacks, neither the replay of an old login message $C = (ID_i, T, h)$ in the login phase will work, as it will fail in Steps (1) of the authentication phase due to the time interval $(T' - T) \geqslant \Delta T$.
(4) The proposed scheme can resist impersonation attacks. An attacker can attempt to modify message $C = (ID_i, T, h)$ into $C = (ID_i, T_A, h_A)$, where T_A is the attacker's current date and time, so as to succeed in Step (1) of the authentication phase. However, such a modification will fail in Step (4) of the authentication phase, because an attacker has no way of obtaining the master key s^* to compute the valid parameter $h_A = H_1(T_A, s^*)$. Furthermore, assume that a user loses his smart card and it is found by an attacker

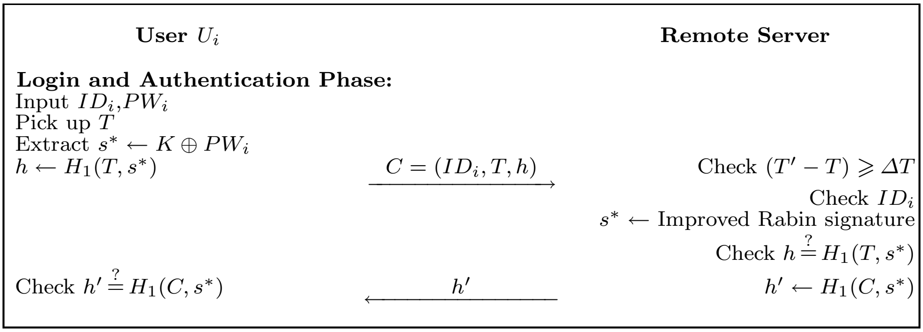

User U_i
Remote Server
Login and Authentication Phase:
Input ID_i,PW_i
Pick up T
Extract $s^* \leftarrow K \oplus PW_i$
$h \leftarrow H_1(T, s^*)$
$\xrightarrow{C = (ID_i, T, h)}$
Check $(T' - T) \geqslant \varDelta T$
Check ID_i
$s^* \leftarrow$ Improved Rabin signature
Check $h \overset{?}{=} H_1(T, s^*)$
Check $h' \overset{?}{=} H_1(C, s^*)$
$\xleftarrow{h'}$
$h' \leftarrow H_1(C, s^*)$

Fig. 3. Proposed mutual authentication scheme

or an attacker steals a user's smart card. However, the attacker cannot impersonate a legitimate user U_i by using the smart card because no one can reveal the PW_i from value K in the smart card without knowing U_i's master key s^*.

(5) If an attacker tries to forge a message $C = (ID_i, T, h)$, he or she must have U_i's master key s^*, because h must be derived from s^*. However, this is infeasible, as s^* has to be obtained from improved Rabin signature scheme [10].

(6) Even though users can freely choose their password PW_i, the proposed scheme can provide user password change without any help from the remote server. If U_i wants to change his or her old password PW_i to a new password PW_i^*, U_i inserts his or her smart card into the smart card reader of a terminal, and enters ID_i, PW_i and PW_i^*. Then, U_i's smart card computes $K' = K \oplus PW_i \oplus PW_i^*$ and stores K' in the smart card in place of the old K. In this case, when a smart card is stolen, an unauthorized user can easily change a new password for the card. To preventing this, in the Step (3) of the registration phase, if the server additionally stores the hash value $H_1(s^*)$ besides the secret value K and public parameter $H_1(\cdot)$ in a smart card, since the smart card can verify the computed value $K \oplus PW_i$ using the stored $H_1(s^*)$, when the smart card was stolen, unauthorized users cannot change the card's password. However, this is a requirement for providing additionally security.

(7) Several server spoofing attacks have been recently proposed [13]. The attacker can manipulate the sensitive data of legitimate users via setting up fake servers. Therefore, a secure remote authentication scheme using a smart card must have the ability to work against such attacks. The proposed scheme provides mutual authentication by adding two hash operations to withstand server spoofing attack as follows: for providing mutual authentication, the remote server computes $h' = H_1(C, s^*)$ by using the received value $C = (ID_i, T, h)$ and computed value s^*, and then sends back h' to the user U_i. Upon receiving message h', the user U_i computes $H_1(C, s^*)$ and compares it with received h'. If they are equal, the user U_i believes that the responding part is the real remote server and the mutual authentication is complete; otherwise, the user U_i interrupts the connection. Figure 3 shows

proposed mutual authentication scheme. If a masqueraded server tries to cheat the requesting user U_i, it has to prepare a valid message h'. However, this is infeasible, as there is no way to derive the master key s^* to compute the hash value h', due to the security property of the improved Rabin signature scheme and the one-way property of the secure one-way hash function.

6 Performance Analysis

Comparisons between Hwang-Li's scheme [2], Lu-Cao's scheme [7], and our proposed scheme are shown in Table 1. Hwang-Li's scheme requires a total of six modular exponential operations, two modular multiplication operations, one modular inversion operation, two hashing operations, and two exclusive-or operations. Lu-Cao's scheme requires a total of two modular square root operations, two modular multiplication operations, one modular inversion operation, and four hashing operations. However, the proposed scheme requires a total of two modular square root operations, two hashing operations, and two exclusive-or operations. Furthermore, the user is only required to perform one hashing operation and one exclusive-or operation during the login and authentication phase of the proposed scheme. Obviously, the proposed scheme is more efficient than Hwang-Li's and Lu-Cao's. In addition, the proposed scheme uses minimum communication bandwidth unlike the other two schemes. Among three messages $C = (ID_i, T, h)$, one is the user's identifier, one is a timestamp and one is a hash output bit (160 bit). These are very low communication messages.

Table 1. A comparison of computation costs

Computational type	Hwang-Li's Scheme [2]		Lu-Cao's Scheme [7]		Proposed Scheme	
	User	Server	User	Server	User	Server
Modular exponential	3	3	0	0	0	0
Modular square root	0	0	0	2	0	2
Modular multiplication	1	1	1	1	0	0
Modular inversion	0	1	0	1	0	0
Hash operation	1	1	1	3	1	1
XOR operation	1	1	0	0	1	1

7 Conclusions

The current paper demonstrated that Lu-Cao's scheme has some drawbacks; that is, the password of a user has to be computed by the server and the scheme has unnecessary computation costs. To achieve the aim of user friendliness as well as low communication and low computation costs, we have presented an improved authentication scheme, in order to isolate such problems while still preserving the advantages of Lu-Cao's scheme. The proposed scheme also provides mutual authentication between the user and a remote server. As a result, the proposed scheme is more secure and efficient than Lu-Cao's scheme.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments in improving our manuscript. This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

References

1. Lamport, L.: Password Authentication with Insecure Communication. Communications of the ACM. Vol. 24. No. 11. (1981) 770-772
2. Hwang, M.S., Li, L.H.: A New Remote User Authentication Scheme Using Smart Cards. IEEE Trans. On Consumer Electronics. Vol. 46. No. 1. (2000) 28-30
3. Chan, C.K., Cheng, L.M.: Cryptanalysis of a Remote User Authentication Scheme Using Smart Cards. IEEE Trans. Consum. Electr. Vol. 46. No. 4. (2000) 992-993
4. Chang, C.C., Hwang, K.F.: Some Forgery Attacks on a Remote User Authentication Scheme Using Smart Cards. Informatics. Vol. 14. No. 3. (2003) 289-294
5. Shen, J.J., Lin, C.W., Hwang, M.S.: A Modified Remote User Authentication Scheme Using Smart Cards. IEEE Trans. Consum. Electr. Vol. 49. No. 2. (2003) 414-416
6. Leung, K.C., Cheng, L.M., Fong, A.S., Chan, C.K.: Cryptanalysis of a Modified Remote User Authentication Scheme Using Smart Cards. IEEE Trans. Consum. Electr. Vol. 49. No. 4. (2003) 1243-1245
7. Lu, R., Cao, Z.: Efficient Remote User Authentication Scheme Using Smart Card. Computer Networks. Vol. 49. (2005) 535-540
8. Ding, Y., Horster, P.: Undetectable On-line Password Guessing Attacks. ACM Operating Systems Review. Vol. 29. No. 4. (1995) 77-86
9. Cao., Z.F.: A Threshold Key Escrow Scheme based on Public Key Cryptosystem. Sci. China. Vol. 44. No. 4. (2001) 441-448
10. Rabin., M.O.: Digitalized Signature and Public Key Functions as Intractable as Factorization. Technical Report 212. MIT Laboratory for Computer Science. (January 1979)
11. Needham, R.M., Schroeder, M.D.: Using Encryption for Authentication in Large Networks of Computers. Communications of the ACM. Vol. 21. No. 12. (1978) 993-999
12. Menezes, A.J., Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptograph. CRC Press. New York. (1997)
13. Aoskan, N., Debar, H., Steiner, M., Waidner, M.: Authentication Public Terminals. Computers Networks. Vol. 31. (1999) 861-970

Forward Secure Password-Enabled PKI with Instant Revocation*

Seung Wook Jung[1] and Souhwan Jung[2],**

[1] Institute for Data Communications Systems, University of Siegen
Hölderlinstraße 3, 57076 Siegen, Germany
[2] School of Electronic Engineering, Soongsil University, 1-1, Sangdo-dong,
Dongjak-ku, Seoul 156-743, Korea
seung-wook.jung@uni-siegen.de, souhwanj@ssu.ac.kr

Abstract. Recently the concept of password-enabled PKI is an emerging issue to support user mobility. Virtual soft token and virtual smartcard were proposed as the password-enabled PKI. However, the virtual soft token does not support key disabling. In the virtual smartcard, the user must interact with remote entity per signing operation. In addition, both schemes do not support forward secrecy and instant revocation.

In this paper, we propose a new approach that supports user mobility. The proposed approach supports key disabling and the user does not need interaction with the remote entity for each signature. Moreover, the proposed scheme allows instant key revocation. Thereby, the distribution of CRL is not required. Furthermore, the proposed scheme supports forward secrecy. In this sense, our scheme, implemented only software, is stronger than a long-term private key with physical smart cards. By forward secrecy and instant revocation, signing documents using a timestamp provided by a trusted authority is not required to protect from modifying signed document by the adversary who knows private key.

Keyword: Password, PKI.

1 Introduction

The Public Key Infrastructure (PKI) is the basis of a pervasive security infrastructure for ensuring user's digital identity. However, the user mobility, also called the roaming user, and private key management for the pervasive security is still issue. Ideally, the private key is stored in a hardware smartcard and the user moves amongst multiple PCs. However, in reality, smartcard readers are not available at every computers. Given the cost and availability problems of hardware smartcard, the concept of password-enabled PKI, which relies on passwords to enable the usage of private keys for providing user mobility, is recently focused on by PKI vendors and researchers as an interesting issue for true pervasive security.

* This work was supported by the Soongsil University Research Fund.
** Corresponding author.

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 54–67, 2006.

Recently, to support the roaming user, virtual soft tokens [1][2][3], and virtual smartcards[4][5][6][7] are proposed[5] as password-enabled PKI. Both approaches assume that there exists an online network server.

In the virtual soft token, the private key of a public/private key pair is encrypted with a password and the encrypted private key is stored on an online network server. With the password, the user and the server establish an authenticated and confidential channel using a password-authenticated key exchange protocol [8][9][10], and the user downloads the encrypted private key. The user decrypts it and uses the private key as in the conventional PKI.

In the virtual smartcard, the user's private key is split into two parts: the password that the user holds and the secret component stored in the server. In [4][5][6][7], The RSA private key for which the corresponding public key is (N, e) is split into a password-derived private key d_1 and another value d_2, $d = d_1 \cdot d_2 \bmod \phi(N)$, and d_2 is stored on a server, where d is a private key corresponding to the public key (N, e). Therefore, the user and the server have to communicate for each private key operation (signing or decryption).

The disadvantage of virtual smartcard schemes is that the user interacts with the remote server per signing, while the virtual soft token does not require any interaction after downloading the encrypted private key from the online server. The disadvantage of virtual soft token schemes is that they does not support key disabling, while the virtual smartcard supports key disabling. After receiving the key disabling message, the server will not generate a signature or decrypt a message with given user's private value d_2. Therefore, even though the adversary gets the password, the adversary cannot generate the signature for a given user, after key disabling. However, in both schemes, when the private key is disclosed, the user must revoke the private key and the Certification Authority (CA) has to distribute the Certificate Revocation List (CRL). Otherwise, the adversary can generate the signature for a given user. Furthermore, even though CRL is distributed, the adversary, who knows the private key, can modify the existing signatures that are generated before revocation, if the time-stamp from the trusted third party is not included in the signatures[11].

In this paper, we propose a new scheme for the roaming user. The long-term key of the proposed scheme is the only password and the password is used in a part of a session private key. In the proposed scheme, the CA issues a Master Password Certificate (MPC) that certifies the user's password. For the user mobility, a part of MPC, which is encrypted by the password, is stored in an online server and the other part is stored in an online CA. The user downloads the encrypted part of MPC, decrypts it, and modify decrypted part of MPC. Only the legitimate user, who knows the password, can generate a session private key and can properly modify decrypted part of MPC corresponding to the session private key in order to request Short-Lived Certificate (SLC) to the online CA. After receiving the SLC, the user does not need interaction with any entity for signature function.

The properties to be achieved are summarized as follows:

- The proposed scheme provides user mobility, so users can move amongst the multiple computers in a way that the user remembers only the ID and the password pair.
- No interaction is required after receiving the SLC. Therefore, the proposed scheme is efficient compared to the virtual smartcard.
- After receiving revocation, the online CA will not generate SLC for the given user as key disabling does in the virtual smartcard scheme. Furthermore, in the proposed scheme, the CRL is not required to be distributed as any schemes based on SLC do not need to distribute CRL. However, existing password-enabled PKI schemes are required to distribute the CRL when the private key is compromised. When revocation does not require distribution of CRL, the revocation is called *instant revocation* in this paper.
- The compromise of the session private key does not compromise of the long-term key and other session keys. Also, the compromise of the password does not compromise the session private key. Therefore, when the private key is compromised, the adversary cannot modify the previous signatures that were generated in other sessions, after the instant key revocation without assuming existing timestamp authority. In this sense, our scheme is more secure than any scheme that uses long-term private in the physical smart card.

The remaining of this paper is organized as follows. Section 2 presents the background knowledge. The Password Certificate (PC) scheme ((MPC) issuing, signing a document and verify document using PC), along with the security proof, are presented in Section 3. In section 4, the PC scheme for user mobility is presented and the security is analyzed. The section 5 compare the proposed scheme with existing schemes. Finally, in Section 6, we summarize this paper.

2 Preliminaries

2.1 Notation

In this paper we use the following notation

- $a \in A$ denotes that the value a is in the set A .
- $a \in_R A$ denotes the choice of a uniform and independent random value a from the set A.
- Z_p denotes the set $\{0, 1, \ldots, p-1\}$, Z_p^* denotes the set $\{1, 2, \ldots, p-2, p-1\}$ with prime p.
- $S_k(m)$ denotes a signature on a message m with key k and outputs σ.
- $V_y(\sigma, m)$ denotes a verification algorithm of a signature σ using a public key y and the outputs true or false.
- $SGen(), SE_k()$ and $SD_k()$ denote a symmetric key encryption scheme, where $SGen()$ is a key generation algorithm, $SE_k()$ is an encryption algorithm using a symmetric key k, $SD_k()$ is a decryption algorithm using a symmetric key k.

- $AGen(), AE_y()$ and $AD_x()$ denote an asymmetric encryption scheme, where $AGen()$ is a key generation algorithm, $AE_y()$ is an encryption algorithm using a public key y and $AD_x()$ denotes a decryption algorithm using a private key x.

2.2 The Schnorr Signature

In order to prove the security of our scheme, the Schnorr signature scheme is used, so we review Schnorr signature scheme briefly. Let p and q be large primes with $q|p-1$. Let g be a generator of a multiplicative subgroup of Z_p^* with order q. A signer choose a private key x and computes a public key $y = g^x \bmod p$. In order to sign a message m, the signer chooses a random number $k \in_R Z_q^*$ and computes $r = g^k \bmod p, s = -x \cdot h(m||r) + k \bmod q$, where $h()$ is a collision resistant hash function. The verification algorithm checks $r \overset{?}{=} g^s \cdot y^{h(m||r)} \bmod p$. [12] proved the security of the Schnorr signature scheme in the random oracle model under the adaptive chosen message attacks. If there exists an Probabilistic Polynomial Time (PPT) adversary breaking the Schnorr signature, then we can solve the discrete logarithm in a subgroup of large prime within the PPT bound. Therefore, the Schnorr signature scheme is as secure as solving the Discrete Logarithm Problem (DLP).

3 Password-Certificate(PC) Scheme

This scheme is an application of the Schnorr signature scheme for issuing certificates. In this scheme, the user has to keep MPC confidential in a hardware smartcard. Therefore, this scheme does not provide user mobility, because the smartcard readers are not deployed in most computers. However, this scheme is useful to explain how our scheme works and for the security proof.

3.1 Setup

1. The CA chooses large primes p, q such that $q|p-1$, a generator g of a multiplicative subgroup of Z_p^* with order q and a collision resistant hash function $h()$ where $h() : \{0,1\}^* \rightarrow Z_q^*$.
2. The CA chooses $x_{CA} \in_R Z_q^*$ as a private key and computes $y_{CA} = g^{x_{CA}} \bmod p$ as a public key.
3. The CA must keep x_{CA} confidential. The public key and parameters $(p, q, g, y_{CA}, ID_{CA})$ and $h()$ are publicly known in the network, where ID_{CA} is the CA's identifier.

3.2 Registration and Issuing of the Master Password Certificate(MPC)

Assume that the communication channel between the user and the CA for the registration procedure is authenticated and confidential such as Transport Layer Security(TLS)[13] with the server certificate. Furthermore, the user must be authenticated appropriately e.g., using the out-of-band mechanism. As a result

User CA

$v_i, a \in_R Z_q^*$
$\pi_i = h(pw_i||v_i)$
$r' = g^{a+\pi_i} \bmod p$

$\rightarrow u_i, r'$

$k \in_R Z_q^*$
$r_i = g^k \cdot r' \bmod p$
$CI_i = u_i||ID_{CA}||UCID||etc$
$s_i' = -x_{CA} \cdot h(CI_i||r_i) - k \bmod q$

$s_i', r_i, CI_i \leftarrow$

$s_i = s_i' - a \bmod q$
check $g^{\pi_i} \overset{?}{=} g^{s_i} \cdot y_{CA}^{h(CI_i||r_i)} \cdot r_i$
keeps $MPC_i = (s_i, r_i, CI_i)$

success $\rightarrow$

keeps u_i

Fig. 1. Registration and Issuing Master Password Certificate

of successful registration, the user has a MPC. The procedures of registration and Issuing MPC are described as follows. (Fig. 1 also shows the procedures of registration and issuing of the MPC).

1. A user u_i, where i is the index of the user in the set of users and u_i is the identifier of the user i, types a password pw_i, chooses a salt $v_i \in_R Z_q^*$, and a random number $a \in_R Z_q^*$. The user computes $\pi_i = h(pw_i||v_i)$ and $r' = g^{a+\pi_i} \bmod p$.
2. The user sends u_i and r' to the CA through an authenticated and confidential channel.
3. The CA chooses $k \in_R Z_q^*$ and computes $r_i = g^k \cdot r' \bmod p$.
4. The CA computes $s_i' = -x_{CA} \cdot h(CI_i||r_i) - k \bmod q$, where certificate information CI_i is the concatenation of the user identifier u_i, the CA's identifier ID_{CA}, the unique certificate identifier $UCID$, and so on. That is, $CI_i = u_i||ID_{CA}||UCID||etc$. The CA returns s_i', r_i and CI_i to the user through an authenticated and condifential channel.
5. The user computes $s_i = s_i' - a \bmod q$ and checks $g^{\pi_i} \overset{?}{=} g^{s_i} \cdot y_{CA}^{h(CI_i||r_i)} \cdot r_i \bmod p$. Note that $s_i = -x_{CA} \cdot h(CI_i||r_i) - k - a \bmod q$ and $r_i = g^{k+a+\pi_i} \bmod p$. If true, then the user generates a Master Password Certificate $MPC_i = (s_i, r_i, CI_i)$. The user keeps MPC_i and v_i confidential. The user sends a success message.

3.3 Session Private Key, Short-Lived Certificate (SLC), and Public Key Generation

Session Private Key and SLC Generation: The user chooses $b \in_R Z_q^*$. The user computes $SLC_i = (s_i''(= s_i + b \bmod q), r_i, CI_i)$ and a session private key

$x_i = b + \pi_i \bmod q$. To sign a message m, the user generates a signature $S_{x_i}(m)$ using ElGamal-like signature with x_i and sends $SLC_i, m, S_{x_i}(m)$ to a verifier.

Session Public Key Generation: The verifier generates a session public key $y_i = g^{b+\pi_i} = g^{s''_i} \cdot y_{CA}^{h(CI_i||r_i)} \cdot r_i \bmod p$. Note, $s''_i = -x_{CA} \cdot h(CI_i||r_i) - k - a + b \bmod q$ and $r_i = g^{k+a+\pi_i} \bmod p$. The verifier can verify the signature $S_{x_i}(m)$ with y_i.

3.4 Security Proof

Adversary Model. The adversary is network wire, so the adversary can eavesdrop, modify, and delete message. Additionally the adversary can query and run any protocol. Also, the adversary has PPT bounded computing power.

Assumption 1. *There is no PPT adversary who can solve the DLP.*

Assumption 2. *The signature scheme* $(S_{x_i}(m), V_{y_i}(S_x(m), m)$ *with the user's session private key* x_i *and public key* y_i *is secure against adaptive chosen message attacks.*

Lemma 1. *There is a PPT adversary* u_i *who can register up to the number of the registration query,* q_{reg}*, in polynomial time* t*, where the registration query is* (u, r')*. The probability is taken over the coin flips of* u_i*. If* u_i *generates a forgery certificate* $MPC_k(= s_k, r_k, CI_k)$ *along with a corresponding password* π_k *within polynomial time* t *and (non-negligible) with the success probability* ϵ*, where* $CI_k = u_k||ID_{CA}||UCID||etc$ *and* $k \neq i$*, then Schnorr Signature scheme is* (t', q_{sig}, ϵ)*-breakable, where* $q_{sig} = q_{reg}$*,* $t' = t + t(s)$*, and* $t(s)$ *denote a computation time for one subtraction operation.*

Proof. If the adversary can generate $MPC_k(= s_k, r_k, CI_k)$ and the corresponding password π_k with (t, q_{sig}, ϵ), then the adversary can easily generate a tuple $(s_k - \pi_k, r_k)$. The tuple $(s_k - \pi_k, r_k)$ is a signature over CI_k of the Schnorr signature scheme. Therefore, the PPT adversary can generate a Schnorr signature by generating a forgery certificate with polynomial time bound $t' = t + t(s)$ and probability ϵ after querying q_{sig} to the CA. Note, $s_k - \pi_k = -x_{CA} \cdot (h(CI_k||r_k)) - k - a - \pi_i \bmod q$ and $r_k = g^{k+a+\pi_k}$. □

Lemma 2. *There is a PPT adversary* A *who can ask the short-lived certificate* SLC_i *up to the number of the short-lived certificate query,* q_{SLC}*, in polynomial time* t*. If* A *generates a forgery certificate* $SLC'_k(= s'_k, r'_k, CI'_k)$ *along with a corresponding session key* x'_k *within polynomial time* t *and (non-negligible) success probability* ϵ *, then the Schnorr signature scheme is* (t', q_{sig}, ϵ)*-breakable, where* $q_{sig} = q_{SLC}, t' = t + t(s)$ *and* k *is any user, including* i*, in the network.*

Proof. If A can generate (s'_k, r'_k, CI'_k) with a corresponding session key x'_k, then A can generate a tuple $(s'_k - x'_k(= b + \pi_k), r'_k)$ with simple subtraction. The tuple is a signature over CI'_k of the Schnorr signature scheme. Therefore, the PPT adversary can generate the Schnorr signature by generating forgery certificate with the time bound $t' = t + t(s)$ with a subtraction operation. Note, $s'_k - x'_k = -x_{CA} \cdot (h(CI'_k||r'_k)) - k - a + b - b - \pi_k \bmod q$ and $r'_k = g^{k+a+\pi_k}$. □

For the same reason, it is infeasible that the adversary A generate the MPC'_k and x'_k corresponding to MPC'_k.

So far, we explained the security of certificates(MPC and SLC). [12] proved the security of the Schnorr signature scheme in the random oracle model under the adaptive chosen message attack. The Schnorr signature scheme is as secure as DLP[12]. Therefore, generating a forgery certificate is infeasible for the network adversary under the assumption 1 and the assumption 2. The other way of breaking our scheme is to disclose the password-derived value π_i or session keys x_i.

Lemma 3. *The disclosing a session private key $x_i = b+\pi_i$ by network adversary is as hard as solving DLP .*

Proof. If the adversary can get the private key x_i from the public data $(SLC_i, S_{x_i}(m), m, g^{x_i})$, then adversary can solve DLP of g^{x_i} or break the underlying signature scheme. Therefore, it is infeasible that adversary get session private key under assumption 1 and assumption 2. □

Lemma 4. *The off-line dictionary attack is impossible.*

Proof. The password π_i is always encrypted with a and b which are uniformly and independently chosen from Z_q^* and sent to communication party. Therefore, dictionary attack by the network adversary is impossible. □

Note that the off-line dictionary attack is impossible in the information theoretical sense rather than infeasible in the complexity theoretical sense because of random numbers a and b.

Fact 1. *From Lemma 4, even if adversaries know x_i, they cannot get the π_i. The random value $b(= x_i - \pi_i \bmod q)$ are chosen uniformly and independently from Z_q^*. Therefore, compromise of x_i does not compromise previous a session private key. However, if adversaries know x_i, then they can generate valid certificates $(s''_i + c, r_i, CI_i)$ and private keys $x_i + c$, where c is a random number. Therefore, u_i must revoke u_i's certificate, so adversaries cannot generate further signatures.*

4 PC for User Mobility (PC+UM)

When an adversary compromises MPC_i in the PC scheme, the adversary can conduct the off-line dictionary attack. Therefore, in the PC scheme, the user has to store MPC_i confidential in the hardware smartcard. The smartcard reader is not available in most computers, so the user mobility is limited.

The PC+UM scheme supports the user mobility, so the user moves amongst multiple machines with a memorable password. In this scheme, there are four entities: Download Server (DS) for downloading the part of certificate, the online CA for issuing the SLC, the CA for issuing the MPC, and the user. In this section, for simplicity, we assume that the online CA for issuing SLC and the CA for issuing MPC are the same entity, so there are three entities for the PC+UM scheme in this paper. The PC+UM scheme consists of (1)setup, (2)registration

User

CA

$v_i, a \in_R Z_q^*$

$\pi_i = h(pw_i||v_i)$

$r' = g^{a+\pi_i} \bmod p$

$\rightarrow u_i, r'$

$k \in_R Z_q^*$

$r_i = g^k \cdot r' \bmod p$

$CI_i = u_i||ID_{CA}||UCID||etc$

$s_i' = -x_{CA} \cdot h(CI_i||r_i) - k \bmod q$

$s_i', r_i \leftarrow$

$s_i = s' - a \bmod q$

checks $g^{\pi_i} \overset{?}{=} g^{s_i} \cdot y_{CA}^{h(CI_i||r_i)} \cdot r_i \bmod p$

success $\rightarrow$

keeps (CI_i, r_i)

DS

$sk_i = SGen(\pi_i)$

$e_i = SE_{sk_i}(s_i), v_i, u_i \rightarrow$

keeps (e_i, v_i, u_i)

Fig. 2. Registration and Issuing a Master Password Certificate of PC+UM

and issuing MPC, (3) issuing SLC and generating the session private key, (4) verify signature and (5) instant revocation.

4.1 Setup

1. The CA chooses large primes p, q such that $p = 2q + 1$, a generator g of a multiplicative subgroup of Z_p^* with order q and a collision resistant hash function $h()$ where $h() : \{0, 1\}^* \rightarrow Z_q^*$.
2. The CA chooses $x_{CA} \in_R Z_q^*$ as a private key for signing the certificate and computes $y_{CA} = g^{x_{CA}} \bmod p$ as a corresponding public key.
3. The CA chooses an ElGamal asymmetric encryption scheme [14][15] and chooses $x_{CA.E} \in_R Z_q^*$ as a private key for the asymmetric encryption scheme and computes $y_{CA.E} = g^{x_{CA.E}} \bmod p$ as a corresponding public key.
4. The CA must keep x_{CA} and $x_{CA.E}$ confidential. The public key and parameters $(p, q, g, y_{CA}, y_{CA.E}, ID_{CA})$ are publicly known in the network, where ID_{CA} is the identifier of the CA.

4.2 Registration and Issuing Master Password Certificate in the PC+UM Scheme

Assume that the communication channel between the user and CA for registration procedure is authenticated and confidential. Furthermore, the user must be identified appropriately e.g., using the out-of-band mechanism. The procedures of registration are as follows. (The whole procedures are described in the Fig. 2).

1. A user u_i, where i is the index of user in the set of users and u_i is the identifier of the user i, types a password pw_i, chooses a salt $v_i \in_R Z_q^*$, and a random number $a \in_R Z_q^*$. The user computes $\pi_i = h(pw_i||v_i)$ and $r' = g^{a+\pi_i} \bmod p$.
2. The user sends u_i and r' to the CA through an authenticated and confidential channel.
3. The CA chooses $k \in_R Z_q^*$ and computes $r_i = g^k \cdot r' \bmod p$.
4. The CA computes $s_i' = -x_{CA} \cdot h(CI_i||r_i) - k \bmod q$, where certificate information CI_i is the concatenation of the user identifier u_i, the CA's identifier ID_{CA}, the unique certificate identifier $UCID$, and so on. That is, $CI_i = u_i||ID_{CA}||UCID||etc$. The CA returns s_i', r_i and CI_i to the user through an authenticated and confidential channel.
5. If $g^{\pi_i} \overset{?}{=} g^{s_i} \cdot y_{CA}^{h(CI_i||r_i)} \cdot r_i \bmod p$ is true, the user sends success message to the CA.
6. The CA keeps CI_i, r_i confidential.
7. The user generates a symmetric key $sk_i = SGen(\pi_i)$.
8. The user sends $e_i = SE_{sk_i}(s_i), v_i$, and u_i to the DS.
9. The DS keeps (e_i, v_i, u_i).

User		DS
	$u_i \rightarrow$	searches (e_i, v_i, u_i)
	$e_i, v_i \leftarrow$	
$s_i = SD_{sk_i}(e_i)$		
$b \in_R Z_q^*$		
$s_i'' = s_i + b \bmod q$		
$ae_i = AE_{y_{CA.E}}(g^{s_i''})$		
$x_i = \pi_i + b \bmod q$		
$\sigma = S_{x_i}(ae_i, u_i, ID_{CA}, T)$		CA
	$ae_i, u_i, T, \sigma \rightarrow$	
		searches (u_i, CI_i)
		$g^{s_i''} = AD_{x_{CA.E}}(ae_i)$
		$y_i = g^{s_i''} \cdot r_i \cdot y_{CA}^{h(CI_i' \Vert r_i)} \bmod p$
		$V_{y_i}(\sigma, ae_i, u_i, ID_{CA}, T)$
	$\leftarrow SLC_i = S_{x_{CA}}(CI_i' \Vert y_i)$	

Fig. 3. Short-lived Certificate Generation of PC+UM

4.3 Issuing Short Lived Certificate and Generating Session Private Key in the PC+UM Scheme

The following shows the procedures of issuing SLC and generating a session private key. Also, Fig. 3 shows the procedures of issuing a SLC and generating a session private key.

1. The user sends u_i to the DS, and the DS returns e_i and v_i.
2. The user generates $\pi_i = h(pw_i||v_i)$ and $sk_i = SGen(\pi_i)$. The user computes $s_i = SD_{sk_i}(e_i)$.
3. The user chooses $b \in_R Z_q^*$. The user computes $s_i'' = s_i + b \bmod q$, $ae_i = AE_{y_{CA.E}}(g^{s_i''})$, and a session private key $x_i = \pi_i + b \bmod q$.
4. The user generates a signature $\sigma_i = S_{x_i}(ae_i, u_i, ID_{CA}, T)$ to prove knowledge of the session private key, where T is a timestamp.
5. The user sends (ae_i, u_i, T, σ_i) to the CA.
6. The CA searches (CI_i, r_i) with u_i, decrypts $g^{s_i''} = AD_{x_{CA.E}}(ae_i)$, and generates a session public key $y_i = g^{s_i''} \cdot y_{CA}^{h(CI_i||r_i)} \cdot r_i \bmod p$. Note that $s_i'' = -x_{CA} \cdot h(CI_i||r_i) - k - a + b \bmod q$, $r_i = g^{k+a+\pi_i} \bmod p$, and $y_i = g^{\pi_i + b}$.
7. The CA checks the timestamp and verifies the signature σ. If the verification is false, the CA adds 1 to the number of consecutive error and 1 to the number of the total error. If the consecutive error is reached to the limitation or total error is reached to the limitation, the CA considers the request as an online dictionary attack and blocks the user account. Otherwise, the CA resets the number of consecutive errors to 0. The CA generates the short-lived certificate $SLC_i = S_{x_{CA}}(CI_i'||y_i)$ and sends it to the user, where CI_i' is a new short-lived certificate information, e.g., including the attributes of the user for the privilege management.

4.4 Instant Revocation

When the password is disclosed, the user must revokes the (CI_i, r_i), so that the CA will not issue the SLC for a given (CI_i, r_i). The instant revocation procedures are the following.

1-3. is same as the Sec. 4.3.

4. The user generates a signature $\sigma_i = S_{x_i}(ae_i, u_i, ID_{CA}, IR, T)$ to prove knowledge of the session private key, where T is a timestamp and IR represents the Instant Revocation.
5. The user sends $(ae_i, u_i, IR, T, \sigma_i)$ to the CA.
6. The CA searches (CI_i, r_i) with u_i, decrypts $g^{s_i''} = AD_{x_{CA.E}}(ae_i)$, and generates a session public key $y_i = g^{s_i''} \cdot y_{CA}^{h(CI_i||r_i)} \cdot r_i \bmod p$. Note that $s_i'' = -x_{CA} \cdot h(CI_i||r_i) - k - a + b \bmod q$, $r_i = g^{k+a+\pi_i} \bmod p$, and $y_i = g^{\pi_i + b}$.
7. The CA checks the timestamp and verifies the signature σ. If the verification is false, the CA adds 1 to the number of consecutive errors and 1 to the number of the total errors. If the consecutive error is reached to the limitation or total error is reached to the limitation, the CA considers the request as an online dictionary attack and blocks the user account. Otherwise, the CA blocks the user account (CI_i, r_i).

4.5 Security(Sketch)

It is trivial to see that the separation of s_i and r_i itself does not reduce the security of the Schnorr signature scheme. However, when s_i and s_i'' are disclosed, an adversary knows $b(= s_i'' - s_i \bmod q)$, which is a part of $x_i(= \pi_i + b \bmod q)$.

User		DS
	$u_i \rightarrow$	searches (e_i, v_i, u_i)
	$e_i, v_i \leftarrow$	
$s_i = SD_{sk_i}(e_i)$		
$b \in_R Z_q^*$		
$s_i'' = s_i + b \bmod q$		
$ae_i = AE_{y_{CA.E}}(g^{s_i''})$		
$x_i = \pi_i + b \bmod q$		
$\sigma = S_{x_i}(ae_i, u_i, ID_{CA}, IR, T)$		CA
	$IR, ae_i, u_i, T, \sigma \rightarrow$	
		searches (u_i, CI_i)
		$g^{s_i''} = AD_{x_{CA.E}}(ae_i)$
		$y_i = g^{s_i''} \cdot r_i \cdot y_{CA}^{h(CI_i' \| r_i)} \bmod p$
		if true $= V_{y_i}(\sigma, ae_i, u_i, ID_{CA}, T)$
		block (CI_i, r_i)
	$\leftarrow$ success	

Fig. 4. Instant Revocation of PC+UM

Therefore, the adversary can conduct a dictionary attack to $g^{\pi_i} \cdot g^b \bmod p$. If the dictionary attack succeeds, then the adversary can generate valid signatures for the user u_i with π_i and s_i. Therefore, the s_i and s_i'' must be kept confidential.

In PC+UM scheme, the solution for this problem is encrypting s_i and s_i''. s_i and s_i'' are unknown to the adversary. Therefore, the adversaries have no password verification data to check the correctness of the guessing.

If a session key $x_i^j (= \pi_i + b^j \bmod p)$ is disclosed, where j is a session index, then the adversary computes, for a session k, a session private key $x_i^k = x_i^j (= \pi_i + b^j) - ({s_i^{''}}^j - {s_i^{''}}^k (= b^j - b^k)) \bmod q$. In order to get the ${s_i^{''}}^k$ and ${s_i^{''}}^j$, the adversary has to break the underlying asymmetric encryption scheme. Therefore, it is infeasible that the adversary, who knows a session private key, gets the other session key under the assumption that the breaking underlying asymmetric encryption scheme is infeasible.

The adversary who knows (π_i, s_i) can generate session keys $x_i^j = \pi_i + b^j \bmod q$ for session j, if the adversary can get a random number b^j. To get b^j from public values $(v_i, u_i, e_i (= SE_{sk_i}(s_i)), y^{x_i^j}, AE^j_{y_{CA.E}}(g^{{s_i^{''}}^j}))$ with (π_i, s_i), the adversary has to break the underlying asymmetric encryption of $AE^j_{y_{CA.E}}(g^{{s_i^{''}}^j})$ and solve DLP problem of $g^{{s_i^{''}}^j}$ because of $b = s_i'' - s_i \bmod q$. Therefore, it is infeasible that the adversary, who knows π_i, gets the session private key under the assumption 1 and the assumption that breaking underlying asymmetric encryption scheme is infeasible.

4.6 Instant Revocation and Forward Secrecy

In conventional PKI, when the private key is compromised, the user has to revoke the corresponding certificate and the CA adds the certificate into the

CRL and distributes CRL. After revocation, the adversary cannot generate valid signatures. However, the overhead of distribution and verification of CRL is costly in various ways. When the password π_i is disclosed in the proposed scheme, the user has to send the revocation message, which includes user signature to the CA as described in Fig. 4. The CA deactivates the corresponding (CI_i, r_i) and never generates a SLC for the given (CI_i, r_i). Therefore, our scheme does not need to distribute a CRL, so the overhead of distributing and verification of the CRL is eliminated. Note when disclosing the session key, the adversary generates signatures only for the short period, e.g., 8 hour, specified by the SLC because of forward secrecy. Note when the password is disclosed in virtual smartcard schemes, the server does not operate a cryptographic function after receiving key disabling from the user. However, in virtual smartcards and virtual soft tokens based on the conventional PKI, when the private key is disclosed, the user has to revoke the corresponding certificate and the CA must distribute the CRL.

In conventional PKI, even though the private key is revoked, the adversary can modify the signatures generated before revoking the certificate. Time-stamping signed documents via a trusted time stamping authority [11] can provide the solution against this attack, but this requires that one need to communicate with authority for each signature, which is costly in various ways. The other solution for protecting these attacks is using forward secure signature schemes [16][17]. However, the computation overhead of forward secure signature schemes are much higher than the ordinary signature schemes such as the Schnorr signature. The proposed scheme provides the forward secrecy property, as discussed at section 4.5 and Fact 1. Therefore, after instant revocation, the adversary cannot modify any signatures that are generated before revoking.

5 Comparison

In this section, we compare the existing schemes (virtual smartcard and virtual soft token) to the proposed scheme. Table 1 shows the summary of the comparison. The virtual smartcard support key disabling but interaction is required per signature operation. The virtual soft token schemes and virtual smartcard schemes do not support forward secrecy and instant revocation, while the proposed scheme supports forward secrecy and instant revocation. The virtual smartcard schemes and virtual soft token schemes requires a trusted time-stamp

Table 1. Comparison with existing schemes

	virtual soft token	virtual smartcard	the proposed scheme
key disabling		✓	✓
Interaction per signature		✓	
Forward secrecy			✓
Instant Revocation			✓
Timestamp authority	✓	✓	

authority to protect that the adversary, who knows the private key, modifies the existing signature after key revocation, while the proposed scheme does not requires a trusted time-stamp authority.

6 Conclusion

There are two existing approaches (the virtual soft token and the virtual smart card) for the user mobility. In this paper, we proposed the third approach for supporting user mobility in the PKI involving the online CA. In our scheme, a long-term private key is only a password. A user receives a part of the certificate from the CA, encrypts it with the password, and uploads it to the remote server. When the user wants to operate a signature function, the user downloads the encrypted part of certificate and decrypts it. The user changes the part of certificate and generates a session private key corresponding to the changed part of certificate. The user requests the short-lived certificate from the online CA with the changed part of certificate and signature to prove the knowledge of the session private key. The online CA generates a session public key corresponding to the request and issues a short-lived certificate. Only the legitimate user can know the session private key corresponding to the session public key in our scheme.

Application of our scheme is single sign on where the online CA issues a short-lived attribute certificate for privileged management without assuming hardware smartcard. In general, the lifetime of the attribute certificate is shorter than the key certificate.

The advantages of our scheme are fourfold: First, the user roaming is supported. Second, no interaction is required for the signature function as a virtual soft token does. Third, instant revocation is supported, so that the CA does not need to distribute a CRL. Fourth, the compromise of a session private key does not compromise the password or previous session private key and vise versa. Moreover, after instant revocation, the adversary who knows a password(or a session private key) cannot modify the existing signature(that are generated using the other session private key) without a timestamp authority or equivalent technique. Therefore, in this sense, the proposed scheme is more secure than a long-term private key with a physical smart card. Conclusively, in the proposed scheme, the user mobility is securely supported.

References

1. Perlman, R., Kaufman, C.: Secure password-based protocol for downloading a private key. Proc. ISOC Network and Distributed System Security Sympositum (1999)
2. Ford, W., Kaliski, B.: Server-assisted generation of a strong secret from a password. Proc. IEEE International Workshop on Enterprise Security (2000)
3. Jablon, D.: Password authentication using mulitple servers. Lecture Notes in Computer Science **2020** (2001) 344–360

4. Ganesan, R., Yaksha: Argumenting kerberos with public-key cryptography. Proceedings of the ISOC Network and Distributed System Security Symposium (1995)
5. R. Sandhu, M.B., Ganesan, R.: Password-enabled pki: Virtual smart-cards versus virtual soft token. Proc. of 1th Annual PKI Resarch Workshop (2002) 89–96
6. Wang, X.: Intrusion-tolerant passwqord-enabled pki. In: In Proceedings of the 2nd Annual PKI Research Workshop. (2004) 44–53
7. Kwon, T.: Virtual software tokens - a practical way to secure pki roaming. In Davida, G.I., Frankel, Y., Rees, O., eds.: InfraSec. Volume 2437 of Lecture Notes in Computer Science., Springer (2002) 288–302
8. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: Proc. IEEE Symposium on Research in Security and Privacy. (1992) 72–84
9. Jablon, D.: Strong password-only authenticated key exchange. In: Proceedings RSA Conference. Lecture Notes in Computer Science, Internet Society (2001)
10. Wu, T.: The secure remote password protocol. In: Proceedings of the Symposium on Network and Distributed Systems Security (NDSS '98), San Diego, California, Internet Society (1998) 97–111
11. Haber, S., Stornetta, W.S.: How to time-stamp a digital document. J. Cryptology **3** (1991) 99–111
12. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology **13** (2000) 361–396
13. Dierks, T., Allen, C.: The TLS Protocol Version 1.0. IETF. (1999)
14. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inform. Theory **31** (1985) 469–472
15. Tsiounis, Y., Yung, M.: On the security of elgamal based encryption. In: Public Key Cryptography. (1998) 117–134
16. Bellare, M., Miner, S.K.: A forward-secure digital signature scheme. In Wiener, M.J., ed.: CRYPTO. Volume 1666 of Lecture Notes in Computer Science., Springer (1999) 431–448
17. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In Biham, E., ed.: EUROCRYPT. Volume 2656 of Lecture Notes in Computer Science., Springer (2003) 255–271

Separable Identity-Based Deniable Authentication: Cryptographic Primitive for Fighting Phishing[⋆]

Willy Susilo[⋆⋆] and Yi Mu

Centre for Information Security Research
School of Information Technology and Computer Science
University of Wollongong
Wollongong 2522, Australia
{wsusilo, ymu}@uow.edu.au

Abstract. Phishing emails are one of today's most common and costly forms of digital identity theft. They are now very convincing that even experts cannot tell what is and is not genuine. In a phishing attack, victims are lured by an official looking email to a fraudulent website that appears to be that of a legitimate service provider. Such attacks can be mitigated with digitally-signed emails. Unfortunately, traditional digital signatures will destroy the traditional repudiability of email and they also require the unrelialistic adoption of a Public Key Infrastructure. To overcome this problem, we introduce a new cryptographic primitive called *separable identity-based deniable authentication*. Firstly, we present a generic construction of such a scheme, and proceed with an efficient construction based on bilinear pairing, which is an instantiation of our generic construction. This construction is an affirmative answer to the open question proposed by Adida, Hohenberger and Rivest [AHR05$^+$].

Keywords: phishing, email, repudiable, separable, ID-based, deniable, authentication.

1 Introduction

Phishing attacks are the act of sending an e-mail to a user falsely claiming to be an established genuine enterprise in an attempt to lure the user to a fraudulent website so that the user will surrender his/her private information. Over the past year, phishing attacks were launched pretending to be known services, such as AOL, eBay and many bank institutions, with an estimated cost of identity theft to these companies and their consumers surpassing \$ 10 billion dollars [APWG$^+$]. The consequences of phishing attacks are devastating to email as a communication medium. Banking institutions have been reduced to recommending that their users not selecting on links in emails [AHR05]. The

[⋆] This work is partially supported by ARC Linkage Project Grant LP0667899.

[⋆⋆] This work is partially supported by ARC Discovery Grant DP0663306.

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 68–80, 2006.

very openness that originally made email easy to use is now threatening to make the medium completely unusable.

Defenses against phishing attacks exist, but none of them is satisfactory. For instance, the Anti-Phishing Working Group suggests to authenticate all emails using standard digital signatures like PGP or S/MIME [APWG]. We note that this simple solution will *not* solve the problem entirely since its adoption is unlikely due to the need of a widespread public key infrastructure (PKI) and the non-repudiability of digital signatures that will *destroy* the property of traditional email (i.e. deniability). We aim to find an alternative to the traditional digital signature solution *without* losing the inherent properties of the email. Email is currently repudiable. The use of traditional digital signatures will harm this property and strip email users of their privacy, since emails might become legally binding.

Adida, Hohenberger and Rivest suggested a notion of separable identity-based ring signature (SIBR) to fight phishing attacks [AHR05$^+$]. In their construction, they incorporate a ring signature scheme (eg. [AOS02, ZK02]) to construct a SIBR. Their solution is acceptable since it retains the email property together with not relying on a PKI infrastructure. Nonetheless, their construction depends on the existence of a ring signature scheme. Additionally, they also pointed out that an identity-based deniable signatures could be one of the possible solutions to solve phishing attacks, but there is *no* known construction available [AHR05$^+$], and the construction has been posed as an open problem.

Our Contribution

In this paper, we answer the question proposed in [AHR05$^+$] affirmatively, by presenting a generic construction of separable identity-based deniable signature. The term *separable* in this context refers to cross domains between the two parties, namely the sender and the receiver. We cannot expect both sender and receiver to use an agreed public parameter as this will become unrealistic. The notion of separability makes our new notion practical, since users select a *master* of their choice and cryptographic schemes operate across various masters. In the context of email, a user's master will simply be her email domain, for example Alice with email address `alice@earth.com`, will derive her public key from the Private Key Generator (PKG) at `earth.com`. In our new notion, we retain the property of traditional email system, namely *sender repudiability* based on *recipient forgeability.* Intuitively, a user Alice can send an email to Bob with a separable ID-based deniable signature attached to it, so that Bob will believe that the email is indeed from Alice, but Bob cannot convince anyone else about the fact that Alice was the real signer.

1.1 Related Work

In [RST01], the notion of *ring signatures* was formalized and an efficient scheme based on RSA was proposed. A ring signature scheme allows a signer who knows at least one piece of secret information (or trapdoor information) to produce a sequence of n random permutations and form them into a ring. This signature can be used to convince any third party that one of the people in the group (who knows the trapdoor information) has authenticated the message on behalf of the group. The authentication provides *signer ambiguity*, in the sense that no one can identify who has actually signed the message. In [AOS02], a method to construct a ring signature from different types of public keys, such as these for integer factoring based schemes and discrete log based schemes, was proposed. The proposed scheme is more efficient than [RST01]. The formal security definition of a ring signature is also given in [AOS02].

The notion of *undeniable signature* is proposed by Chaum and van Antwerpen [C89] in 1989, to allow a signer to have complete control over her signature. In this scheme, the verification of the signer's signature requires the participation of the signer in an interactive protocol. The signer is able to reject invalid signatures, but she must not be able to deny valid signatures. If the signer is unavailable or unwilling to cooperate, the signature would not be longer verifiable. To overcome this shortcoming, the notion of *confirmer signature* [C94] is proposed. In confirmer signatures, the ability to verify or deny signatures is transferred to a designated confirmer. A generic construction of a confirmer signature from an ordinary signature scheme is proposed in [CM00].

Motivated by the above problem, Jakobsson, Sako and Impagliazzo proposed a *designated verifier* signatures in [JSI96]. This signature scheme is the first non-interactive undeniable signature scheme that transforms Chaum's scheme [C90] into non-interactive verification using a designated verifier proof. In a designated verifier scheme, the signature provides authentication of a message without providing a non-repudiation property of traditional signatures. A designated verifier scheme can be used to convince a single third party, i.e. the designated verifier, and only the designated verifier who can be convinced about its validity or invalidity. This is due to the fact that the designated verifier can always create a signature intended for himself that is indistinguishable from an original signature. This scheme does not require any interaction with the presumed signer to verify the authenticity of the message. Following this idea, Galbraith and Mao proposed a non-interactive undeniable signature scheme in finite fields [GM03] in the multi-user setting to have *invisibility* and *anonymity*. In [LQ04], Libert and Quisquater proposed an identity based undeniable signature scheme that can be regarded as identity based version of Galbraith-Mao's scheme using pairings.

As noted in [RST01], ring signature schemes can be used to provide this mechanism by joining the verifier in the ring. However, it might not be practical in the real life since the verifier might not have any public key setup. In [Des03], Desmedt raised the problem of generalizing the designated verifier signature concept to a multi designated verifier scheme. This question was answered in [LV04], where a construction of multi designated verifiers signature scheme was proposed. The main idea of this scheme is to use a ring signature scheme to convince a group of verifiers on the authenticity of a signed message.

Dwork, Naor and Sahai proposed *deniable authentication* in [DNS98]. Deniable authentication provides a system that addresses the deniability aspects, i.e. the protocol does not leave any paper trail for the authentication of the message. This work allows a single signer to achieve this property.

In [Naor02], the notion of ring signatures was combined with deniable authenticaton [DNS98]. The result is called *Deniable Ring Authentication* that allows a signer to authenticate a message m on behalf of an ad hoc collection of users and to convince a verifier that this authentication is done correctly. Moreover, the verifier cannot convince any third party that the message m was indeed authenticated. There is no 'paper trail' of the conversation, other than what could be produced by the verifier alone, as in zero-knowledge [Naor02]. However, the verification is done interactively, and hence, the requirement of having an anonymous routing, such as MIX-nets, is essential. Moreover, as a result of the requirement of this new notion, the message size is longer compared to a normal ring signature. We presented the non-interactive version of this notion in [SM03, SM04].

1.2 Organization of the Paper

The rest of this paper is organized as follows. In the next Section, we will review some cryptographic tools required throughout this paper. In Section 3, we present the model of separable identity-based (or ID-based, for short) deniable signature schemes and their security requirements. In Section 4, we present a generic construction of separable ID-based deniable signature schemes. We also provide an instantiation of our generic construction based on the ID-based signature scheme proposed in [CC03] and ID-based chameleon hash function proposed in [ZSS03]. Section 5 concludes the paper.

2 Cryptographic Tools

2.1 Basic Concepts of Bilinear Pairings

Let $\mathbb{G}_1, \mathbb{G}_2$ be cyclic additive groups generated by P_1, P_2, respectively, whose order are a prime q. Let $\mathbb{G}_M$ be a cyclic multiplicative group with the same order q. We assume there is an isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ such that $\psi(P_2) = P_1$. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_M$ be a bilinear mapping with the following properties:

1. *Bilinearity*: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P \in \mathbb{G}_1, Q \in \mathbb{G}_2, a, b, \in \mathbb{Z}_q$.
2. *Non-degeneracy*: There exists $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ such that $e(P, Q) \neq 1$.
3. *Computability*: There exists an efficient algorithm to compute $e(P, Q)$ for all $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$.

For simplicity, hereafter, we set $\mathbb{G}_1 = \mathbb{G}_2$ and $P_1 = P_2$. We note that our scheme can be easily modified for a general case, when $\mathbb{G}_1 \neq \mathbb{G}_2$.

Bilinear pairing instance generator is defined as a probabilistic polynomial time algorithm $\mathcal{IG}$ that takes as input a security parameter ℓ and returns a uniformly random tuple $param = (p, \mathbb{G}_1, \mathbb{G}_M, e, P)$ of bilinear parameters, including a prime number p of size ℓ, a cyclic additive group $\mathbb{G}_1$ of order q, a multiplicative group $\mathbb{G}_M$ of order q, a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_M$ and a generator P of $\mathbb{G}_1$. For a group $\mathbb{G}$ of prime order, we denote the set $\mathbb{G}^* = \mathbb{G} \setminus \{\mathcal{O}\}$ where $\mathcal{O}$ is the identity element of the group.

2.2 Chameleon Hashing and ID-Based Chameleon Hashing

Chameleon hashing (or *trapdoor commitment*) is basically non-interactive commitment schemes as proposed by Brassard, Chaum and Crepeau [BCC88]. The idea of chameleon hash functions was introduced and formalized in [KR97] in the construction of their chameleon signature schemes. The name "chameleon" refers to the ability of the owner of the trapdoor information to change the input to the function to any value of his choice without changing the resulting output.

A chameleon hash function is associated with a pair of public and private keys and has the following properties [KR97]: (1) Anyone who knows the public key can compute the associated hash function. (2) For people who do not have the knowledge of the trapdoor (i.e. the secret key), the hash function is collision resistant: it is infeasible to find two inputs which are mapped to the same output. (3) The trapdoor information's holder can easily find collisions for every given input. Several constructions of chameleon hashing have been proposed by Krawczyk and Rabin [KR97] which are based on discrete log, Catalano *et al.* [CGHN01] which is based on the hardness of deciding whether an element is a "small" e-th residue modulo N^2 and Bresson *et al.* [BCP03] which is based on modulo N^2.

The idea of chameleon hashing has been extended in [AM04] to construct an identity-based chameleon hash. An ID-based chameleon hash scheme is defined by a family of efficiently computable algorithms ($\mathsf{Setup}, \mathsf{Extract}, \mathsf{Hash}, \mathsf{Forge}$) as follows.

- Setup: A probabilistic algorithm that is run by a trusted authority TA to generate a pair of keys $\mathcal{SK}$ and $\mathcal{PK}$ defining the scheme. TA publishes $\mathcal{PK}$ and keeps $\mathcal{SK}$ secret.
- $\mathsf{Extract}$: A deterministic algorithm that accepts $\mathcal{SK}$ and an identity string ID and outputs the trapdoor information $\mathcal{T}$ associated with the identity ID.
- Hash: A probabilistic algorithm that accepts $\mathcal{PK}$, an identity string ID and a message m to produce a hash value h.
- Forge: An algorithm that, on input $\mathcal{PK}$, an identity string ID, the trapdoor information $\mathcal{T}$ associated with ID, a message m', and a hash value $h = \mathsf{Hash}(\mathcal{PK}, \mathsf{ID}, m)$, where $m \neq m'$, outputs a sequence of random bits that correspond to a valid computation of $\mathsf{Hash}(\mathcal{PK}, \mathsf{ID}, m')$ yielding a collision on the same target value h.

Related to this definition is the notion of *collision forgery* defined [AM04] as follows.

Definition 1. *A collision forgery strategy is a probabilistic algorithm that given identity string* ID*, a message* m *and random bits* r*, outputs another message* m' *and random bits* r'*, where* $m \neq m'$ *and* $r \neq r'$*, such that* $\mathsf{Hash}(\mathsf{ID}, m, r) = \mathsf{Hash}(\mathsf{ID}, m', r')$ *with non-negligible probability.*

A hashing scheme is said to be *secure against existential collision forgery by passive attacks* if no collision-forgery strategy against it exists.

The semantic security for chameleon hashing scheme is defined as follows [AM04].

Definition 2. *The chameleon hashing scheme is said to be semantically secure if for all identity strings* ID *and all pairs of messages* (m, m')*, the probability distributions of the random variables* $\mathsf{Hash}(\mathsf{ID}, m, r)$ *and* $\mathsf{Hash}(\mathsf{ID}, m', r')$ *are computationally indistinguishable.*

In [AM04], an ID-based chameleon hash function based on factorization is proposed. It is also shown an application of ID-based chameleon hash function for a sealed-bid auction system.

An ID-based chameleon hash function from bilinear pairing has been constructed in [ZSS03], which is defined as follows.

- **Setup:** PKG chooses a random number $s \in \mathbb{Z}_q^*$ and sets $P_{pub} = sP$. Define a cryptographic hash function: $H_0 : \{0,1\}^* \rightarrow \mathbb{G}_1$. Define another cryptographic hash function: $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$. PKG publishes $\{\mathbb{G}_1, \mathbb{G}_2, e, q, \lambda, P, P_{pub}, H_0, H_1\}$ and keeps s as the *master-key*, which is known only by itself.
- **Extract:** A user submits his identity information ID to PKG. PKG computes the user's public key as $\mathsf{Q}_{\mathsf{ID}} = H_0(ID)$, and returns $\mathcal{S}_{\mathsf{ID}} = s\mathsf{Q}_{\mathsf{ID}}$ to the user as his private key.
- **Hash:** Given a message m, choose a random element R from $\mathbb{G}_1$, define the hash as

$$Hash(P_{pub}, \mathsf{ID}, m, R) = e(R,\ P)e(H_1(m)H_0(\mathsf{ID}),\ P_{pub}).$$

- **Forge**

$$Forge(P_{pub}, \mathsf{ID}, \mathcal{S}_{\mathsf{ID}}, m, R, m') = R' = (H_1(m) - H_1(m'))\mathcal{S}_{\mathsf{ID}} + R.$$

We refer the reader to [ZSS03] for the correctness and security of this scheme.

3 Separable Identity-Based Deniable Signatures

In this section, firstly we provide a formal definition of a separable ID-based deniable signature scheme.

Definition 3 (Separable ID-based Deniable Signature Scheme). *Given an integer* ℓ*, a separable ID-based deniable signature scheme* `SIDDS` *with security parameter* ℓ *is defined by the following.*

- *A common parameter generation algorithm* Setup*: it is a probabilistic algorithm which takes* ℓ *as input. The outputs are public parameters. The public parameters for the sender can be different from the ones for the receiver.*
- *PKG's key generation algorithm* `Keygen`*: it is a probabilistic algorithm that takes the public parameters and a security parameter* ℓ *as inputs, and outputs a pair of keys* $(\mathcal{PK}_{\mathtt{PKG}}, \mathcal{SK}_{\mathtt{PKG}})$.

- *Key extraction* $\mathsf{Extract}$*: it is a deterministic algorithm that takes an identity of a user* ID *and a PKG's secret key* $\mathcal{SK}_{\mathtt{PKG}}$ *and outputs the secret key of the user* $\mathcal{S}_{\mathsf{ID}}$*.*
- *Linking algorithm* `Link`*: it is a deterministic algorithm that "link" the two domains between the public parameters of the signer and the receiver. The input of this algorithm is the domain used by the sender, and the output of this algorithm is the domain used by the receiver.*
- *Signing algorithm* `Sign`*: it is a probabilistic algorithm that takes a message* $m \in \{0,1\}^*$*, an identity of the receiver, an identity of the signer, a signer's secret key and output an ID-based deniable signature* σ*. Upon the creation of the signature, the* `Link` *algorithm is invoked.*
- *Verification algorithm* `Verify`*: it is a deterministic algorithm that takes a message* m*, a signature* σ*, an identity of the receiver, an identity of the signer and outputs* `True` *if the signature is valid, or* $\perp$ *otherwise. Upon the verification algorithm, the* `Link` *algorithm is invoked.*

Additionally, an ID-based deniable signature scheme must satisfy the following properties:

- *Correctness*

$$Pr\left[True \leftarrow \mathtt{Verify}(\mathtt{m}, \sigma, \mathsf{ID}_{\mathtt{R}}, \mathsf{ID}_{\mathtt{S}}) | \sigma \leftarrow \mathtt{Sign}(\mathtt{m}, \mathsf{ID}_{\mathtt{R}}, \mathsf{ID}_{\mathtt{S}}, \mathcal{S}_{\mathsf{ID}_{\mathtt{S}}})\right] = 1$$

- *Unforgeability*: Given an identity ID_A, it is computationally infeasible without the knowledge of the secret key of either A or B to produce a valid ID-based deniable signature for B that will be accepted by the verifying algorithm.
- *Signer's identity privacy*: Given a message $m \in \{0,1\}^*$ and a valid ID-based deniable signature σ between A and B, it is *infeasible* for a third party to determine who created the signature, even if one knows all A's and B's secret keys.
- *Separability*: The public parameters used by the sender (or signer) can be *separate* from the ones used by the receiver. The link between the two domains is connected by the `Link` algorithm.

3.1 Formal Security Notion

Existential Unforgeability Under a Chosen Message Attack
We provide a formal definition of existential unforgeability of a separable ID-based deniable signature scheme under a chosen message attack. To achieve this, we extend the definition of existential unforgeability against a chosen message attack of [GMR88]. Our extension can capture an adversary who can simulate and observe the scheme. It is defined using the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

- Setup: $\mathcal{C}$ runs Setup for a given security parameter ℓ to obtain a public parameter `param`, and the PKG's public key P_{pub}. The associated PKG's secret key is kept secret by $\mathcal{C}$. $\mathcal{C}$ also runs `Keygen` algorithm. The public key P_{pub} is provided to $\mathcal{A}$.

- $\mathsf{Extract}$ Queries: $\mathcal{A}$ can request the private keys corresponding to any identity ID_i, for $1 \leq i \leq q_{ex}$, where q_{ex} denotes the number of extraction queries, polynomially bounded in ℓ. As a response to each query, $\mathcal{C}$ runs $\mathsf{Extract}$ using ID_i as input and returns a resulting secret key $\mathcal{S}_{\mathsf{ID}_i}$.
- $\mathtt{Sign}$ Queries: $\mathcal{A}$ can request a signature on a message m_j, $1 \leq j \leq q_m$, from an identity ID_i to ID_k, where q_m denotes the number of signature queries polynomially bounded in ℓ. In response, $\mathcal{C}$ runs $\mathsf{Extract}$ to obtain the secret key of ID_i, $\mathcal{S}_{\mathsf{ID}_i}$, and then runs $\mathtt{Sign}$ using ID_i, $\mathcal{S}_{\mathsf{ID}_i}$ and m_j as inputs and returns a resulting signature σ_j for the message m_j.
- $\mathtt{Verify}$ Queries: Answers to these queries are not provided by $\mathcal{C}$ since $\mathcal{A}$ can compute them for himself using the $\mathtt{Verify}$ algorithm.
- $\mathtt{Output}$: Finally, $\mathcal{A}$ outputs a tuple $(\mathsf{ID}_i, \mathsf{ID}_j, \sigma)$ for a signer ID_i to a receiver ID_j. $\mathcal{A}$ wins the game if $\mathtt{Verify}(\mathsf{ID}_\mathtt{i}, \mathsf{ID}_\mathtt{j}, \sigma, \mathtt{m}) \stackrel{?}{=} \mathtt{True}$ holds; no secret key for ID_i and ID_j were issued during the $\mathsf{Extract}$ queries stage and σ was not obtained in $\mathtt{Sign}$ queries stage.

The success probability of an adversary to win the game is defined by

$$\mathtt{Succ}_{\mathcal{A}}^{UF-SIDDS-CMA}(\ell).$$

Definition 4. *We say that a separable ID-based deniable signature scheme is existentially unforgeable under a chosen message attack if the probability of success of any polynomially bounded adversary in the above game is negligible (ϵ). That is,*

$$\mathtt{Succ}_{\mathcal{A}}^{UF-SIDDS-CMA}(\ell) \leq \epsilon.$$

Signer's Identity Privacy
A formal notion of signer's identity privacy under a chosen message attack ($\mathtt{SIP\text{-}SIDDS\text{-}CMA}$) is defined as follows. We consider a $\mathtt{SIP\text{-}SIDDS\text{-}CMA}$ attacker $\mathcal{A}$ in the random oracle model. During the learning stages, the attacker takes two signing identities ID_{A_0} and $\mathsf{ID}_{\mathcal{A}_1}$, and a receiver's identity ID_B, and outputs a message m^* together with some state information t. After the learning stage, $\mathcal{A}$ obtains a challenge signature σ^* directed to ID_B, which is formed by signing the message m^* at random under one of the two secret keys from the identities and the information t. $\mathcal{A}$'s task is to determine which key was chosen during this stage. The adversary has access to the random oracles H, to the signing oracles and to the verifiying oracles, and is allowed to invoke them on any message with the restriction of not querying (m^*, σ^*) from the verifying oracle in the second stage. The success probability of an adversary to win the game is defined by

$$\mathtt{Succ}_{\mathcal{A}}^{SIP-SIDDS-CMA}(\ell)$$

Definition 5. *We say that a separable ID-based deniable signature scheme provides signer's identity privacy under a chosen message attack if the probably of success of any polynomially bounded adversary in the above game is negligible (ϵ). That is,*

$$\mathtt{Succ}_{\mathcal{A}}^{SIP-SIDDS-CMA}(\ell) \leq \frac{1}{2} + \epsilon$$

4 Generic Construction of Separable ID-Based Deniable Signatures

In this section, we provide a generic construction of separable ID-based deniable signature.

Let ($\texttt{IDSetup}, \texttt{IDExtract}, \texttt{IDSign}, \texttt{IDVerify}$) be ID-based signature setup algorithm, ID-based key extraction algorithm, ID-based signing and ID-based verifying algorithm, respectively. We note that any ID-based signature scheme can be used for this purpose (eg. [Sha85,CC03]). Let ($\texttt{Hash}$– $\texttt{Setup}$, $\mathsf{Extract}$, Hash, Forge) be an ID-based chameleon hash function as defined in [AM04]. The generic construction of separable ID-based deniable signatures is as follows.

- Setup: Run $\texttt{IDSetup}$ for both sender and verifier.
- $\texttt{KeyGen}$: Run key generation algorithm by the PKG for both sender and verifier.
- $\texttt{Extract}$: Run the $\mathsf{Extract}$ algorithm for sender and verifier.
- $\texttt{Link}$: Define a public *full-domain hash function* that maps from the domain of the group used by the sender to the domain of the group used by the receiver.
- $\texttt{Sign}$: To sign a message $m \in \{0,1\}^*$, the signer performs the following.

$$\begin{cases} \texttt{Hash} - \texttt{Setup}; & \\ \tilde{m} \leftarrow \mathsf{Hash}(m,r) & for\ a\ random\ r; \\ \bar{m} \leftarrow \texttt{Link}(\tilde{m}); & \\ \sigma \leftarrow \texttt{IDSign}(\bar{m}, \mathcal{S}_{\mathsf{ID}_A}, \mathsf{ID}_B). & \end{cases}$$

The signature on m is (σ, r).

- $\texttt{Verify}$: To verify a message signature pair (m, σ, r), the receiver performs the following.

$$\begin{cases} \texttt{Hash} - \texttt{Setup}; & \\ \tilde{m} \leftarrow \mathsf{Hash}(m,r) & for\ a\ random\ r; \\ \bar{m} \leftarrow \texttt{Link}(\tilde{m}); & \\ Result \leftarrow \texttt{IDVerify}(\bar{m}, \mathsf{ID}_A, \mathsf{ID}_B). & \end{cases}$$

Check whether *Result* is $True$ or not. If it is $True$, then accept the signature. Otherwise, reject.

Sender Repudiability Based on Recipient Forgeability

We note that the above generic construction satisfies sender repudiability. This is achieved by the ability of the recipient to forge the signature by generating m^* of his choice and execute Forge algorithm to find another pair of (m^*, r^*), where $m^* \neq m$, that also satisfies the verification algorithm. As a result of the Forge algorithm, another message that will collide with the same chameleon hash value will be computed. Therefore, any third party cannot be convinced with the authenticity of the message-signature pair as the recipient can always forge the message. On the other hand, the recipient will be convinced with the authenticity of the message-signature pair since he/she has not generated the pair himself/herself.

Existential Unforgeability Under a Chosen Message Attack
The notion of existential unforgeability under a chosen message attack for our generic construction is ensured by the underlying identity-based signature scheme that is used in the construction.

4.1 An Example

In this Section, we provide an instantiation of our generic construction and present a separable ID-based deniable signature based on bilinear pairing. Our construction is based on the ID-based signature scheme proposed in [CC03] and ID-based chameleon hash function proposed in [ZSS03]. The scheme is as follows.

- `Setup` & `Keygen`: The setup algorithm includes the setup algorithms for both signer and receiver. In the following, we use an index S to indicate the signer, and an index R to indicate the receiver. For both signer and receiver, PKG_i, $i \in \{S, R\}$, chooses a random number $s_i \in \mathbb{Z}_q^*$ and sets $P_{pub_i} = s_i P$. PKG_i also publishes system parameter $\{\mathbb{G}_{1_i}, \mathbb{G}_{2_i}, e_i, q_i, \ell_i, P_i, H_{0_i}, H_{1_i}\}$ and keeps s_i as the master key, which is only known to PKG_i. Here $H_{1_i} : \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{Z}_{q_i}^*$, $H_{0_i} : \{0,1\}^* \to \mathbb{G}_{1_i}^*$.
- `Extract`: A user submits his/her identity information ID_i to PKG_i, and PKG_i computes the user's public key as $\mathsf{Q}_{\mathsf{ID}_i} = H_{0_i}(\mathsf{ID}_i)$ and returns the user's secret key $\mathcal{S}_{\mathsf{ID}_i} = s_i \mathsf{Q}_{\mathsf{ID}_i}$ to the user via a private channel.
- `Link`: The linking algorithm is defined as follows:

  $$Link : \mathbb{G}_{1_S}^* \to \mathbb{G}_{1_R}^*$$

  which is a cryptographic hash function that maps from the group $\mathbb{G}_{1_S}^*$ used by the signer to the group $\mathbb{G}_{1_R}^*$ used by the receiver.
- `Sign`: To sign a message $m \in \{0,1\}^*$, the signer performs the following.
  1. Generate an ID-based chameleon hash function from the receiver's identity as follows.

     $$\mathsf{Hash}(P_{pub_R}, \mathsf{ID}_R, m, T) = e_R(T, P_R) e(H_{1_R}(m) H_{0_R}(\mathsf{ID}_R), P_{pub_R})$$

     for a randomly chosen $T \in \mathbb{G}_{1_R}$.
  2. Select a random $T \in \mathbb{G}_{1_R}$ and compute

     $$\tilde{m} = \mathsf{Hash}(P_{pub_R}, \mathsf{ID}_R, m, T)$$

  3. Execute the `Link` algorithm as follows.

     $$\bar{m} = \mathtt{Link}(\tilde{m})$$

  4. Select an integer $r \in \mathbb{Z}_{q_S}^*$ and perform the following.
     - $U = r\mathsf{Q}_{\mathsf{ID}_S}$.
     - $h = H_{0_S}(\bar{m} || U)$.
     - $V = (r + h)\mathcal{S}_{ID_S}$.

     The signature on a message m is $(U, V, T) \in \mathbb{G}_{1_S}^3$.

– `Verify`: To verify the signature, the receiver will perform the following.
  1. Generate an ID-based chameleon hash function from the receiver's identity as follows.
  $$\mathsf{Hash}(P_{pub_R}, \mathsf{ID}_R, m, T) = e_R(T, P_R)e(H_{1_R}(m)H_{0_R}(\mathsf{ID}_R), P_{pub_R})$$
  for a randomly chosen $T \in \mathbb{G}_{1_R}$.
  2. Compute $\bar{m} \leftarrow \mathtt{Link}(\mathsf{Hash}(P_{pub_R}, \mathsf{ID}_R, m, T))$.
  3. Compute $h = H_{0_S}(\bar{m}||U)$.
  4. Test whether
  $$e_S(P, V) \overset{?}{=} e_S(P_{pub_S}, U + h\mathsf{Q}_{\mathsf{ID}_S})$$
  holds with equality. If so, then output `True`. Otherwise, output $\perp$.

Sender Repudiability Based on Recipient Forgeability
We note that the above construction is *repudiable.* This is due to the chameleon hash function used in the construction. The receiver can always execute Forge algorithm to find another message $m^* \neq m$ that will also satisfy the signature verification. This way, the signature is repudiable. □

Application of Separable ID-Based Deniable Signatures to Mitigate Phishing
Using a separable ID-based deniable signature scheme, we achieve the following. The sender (Alice) can send a message to a receiver (Bob), in such a way that (1) Bob believes that the message is indeed sent by Alice; and (2) Bob cannot convince any other third party about this fact. We note that we have achieved the deniable property of email systems using the sender repudiability feature. The reason why any third party will *not* believe with the fact that Alice has produced a valid signature is due to the problem that Bob can create such a signature which is indistinguishable from what Alice has produced.

Consider a situation where Mastercard would like to inform Bob that Bob needs to change his password. Mastercard will act as a *sender* in our scheme, and Bob (i.e. the recipient) will believe the authenticity of the message (since the signature can only be generated either by Mastercard or Bob himself, but since he does not generate the signature, then it must really come from Mastercard). Additionally, we do not lose the email property, namely repudiability (due to the sender repudiability based on recipient forgeability). Now, consider another situation where phishing happens. A sender will pretend to be Mastercard and send a message to Bob. However, Bob can verify that the message is *not* sent by Mastercard and therefore, phishing can be avoided. We note that in this situation, the verification algorithm will *not* hold.

Efficiency Comparison
Let $|q|$ denote the number of bits used to represent q. In the scheme presented in this Section, the signature length is $|2\mathbb{G}_{1_S} + \mathbb{G}_{1_R}|$. In practice, this can be upperbounded by $3|q|$. In the construction presented in [AHR05$^+$] that is based on ring signature scheme, we also require $3|q|$ bits signature length, which is comparable to our construction.

5 Conclusion

In this paper, we presented a new notion of separable identity-based deniable signature. Our new notion can be used to mitigate phishing. We also provide a generic construction of such scheme, and we conclude with an instantiation of our generic construction based on the schemes presented in [CC03] and [ZSS03].

Acknowledgement

We would like to express our gratitude thanks to Yong Li and the anonymous referees of EuroPKI 2006 for their invaluable suggestions to improve this paper.

References

[AOS02] M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n Signatures from a Variety of Keys. *Advances in Cryptology - Asiacrypt 2002, Lecture Notes in Computer Science 2501*, pages 415 – 432, 2002.

[AHR05] B. Adida, S. Hohenberger, and R. L. Rivest. Fighting Phishing Attacks: A Lightweight Trust Architecture for Detecting Spoofed Emails. *DIMACS Workshop on Theft in E-Commerce, April 2005*, April 2005.

[AHR05+] B. Adida, S. Hohenberger, and R. L. Rivest. Separable Identity-Based Ring Signatures: Theoretical Foundations for Fighting Phishing Attacks. *DIMACS Workshop on Theft in E-Commerce*, April 2005.

[APWG] Anti-Phishing Working Group. Digital Signatures to Fight Phishing Attacks. *Available at http://www.anthiphishing.org/smim-dig-sig.htm.*

[APWG+] Anti-Phishing Working Group. Phishing Activity Trends Report. *Available at http://www.antiphishing.org*, November 2004.

[AM04] G. Ateniese and B. de Medeiros. Identity-based Chameleon Hash and Applications. *Financial Cryptography 2004, Lecture Notes in Computer Science 3110*, pages 164 – 180, 2004.

[BCC88] G. Brassard, D. Chaum, and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *JCSS, 37(2)*, pages 156 – 189, 1988.

[BCP03] E. Bresson, D. Catalano, and D. Pointcheval. A Simple Public-Key Cryptosystem with a Double Trapdoor Decryption Mechanism and Its Applications. *Advances in Cryptology - Asiacrypt 2003, Lecture Notes in Computer Science 2894*, pages 37 - 54, 2003.

[CM00] J. Camenisch and M. Michels. Confirmer signature schemes secure against adaptive adversaries. *Advances in Cryptology - Eurocrypt 2000, Lecture Notes in Computer Science 1807*, 2000.

[CGHN01] D. Catalano, R. Gennaro, N. Howgrave-Graham, and P. Q. Nguyen. Paillier's Cryptosystem Revisited . *ACM CCS 2001*, 2001.

[CC03] J. Cha and J. H. Cheon. An Identity-based Signature from Gap Diffie-Hellman Groups. *Public Key Cryptography (PKC 2003), Lecture Notes in Computer Science 2567*, pages 18 – 30, 2003.

[C90] D. Chaum. Zero-knowledge undeniable signatures. *Advances in Cryptology - Eurocrypt '90*, pages 458–464, 1990.

[C94] D. Chaum. Designated Confirmer Signatures. *Advances in Cryptology - Eurocrypt '94, Lecture Notes in Computer Science 950*, pages 86 – 91, 1994.

[C89] D. Chaum and H. van Antwerpen. Undeniable signatures. *Advances in Cryptology - Crypto '89, Lecture Notes in Computer Science 435*, pages 212–216, 1990.

[Des03] Y. Desmedt. Verifier-Designated Signatures. *Rump Session, Crypto 2003*, 2003.

[DNS98] C. Dwork, M. Naor, and A. Sahai. Concurrent Zero-Knowledge. *Proc. 30th ACM Symposium on the Theory of Computing*, pages 409 – 418, 1998.

[GM03] S. Galbraith and W. Mao. Invisibility and Anynimity of Undeniable and Confirmer Signatures. *CT-RSA 2003, Lecture Notes in Computer Science 2612*, pages 80 – 97, 2003.

[GMR88] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal of Computing*, 17/2:281–308, 1988.

[JSI96] M. Jakobsson, K. Sako, and R. Impagliazzo. Designated Verifier Proofs and Their Applications. *Advances in Cryptology - Eurocrypt '96, Lecture Notes in Computer Science 1070*, pages 143 – 154, 1996.

[KR97] H. Krawczyk and T. Rabin. Chameleon hashing and signatures. *Network and Distributed System Security Symposium, The Internet Society*, pages 143 – 154, 2000.

[LV04] F. Laguillaumie and D. Vergnaud. Multi-Designated Verifiers Signatures. *Sixth International Conference on Information and Communications Security (ICICS 2004), Lecture Notes in Computer Science*, pages 495 – 507, 2004.

[LQ04] B. Libert and J. J. Quisquater. Identity based Undeniable Signatures. *Topics in Cryptology, CT-RSA 2004, Lecture Notes in Computer Science 2964*, pages 112 – 125, 2004.

[Naor02] M. Naor. Deniable Ring Authentication. *Advances in Cryptology - Crypto 2002, Lecture Notes in Computer Science 2442*, pages 481 – 498, 2002.

[RST01] R. L. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret. *Advances in Cryptology - Asiacrypt 2001, Lecture Notes in Computer Science 2248*, pages 552 – 565, 2001.

[Sha85] A. Shamir. Identity-based cryptosystems and signature schemes. *Advances in Cryptology - Crypto '84, Lecture Notes in Computer Science 196*, pages 47–53, 1985.

[SM03] W. Susilo and Y. Mu. Non-Interactive Deniable Ring Authentication. *The 6th International Conference on Information Security and Cryptology (ICISC 2003), Lecture Notes of Computer Science 2971*, pages 386 - 401, 2003.

[SM04] W. Susilo and Y. Mu. Deniable Ring Authentication Revisited. *The Second Applied Cryptography and Network Security (ACNS04), Lecture Notes in Computer Science 3089*, pages 149 - 163, 2004.

[ZK02] F. Zhang and K. Kim. ID-based Blind Signature and Ring Signature from Pairings. *Advances in Cryptology - Asiacrypt 2002, Lecture Notes in Computer Science 2501*, pages 533 – 437, 2002.

[ZSS03] F. Zhang, R. Safavi-Naini, and W. Susilo. ID-Based Chameleon Hashes from Bilinear Pairings. *Cryptology ePrint Archive, Report 2003/208*, 2003.

Breaking Yum and Lee Generic Constructions of Certificate-Less and Certificate-Based Encryption Schemes

David Galindo[1], Paz Morillo[2], and Carla Ràfols[2]

[1] Institute for Computing and Information Sciences,
Radboud University Nijmegen, P.O. Box 9010,
6500 GL, Nijmegen, The Netherlands
d.galindo@cs.ru.nl

[2] Universitat Politècnica de Catalunya,
C/Jordi Girona, 1-3 08034 Barcelona
{paz, crafols}@ma4.upc.edu

Abstract. Identity-based public key cryptography is aimed at simplifying the management of certificates in traditional public key infrastructures by means of using the identity of a user as its public key. The user must identify itself to a trusted authority in order to obtain the secret key corresponding to its identity. The main drawback of this special form of public key cryptography is that it is key escrowed. Certificate-based and certificate-less cryptography have been recently proposed as intermediate paradigms between traditional and identity-based cryptography, seeking to simplify the management of certificates while avoiding the key escrow property of identity-based cryptography. In this work we cryptanalyse the certificate-based and certificate-less encryption schemes presented by Yum and Lee at EuroPKI 2004 and ICCSA 2004 conferences.

Keywords: public-key infrastructure, identity-based encryption, certificate-based and certificate-less encryption, cryptanalysis.

1 Introduction

In traditional public key cryptography (PKC) the authenticity of the public keys must be certified by a trusted third party, which is called Certification Authority (CA). The infrastructure required to support traditional PKC is the main difficulty in its deployment. Many of the problems of any public key infrastructure arise from the management of certificates, which includes storage, revocation and distribution.

In 1984, Shamir proposed the concept of identity-based PKC, which sought to reduce the requirements on the public key infrastructure by using a well-known aspect of the client's identity as its public key. With this approach, certification becomes implicit. For instance, in the case of identity-based encryption (IBE), the sender of a message does not need to check whether the receiver is certified or not. Instead, prior to decryption, the receiver must identify himself to a trusted

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 81–91, 2006.

authority who is in possession of a master key. If the identification is successful, the authority sends the user his private key. The first practical provably secure IBE scheme was proposed by Boneh and Franklin in 2001, using bilinear maps on elliptic curves and it was proven secure in the random oracle model [7]. The main drawback of IBE is that it is inherently key escrowed, which limits the applicability of IBE.

Motivated by the above problem, the concept of certificate-based PKC was introduced by Gentry in [11]. In this model, certificates are needed to generate the user's secret key, so certification becomes implicit. In addition there is no key escrow, since the user's secret key is generated by joining both the certificate and a private information only known to the user. In a certificate-based encryption (CBE) scheme, senders are not required to obtain fresh information of receivers' certificate status; the receiver will be able to decrypt only if its public key is certified.

Independently from the previous work, the concept of certificate-less PKC was introduced by Al Riyami and Paterson in [1]. In contrast to traditional public key cryptographic systems, CL-PKC does not require the use of certificates to guarantee the authenticity of public keys. It does rely on the use of a trusted authority who is in possession of a master key. On the other hand, CL-PKC does not suffer from key escrow, since the authority does not have access to the user's private key. Several cryptographic primitives for certificate-less PKC were proposed in [1], including a certificate-less public key encryption (CL-PKE) scheme.

In contrast to IBE, the confidentiality of CBE and CL-PKE schemes must be protected against dishonest users as well as against the trusted authorities. Security notions taking into account these new scenarios were proposed in the seminal works [11, 1].

Thus, certificate-less PKC and certificate-based PKC can be conceptually seen as intermediates between traditional PKC and identity-based PKC. This idea motivated the work by Yum and Lee [15, 16], in which they tried to show a formal equivalence among IBE, CBE and CL-PKE. In particular, their intention was to show that IBE implies both CBE and CL-PKE by giving a generic construction from IBE to those primitives. To do so, they defined a weaker security model for CL-PKE than the original model introduced in [1]. Their generic constructions have been cited as sound constructions in the works [2, 3, 9, 12, 13][1].

Our contribution. In this paper we show that a dishonest authority can break the security of the three generic constructions of CBE and CL-PKE schemes given in [15, 16]. These constructions are inherently flawed due to a naive use of double encryption as highlighted in [10]. We stress that our attacks are within the restricted security model proposed by Yum and Lee, that is, *our results contradict* three of their theorems.

Related work. In a recent work, Libert and Quisquater [13] show that the transformation from IBE to CL-PKE in [15] due to Yum and Lee is insecure in

[1] In the work [13] only the transformations in [16] are regarded as valid constructions in the restricted security model.

the full original security model [1]. Their attack does not apply to the restricted security model of [15], and then it *does not contradict* Yum and Lee claim.

A generic construction from IBE to CBE was outlined by Dodis and Katz in [10]. They study how to perfom secure *multiple encryption*, i.e. the encryption of data using multiple, independent encryption schemes. They provide a generic construction of multiple encryption for public key encryption schemes and suggest how to use their ideas to obtain CBE secure constructions. In [2] a transformation from CL-PKE to CBE was proposed, but the security proof was only given for one of the two attacks that a CBE scheme has to withstand. Recent work [12] pointed out the impossibility of using the same techniques to prove security against the other type of attacks, calling into question the meaningfulness of that transformation.

Regarding CL-PKE, the generic constructions from IBE to CL-PKE we are aware of are to be found in [5, 13]. The drawback of these constructions is that they use the random oracle model heuristic, and therefore it is not actually guaranteed they are sound in the standard complexity model [8]. In [13] it is also pointed out that the generic construction IBE-to-CBE suggested in [10] also applies to the CL-PKE case, as long as the restricted security model of Yum and Lee is considered.

Therefore, designing a generic transformation from IBE to CL-PKE without random oracles in the full security model proposed in [1] remains an open problem to the best of our knowledge.

2 Definitions for Identity-Based Encryption

We begin by fixing some notation. If A is a non-empty set, then $x \leftarrow A$ denotes that x has been uniformly chosen in A. If A is a finite set, then $|A|$ denotes its cardinality.

An identity-based encryption scheme is specified by four probabilistic polynomial time (PPT) algorithms (see for instance [6]):

- ID.Gen takes a security parameter k and returns the system parameters `ID.pms` and master-key `ID.msk`. The system parameters include the description of sets $\mathcal{M}, \mathcal{C}$, which denote the set of messages and ciphertexts respectively. `ID.pms` is publicly available, while `ID.msk` is kept secret by the trusted authority.
- ID.Ext takes as inputs `ID.pms`, `ID.msk` and an arbitrary string $ID \in \{0,1\}^*$ and returns a private key d_{ID} to the user with identity ID. This must be done over a secure channel, since d_{ID} enables to decrypt ciphertexts under the identity ID.
- ID.Enc takes as inputs `ID.pms`, $ID \in \{0,1\}^*$ and $M \in \mathcal{M}$. It returns a ciphertext $C \in \mathcal{C}$.
- ID.Dec takes as inputs `ID.pms`, $C \in \mathcal{C}$ and a private key d_{ID}, and it returns $M \in \mathcal{M}$ or rejects.

Chosen ciphertext security. An IBE scheme is said to have indistinguishability against an adaptive chosen ciphertext attack (IND-ID-CCA) if any PPT algorithm $\mathcal{A}$ has a negligible advantage in the following game:

Setup. The challenger takes a security parameter k and runs the $\mathsf{ID.Gen}$ algorithm. It gives $\mathtt{ID.pms}$ to the adversary. It keeps $\mathtt{ID.msk}$ to itself.

Phase 1. The adversary issues queries of the form

- Extraction query $\langle ID_i \rangle$. The challenger runs algorithm $\mathsf{ID.Ext}$ to generate the private key d_i corresponding to ID_i. It sends d_i to the adversary.
- Decryption query $\langle ID_i, C_i \rangle$. The challenger generates the private key d_i. It then runs $\mathsf{ID.Dec}$ to decrypt C_i under ID_i.

These queries may be asked adaptively, that is, each query may depend on the answers obtained to the previous queries.

Challenge. The adversary outputs equal length plaintexts $M_0, M_1 \in \mathcal{M}$ and an identity ID_{ch}. The only constraint is that the private key for ID_{ch} was not requested in Phase 1. The challenger picks $b \leftarrow \{0,1\}$ and sets $C = \mathsf{ID.Enc}(\mathtt{ID.pms}, ID_{\mathsf{ch}}, M_b)$. It sends C to the adversary.

Phase 2. The adversary issues extraction and decryption queries as in Phase 1, with the restriction $\langle ID_i \rangle \neq \langle ID_{\mathsf{ch}} \rangle$ and $\langle ID_i, C_i \rangle \neq \langle ID_{\mathsf{ch}}, C \rangle$.

Guess. The adversary outputs a guess $b' \in \{0,1\}$.

Such an adversary is called an $\mathsf{IND\text{-}ID\text{-}CCA}$ adversary $\mathcal{A}$, and its advantage is defined as $\mathsf{Adv}_{\mathcal{E},\mathcal{A}}^{\mathsf{ID-CCA}}(1^k) = |\Pr[b = b'] - 1/2|$.

Definition 1. *An* IBE *system $\mathcal{E}$ is* secure under chosen ciphertext attacks *if for any probabilistic polynomial time* $\mathsf{IND\text{-}ID\text{-}CCA}$ *adversary $\mathcal{A}$ the function* $\mathsf{Adv}_{\mathcal{E},\mathcal{A}}^{\mathsf{CCA}}(1^k)$ *is negligible.*

3 Definitions for Certificate-Based Encryption

A certificate-based encryption scheme is a tuple of five PPT algorithms:

- $\mathsf{CB.Gen}$ is a probabilistic algorithm taking as input a security parameter k. It returns $\mathtt{CB.msk}$ (the certifier's master-key) and public parameters $\mathtt{CB.pms}$ that include the description of a string space Λ. Usually this algorithm is run by the CA. The system parameters include the description of sets $\mathcal{M}, \mathcal{C}$, which denote the set of messages and ciphertexts respectively.
- $\mathsf{CB.SetKeyPair}$ is a probabilistic algorithm that takes $\mathtt{CB.pms}$ as input[2]. It returns a pair public key - private key (PK, SK).
- $\mathsf{CB.Certify}$ is an algorithm that takes as input $\langle \mathtt{CB.msk}, \mathtt{CB.pms}, i, \mathtt{user}, PK \rangle$. It returns $Cert_i$, which is sent to the client. Here i identifies i-th time period, while $\mathtt{user} \in \Lambda$ contains other information needed to certify the client such as the client's identifying information, and PK is a public key.
- $\mathsf{CB.Enc}$ is a probabilistic algorithm taking as inputs $\langle \mathtt{CB.pms}, M, i, \mathtt{user}, PK \rangle$ where $M \in \mathcal{M}$ is a message. It returns a ciphertext $C \in \mathcal{C}$ for message M or $\perp$ if PK is not a valid public key.

[2] Actually, in the CBE generic construction by Yum and Lee [15], it is additionally assumed that user is also part of the input.

- CB.Dec is a deterministic algorithm taking as inputs $\langle \mathtt{CB.pms}, Cert_i, SK, C\rangle$ as input in time period i. It returns either a message $M \in \mathcal{M}$ or the special symbol $\perp$ indicating a decryption failure.

Naturally, we require that if C is the result of applying algorithm CB.Enc with input $\langle \mathtt{CB.pms}, M, i, \mathtt{user}, PK\rangle$ and (PK, SK) is a valid key-pair, then M is the result of applying algorithm CB.Dec on input $\langle \mathtt{CB.pms}, Cert_i, SK, C\rangle$, where $Cert_i$ is the output of the CB.Certify. We write

$$\mathsf{CB.Dec}\big(\mathtt{CB.pms}, Cert_i, SK, \mathsf{CB.Enc}(\mathtt{CB.pms}, M, i, \mathtt{user}, PK)\big) = M.$$

3.1 Security

The security of a certificate-based encryption scheme is defined against two different types of adversaries. The Type I adversary $\mathcal{A}_I$ has no access to the master key, but may make certification queries and decryption queries. This adversary models the security against non-certified users and general eavesdroppers. Secondly, the Type II adversary $\mathcal{A}_{II}$ is equipped with the master key and models an eavesdropping CA. In the following we give the definitions corresponding to the second type of adversary, since this is the adversary for which the attack presented in this paper is successful. For the full security definition of a CBE scheme we refer the reader to [2], which slightly weakened the attack of the certifier on the original definition of [11], which was inconsistent with the concrete scheme that [11] itself presented.

CBE Game 2. Attack of the certifier

Setup. The challenger runs CB.Gen, gives $\mathtt{CB.pms}$ and $\mathtt{CB.msk}$ to the adversary $\mathcal{A}_{II}$. The challenger then runs CB.SetKeyPair to obtain a key-pair $\langle PK, SK\rangle$ and gives PK to the adversary $\mathcal{A}_{II}$.

Phase 1. The adversary issues decryption queries $q_1, \ldots, q_m$ where each q_j is a decryption query $\langle i, \mathtt{user}, PK, C\rangle$. On this query, the challenger generates $Cert_i$ by using algorithms CB.Certify with inputs $\langle \mathtt{CB.msk}, \mathtt{CB.pms}, i, \mathtt{user}, PK\rangle$ and outputs $\mathsf{CB.Dec}_{Cert_i, SK}(C)$, else it returns $\perp$. These queries may be asked adaptively, that is, they may depend on the answers to previous queries.

Challenge. On challenge query $\langle i^*, \mathtt{user}^*, M_0, M_1\rangle$, where $M_0, M_1 \in \mathcal{M}$ are of equal length, the challenger checks that $\mathtt{user}^* \in \Lambda$. If so, it chooses a random bit b and returns $C^* = \mathsf{CB.Enc}_{i^*, \mathtt{user}^*, PK^*}(M_b)$; else it returns $\perp$.

Phase 2. As in phase 1, with the restriction

$$\langle i, \mathtt{user}, PK, C\rangle \neq \langle i^*, \mathtt{user}^*, PK, C^*\rangle.$$

Guess. The adversary $\mathcal{A}_{II}$ outputs a guess $b' \in \{0, 1\}$. The adversary wins the game if $b = b'$.

We define the advantage of an adversary $\mathcal{A}_{II}$ as

$$\mathsf{Adv}^{\mathsf{CBE-CCA}}_{\mathcal{E}, \mathcal{A}_{II}}(1^k) = \left|\Pr[b = b'] - 1/2\right|.$$

Definition 2. *A* CBE *scheme is said to be* secure against adaptive chosen ciphertext attacks from the certification authority *if no probabilistic polynomially bounded adversary has non-negligible advantage in CBE Game 2.*

4 Certificate-Less Public Key Encryption Definitions

A certificate-less public key encryption scheme is a tuple of seven PPT algorithms:

–CL.Gen is a probabilistic algorithm taking as input a security parameter k. It returns the system parameters CL.pms and CL.msk. The system parameters include the message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$.

–CL.PartialKey is a probabilistic algorithm that takes CL.pms, CL.msk and an identifier $ID_A \in \{0,1\}^*$ for entity A as inputs. It returns a partial private key D_A.

–CL.SecretVal is a probabilistic algorithm that takes as inputs[3] CL.pms and returns a secret value x_A.

–CL.SetPrivKey is a deterministic algorithm that takes as inputs CL.pms, D_A and x_A and returns S_A, a (full) private key.

–CL.SetPubKey is a deterministic algorithm taking as input CL.pms, x_A. It returns a public key P_A.

–CL.Enc is a probabilistic algorithm taking as inputs CL.pms, M, P_A, ID_A where $M \in \mathcal{M}$ is a message. It returns a ciphertext $C \in \mathcal{C}$ for message M or $\perp$ indicating a encryption failure.

–CL.Dec is a deterministic algorithm taking as inputs CL.pms, S_A, C. It returns either a message $M \in \mathcal{M}$ or the special symbol $\perp$ indicating a decryption failure.

Naturally, we require that if C is the result of applying algorithm CB.Enc with input CL.pms, P_A, ID_A, M, then M is the result of applying algorithm CB.Dec on input CL.pms, S_A, C. That is,

$$\mathsf{CB.Dec}\big(\mathtt{CL.pms}, S_A, \mathsf{CB.Enc}(\mathtt{CL.pms}, M, P_A, ID_A)\big) = M.$$

Algorithms CL.SetPrivKey and CL.SetPubKey are normally run by an entity A for itself, after running CL.SecretVal. Usually A is the only entity in possession of S_A and x_A. Algorithms CL.Gen and CL.PartialKey are usually run by a trusted authority, called key generation center (KGC).

4.1 Security

The security of a certificate-less encryption scheme is defined against two different types of adversaries. The Type I adversary $\mathcal{A}_I$ has no access to the master-key CL.msk, but may replace public keys, extract partial private and private keys,

[3] Actually, in the CL-PKE generic constructions by Yum and Lee [15, 16], it is additionally assumed that ID_A is also part of the input.

and make decryption queries. This adversary models a non-registered user and a general eavesdropper. The Type II adversary $\mathcal{A}_{II}$ is equipped with the master-key and models an eavesdropping KGC. $\mathcal{A}_{II}$ is not allowed to replace public keys. In the following we give the definitions corresponding to the second type of adversary, since the attack we describe in this paper is carried out by the KGC. We stress that Yum and Lee security model for this adversary is unchanged from [1].

CL Game 2. Attack of a Type II Adversary

Setup. The challenger runs $\mathsf{CL.Gen}$, and gives $\mathtt{CL.pms}$ and $\mathtt{CL.msk}$ to the adversary $\mathcal{A}_{II}$.

Phase 1. The adversary issues queries $q_1, \ldots, q_m$ where each q_j is one of public key, private key and decryption query.

Challenge. On challenge query $\langle ID_{\mathsf{ch}}, M_0, M_1 \rangle$, where $M_0, M_1 \in \mathcal{M}$ are of equal length and the private key of ID_{ch} was not queried in phase 1, the challenger chooses a random bit b and returns $C^* = \mathsf{CL.Enc}(M_b)$ the encryption of M_b under the current public key P_{ch} for ID_{ch}. Then C^* is delivered to the adversary.

Phase 2. As in phase 1, except that $\mathcal{A}_{II}$ can not make a decryption query on the challenge ciphertext C^* for $(ID_{\mathsf{ch}}, P_{\mathsf{ch}})$ nor a private key query on the challenge identity ID_{ch}.

Guess. Finally, $\mathcal{A}_{II}$ outputs a guess $b' \in \{0, 1\}$. The adversary wins the game if $b = b'$.

We define the advantage of an adversary $\mathcal{A}_{II}$ in CL Game 2 as $\mathsf{Adv}^{\mathsf{CL-CCA}}_{\mathcal{E},\mathcal{A}_{II}}(1^k) = |\Pr[b = b'] - 1/2|$.

Definition 3. *A* CL-PKE *scheme is said to be* secure against adaptive chosen ciphertext attacks from the key generation center *if no probabilistic polynomially bounded adversary has non-negligible advantage in CL Game 2.*

5 An Attack Against the Generic Construction for CBE from EuroPKI 2004

At EuroPKI 2004, Yum and Lee [16] proposed a generic construction for IND-CBE-CCA certificate-based encryption schemes from IND-ID-CCA identity-based encryption schemes. Their construction is depicted in Figure 1. The main idea of their construction is to use double encryption with respect to IBE. One of the decryption keys is known by the certifier, while the other decryption key is only known to the user. Unfortunately, the double encryption design used in [16] is insecure in the light of [10].

We note that this construction does not achieve the required security for certificate-based schemes, at least in the case of an attack of the certifier, as defined in Section 3.1. Remember that the certifier is equipped with his own secret key $\mathtt{CB.msk}$ and that it is allowed to make decryption queries, with the

```
$\mathsf{CB.Gen}(1^k)$
    $(\texttt{ID.pms}_{CA}, \texttt{ID.msk}_{CA}) \leftarrow \mathsf{ID.Gen}(1^k)$
    $\texttt{CB.msk} \leftarrow \texttt{ID.msk}_{CA}$
    $\texttt{CB.pms} \leftarrow \texttt{ID.pms}_{CA}$
    Return $(\texttt{CB.pms}, \texttt{CB.msk})$

$\mathsf{CB.SetKeyPair}(\texttt{CB.pms}, \texttt{user})$
    $(\texttt{ID.pms}_U, \texttt{ID.msk}_U) \leftarrow \mathsf{ID.Gen}(1^k)$
    $d_U \leftarrow \mathsf{ID.Ext}(\texttt{ID.pms}_U, \texttt{ID.msk}_U, \texttt{user})$
    $SK_U \leftarrow (d_U, \texttt{ID.pms}_U)$
    $PK_U \leftarrow \texttt{ID.pms}_U$
    Return $(PK_U, SK_U)$

$\mathsf{CB.Certify}(\texttt{CB.msk}, \texttt{CB.pms}, i, \texttt{user}, PK_U)$
    $Cert_i^U$
    $\leftarrow \mathsf{ID.Ext}\big(\texttt{CB.pms}, \texttt{CB.msk}, (\texttt{user}, i, PK_U)\big)$
    Return $Cert_i^U$

$\mathsf{CB.Enc}(\texttt{CB.pms}, M, i, \texttt{user}, PK_U)$
    $C' \leftarrow \mathsf{ID.Enc}(PK_U, \texttt{user}, M)$
    $C \leftarrow \mathsf{ID.Enc}(\texttt{CB.pms}, (\texttt{user}, i, PK_U), C')$
    Return $C$
$\mathsf{CB.Dec}(\texttt{CB.pms}, Cert_i^U, SK_U, C)$
    Parse $SK_U$ as $(d_U, \texttt{ID.pms}_U)$
    $C' \leftarrow \mathsf{ID.Dec}(\texttt{CB.pms}, Cert_i^U, C)$
    $M \leftarrow \mathsf{ID.Dec}(PK_U, d_U, C')$
    Return $M$
```

Fig. 1. Yum-Lee transformation from IBE to CBE

natural limitation that he cannot ask for the challenge ciphertext. The attack begins once the certifier (called adversary $\mathcal{A}_{II}$ in the CBE game 2) obtains the challenge ciphertext $C^* = \mathsf{CB.Enc}(\texttt{CB.pms}, M_b, i^*, \texttt{user}^*, PK_U^*)$ for M_0, M_1 and unknown $b \in \{0,1\}$ chosen by the challenger. The attack works as follows:

1. $\mathcal{A}_{II}$ generates the certificate $Cert_{i^*}^U$ for $\texttt{user}^*, i^*, PK_U^*$ by running
$$\mathsf{CB.Certify}(\texttt{CB.msk}, \texttt{CB.pms}, i^*, \texttt{user}^*, PK_U^*).$$
2. This certificate is used to decrypt and obtain $C' \leftarrow \mathsf{ID.Dec}(\texttt{CB.pms}, Cert_{i^*}^U, C^*)$.
3. Since $\mathsf{ID.Enc}$ is a probabilistic algorithm, $\mathcal{A}_{II}$ reencrypts C' until obtains $C'' = \mathsf{ID.Enc}(\texttt{CB.pms}, (\texttt{user}^*, i^*, PK_U^*), C')$ such that $C'' \neq C^*$.
4. $\mathcal{A}_{II}$ asks the decryption oracle for the decryption of C''. Since $C'' \neq C^*$, this is a valid decryption query and $\mathcal{A}_{II}$ gets back M_b.

The advantage of this adversary is 1/2, so the scheme in Figure 1 is not secure in the sense of against adaptive chosen ciphertext attacks from the certification authority.

This attack can be easily avoided following [10]. In fact, the proposal of [10] for a generic construction of CBE is very similar to [15]. The main difference is that it uses parallel encryption instead of sequential encryption, but the idea to obtain full security are the same. Informally, this idea is to use the verifier's key of a one-time signature scheme as a label when encrypting and then sign the whole ciphertext. The non-malleability of the ciphertext and the security of the signature scheme prevent the attack from being successful.

6 An Attack Against Yum and Lee Generic Constructions for CL-PKE Schemes

In the same paper [16], Yum and Lee gave a generic transformation from IBE to CL-PKE. The security model they considered for CL-PKC is much more restricted than the original one of [2]. The transformation [16] is depicted in Figure 2. In the same vein as in the previous construction, a double identity-based encryption mechanism is used. One of the decryption keys is known by the key generation center, while the other decryption key is only known to the user. Unfortunately, the double encryption is done with the naive technique described in [10], which is insecure even in the weaker security model considered by [16].

```
$\mathsf{CL.Gen}(1^k)$
    $(\mathtt{ID.pms}_{KGC}, \mathtt{ID.msk}_{KGC}) \leftarrow \mathsf{ID.Gen}(1^k)$
    $\mathtt{CL.msk} \leftarrow \mathtt{ID.msk}_{KGC}$
    $\mathtt{CL.pms} \leftarrow \mathtt{ID.pms}_{KGC}$
    Return $(\mathtt{CL.pms}, \mathtt{CL.msk})$

$\mathsf{CL.PartialKey}(\mathtt{CL.pms}, \mathtt{CL.msk}, ID_A)$
    $d_A \leftarrow \mathsf{ID.Ext}(\mathtt{CL.pms}, \mathtt{CL.msk}, ID_A)$
    $D_A \leftarrow (d_A, ID_A)$
    Return $D_A$

$\mathsf{CL.SecretVal}(\mathtt{CL.pms}, ID_A)$
    $(\mathtt{ID.pms}_A, \mathtt{ID.msk}_A) \leftarrow \mathsf{ID.Gen}(1^k)$
    $x_A \leftarrow (\mathtt{ID.pms}_A, \mathtt{ID.msk}_A, ID_A)$
    Return $x_A$

$\mathsf{CL.SetPrivKey}(\mathtt{CL.pms}, D_A, x_A)$
    Parse $x_A$ as $(\mathtt{ID.pms}_A, \mathtt{ID.msk}_A, ID_A)$
    Parse $D_A$ as $(d_A, ID_A)$
    $d'_A \leftarrow \mathsf{ID.Ext}(\mathtt{ID.pms}_A, \mathtt{ID.msk}_A, ID_A)$
    $S_A \leftarrow (d_A, d'_A, \mathtt{ID.pms}_A)$
    Return $S_A$

$\mathsf{CL.SetPubKey}(\mathtt{CL.pms}, x_A)$
    Parse $x_A$ as $(\mathtt{ID.pms}_A, \mathtt{ID.msk}_A, ID_A)$
    $P_A \leftarrow \mathtt{ID.pms}_A$
    Return $P_A$

$\mathsf{CL.Enc}(\mathtt{CL.pms}, M, P_A, ID_A)$
    $C' \leftarrow \mathsf{ID.Enc}(P_A, ID_A, M)$
    $C \leftarrow \mathsf{ID.Enc}(\mathtt{CL.pms}, ID_A, C')$
    Return $C$

$\mathsf{CB.Dec}(\mathtt{CL.pms}, S_A, C)$
    Parse $S_A$ as $(d_A, d'_A, \mathtt{ID.pms}_A)$
    $C' \leftarrow \mathsf{ID.Dec}(\mathtt{CL.pms}, d_A, C)$
    $M \leftarrow \mathsf{ID.Dec}(\mathtt{ID.pms}_A, d'_A, C')$
    Return $M$
```

Fig. 2. Yum-Lee transformation from IBE to CL-PKE

Indeed, it is not hard to see that their construction suffers from exactly the same problem as the one for certificate-based encryption and that the attack of the Type II adversary succeeds for exactly the same reason. The attack begins once the adversary $\mathcal{A}_{II}$ in the CL Game 2 described in Section 4.1 obtains the challenge ciphertext $\mathsf{CL.Enc}(\mathtt{CL.pms}, M_b, P_A^*, ID_A^*)$ for M_0, M_1 and unknown $b \in \{0,1\}$ chosen by the challenger. The attack works as follows:

1. Since the challenger has given $\mathcal{A}_{II}$ the KGC's master-key $\mathtt{CB.msk}$, the adversary can generate the partial private key $D_A^* = (d_A^*, ID_A^*)$ for user ID_A^* by running $D_A^* \leftarrow \mathsf{CL.PartialKey}(\mathtt{CL.pms}, \mathtt{CL.msk}, ID_A^*)$.
2. This partial private key is used to decrypt and obtain
$$C' \leftarrow \mathsf{ID.Dec}(\mathtt{CL.pms}, d_A^*, C^*).$$
3. Since $\mathsf{ID.Enc}$ is a probabilistic algorithm, $\mathcal{A}_{II}$ reencrypts C' until obtains $C'' \leftarrow \mathsf{ID.Enc}(P_A^*, ID_A^*, C')$ such that $C'' \neq C^*$.

4. $\mathcal{A}_{II}$ asks the decryption oracle for the decryption of C''. Since $C'' \neq C^*$, this is a valid decryption query and $\mathcal{A}_{II}$ gets back M_b.

The advantage of this adversary is 1/2, so the scheme in Figure 2 is not secure in the sense of against adaptive chosen ciphertext attacks from the key generation center.

In the work [15], the authors give another transformation from identity-based encryption to certificate-less encryption. In this case, the user employs a traditional public key encryption scheme [4] instead of an identity-based encryption scheme. The rest of the construction exactly resembles the one described in the previous figure and therefore the attack just presented also applies to [15].

In a recent work [13], a similar attack against [15] is proposed. However, the attack is for a type I adversary and only works in the full security model.

References

1. S. Al-Riyami and K.G. Paterson. Certificateless public key cryptography. In *Advances in Cryptology - ASIACRYPT 2003*, LNCS vol. 2894, pp. 452-473, Springer-Verlag, 2003.
2. S. Al-Riyami and K.G. Paterson. CBE from CL-PKE: A generic construction and efficient scheme. *Public Key Cryptography - PKC 2005*, LNCS vol. 3386, pp. 398-415, Springer-Verlag, 2005.
3. J. Baek, R. Safavi-Naini and W. Susilo. Certificateless Public Key Encryption Without Pairing. *Information Security Conference - ISC 2005*, LNCS vol. 3650, pp. 134-148, Springer-Verlag, 2005.
4. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. *Advances in Cryptology - CRYPTO 1998*, LNCS vol. 1462, pp. 26-45, Springer-Verlag, 1998.
5. K. Bentahar and P. Farshim and J. Malone-Lee and N.P. Smart. Generic Constructions of Identity-Based and Certificateless KEMs. Cryptology ePrint Archive, Report 2005/058.
6. D. Boneh and M. Franklin. Identity-Based Encryption. From The Weil Pairing, *Advances in Cryptology - Crypto 2001*, LNCS vol. 2139, pp. 213-229, Springer-Verlag, 2001.
7. M. Bellare and P. Rogaway. Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. *ACM CCS 93*, pp. 62-73, ACM Press, 1993.
8. R. Canetti, O. Goldreich and S. Halevi. The random oracle methodology, revisited. *Journal of the ACM* vol. 51, num. 4, pp. 557-594, ACM press, 2004.
9. A. Dent and C. Kudla. On Proofs of Security for Certificateless Cryptosystems. Cryptology ePrint Archive, Report 2005/348.
10. Y. Dodis and J. Katz. Chosen-Ciphertext Security of Multiple Encryption, *Theory of Cryptography Conference - TCC 2005*, LNCS vol. 3378, pp. 188-209, Springer-Verlag, 2005.
11. C. Gentry. Certificate-Based Encryption and the Certificate-Revocation Problem, *Advances in Cryptology - Eurocrypt 2003*, LNCS vol. 2656, pp. 272-291, Springer-Verlag, 2003.
12. B.G. Kang and J.H. Park. It is possible to have CBE from CL-PKE? Cryptology ePrint Archive, Report 2005/431, 2005. http://eprint.iacr.org/.

13. B. Libert and J.J. Quisquater. On Constructing Certificateless Cryptosystems from Identity Based Encryption. *Public Key Cryptography 2006 - PKC 2006*, LNCS vol. 3958, pp. 474-490, Springer-Verlag, 2006.
14. A. Shamir. Identity-based cryptosystems and signature schemes, *Advances in Cryptology - Crypto 1984*, LNCS vol. 196, pp. 47-53, Springer-Verlag, 1985.
15. D.H. Yum and P.J. Lee. Generic Construction of Certificateless Encryption. In *Computational Science and Its Applications - ICCSA 2004*, LNCS vol. 3043, pp. 802-811, Springer-Verlag, 2004.
16. D.H. Yum and P.J. Lee. Identity-based cryptography in public key management. In *EuroPKI 2004*, LNCS vol. 3093, pp. 71-84, Springer-Verlag, 2004.

On the Security of Multilevel Cryptosystems over Class Semigroups of Imaginary Quadratic Non-maximal Orders*

Yongtae Kim[1], Chang Han Kim[2], and Taek-Young Youn[3]

[1] Dept. of Mathematics Education,
Gwangju National Univ. of Education,
Gwangju, Korea
ytkim@gnue.ac.kr

[2] Dept. of Information and Security,
Semyung Univ., Jecheon, Korea
chkim@semyung.ac.kr

[3] Center for Information Security Technologies(CIST),
Korea Univ., Seoul, Korea
taekyoung@cist.korea.ac.kr

Abstract. A cryptography for enforcing multilevel security in a system where hierarchy is represented by a partially ordered set was introduced by Akl et al. But the key generation algorithm of Akl et al. is infeasible when there is a large number of users. To overcome this shortage, in 1985, MacKinnon et al. proposed a paper containing a condition which prevents cooperative attacks and optimizes the assignment. In 2005, Kim et al. proposed key management systems for multilevel security using one-way hash function, RSA algorithm, Poset dimension and Clifford semigroup in the context of modern cryptography. In particular, the key management system using Clifford semigroup of imaginary quadratic non-maximal orders is based on the fact that the computation of a key ideal K_0 from an ideal EK_0 seems to be difficult unless E is equivalent to O. We, in this paper, show that computing preimages under the bonding homomorphism is not difficult, and that the multilevel cryptosystem based on the Clifford semigroup is insecure and improper to the key management system.

Keywords: Hierarchy, Key generation algorithm, Class semigroup, Key exchange system.

1 Introduction

An organization with hierarchical structure such as government, diplomacy and military can require users highly placed in the hierarchy to keep a security clearance lower than or equal to those lowly placed. In this context a cryptography

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 92–100, 2006.

for enforcing multilevel security in a system where hierarchy is represented by a partially ordered set(poset) was introduced by Akl et al. [1]. They generate the keys K_i relying on the fundamental assumption behind the RSA. The key generation algorithm of Akl et al. [1] has the advantage that only copy of a piece of information is stored or broadcast and its disadvantage is the large number of keys held by each user. In an effort to overcome this shortage, MacKinnon et al. [9] proposed a paper containing an additional condition which prevents cooperative attacks and optimizes the assignment by giving an improved algorithm to remove the nodes of the longest chain. In 2005, Kim et al. [8] proposed key management systems for multilevel security using one-way hash function, RSA algorithm, Poset dimension and Clifford semigroups. In particular, the key management system using Clifford semigroups of imaginary quadratic non-maximal orders is based on the fact that the computation of the key ideal K_0 from an ideal EK_0 seems to be difficult unless E is equivalent to O. Using the properties of commutative semilattice of idempotents, in this paper, we show that computing preimages of the key ideal K_0 under the bonding homomorphism is not difficult, and that the multilevel cryptosystem based on the Clifford semigroup is insecure and improper to the key management system.

2 Multilevel Security Problem and Its Cryptographic Solution

The notion of the multilevel security and the key management can be found in [1,9]. Assume that the users of computer system are divided into a number of disjoint sets, $U_1, U_2, \cdots, U_n$, which are called security classes. By the partially ordered relation $\leq$ on the set $S = \{U_1, U_2, \cdots, U_n\}$ of classes, the relation $U_i \leq U_j$ in the partially ordered set $(S, \leq)$ means that users in U_i have a security clearance lower than or equal to those in U_j, in other words, users in U_j can have access to information held by users in U_i, while the opposite is not allowed. Let x_m be a piece of information, that a central authority(CA) desires to store in (or broadcast over) the system. Then the meaning of the subscript m is that object x is accessible to users in class U_m and the users in all classes U_i such that $U_m \leq U_i$. In addition to above conditions, the access to information should be as decentralized as possible so that authorized users are able to independently retrieve x_m as soon as it is stored or broadcast by the CA. In [1], Akl et al. proposed a cryptographic solution to the multilevel security problem in three steps as follows.

Step 1 : The CA generates n (deciphering) keys, $K_1, K_2, \cdots, K_n$, for use with the crytoalgorithm.
Step 2 : For $i = 1, 2, \cdots, n$, key K_i is distributed to all users in U_i who keep it secret.
Step 3 : In addition, for $i, j = 1, 2, \cdots, n$, all users in U_j also obtain K_i if $U_i \leq U_j$.

Let E_K and D_K be enciphering and deciphering procedure under the control of the ciphering key K. When an information x_m is to be stored (or broadcast) it is first encrypted with K_m to obtain $x' = E_{K_m}(x_m)$ and then stored or broadcast as the pair $[x', m]$. This guarantees that only users in possession of K_m will be able to retrieve x_m from $x_m = D_{K_m}(x')$. This solution has the advantage that only copy of x_m is stored or broadcast and its disadvantage is the large number of keys held by each user. In order to solve the key storage problem, Akl et al.[1] proposed a key management system in which a user in U_j stores only own key K_j, and can compute from this the key K_i if and only if $U_i \leq U_j$. In such a system, however, there exists the possibility of two users collaborating to compute a key to which they are not entailed. In [9], MacKinnon et al. formulate a condition which prevent such cooperative attacks and characterize all keys assignments which satisfy the condition, and they proposed the following algorithm;

Algorithm : Longest Chain
Step 4 : Find the longest chain $\{i_1, \cdots, i_k\}$ in the poset.
Step 5 : Assign to this chain the smallest available prime p (which now becomes unavailable).
Step 6 : Remove nodes $i_1, \cdots, i_k$ from the poset.
Step 7 : If the poset is not empty, go to Step 4.

Although its running time is $O(|S|^2)$, this algorithm is just an heuristic and the authors generate the keys K_i relying on the fundamental assumption behind the RSA.

3 The Structure of the Class Semigroup $Cls(O)$

In this section, we introduce some facts concerning class semigroups of orders in imaginary quadratic fields. Most of the terminologies, throughout this paper, are due to Gauss[6], and notations and some preliminaries are due to Cox[4], Zanardo and Zannier[12] and Jacobson[7]. The notations O, $\mathbb{Z}$ and $\mathbb{Q}$ denote the imaginary quadratic non-maximal order, the ring of integers and the field of rational numbers respectively. Let $D_1 < 0$ be a square free rational integer, $D = 4D_1/r^2$, where $r = 2$ if $D_1 \equiv 1 \bmod 4$, and $r = 1$ if $D_1 \equiv 2, 3 \bmod 4$. Then $K = \mathbb{Q}(\sqrt{D_1})$ is an imaginary quadratic field of discriminant D. Note that $K = \mathbb{Q}(\sqrt{D})$. If $\alpha, \beta \in K$, we denote by $[\alpha, \beta]$ the set $\alpha\mathbb{Z} + \beta\mathbb{Z}$. An order in K having conductor f with discriminant $D_f = f^2 D$ is denoted by $O = [1, f\omega]$, where $\omega = (D + \sqrt{D})/2$. An (integral)ideal A of O is a subset of O such that $\alpha + \beta \in A$ and $\alpha\lambda \in A$ whenever $\alpha, \beta \in A, \lambda \in O$. For $\alpha \in K, \alpha', N(\alpha)$ and $Tr(\alpha)$ denote the complex conjugate, norm and trace of α respectively. Let $\gamma = f\omega$. Then any ideal A of O (any O-ideal) is given by $A = [a, b + c\gamma]$, where $a, b, c \in \mathbb{Z}$, $a > 0, c > 0$, $c \mid a, c \mid b$ and $ac \mid N(b + c\gamma)$. If $c = 1$, then A is called primitive, which means that A has no rational integer factors other than 1. Then $A = [a, b + \gamma]$ is O-ideal if and only if a divides $N(b + \gamma)$. We say that A and B are equivalent ideals of O and denote $A \sim B$ if there exist non-zero $\alpha, \beta \in K$ such that $(\alpha)A = (\beta)B$ (this relation actually is equivalent relation). We denote the equivalence class of an ideal A by $\overline{A}$. An ideal class $\overline{I}$ is called idempotent if $\overline{I}^2 = \overline{I}$ and the ideal I is also called idempotent. Let $I(O)$ be the set of non-zero fractional ideals of O, and $P(O)$ the set of non-zero principal ideals of O. Then $Cls(O) = I(O)/P(O)$ will be the class semigroup of the order O. We remind that the commutative semigroup $\mathcal{S}$ is called a Clifford commutative semigroup if one of the following equivalent statements holds (Confer [12]).

C1) every element x of $\mathcal{S}$ is contained in a subgroup G of $\mathcal{S}$,
C2) every element x of $\mathcal{S}$ is regular, i.e. there exists $y \in \mathcal{S}$ such that $x = x^2y$ (such an x is called von Neumann regular),
C3) $\mathcal{S}$ is a semilattice of groups.

In the sequel, we will set the positive definite quadratic form $u(x, y) = ax^2 + bxy + cy^2$ as (a, b, c) for brevity, and call η the root of $u(x, y)$ if $u(\eta, 1) = 0$, where η lies in the upper half plane. We begin with introducing a lemma which is a generalization of Proposition 7.4 of Cox[4].

Lemma 1. *Let $u(x, y) = (a, b, c)$ be a positive definite quadratic form with discriminant D_f, where $k = \gcd(a, b, c)$. Let η be the root of $u(x, y)$. Then the ideal $[a, a\eta]$ is invertible if and only if $k = 1$ in the order $O = [1, \gamma]$ of K.*

Proof. First, we note that $[1, a\eta]$ is an order of K, since $a\eta$ is an algebraic integer. We can now show whether $[a, a\eta]$ is a invertible ideal or not in $[1, a\eta]$ according to $k = 1$ or not. For a given $\beta \in K$, $\beta[a, a\eta] \subset [a, a\eta]$ is equivalent to (i) $\beta a \in [a, a\eta]$ and (ii) $\beta(a\eta) \in [a, a\eta]$. Since $a\beta$ belongs to $[a, a\eta]$, we have $a\beta = ma + n(a\eta)$, that is , $\beta = m + n\eta$ for some rational integers m and n.

Conversely, for any rational integers m and n, $a(m + n\eta)$ clearly belongs to $[a, a\eta]$. For (ii), note that

$$\beta(a\eta) = ma\eta + na\eta^2 = ma\eta + n(-b\eta - c) = -nc + (ma - nb)\eta.$$

Thus, $\beta(a\eta) \in [a, a\eta]$ if and only if $a \mid nc$, $a \mid nb$ and m is arbitrary. If $k = 1$, then $a \mid n$. However, if $k > 1$, then $\gcd(a, b)$ and $\gcd(a, c) \geq k$. Thus, there exists a non-trivial divisor s of a and an arbitrary rational integer m such that $a\eta(m + s\eta) \in [a, a\eta]$. These facts say that

$$\{\beta \in K \mid \beta[a, a\eta] \subset [a, a\eta]\} = [1, a\eta]$$

if and only if $k = 1$. From this fact, $[a, a\eta]$ is invertible in $[1, a\eta]$ if and only if $k = 1$. Since f is the conductor of O with discriminant D_f, and fD and b have the same parity, we have $a\eta = -(b+fD)/2+\gamma$, and $(b+fD)/2 \in \mathbb{Z}$. Consequently it follows that $[1, a\eta] = [1, \gamma]$, and thus $[a, a\eta] = [a, -(b+fD)/2+\gamma]$ is an O-ideal.

In particular, if $a = k$, then we denote the ideal $[k, k\eta]$ by E_k. By simple calculations and Lemma 1, it is easily shown that $E_k = [k, \gamma]$ for any divisor $k \mid f$. To clarify the structure of $Cls(O)$, we need the following lemmas.

Lemma 2. *([12, Theorem 10]) Let $I = [a, b + \gamma]$ be a non-zero O-ideal and $\gcd(I) = k$. Then we have $E_k^2 = kE_k, II' = aE_k, IE_k = kI$.*

Note that $\overline{E_k}$'s are the only idempotent elements in the order O. For a quadratic form $u(x,y)=(a,b,c)$, we define

$$\gcd(u(x,y))=\gcd(a,b,c), u_1(x,y)=(1/\gcd(u(x,y)))u(x,y),$$

$$\gcd(I)=\gcd(a,Tr(b+\gamma),N(b+\gamma)/a)$$

for a non-zero O-ideal $I=[a,b+\gamma]$, and denote the discriminant of I by $Tr(b+\gamma)^2-4N(b+\gamma)$.

Lemma 3. *Suppose that I and J are O-ideals with same discriminant D_f such that* $\gcd(I)=k_1, \gcd(J)=k_2$. *Then* $\gcd(IJ)=lcm(k_1,k_2)$.

Proof. Let $u(x,y)$ and $v(x,y)$ be positive definite quadratic forms with discriminant D_f corresponding to the ideals I and J respectively. We now define $u(x,y)=k_1u_1(x,y)$ and $v(x,y)=k_2v_1(x,y)$, where $k_1=\gcd(u(x,y))$ and $k_2=\gcd(v(x,y))$. In this case, if $f=k_1d_1=k_2d_2$, then $u_1(x,y)$ and $v_1(x,y)$ are primitive with discriminant d_1^2D and d_2^2D respectively. From Gauss[6, art.236], the direct composition $U_1(x,y)$ of $u_1(x,y)$ and $v_1(x,y)$ has the discriminant d^2D, where $d=\gcd(d_1,d_2)$. From elementary number theory, we have $f=kd$, where $k=lcm(k_1,k_2)$. From this fact, if we denote $U(x,y)$ the direct composition of $u(x,y)$ and $v(x,y)$, then we have $\gcd(U(x,y))=k$. This completes the proof.

An important property of $\gcd(I)$ is given below.

Lemma 4. *(See [12, Proposition 13])*
If $I=[a,b+\gamma]$ is a non-zero primitive O-ideal, then $\gcd(I)$ *divides f.*

Note that [12,Proposition 14] says that $G_\alpha G_\beta$ is contained in G_δ, where $\delta=lcm(\alpha,\beta)$(Lemma 3 of this paper is equivalent to that). It is well-known that the cardinality of $Cls(O)$ is finite. Now we are ready to clarify the structures of the group G_δ and the semigroup $Cls(O)$.

Theorem 1. *The class semigroup $Cls(O)=\cup_{k|f}G_{\overline{E_k}}$, where $G_{\overline{E_k}}$ is the set of all classes containing O-ideals I with* $\gcd(I)=k$.

Proof. For any O-ideal $I=[a,b+\gamma]$ with $\gcd(A)=k$, $I^2I'=I(aE_k)=akI$ by Lemma 2, that is $\overline{I}=\overline{I}^2\overline{I'}$. Equivalently, $\overline{I}$ is von Neumann regular, which leads that $Cls(O)$ is a Clifford semigroup by the equivalence relation (C2), and thus $Cls(O)$ is a finitely disjoint union of groups of the form G_e, where e is an idempotent element of $Cls(O)$, which leads that $Cls(O)$ has a semilattice structure (C3) with a bonding homomorphism between groups. From Lemma 2 and Lemma 4, the set of idempotents $\mathcal{E}=\{\overline{E_k}\mid k\mid f\}$, and thus the group $G_{\overline{E_k}}=\{\overline{I}\mid\overline{IE_k}=\overline{I}$ and $\overline{IJ}=\overline{E_k}$ for some $\overline{J}\in Cls(O)\}$. Let G be the set of all O-ideals I such that $\gcd(I)=k$. Then, we claim that $G_{\overline{E_k}}=G$. In fact; For any O-ideal I, $\gcd(I)$ divides f by Lemma 4. Suppose that $\gcd(I)=k$, then we have $\overline{IE_k}=\overline{I}$ and $\overline{II'}=\overline{E_k}$ by Lemma 2, which implies that $\overline{I}\in G_{\overline{E_k}}$. Conversely suppose that $\overline{J}\in G_{\overline{E_k}}$ and $\gcd(J)=h$. Then we have $\overline{JJ'}=\overline{E_k}$ by Lemma 2. From the fact that $\gcd(I)=\gcd(I')$ and Lemma 3, we have $\gcd(II')=\gcd(I)$, and thus $h=\gcd(J)=\gcd(JJ')=\gcd(E_k)=k$. This completes the proof.

Note that $G_{\overline{E_1}}(= Cl(O)$ the class group) contains all the equivalence classes of invertible ideals in O and $\mathcal{E}$, which is the set of all the equivalence classes of idempotent in O, is the semilattice since $Cls(O)$ is the Clifford semigroup. In $Cls(O)$, for $\overline{E_i}, \overline{E_j} \in \mathcal{E}$ such that $\overline{E_j} \leq \overline{E_i}$ in the partial order defined on $\mathcal{E}$, there exists a bonding homomorphism $\phi_{\overline{E_iE_j}} : G_{\overline{E_i}} \to G_{\overline{E_j}}$. In [12], Zanardo and Zannier proved the following theorem which ensures the existence of the surjective bonding homomorphisms among the groups $G_{\overline{E_k}}$, and gave the method for finding a preimage of a non-invertible ideal under the bonding homomorphism.

Theorem 2. *(Confer [12, Theorem 16 and Theorem 17]) Let $E_k = [k, \gamma]$, where $k \mid f$, and let I be an O-ideal such that $\overline{I} \in G_{\overline{E_k}}$. Then $JE_k = kI$ for some invertible ideal J. Therefore all the bonding homomorphisms of the Clifford semigroup $Cls(O)$ are surjective.*

The general and efficient algorithms for multiplication of ideals are referred to [3,4,5].

4 Analyses of KMS Using the Clifford Semigroups

In [8], Kim et al. proposed four key management systems(KMS) for multilevel security. Among them, we now revisit the KMS using the Clifford semigroups of imaginary quadratic non-maximal orders to consider its security. The KMS proceeds as described in [8].

4.1 KMS Using the Clifford Semigroups

The parameters needed to class semigroups of imaginary quadratic non-maximal orders are first selected, and then the idempotents of the class semigroups are introduced.

1. a sufficiently large conductor f.
2. an idempotents $\overline{E_k}$ of $Cls(O)$ is the equivalent class of an ideal of the form $E_k = [k, \gamma]$, where k is a divisor of f.
3. for $\overline{E_h}, \overline{E_k} \in \mathcal{E}$, where the ideal $E_h = [h, \gamma]$, the partial order $\leq$ on $\mathcal{E}$ defined by $\overline{E_k} \leq \overline{E_h}$ if $h|k$.
4. a key ideal K_0.

If $\overline{E_i}, \overline{E_j}$ are idempotents, where $\overline{E_j} \leq \overline{E_i}$, then the bonding homomorphism $\phi_{\overline{E_iE_j}} : G_{\overline{E_i}} \to G_{\overline{E_j}}$ is defined by $\phi_{\overline{E_iE_j}}(\overline{K}) = \overline{E_jK}$, where $\overline{K} \in G_{\overline{E_i}}$. First, the CA assigns an idempotent ideal E_{k_i} to each class U_i (confer Fig.1), and selects a random key K_0, and computes $E_{k_2}K_0, E_{k_3}K_0$, and then distributes each of them to the classes U_2 and U_3 respectively. The CA next computes $E_{k_2}E_{k_4}K_0, E_{k_2}E_{k_5}K_0, E_{k_3}E_{k_6}K_0$, and $E_{k_3}E_{k_7}K_0$, and then distributes them to U_4, U_5, U_6 and U_7 respectively in the third row of Fig.1. In this way, the CA computes the keys of all classes, and distributes each of them respectively. Then the users in an upper class can compute all keys belonging to classes lower than itself. In particular, the authors in [8] claimed that the computation of K_0 from $E_{k_i}K_0$ seems to be difficult unless E_{k_i} is equivalent to O.

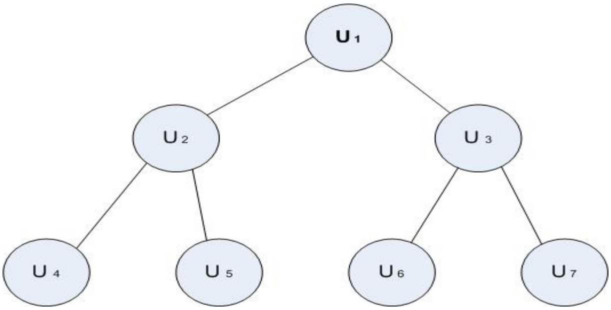


Fig. 1. A Lower Tree

4.2 Analyses of the KMS

In this section, we like to analyze the KMS above by considering the structure the class semigroups and the properties of their ideals in the following points of view. Let E_h, E_k, K_0 and the corresponding bonding homomorphism $\phi_{\overline{E_h E_k}} : G_{\overline{E_h}} \rightarrow G_{\overline{E_k}}$ be the same as above, and we assume that $\overline{E_k} \leq \overline{E_h}$, where $\overline{E_h} \in G_{\overline{E_h}}$.

Computing Preimages Under the Bonding Homomorphism. 1. Kim et al.[8] are right in saying that the users in an upper class can compute all keys belonging to classes lower than itself.
2. The authors claimed that the computation of K_0 from $E_k K_0$ seems to be difficult unless E_k is equivalent to O. It, however, is not difficult to calculate K_0 from $J = E_k K_0$. In fact; Jacobson[7] says that the algorithm in Theorem 2 is one to one on the level of ideals, but given an equivalence class $\overline{J} \in G_{\overline{E}_k}$, one can apply it to any ideal representative equivalent to J, thereby randomizing over the ideal classes in $Cl(O)$ whose images under ϕ_k are equal to $\overline{J}$.

Choosing the Key Ideal. 1. In [8], the authors choose the key ideal K_0 arbitrarily. It, however, is not easy to select a non-invertible ideal of a non-maximal order(confer the open problem of this paper).
2. In general, for an (invertible or not) ideal K_0 with $\gcd(K_0) = h$, Theorem 2 ensures that there exists an invertible O-ideal K such that $KE_h = hK_0$, and thus $\overline{K_0 E_k} = \overline{KE_k}$ by Lemma 3. From this fact, without loss of generality, $G_{\overline{E_h}}$ can be replaced by $Cl(O)$, and h can be always taken 1. For brevity, we denote ϕ_k the bonding homomorphism of $Cl(O)$ to $G_{\overline{E_k}}$.

Security of the KMS. 1. Theorem 2 describes an algorithm for computing the required preimages given only a representative of an ideal class in $G_{\overline{E_k}}$ and k under ϕ_k. In general, we have $\mid G_{\overline{E_k}} \mid < \mid Cl(O) \mid$, which means that the preimage of a representative of an ideal class in $G_{\overline{E_k}}$ under ϕ_k is not unique. Since there are $|Ker(\phi_k)|$ different preimages of $\overline{J}$ under ϕ_k, the worst case number of attempts before one expect to succeed with this strategy is at most $|Ker(\phi_k)|$,

which is significantly small in general. The procedure for computing preimage by changing under ϕ_k can be randomized by changing the representative of the ideal equivalence class. If the first chosen preimage does not find I, the process is simply repeated until it is found.
2. On the other hand, if the number of users U_i of the KMS are large, then so are the number of idempotents E_{k_i} of the class semigroup $Cls(O)$ used. From Theorem 1, the number of prime factors of f becomes large, and thus each length of the prime factor is relatively small if f is fixed, which means that the multilevel security problem in $Cls(O)$ of the above KMS is reduced to the multilevel security problem in the class group $Cl(O)$(Recall that the class group $Cl(O)$ is a proper subgroup of $Cls(O)$ by Theorem 1) and a lot of number of finite fields corresponding to the prime factors of f. Thus, the cryptosystems in the class semigroup $Cls(O)$ using non-invertible ideal offer less security than cryptosystems in class group $Cl(O)$. In this case, the conductor f can be factored completely so that the structure of $Cls(O)$ can be easily revealed by Theorem 1, and thus the cryptosystem based on $Cls(O)$ can be easily broken.
3. By Lemma 3, we have $\overline{E_{k_1}E_{k_2}} = \overline{E_{k_2}}$ if $k_2|k_1$, and thus the deciphering key $\overline{E_{k_1}E_{k_2}K_0}$ of the user U_2 in Step 1 and Step 2 is equal to $\overline{E_{k_2}K_0}$. That is, the multiplication of two idempotents which are totally ordered by the partial order $\leq$ on $\mathcal{E}$ becomes to be the idempotent of lower user in the level of class. Thus, the possibility of finding the key K_0 is equal to all users.
4. In addition, if $\overline{E_{k_2}} \leq \overline{E_{k_1}}$, where $E_{k_1} = [k_1, \gamma]$ and $E_{k_2} = [k_2, \gamma]$, then k_1 is a divisor of k_2, which means that a user in U_2 of the lower class in Step 3 is able to calculate the ideal E_{k_1} by factoring k_2 of the upper class. Consequently, the meaning of the level of information security will be lost under the multilevel cryptosystem based on the Clifford semigroup.

5 Open Problem

Although the KMS using Clifford semigroup of an imaginary quadratic non-maximal order proposed by Kim et al. is easily broken or improper, there is an interesting open problem which is related to the RSA cryptosystem.

Open Problem: Is there an efficient algorithm for finding a non-invertible ideal class $\overline{I} \in G_{\overline{E_k}}$, where k is a non-trivial divisor of f, in an imaginary quadratic non-maximal order?

In fact, let the conductor f be pq, where p, q are sufficiently large. We can construct a class semigroup of an imaginary quadratic non-maximal order with conductor f by section 3. Now, if one can choose a non-invertible ideal class $\overline{I} \in G_{\overline{E_k}}$, where k is a non-trivial divisor of f, then $\gcd(I)(= k)$ should be either p or q by Theorem 1. In particular, our open problem can be related to the open problem 1 proposed by Jacobson[7] on the possibility of choosing k such that $|G_{\overline{E_k}}|$ is not significantly smaller than $|Cl(O)|$.

6 Conclusion

A cryptographic scheme for enforcing multilevel security in a system where hierarchy is represented by a partially ordered set was introduced by Akl et al. They generate the keys K_i relying on the fundamental assumption behind the RSA. But the key generation algorithm of Akl et al. is infeasible when there is a large number of users. To overcome this shortage, in 1985, MacKinnon et al. proposed a paper containing a condition which prevents cooperative attacks and optimizes the assignment. In 2005, Kim et al. proposed key management systems for multilevel security using one-way hash function, RSA algorithm, Poset dimension and Clifford semigroup in the context of modern cryptography. In particular, the key management system in [8] using Clifford semigroup of imaginary quadratic non-maximal orders is based on the fact that the computation of a key ideal K_0 from an ideal EK_0 seems to be difficult unless E is equivalent to O. Using the properties of commutative semilattice of idempotents, in this paper, we show that computing preimages of the key ideal K_0 under the bonding homomorphism is not difficult, and that the multilevel cryptosystem based on the Clifford semigroup is insecure and improper to the key management system. In section 5, we propose an open problem whether one can easily choose a non-invertible ideal in an imaginary quadratic non-maximal order, which will be related to the RSA cryptosystem.

References

1. Selim G. Akl, Peter D. Taylor, *Cryptographic Solution to a Multilevel Security Problem*, CRYPTO 1982 (1982) pp. 237-249.
2. Selim G. Akl, Peter D. Taylor, *Cryptographic Solution to a Problem of Access Controlin Hierarchy*, ACM Trans. Computer System 1(3) (1983) pp. 239-248.
3. J. Buchmann, H. C. Williams, *A key-exchange system based on imaginary quadratic fields*, J. Cryptology 1 (1988) pp. 107-118.
4. D. Cox, *Primes of the form* $x^2 + ny^2$, Wiley, New York (1989).
5. H. Cohen, *A course in computational algebraic number theory*, Springer-Verlag, Berlin (2000).
6. C. F. Gauss, *Disquisitiones Arithmeticae* , translated by Clarke A. A., Springer-Verlag, New York (1986).
7. Michael J. Jacobson Jr. , *The security of cryptosystems based on class semigroups of imaginary quadratic non-maximal orders* ASISP 2004, LNCS 3108 (2004) pp. 149-156.
8. Hwankoo Kim, Bongjoo Park, JaeCheol Ha, Byoungcheon Lee, DongGook Park, *New Key Management Systems for Multilevel Security*, ICCSA 2005, LNCS 3481 (2005) pp. 245-253.
9. Stephen J. MacKinnon, Peter D. Taylor, Henk Meijer, Selim G. Akl, *An Optimal Algorithm for Assignment Cryptographic Keys to Control Access in a Hierarchy*, IEEE Trans. on Computers vol.34 no.9 (1985) pp. 797-802.
10. S. Paulus, T. Tagaki, *A new public-key cryptosystem over a quadratic order with quadratic decryption time*, Journal of Cryptology 13 (2000) pp. 263-272.
11. R. L. Rivest, A. Shamir, L. Adelman, *A method for obtaining digital signatures and public key cryptosystems*, Communications of ACM 21 (1978) pp. 120-126.
12. P. Zanardo, U. Zannier, *The class semigroup of orders in number fields*, Math. Proc. Camb. Phil. Soc. 115 (1994) pp. 379-391.

Short Linkable Ring Signatures Revisited

Man Ho Au[1], Sherman S.M. Chow[2], Willy Susilo[1], and Patrick P. Tsang[3]

[1] Center for Information Security Research
School of Information Technology and Computer Science
University of Wollongong, Wollongong 2522, Australia
{mhaa456, wsusilo}@uow.edu.au
[2] Department of Computer Science
Courant Institute of Mathematical Sciences
New York University, NY 10012, USA
schow@cs.nyu.edu
[3] Department of Computer Science
Dartmouth College, Hanover, NH 03755, USA
patrick@cs.dartmouth.edu

Abstract. Ring signature is a group-oriented signature in which the signer can spontaneously form a group and generate a signature such that the verifier is convinced the signature was generated by one member of the group and yet does not know who actually signed. Linkable ring signature is a variant such that two signatures can be linked if and only if they were signed by the same person.

Recently, the first short linkable ring signature has been proposed. The short signature length makes it practical all of a sudden to use linkable ring signature as a building block in various cryptographic applications. However, we observed a subtle and yet imperative blemish glossed over by their security model definition which, if not carefully understood and properly handled, could lead to unanticipated security threats.

Inspired by the recent refinement of security definitions in conventional ring signatures, we formalize a new and better security model for linkable ring signature schemes that takes into account realistic adversarial capabilities. We show that the new model is *strictly stronger* than all existing ones in the literature. Under our new model, we propose a new short linkable ring signature scheme, improved upon the existing scheme.

Keywords: ring signature, linkable ring signature, short signature.

1 Introduction

Ring signatures, introduced by Rivest, Shamir and Tauman [19], are characterized by three main properties: anonymity, spontaneity, and unlinkability. Anonymity in ring signatures means 1-out-of-n signer verifiability, which enables the signer to keep anonymous in these "rings" of diversion signers. Spontaneity is a property which makes distinction between ring signatures and group signatures [8]. In group signature schemes, there exists a trusted third party (TTP), usually known as the group manager, who handles the joining of group members

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 101–115, 2006.

by interacting with them. In ring signature schemes, no such trusted party exists and the rest of the $n-1$ members in the ring can be totally unaware that they have been included in the ring. Unlinkability is another notion related to privacy – two ring signatures issued by the same signer are unlinkable in any way, except the very fact that this signer appears in the rings of both ring signatures. These three properties make ring signatures widely applicable to various cryptographic schemes [2, 9, 13]. Taking the example of concurrent signatures [9, 13] which is a partial solution to the fair exchange of signatures without TTPs, anonymity provides the signer-ambiguity of signatures (before they are exchanged) and the spontaneity enables a solution without TTPs. Survey of ring signatures and related applications can be found in [12, 20].

A twist in this paradigm is linkable ring signatures [16], which make it possible to identify whether two ring signatures were actually issued by the same signer, but still impossible to identify who the signer was. This reduced level of anonymity is known as linkable-anonymity, or pseudonymity. Linkable-anonymity renders ring signatures a useful building block in various cryptographic applications with privacy concerns. In [21], applications of linkable ring signatures in e-cash, e-voting and attestation were discussed. We briefly describe the case for e-cash here. Two obvious security requirements of an e-cash system are user anonymity and the capability of detecting double spending. The anonymity set of ring signature is the set of e-coins issued by the bank thus far (i.e. each pair of keys represents a coin). When a user spends, he/she use one of the signing keys among the e-coins to sign a ring signature on a transaction transcript. Ring signatures guarantee that one e-coin honestly spent in a transaction is (computationally) indistinguishable from another, thereby protecting the anonymity of the user. Linkability comes to play in double spending detection since it is possible to tell whether or not two signatures were signed using the same key, semantically implying whether the same e-coin has been spent twice.

The efficiency of ring signatures obviously determines the efficiency (and thus practicality) of these cryptographic applications. Researches have been directed toward goals such as improving the security of the scheme (e.g. [4, 11], the running time of signature generation (e.g. [10]), or the space complexity of the signature (e.g. [14, 21]). Recently, the first *short* linkable ring signature scheme has been proposed [21]. This nice property increases the practicality of ring signature as a building block of cryptographic applications.

1.1 Our Contributions

In this work we first point out a subtle blemish in the security model for linkable ring signature in [21]. Namely their security model glossed over the existence of an empowered central authority. We discuss the possible security threats when their scheme is instantiated without carefully addressing the issue.

Secondly, we survey the literature on the security models proposed for linkable ring signatures and formalize a new one that is the *strongest* among the all. We do an in-depth comparison among the new one and existing ones, and argue the necessity of our new model by showing the fact that it takes into account realistic adversarial capabilities not considered before. The proposed security model paves the way for future secure linkable ring signature designs.

We also propose a new short linkable ring signature construction based on [21] that is secure under our new security model. The significance is twofold – we now have a short linkable ring signature construction secure against stronger but realistic adversaries; and the proposed stronger security model is not an impractical one as there does exist an efficient construction under the model.

1.2 Paper Organization

Next two sections provide preliminaries needed in the rest of the paper, which includes a review of the short linkable ring signature due to [21] and a discussion on the implications of their security model. We propose a new security model for linkable ring signatures and compare it with the existing ones in Section 4, followed by our new construction in Section 5. Section 6 concludes the paper.

2 Preliminaries

We first give some notations to be used in the rest of the paper. N is a *safe prime product* if $N = pq = (2p' + 1)(2q' + 1)$ for some primes p, q, p', q' such that p' and q' are of the same length. Denote by $QR(N)$ the group of quadratic residues modulo a safe prime product N.

2.1 Mathematical Assumptions

Decisional Diffie-Hellman (DDH) Assumption. Let G be a group where $|G| = q$ and $g \in G$ such that $\langle g \rangle = G$. No PPT algorithm can, on input two distributions $\langle g, g^a, g^b, g^{ab} \rangle$ and $\langle g, g^a, g^b, g^c \rangle$ where $a, b, c \in_R \mathbb{Z}_q$, distinguish them with non-negligible probability over 1/2 in time polynomial in q.

Strong RSA Assumption. There exists no probabilistic polynomial time (PPT) algorithm which, on input a random λ-bit safe prime product N and a random $z \in QR(N)$, returns $u \in \mathbb{Z}_N^*$ and $e \in \mathbb{N}$ such that $e > 1$ and $u^e = z (\text{mod } N)$, with non-negligible probability and in time polynomial in λ.

Link Decisional RSA (LD-RSA) Assumption [21]. There exists no PPT algorithm which, on input a λ-bit safe prime product N, a generator g of $QR(N)$, $n_0 = p_0 q_0$ and $n_1 = p_1 q_1$ where p_0, q_0, p_1, q_1 are sufficiently large random primes of size polynomial in λ, $g^{p_b + q_b}$ where $b \in_R \{0, 1\}$, returns with $b' = b$ with probability non-negligibly over 1/2 and in time polynomial in λ.

2.2 Building Blocks

Signature of Knowledge. Every three-round Proof of Knowledge protocols (PoKs) that is Honest-Verifier Zero-Knowledge (HVZK) can be transformed into a signature scheme by setting the challenge to the hash value of the commitment concatenated with the message to be signed [15]. Signature schemes generated as such are provably secure [18] against existential forgery under adaptively chosen message attack in the random oracle model [3]. They are sometimes referred to as *Signatures based on Proofs of Knowledge*, SPK for short [7]. As an example, we denote by $SPK\{(x) : y = g^x\}(M)$, where M is the message, the signature scheme derived from the zero-knowledge proof of the discrete logarithm of y using the above technique. Such notation will be used in the rest of the paper.

Accumulators with One-Way Domain. An accumulator "accumulates" multiple values into one single value such that, for each value accumulated, there is a witness proving that it has indeed been accumulated. A dynamic accumulator is one that allows values to be added or deleted dynamically. Short ring signatures can be constructed from accumulators by first accumulating the public keys of the ring to form a short value, followed by the signer proving, non-interactively in a zero-knowledge manner, that his public key has indeed been accumulated in the accumulator and he knows the secret key corresponding to that public key.

An *accumulator with one-way domain* is a quadruple $(\{F_\lambda\}_{\lambda\in\mathbb{N}}, \{X_\lambda\}_{\lambda\in\mathbb{N}}, \{Z_\lambda\}_{\lambda\in\mathbb{N}}, \{R_\lambda\}_{\lambda\in\mathbb{N}})$, such that the pair $(\{F_\lambda\}_{\lambda\in\mathbb{N}}, \{X_\lambda\}_{\lambda\in\mathbb{N}})$ is a collision-resistant accumulator [5], and each R_λ is a relation over $X_\lambda \times Z_\lambda$ that is *efficiently verifiable*, *efficiently samplable* and *one-way*.[1] In the following we describe the accumulator with one-way domain given by [14]. For $\lambda \in \mathbb{N}$, the family F_λ consists of the exponentiation functions modulo λ-bit safe-prime products such that $f : QR(n) \times \mathbb{Z}_{n/4} \rightarrow QR(n)$ and $f : (u, x) \mapsto u^x \mod n$ where n is a λ-bit safe-prime product. The accumulator domain $\{X_\lambda\}_{\lambda\in\mathbb{N}}$, the pre-image domain $\{Z_\lambda\}_{\lambda\in\mathbb{N}}$ and the one-way relation $\{R_\lambda\}_{\lambda\in\mathbb{N}}$ are respectively defined as:

$$\begin{aligned}
&X_\lambda \doteq \{e \text{ is prime} | (\tfrac{e-1}{2} \in \mathsf{RSA}_\ell) \wedge (e \in S(2^\ell, 2^\mu))\},\\
&Z_\lambda \doteq \{(e_1, e_2) | e_1, e_2 \text{ are distinct } \ell/2\text{-bit primes and } e_2 \in S(2^{\ell/2}, 2^\mu)\}, \text{ and}\\
&R_\lambda \doteq \{(x, (e_1, e_2)) \in X_\lambda \times Z_\lambda | (x = 2e_1e_2 + 1)\},
\end{aligned}$$

where $S(2^\ell, 2^\mu)$ is embedded within $(0, 2^\lambda)$ with $\lambda - 2 > \ell$ and $\ell/2 > \mu + 1$.

3 Review of Short Linkable Ring Signature Due to [21]

This section is divided into two halves. In the first half, we review the short linkable ring signature construction due to [21]. Then we state some observations on their security model and its possible implications.

3.1 Construction

In [21], a linkable ring signature scheme consists of a tuple (Init, KeyGen, Sign, Verify, Link) of five poly-time algorithms. Init is called to generate the system parameters. KeyGen is responsible for generating key pairs for all users in the

[1] Consult [14] for their definitions.

system. Note that in their scheme KeyGen is executed by a trusted party, as supposed to usual public key generation. Sign is run by a signer to generate a ring signature. Verify can be invoked by anybody to verify a signature. Finally Link can also be executed by anybody to test whether or not two signatures were generated by the same signer.

The authors in [21] instantiated their generic linkable ring signature construction using accumulators [14], resulting in an actual short linkable ring signature construction. We refer to this particular instantiation as SLRS. While only the SPK pertaining to the signing/verification algorithms was explicitly stated in the original paper, we enumerate all the algorithms in SLRS in the following, up to some notation adaptation. An enumeration is necessary not only for a better understanding their scheme, but also an easier appreciation of the material presented in the rest of this paper.

- Init, on input security parameter 1^λ, firstly prepares a collision-resistant accumulator with one-way domain, together with its description desc, according to [14], then picks uniformly at random a generator $\tilde{g} \in QR(N)$ for the group $QR(N)$, where N is defined in desc, and outputs the system parameters param as $(1^\lambda, \mathsf{desc}, \tilde{g})$.
- KeyGen executes, for every user i in the scheme, the sampling algorithm W of the accumulator to obtain $(y_i, (p_i, q_i))$ and hence user i's key pair $(sk_i, pk_i) \doteq ((p_i, q_i), y_i)$, and returns a vector of key pairs of all users.
- Sign, on input a public key set $\mathcal{Y} = \{pk_1, \ldots, pk_n\}$ with $pk_i = y_i$ for all i, a message $M \in \{0,1\}^*$ and a private key $sk_\pi = (p_\pi, q_\pi)$ that corresponds to $pk_\pi \in \mathcal{Y}$, does the following:
  1. Compute the witness w_π for y_π as $w_\pi \leftarrow f(u, \{y_i | i \neq \pi\})$ and then the accumulated value v of all public keys as $v \leftarrow f(w_\pi, y_\pi)$. Recall that f is the accumulating function described in the previous section.
  2. Compute a signature for
  $$SPK\left\{\begin{pmatrix} w, y, \\ p, q \end{pmatrix} : \begin{array}{lll} w^y = v \bmod N & \wedge\ y = 2pq + 1 & \wedge \\ y \in S(2^\ell, 2^\mu) & \wedge\ q \in S(2^{\ell/2}, 2^\mu) & \wedge \\ \tilde{y} = \tilde{g}^{p+q} \bmod N & & \end{array}\right\}(M). \quad (1)$$
  3. Denote by σ' be the output after the execution of the SPK above. The signature σ returned by the algorithm is given by $\sigma \doteq (v, \tilde{y}, \sigma')$. Note that the tag $\tilde{y}$ is uniquely determined by the private key sk_π.
- Verify, on input a public key set $\mathcal{Y} = \{pk_1, \ldots, pk_n\}$ with $pk_i = y_i$ for all i, a message $M \in \{0,1\}^*$ and a signature $\sigma = (v, \tilde{y}, \sigma') \in \Sigma$, where Σ is the signature space corresponding to the output domain of Sign, verifies the statement $v \overset{?}{=} f(u, \{y_i | i \in [1, n]\})$ and the validity of σ' with respect to the SPK represented by Equation (1), and then returns `accept` if both checks pass or `reject` otherwise.
- Link, given two valid signatures, extracts their respective linkability tags and returns `linked` if they are the same or `unlinked` otherwise.

3.2 Security Model

The syntax of the key generation algorithm (LRKg) in [21] implies the existence of a central authority who generates key pairs for, as well as distributing them to, all users in the scheme. A user must acquire a key pair from the authority in order to participate. In other words, the authority governs the formation of the group. Signatures can therefore be signed on behalf of those and only those users who have successfully joined the group, as in the case of group signature. Recall that in ring signature, however, any user generates his/her own key pair on his/her own, possibly followed by acquiring a certificate on the public key from a certification authority (CA), as one would do in any public key infrastructure.

Worse still, the central authority under such a definition of the key generation algorithm could possibly introduce new and unanticipated threats to the construction because its existence was not captured in the definitions of the various security notions in [21]. As a matter of fact, the central authority is capable of forging signatures, slandering honest users, revoking linkable-anonymity and also undermining linkability. We briefly describe how a central authority can perform each of the above malicious actions in the appendix.

4 Our Proposed Security Model

4.1 Syntax

A *Linkable Ring Signature* scheme is a tuple (Init, KeyGen, Sign, Verify, Link) of five poly-time algorithms. In [21], the algorithm KeyGen generates key pairs for all users in the scheme in one shot. As discussed in the previous sections, this makes the scheme non-spontaneous, conflicting a fundamental requirement of ring signatures. Also, one must pay unreasonable trust on the entity who runs this algorithm. As a remedy, we instead define KeyGen as an algorithm executed by each individual user for the generation of his/her own key pair. The following enumerates the syntax.

- $\mathsf{param} \leftarrow \mathsf{Init}(1^\lambda)$, the poly-time *initialization* algorithm which, on input a security parameter $\lambda \in \mathbb{N}$, outputs the system parameters param containing, among other things, 1^λ. All algorithms below have implicitly param as one of their inputs.
- $(sk_i, pk_i) \leftarrow \mathsf{KeyGen}()$, the PPT *key generation* algorithm which outputs a secret/public key pair (sk_i, pk_i). We denote by $\mathcal{SK}$ and $\mathcal{PK}$ the domains of possible secret keys and public keys respectively. When we say that a public key corresponds to a secret key or vice versa, we mean that the secret/public key pair is an output of KeyGen.
- $\sigma \leftarrow \mathsf{Sign}(\mathcal{Y}, M, x)$, the PPT *signing* algorithm which, on input a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$, where $n \in \mathbb{N}$ is of size polynomial in λ, a message $M \in \{0,1\}^*$, and a private key $x \in \mathcal{SK}$ whose corresponding public key is contained in $\mathcal{Y}$, produces a signature σ. We denote by Σ the domain of possible signatures.

- $1/0 \leftarrow \mathsf{Verify}(\mathcal{Y}, M, \sigma)$, the poly-time *verification* algorithm which, on input a set $\mathcal{Y}$ of n public keys in $\mathcal{PK}$, where $n \in \mathbb{N}$ is of size polynomial in λ, a message $M \in \{0,1\}^*$ and a signature $\sigma \in \Sigma$, returns 1 or 0 meaning `accept` or `reject` respectively. If the algorithm returns `accept`, the message-signature pair (M, σ) is said to be *valid.*
- $1/0 \leftarrow \mathsf{Link}(\sigma_0, \sigma_1)$, the poly-time *linking* algorithm which, on input two valid signatures, outputs 1 or 0 meaning `linked` or `unlinked` respectively.

Linkable ring signature schemes must satisfy:

1. *Verification Correctness* – Signatures signed according to specification are accepted during verification, with overwhelming probability; and
2. *Linking Correctness* – Two signatures signed according to specification are `linked` with overwhelming probability if the two signatures share a common signer. On the other hand, two signatures signed according to specification are `unlinked` with overwhelming probability if the two signatures do *not* share a common signer.

4.2 Security Notions

Recently Bender *et al.* [4] proposed more-refined definitions of security notions for conventional ring signatures. Specifically, they differentiated various attacks against unforgeability and anonymity. The separation between security notions resulted from the refinement promotes a better evaluation of ring signatures.

We note that the same classification can be, upon certain adaptation, applied to linkable ring signatures as well. Consequently, we reconsider various security notions borrowing the nomenclature used in [4] for easy analogy. Whenever it deems necessary, we reformulate security notions strengthened to better capture the attacking capabilities of the adversary in the real world.

Unforgeability. Bender *et al.* classified in [4] unforgeability for ring signatures into (1) against fixed-ring attacks, (2) against chosen-subring attacks, and (3) with respect to (w.r.t.) insider corruption. Among all linkable ring signature schemes in the literature [16, 17, 21], [16] and [17] are unforgeable against chosen-subring attacks[2]. In [21], the adversary is further allowed to corrupt users and acquire their private keys. i.e. [21] is unforgeable also w.r.t. insider corruption.

Our definition of unforgeability in the following resembles that in [21].

Definition 1 (Unforgeability). *A linkable ring signature scheme is unforgeable if for any PPT adversary $\mathcal{A}$ and for any polynomial $n(\cdot)$, the probability that $\mathcal{A}$ succeeds in the following game is negligible:*

1. *(Initialization Phase.) Key pairs $\{(PK_i, SK_i)\}_{i=1}^{n(k)}$ are generated by executing $\mathsf{KeyGen}(1^k)$, and the set of public keys $S \doteq \{PK_i\}_{i=1}^{n(k)}$ is given to $\mathcal{A}$.*

[2] The attack was named “chosen-public-key attacks” in [16, 17].

2. *(Probing Phase.) $\mathcal{A}$ is given access to a signing oracle $\mathcal{SO}(\cdot,\cdot,\cdot)$, where $\mathcal{SO}(s, M, R)$ outputs $\textsf{Sign}_{s,SK_s}(M, R)$ and we require that $R \subseteq S$ and $PK_s \in R$. $\mathcal{A}$ is also given access to a corrupt oracle $\mathcal{CO}(\cdot)$, where $\mathcal{CO}(i)$ outputs SK_i.*
3. *(Output Phase.) $\mathcal{A}$ outputs (M^*, σ^*, R^*), and succeeds if $\textsf{Verify}_{R^*}(M^*, \sigma^*) = 1$, $\mathcal{A}$ never queried $(\cdot, M^*, R^*)$ to its signing oracle, and $R^* \subseteq S \backslash C$, where C is the set of corrupted users.*

Linkable-Anonymity. Comparing to [21], the corruption oracle in our definition gives the random coin used in the private keys generation, which is at least as strong as giving out the private key directly, since a private key can be reconstructed from the random coin but the reverse is not necessarily the case.

Attribution attacks or key exposure attacks of anonymity have been considered in [4]. We argue that, however, these two scenarios are not applicable in linkable ring signatures, for reasons as follows. It can be shown that, in any secure and practical linkable ring signature scheme, a signature must contain a tag which is a correct, unique and efficiently-computable one-way mapping of the signer's secret. Thus, a non-signer who is willing to reveal his/her private key can convince anyone that fact that he/she did not generate a particular signature. For a similar reason, if a signer is forced to divulge his/her private key, everyone can decide if he/she is the signer of a particular signature or not. As a consequence, a secure linkable ring signature cannot be anonymous against attribution attacks/against key exposure.

We now define linkable anonymity as follows.

Definition 2 (Linkable-anonymity). *A linkable ring signature is* linkably anonymous *if for any PPT adversary $\mathcal{A}$ and for any polynomial $n(\cdot)$, the probability that $\mathcal{A}$ succeeds in the following game is negligibly close to* $1/2$*:*

1. *(Initialization Phase.) Key pairs $\{(PK_i, SK_i)\}_{i=1}^{n(k)}$ are generated by executing $\textsf{KeyGen}(1^k; \omega_i)$ for randomly chosen ω_i, and the set of public keys $S \doteq \{PK_i\}_{i=1}^{n(k)}$ is given to $\mathcal{A}$.*
2. *(Probing Phase I.) $\mathcal{A}$ is given access to a signing oracle $\mathcal{SO}(\cdot,\cdot,\cdot)$, where $\mathcal{SO}(s, M, R)$ outputs $\textsf{Sign}_{s,SK_s}(M, R)$ and we require that $R \subseteq S$ and $PK_s \in R$. $\mathcal{A}$ is also given access to a corruption oracle $\mathcal{CO}(\cdot)$, where $\mathcal{CO}(i)$ outputs ω_i and we require that $1 \le i \le n(k)$.*
3. *(Challenge Phase.) $\mathcal{A}$ outputs a message M, distinct indices i_0, i_1, and a ring $R \subseteq S$ for which $PK_{i_0}, PK_{i_1} \in R \cap S$ and all keys in R are distinct. If $(i_0, \cdot, \cdot)$ or $(i_1, \cdot, \cdot)$ was an input to $\mathcal{SO}$, or if i_0 or i_1 was an input to $\mathcal{CO}$, $\mathcal{A}$ fails and the game terminates. Otherwise a random bit b is chosen, and $\mathcal{A}$ is given $\sigma \leftarrow \textsf{Sign}_{i_b,SK_{i_b}}(M, R)$.*
4. *(Probing Phase II.) $\mathcal{A}$ is again given access to $\mathcal{SO}$ and $\mathcal{CO}$. If $(i_0, \cdot, \cdot)$ or $(i_1, \cdot, \cdot)$ is queried to $\mathcal{SO}$, or if i_0 or i_1 is queried to $\mathcal{CO}$, $\mathcal{A}$ fails and the game terminates.*
5. *(Output Phase.) The adversary outputs a bit b', and succeeds if $b' = b$.*

Definition 3 (Linkable-anonymity w.r.t. adversarially-chosen keys). *A linkable ring signature is* linkably anonymous w.r.t. adversarially-chosen keys *if* *for any PPT adversary $\mathcal{A}$ and for any polynomial $n(\cdot)$, the probability that $\mathcal{A}$ succeeds in the previous game, without restricting the ring R to be a subset of S in the challenge phase, and without requiring $R \subseteq S$ for $\mathcal{SO}(\cdot, \cdot, R)$ queries, is negligibly close to* $1/2$.

Linkability. Like unforgeability, linkability can also be classified according to the same three attacks. [16] and [17] are linkable against chosen-subring attack, while [21] is linkable also w.r.t. insider corruption. However, none of the existing models considers the situation when the adversary corrupts a user before key generation and generates the key pair adversarially for the corrupted user. To separate schemes that still remain secure under this new attack from those that does not, we give two definitions below, the first one being a standard linkability definition secure against chosen-subring attack and insider corruption as in [21], and the second secure even w.r.t. adversarially-chosen keys.

Note that linkability w.r.t. adversarially-chosen keys is strictly stronger than standard linkability. In particular, SLRS in [21] is linkable under standard linkability, but not under linkability w.r.t. adversarially-chosen keys. We also demonstrate that the strictly stronger definition of linkability is achievable by giving a concrete construction later on in this paper.

Definition 4 (Linkability). *A linkable ring signature is linkable if for any PPT adversary $\mathcal{A}$ and for any polynomial $n(\cdot)$, the probability that $\mathcal{A}$ succeeds in the following game is negligible:*

1. *(Initialization Phase.) As in Definition 1.*
2. *(Probing Phase.) As in Definition 1.*
3. *(Output Phase.) $\mathcal{A}$ outputs $(M_i^*, \sigma_i^*, R_i^*)$, $i = 1, 2$, and succeeds if it holds that $\mathsf{Verify}_{R_i^*}(M_i^*, \sigma_i^*) = 1$ and $R_i^* \subseteq S$ for $i = 1, 2$, $\mathsf{Link}(\sigma_1^*, \sigma_2^*) = 0$, and $|(R_1^* \cup R_2^*) \cap C| + |(R_1^* \cup R_2^*) \backslash S| \leq 1$, where C is the set of corrupted users.*

Definition 5 (Linkability w.r.t. adversarially-chosen keys). *A linkable ring signature is linkable w.r.t. adversarially-chosen keys if for any PPT adversary $\mathcal{A}$ and for any polynomial $n(\cdot)$, the probability that $\mathcal{A}$ succeeds in the previous game, without restricting the rings R_0^*, R_1^* to be subsets of S in the output phase, and without requiring $R \subseteq S$ in $\mathcal{SO}(\cdot, \cdot, R)$ queries, is negligible.*

Non-slanderability. Non-slanderability was not considered in the first paper on linkable ring signatures [16]. It first appeared in [21]. The same security requirement falls under the notion of linkability as a sub-requirement in [17]. Non-Slanderability in [17] is against chosen-subring attacks while that in [21] is also w.r.t. insider corruption. Again, none of the existing models considers non-slanderability w.r.t. adversarially-chosen keys. It is worth noting that, however, the schemes in [16] and [17] can in fact be proven non-slanderable w.r.t. adversarially-chosen keys.

Definition 6 (Non-slanderability). *A linkable ring signature is non-slanderable if for any PPT adversary $\mathcal{A}$ and for any polynomial $n(\cdot)$, the probability that $\mathcal{A}$ succeeds in the following game is negligible:*

1. *(Initialization Phase.) As in Definition 1.*
2. *(Probing Phase.) As in Definition 1.*
3. *(Output Phase.)* $\mathcal{A}$ *outputs* $(\hat{\sigma}, M^*, \sigma^*, R^*)$ *and succeeds if* $R^* \subseteq S$, $\hat{\sigma}$ *is the output of* $\mathcal{SO}(\hat{s}, \hat{M}, \hat{R})$ *for some* $\hat{R} \subseteq S$ *and* $\hat{s}$ *such that* $PK_{\hat{s}} \in \hat{R} \cap S$, $\mathsf{Verify}_{R^*}(M^*, \sigma^*) = 1$, $\mathsf{Link}(\hat{\sigma}, \sigma^*) = 1$, *and* $\mathcal{A}$ *never queried* $\hat{s}$ *to* $\mathcal{CO}(\cdot)$.

Definition 7 (Non-slanderability w.r.t. adversarially-chosen keys). *A linkable ring signature is non-slanderable w.r.t. adversarially-chosen keys if for any PPT adversary* $\mathcal{A}$ *and for any polynomial* $n(\cdot)$, *the probability that* $\mathcal{A}$ *succeeds in the above game, without the restrictions which the rings* $\hat{R}, R^*$ *are subsets of* S *in the output phase and* $R \subseteq S$ *in* $\mathcal{SO}(\cdot, \cdot, R)$ *queries, is negligible.*

We close this section by a quick summary – our proposed security model is strictly stronger than the one in [21] because the former has:

- equally strong unforgeability,
- at least as strong linkable-anonymity,
- strictly stronger linkability, and
- at least as strong non-slanderability.

5 Our Proposed Construction

Our construction improves, and is thus based on, SLRS of [21]. We first give an overview of the construction, highlighting the amendments made to SLRS.

As opposed to SLRS, KeyGen no longer generates key pairs for all users, and is hence no longer executed by a single entity. It is instead in our construction an algorithm executed by every user to generate only his/her key pair.

We introduce the CA to the scheme. Similar to any Public Key Infrastructure (PKI) [1], the CA in our construction is responsible for certifying user public keys, by signing on the public key and possibly along with some other identifying information. We require the CA in our scheme to certify a public key only if its bearer can prove that the key pair was generated according to specification. Note that the CA is trusted to do the above task honestly.

Recall that in SLRS, a user key pair $(sk, pk) = ((p, q), y)$ is generated in such a way that y is a prime that equals $2pq + 1$ with p, q being primes of same size. Testing the primality of y by the CA is can be done in polynomial time, therefore it suffices for a user to prove in zero-knowledge to the CA that $(y - 1)/2$ is a product of two primes of the same size. We use the protocol in [6] to accomplish this job and call the protocol Prove.

The protocol Prove is thus an interactive two-party protocol between a user and the CA. The input to the user is his/her private key. The common input to both parties is the corresponding user public key. At the end of the protocol run, Prove outputs 1 at the CA side only if the user has the knowledge of the private key, in which case the CA issues the user a certificate.

As a consequence of the above, the signing and verification algorithms are modified to work only with certified user public keys.

5.1 Algorithms

Now we proceed to enumerate the actual algorithms. To save space and enhance readability, we put down only changes made to SLRS, reviewed in section 3.

- Init. Same as in SLRS.
- $\mathsf{Key\text{-}Gen}$. On input the system's parameters param, the algorithm parses param into $(1^\lambda, \mathsf{desc}, \tilde{g})$ and then executes the probabilistic sampling algorithm W of the accumulator to obtain $(y_i, (p_i, q_i))$. Finally it outputs the key pair (sk_i, pk_i), where $sk_i := (p_i, q_i)$ and $pk_i := y_i$. Upon obtaining the key pair, the algorithm executes the Prove protocol with the CA to obtain a certificate. The user public key is then augmented with the certificate and the key pair is returned.
- Sign. Same as in SLRS, except that the algorithm first verifies the certificates and terminates without an output if not all of them are valid.
- Verify. Same as in SLRS, except that the algorithm first verifies the certificates and outputs `invalid` if not all of them are valid.
- Link. Same as in SLRS.

5.2 Security Analysis

The following theorems collectively prove the security of our proposed construction under our proposed security model. Their proofs are in Appendix B.

Theorem 1. *If the DDH in $QR(N)$ problem, the LD-RSA problem and the SRSA problem are hard, our construction is unforgeable in the random oracle model.*

Theorem 2. *If the DDH in $QR(N)$ problem and the LD-RSA problem are hard, then our construction is linkably-anonymous w.r.t. adversarially-chosen keys in the random oracle model.*

Theorem 3. *If the DDH in $QR(N)$ problem, the LD-RSA problem and the SRSA problem are hard, then our construction is linkable w.r.t. adversarially-chosen keys in the random oracle model.*

Theorem 4. *If the DDH in $QR(N)$ problem, the LD-RSA problem and the SRSA problem are hard, then our construction is non-slanderable w.r.t. adversarially-chosen keys in the random oracle model.*

6 Conclusion

Linkable ring signature has found many applications. Short signature size tremendously increases the practicality of the applications. We have presented a thorough review of the first short linkable ring signature due to [21] together with a discussion on a blemish in their security model and its security implications.

In response to the recent refinement of the security model for conventional ring signature [4], we have surveyed all the existing security models for linkable ring signature and formulated a new one which is the strongest among all as in it captures realistic adversarial capabilities not considered in the context of linkable ring signature before. We have also, under the new model, proposed a new secure short linkable ring signature scheme. Future research directions include finding more novel applications of linkable ring signature, and extending other short ring signature schemes (e.g. [22]) to provide linkability.

References

1. Carlisle Adams and Stephen Farrell. Internet X.509 Public Key Infrastructure Certificate Management Protocols. Internet Engineering Task Force: RFC 2510.
2. Ben Adida, Susan Hohenberger, and Ronald L. Rivest. Separable Identity-Based Ring Signatures: Theoretical Foundations For Fighting Phishing Attacks. In *DIMACS Workshop on Theft in E-Commerce: Content, Identity, and Service, April 14 - 15, 2005, Piscataway, NJ, USA*, 2005. To appear.
3. Mihir Bellare and Phillip Rogaway. Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. In *1st ACM Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.
4. Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring Signatures: Stronger Definitions, and Constructions without Random Oracles. In *Theory of Cryptography, TCC 2006*, volume 3876 of *LNCS*, pages 60–79. Springer, 2006.
5. Jan Camenisch and Anna Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76. Springer, 2002.
6. Jan Camenisch and Markus Michels. Separability and Efficiency for Generic Group Signature Schemes. In *CRYPTO 1999*, volume 1666 of *LNCS*, pages 413–430. Springer-Verlag, 1999.
7. Jan Camenisch and Markus Stadler. Efficient Group Signature Schemes for Large Groups (Extended Abstract). In *CRYPTO 1997*, volume 1294 of *LNCS*, pages 410–424. Springer-Verlag, 1997.
8. David Chaum and Eugène van Heyst. Group Signatures. In *EUROCRYPT 1991*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.
9. Liqun Chen, Caroline Kudla, and Kenneth G. Paterson. Concurrent Signatures. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 287–305. Springer, 2004.
10. Sherman S. M. Chow, Siu-Ming Yiu, and Lucas Chi Kwong Hui. Efficient Identity Based Ring Signature. In *Applied Cryptography and Network Security, ACNS 2005*, volume 3531 of *LNCS*, pages 499–512, 2005.
11. Sherman S.M. Chow, Joseph K. Liu, Victor K. Wei, and Tsz Hon Yuen. Ring Signature without Random Oracles. In *ACM Symposium on InformAtion, Computer and Communications Security, ASIACCS 2006*, 2006. To appear, also available at Cryptology ePrint Archive, Report 2005/317.
12. Sherman S.M. Chow, Richard W.C. Lui, Lucas C.K. Hui, and S.M. Yiu. Identity Based Ring Signature: Why, How and What Next. In *Second European PKI Workshop, EuroPKI 2005*, volume 3545 of *LNCS*, pages 144–161. Springer, 2005.
13. Sherman S.M. Chow and Willy Susilo. Generic Construction of (Identity-based) Perfect Concurrent Signatures. In *Information and Communications Security, ICICS 2005*, volume 3783 of *LNCS*, pages 194–206. Springer, 2005.
14. Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous Identification in Ad Hoc Groups. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 609–626. Springer, 2004.

15. Uriel Fiege, Amos Fiat, and Adi Shamir. Zero Knowledge Proofs of Identity. In *STOC '87: 19th Annual ACM conference on Theory of Computing*, pages 210–217, New York, NY, USA, 1987. ACM Press.
16. Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. In *Australasian Conference on Information Security and Privacy, ACISP 2004*, volume 3108 of *LNCS*, pages 325–335, 2004.
17. Joseph K. Liu and Duncan S. Wong. Linkable Ring Signatures: Security Models and New Schemes (Extended Abstract). In *Computational Science and Its Applications - ICCSA 2005*, volume 3481 of *LNCS*, pages 614 – 623. Springer, 2005.
18. David Pointcheval and Jacques Stern. Security Proofs for Signature Schemes. In *EUROCRYPT 1996*, volume 1070 of *LNCS*, pages 387–398, 1996.
19. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to Leak a Secret. In *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, 2001.
20. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to Leak a Secret: Theory and Applications of Ring Signatures. In *Essays in Theoretical Computer Science: in Memory of Shimon Even*, volume 3895 of *LNCS*. Springer, 2006.
21. Patrick P. Tsang and Victor K. Wei. Short Linkable Ring Signatures for E-Voting, E-Cash and Attestation. In *Information Security Practice and Experience, ISPEC 2005*, volume 3439 of *LNCS*, pages 48–60. Springer, 2005.
22. Qianhong Wu, Fangguo Zhang, Willy Susilo, and Yi Mu. An Efficient Static Blind Ring Signature Scheme. In *International Conference on Information Security and Cryptology, ICISC 2005*, LNCS. Springer, 2006. To appear.

A Security Threats in [21]

We show how the central authority in the short linkable ring signature construction due to [21] could violate the security notions defined in the same paper.

- **Unforgeability.** Since user key pairs are generated and thus known by the central authority, the malicious central authority can universally forge any signatures. Note that forgery in ring signatures often goes undetected.[3] Worse still, the above point also renders the scheme repudiable, as users can always deny having generated a signature and frame the central authority.
- **Non-slanderability.** The central authority can slander any user by forging a signature on behalf of the user using his/her private key.
- **Linkable-anonymity.** The central authority possesses all key pairs, and can therefore reveal with convincing evidence the signer-identity of a signature by generating linkable ring signatures for all possible signers and testing to which among the generated signatures the target signature is linked.
- **Linkability.** The central authority is able to maliciously produce user key pairs which can be used to sign multiple times without being linked. We show

[3] When a conventional signature is forged, the victim usually gets to know the forgery sooner or later when he/she is held responsible for the signature, e.g. when he/she receive a bill for some online shopping he/she never did. On the contrary, when a ring signature is forged on behalf of a ring of members, every member in the ring can only suspect that certain other member in the ring has signed.

how this can be done as follows. Suppose the central authority generates a malicious public key $x = 2(\prod_{i=1}^{2k} a_i) + 1$ for some integer k such that $a_i \in S(2^{\ell/k}, 2^{\mu/k})$. Define $\mathcal{J}$ be a set of integers such that $|\mathcal{J}| = k$ and $\mathcal{J} \subset \{1, \cdots, 2k\}$. For the C_k^{2k} choices of $\mathcal{J}$, let $\hat{e}_1 = \prod_{i \in \mathcal{J}} a_i$ and $\hat{e}_2 = \prod_{i \notin \mathcal{J}} a_i$. It is straightforward to show that each of these pair of $(\hat{e}_1, \hat{e}_2)$ can generate ring signature of x without being linked.

B Security Proofs

B.1 Unforgeability

Proof (Theorem 1). It is straightforward to show that non-slanderability and linkability together imply unforgeability. For if an adversary can forge a signature, he can either slander an honest user or collude with any user to break the linkability of the scheme. Therefore, the proof for is a direct consequence of Theorems 3 and 4. □

B.2 Linkable-Anonymity

Proof (Theorem 2). We construct a simulator $\mathcal{S}$ from adversary $\mathcal{A}$, which wins game linkable-anonymity with non-negligible advantage $1/2 + \epsilon$ over random guessing, to solve the LD-RSA problem under the DDH assumption.

$\mathcal{S}$ is given an instance of the LD-RSA problem $(n_0, n_1, T = g^{p_b+q_b})$, with $p_b q_b = n_0$ or n_1. $\mathcal{S}$ creates the system parameters correspondingly and randomly generate a set of key pairs $\mathcal{X} = \{(PK_i, SK_i)\}$ using KeyGen. It randomly chooses a bit $b' = 0$ or 1 and sets $PK^* = 2n'_b + 1$. Denote $\mathcal{X}^* = \mathcal{X} \cup \{PK^*\}$, which is then given to $\mathcal{A}$ as the set of public keys.

$\mathcal{S}$ handles the $\mathcal{SO}$ query as follows. For query involving $PK \in \mathcal{X}$ as the signer, $\mathcal{S}$ is in possession of the secret key and can reply according to the algorithm specification. There are two ways to handle the public key generated adversarially. One is to ban the adversary from having this type of query. The rationale is that since the adversary generate this public key, it should be in possession of the secret key and $\mathcal{SO}$ query on such key provides no useful information for the adversary. The second way is that adversary has to prove the validity of the generated key pair to the simulator before it can be used. This models the scenario in practice when users need to prove the validity of the key pairs before CA issues a certificate for it. In this case, the simulator extracts the secret key of the corresponding public key during the proof of validity of the key and the rest is straightforward. Finally, queries involving PK^* as the signer need special attention. $\mathcal{S}$ sets tag $\tilde{y} = T$ and computes the signature of knowledge in equation (1). Under the DDH assumption in $QR(N)$, the simulated signature of knowledge is indistinguishable from the actual one if T is correctly formed. T is correctly formed if and only if $b = b'$.

If PK^* is chosen to be the challenge signature, then $\mathcal{A}$ wins the game with probability $1/2 + \epsilon$ if T is correctly formed. On the other hand, $\mathcal{A}$ can only win with probability 1/2 since the whole challenge signature is not related to either i_0 or i_1.

In the challenge phase, with probability $2/|\mathcal{X}^*|$, $PK^* \in \{i_0, i_1\}$. If T is correctly formed, then $\mathcal{A}$ wins the game with probability $1/2 + \epsilon$. Otherwise, $\mathcal{A}$ can only win with probability 1/2. Thus, $\mathcal{S}$ solves the LD-RSA problem with probability $1/2 + \epsilon/2|\mathcal{X}^*|$ which is non-negligible over random guessing. □

B.3 Linkability

Proof (Theorem 3). (Sketch.) Since adversary $\mathcal{A}$ is only holding one secret key for the two signatures, there is only one valid $\tilde{y}$ for which $\mathcal{A}$ can produce, with overwhelming probability under the LD-RSA assumption. For $\mathcal{A}$ to break the linkability property, it has to convince a verifier to accept an $\tilde{y}$ for which it cannot generate a correct signature of knowledge with any non-negligible probability.

Then $\mathcal{A}$ must have conducted an *incorrect* proof in the signature of knowledge such that at least one of the following is fake: the first part which is a constant-size ring signature due to Dodis, *et al.* [14], and the second part which is a proof that $\tilde{y}$ is correctly formed.

The first part is fake with negligible probability under the SRSA assumption, as a successful forging implies breaking the unforgeability of the constant-size ring signature due to [14]. The second part is fake with negligible probability under the (computational) LD-RSA assumption. Thus, the total success probability of $\mathcal{A}$ in is negligible. □

B.4 Non-Slanderability

Proof (Theorem 4). (Sketch.) We outline how to construct a simulator $\mathcal{S}$, having black-box access to an adversary $\mathcal{A}$ which can slander an honest user, to solve the LD-RSA problem.

Setup and simulation of $\mathcal{SO}$ is the same as in the proof of linkable-anonymity. With probability $1/|\mathcal{X}^*|$, $\mathcal{A}$ outputs a signature which slanders PK^*. Due to the soundness of the SPK in (1), there exists an extractor which can extract the secret key ($\hat{p}$,$\hat{q}$) corresponds to PK^* such that $PK^* = 2\hat{p}\hat{q} + 1$. With $\hat{p}$ and $\hat{q}$, $\mathcal{S}$ solves the LD-RSA problem. (In fact the computational version of the LD-RSA problem). □

An Infrastructure Supporting Secure Internet Routing

Stephen Kent

BBN Technologies
Cambridge, MA 02138 USA

Abstract. The Border Gateway Protocol (BGP) [1] is the foundation of inter-domain Internet routing. A number of papers have described how BGP is highly vulnerable to a wide range of attacks [2, 3], and several proposals have been offered to secure BGP [4, 5, 6, 7, 8]. Most of these proposed mechanisms rely on a PKI, to provide trusted inputs for routing security mechanisms, to enable BGP routers to reject bogus routing advertisements. This paper provides a detailed proposal for a PKI, including a repository system, representing IP address allocation and Autonomous System number assignment,. This infrastructure offers a near term opportunity to improve routing security, since it does not require changes to routers, while also setting the stage for more comprehensive BGP security initiatives in the future.

1 Background

Inter-domain Internet routing is effected via a distributed system composed of many routers, grouped into management domains called Autonomous Systems (ASes), each identified by an AS number. Routers at the perimeter of each AS are called border routers, and BGP is the protocol executed between them. Routing information, most importantly AS path information (described below), is propagated between ASes using BGP UPDATE messages. Enabling border routers to verify that routes propagated via these messages are "valid" is the primary focus of several proposed BGP security technologies [4, 5, 6, 7, 8].

Although these proposals differ in may respects, all rely on the existence of a PKI attesting to resource holdings, specifically address blocks and AS numbers. Two of the proposals, S-BGP and soBGP, have provided details of how to organize such a PKI, while other proposals have either assumed the existence of the S-BGP PKI or have ignored PKI details and focused on digital signature optimization. To date, there has been essentially no progress in deploying any BGP security enhancements. Some require that more memory, and possibly crypto accelerator hardware, be added to border routers. In the current economic climate for ISPs, this is a very difficult expenditure to justify. The major router vendors no longer garner most of their revenue from sales to ISPs, so they are reluctant to invest in developing routers targeted toward the ISP market. Thus it may be a long time before such BGP security technologies can be deployed.

However, the sort of infrastructure that these security mechanisms assume as an underpinning can offer improved security prior to the deployment of such mechanisms. Creation of the infrastructure described below can be viewed as a first step toward

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 116 – 129, 2006.

more comprehensive security mechanisms. A concerted effort is now (2006) underway to secure agreement on design and deployment details for such a PKI. Staff from all five Regional Internet Registries (RIRs) are meeting to refine the design, and trials are underway. This paper describes the current design, derived from the S-BGP PKI model [9], noting new design aspects and details.

2 Securing Route Origination

Even if one does not deploy a BGP security solution that relies on such an infrastructure, the availability of the infrastructure would allow ISPs to detect bogus route origination. Bogus route origination occurs whenever an AS advertises itself as the origin AS for a prefix, without being authorized to do so by the (legitimate) holder of the prefix. This appears to be one of the most common forms of routing errors today, often arising from configuration errors by network operators. It also can arise from technical or social engineering attacks against ISPs, causing an ISP to advertise a route for a prefix that is not legitimately associated with a subscriber of the ISP. Some spam attacks are facilitated by this so-called “prefix hijacking.” In either case, the bogus route origination will propagate through the Internet if neighboring ISPs do not filter UPDATEs to remove such errors.

Some ISPs use Internet Routing Registry (IRR) data to configure route filters, in an effort to reject bogus routes of various forms. However, network operators complain that the extent and quality of IRR data varies considerably by geopolitical region. There are no intrinsic quality controls on the IRR data, i.e., each ISP or multi-homed subscriber enters its own data into an IRR and, not surprisingly, errors arise. No authority is responsible for quality control of IRR data. Thus the ability to use such data in an automated fashion to create accurate route filters is limited. In contrast, the infrastructure described in this paper provides intrinsic controls on the data to which it attests, allowing automated detection of many forms of errors, as well as protection against attacks on the integrity or authenticity of the data.

The proposed security infrastructure consists of three components: a PKI, route origination authorizations (ROAs), and repositories. The PKI represents the allocation of address blocks and AS numbers to organizations. The ROAs enable an organization to explicitly authorize one or more ASes to originate routes to its address blocks. Repositories provide the means of distributing the PKI and ROA data to interested parties. The intent is that network operators upload to repositories any PKI or ROA data as it changes, and periodically download (e.g., on a daily basis) data uploaded by others. From this data, operators can extract authenticated address prefix origination data, which can be used to construct route filters in a more secure fashion than is currently offered via the IRR system. The following sections describe in greater detail how this data is represented, maintained, and processed.

3 Address Block Allocation and AS Number Assignment

IP addresses and AS numbers are allocated via a geopolitical, tree-structured scheme that ensures uniqueness. The root of the tree is the Internet Assigned Numbers Authority (IANA), which performs this and other operational functions on behalf of the Internet Corporation for Assigned Names and Numbers (ICANN). Under IANA are five Regional Internet Registries (RIRs), each serving a different geopolitical area: ARIN (North America), RIPE NCC (Europe), APNIC (Asia and Pacific), LACNIC (Latin America and Caribbean), and AfriNIC (Africa). In some regions, national or local registries (NIRs/LIRs) form a subordinate registry tier of the hierarchy for address and AS number allocation.

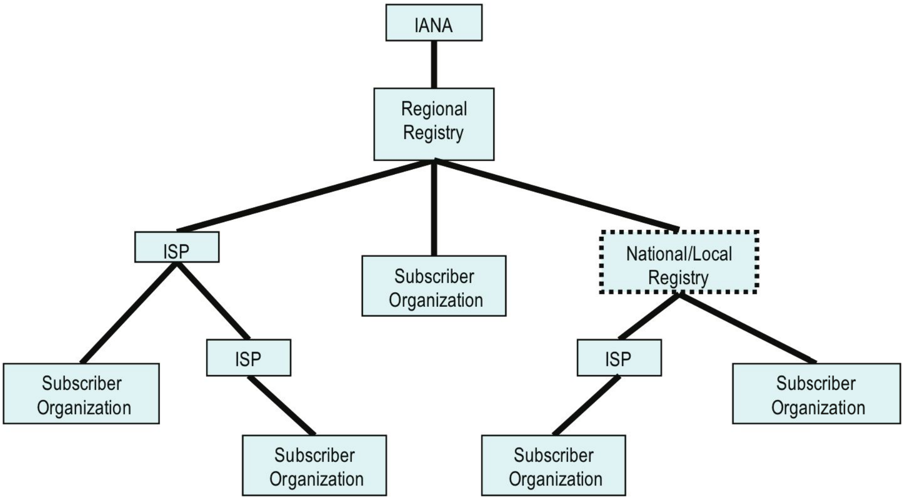


Fig. 1. Address Allocation Hierarchy Structure

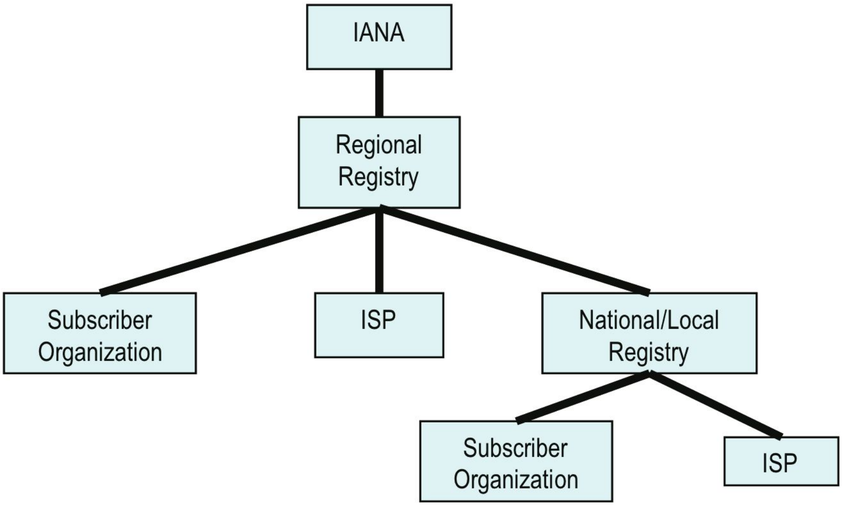


Fig. 2. AS Number Assignment Hierarchy

IANA allocation large address blocks to the RIRs. Registries then allocate addresses to ISPs and to subscribers who wish to receive address blocks not tied to a specific ISP, e.g., because they plan to be multi-homed or because they want strong guarantees of address portability. ISPs allocate addresses to subscribers, some of whom may be smaller ISPs. AS numbers are allocated by IANA to the RIRs, who may sub-allocate them to LIRs/NIRs,. Registries assign AS numbers to ISPs and to subscribers who require them, e.g., a multi-homed subscriber. Figure 1 illustrates the schema for the current address allocation hierarchy. Figure 2 illustrates the schema for the current AS number assignment hierarchy.

Prior to the creation of the RIRs, IANA and others[1] allocated address blocks and assigned AS numbers directly to ISPs and subscribers in a very simple, two-tier hierarchy. Some of these so-called "legacy" allocations of address space remain under IANA, not allocated to any RIR. Some organizations with legacy allocations appear in the whois databases operated by RIRs, but are not formally members of the RIRs. These allocations need to be accommodated in this infrastructure, and they pose some political, though not technical, challenges, as discussed in Section 4.2.

4 The Proposed PKI

The simple structure of the address and AS number allocation hierarchies, and the fact that the organizations that perform the allocation functions are viewed as authoritative, makes this an ideal context for creation of a PKI. The S-BGP design initially proposed [9] creation of two PKIs, one for address allocation and one for AS number assignment. Later, S-BGP [7] proposed a unified PKI, taking advantage of the parallel structure of the address block and AS number allocation hierarchies. ISPs and multi-homed subscribers typically hold both address blocks and AS numbers, and many ISPs have multiple address block allocations. Use of a single hierarchy enables issuance of a single certificate that consolidates both types of holdings, if desired.

Note that the certificates issued under this hierarchy are used for authorization in support of routing security, not for identification. The intent is to allow the holder of a set of address blocks to be able to announce to the Internet, in a secure fashion, the AS number of each entity that is authorized to originate a route to these addresses. The PKI satisfies the first part of this goal by binding a public key to each address block holder, through the use of a new certificate extension [14]. Note that the name of the address block holder need not be "meaningful" to satisfy this requirement. For purposes of routing security, the issuer and subject name in each certificate are not relevant, other than the usual PKI requirements for contextual uniqueness in support of unambiguous certificate path chaining.

This focus is subtly and importantly different from the requirements nominally associated with RIR (e.g., whois) databases. Those databases strive to associate accurate organizational names, and contact information (e.g., an individual's name, postal address, phone number, and e-mail address) with each entry. Experience has shown that it is difficult to maintain all of this information accurately for each address

[1] Prior to the creation of IANA, there were other central registries for IP addresses, including the SRI NIC and the Internic (during the NSFnet era).

block holder, e.g., due to mergers, acquisitions, bankruptcy, personnel turnover, etc. This PKI, because it does not require changes to certificates to track such organizational data changes, is potentially less costly to manage.

Despite the emphasis on use of non-meaningful names for the ultimate resource holders (e.g., ISPs and subscribers), it is not detrimental for the top tier CAs to have traditional X.500-style names. For example, APNIC might be represented as: C = AU, O = Asia Pacific Regional Internet Registry, OU = Resource Holding Certification Authority. This sort of name clearly identifies the RIR in its role as a top tier CA.

Each CA publishes a certification practice statement (CPS) [12]. A CPS for this PKI differs in an important way from a typical CPS. The CAs here are the authoritative record holders for the resource holdings that are the focus of the PKI. The common notion of selecting a CA based on how well it declares it will do its job (via a CPS) is irrelevant here. These CAs state only that they issue certificates consistent with their existing databases and practices. Nonetheless, each CA needs to publish a CPS if only to minimize potential liability.

In the proposed PKI, we expect organizations issuing certificates at or near the top of the hierarchy (e.g., IANA and registries) will employ the most stringent procedures and security measures, whereas lower tier issuers need less stringent procedures and security measures. This is because these organizations manage the allocation of the whole of the address and AS number space, and thus security breaches here have very widespread effects. In contrast, security failures by lower tier issuers, e.g., small ISPs, have limited impact.

This PKI will operate under an explicit certificate policy (CP) that describes the acceptable uses of the certificates issued here. For this PKI, the policy reflects the use of the certificates to attest to resource holdings, primarily in support of routing security. Appropriate uses for these certificates include authentication to an RIR or an ISP in support of database maintenance and the secure Neighbor Discovery protocol [11], which enables an IPv6 host to locate routers, etc.

For the certificates and CRLs that are issued under the PKI, explicit representation of a CP in a certificate, via an X.509 certificate extension [13], helps protect the issuer against liability claims in case of misuse of these certificates.

4.1 The Address and AS Number Extensions

RFC 3779 [14] defines two X.509 extensions that establish a standard format for representing address blocks and AS numbers in certificates. The extensions represent both IPv4 and IPv6 addresses as ranges, not just as prefixes, since some address block allocations are not aligned on the boundaries defined by prefixes and used by BGP. The address extension accommodates multiple IPv4 and IPv6 address blocks. The AS number extension represents blocks of AS numbers, or individual AS numbers. Both the current 16-bit and the newer 32-bit AS number formats are supported.

As explained in RFC 3779, for each organization in the allocation hierarchy, the intent is to associate these extensions with a certificate representing the set of address blocks and AS numbers allocated to the organization. At each successive tier along a path through the hierarchy, the extensions in an organization's certificate represent a subset of the allocation associated with the issuer of the certificate, consistent with the fundamental notion that an organization is not authorized to sub-allocate addresses that have not been allocated to it. The same holds true for AS numbers, except that, by convention, ISPs are not allowed to sub-allocate their AS number allocations.

This subset relationship must be verified by relying parties, an enhancement to normal certificate path validation. Such checking can detect mistakes (or malicious errors) in certificate issuance by the organizations at each tier. The certificate path validation algorithm defined in RFC 3280 (section 6) must be augmented to include the address and AS number extensions as additional trust anchor information, and to perform a subset check for these extensions for each certificate along the path. These checks are analogous to certificate policy checks, but simpler since there are no policy mapping or inhibit policy mapping features to consider. (Developers working on this PKI are adding this validation algorithm extension to OpenSSL software.)

Representing allocation of address blocks and AS numbers via these extensions is relatively simple, but there are some subtleties. For example, an organization may have received address allocations from two different sources, e.g., an RIR and an ISP. In that case, the organization requires two certificates, one issued by the RIR and one by the relevant ISP, in order to preserve the subset relationship. If the organization chooses to use the same name in both certificates, this phenomenon merely looks like a CA with multiple certificates, a common PKI feature. The extensions also provide a flag to express inheritance of the extension values from a higher tier CA, for space efficiency.

4.2 PKI Structure and Operation

The root of the PKI, nominally IANA, will be represented by a self-signed certificate that encompasses all address space and all AS numbers. This certificate should have a very long validity interval, perhaps 10 or more years. The root private key should be large enough to be secure for a long interval, e.g., a 2048-bit RSA key[2], and it requires high assurance protection. One would want the root to employ a crypto module evaluated under FIPS 140-2 [15], e.g., at level 3. Commercial root CAs whose certificates are embedded in browsers typically use similar validity intervals and key sizes, and also employ FIPS-evaluated crypto modules. This self-signed root certificate will be distributed by embedding it in software used to validate certificates and signed objects under this PKI. This is equivalent to the way that SSL root CA public keys are distributed via browsers. It also will be available via the repository described later.

The root issues to each RIR a certificate representing the address blocks and AS numbers allocated to that RIR. These allocations change relatively infrequently, e.g., semi-annually, and thus the root CA will not incur a substantial burden in its interaction with RIRs. Here too it seems appropriate for these certificates to have long validity intervals and to use high assurance crypto modules to protect private keys.

If a region employs national or local registries (NIRs/LIRs), each such registry will receive its certificate from its parent RIR. National/local registries are long-lived organizational entities, so it seems appropriate to use comparably long validity intervals (as for the RIRs), and to employ high assurance crypto modules here as well.

[2] This key size and validity interval is consistent with current browser root CA key parameters. NIST [10] states that a 2048-bit RSA key is suitable for use through 2030.

As noted in Section 3, legacy address allocations are not formally held under any RIR. One means of dealing with these allocations is to have IANA maintain a separate subtree for them. This avoids the political issues associated with issuing certificates for these allocations under a specific RIR, since even the legacy resource holders acknowledge IANA as the (modern) source of their allocations. Several special-case allocations also should be issued certificates under the root. For example, the 10/8, 172.16/12, and , 192.168/16, address blocks are reserved for local use, and are designated as not routable. By issuing certificates for these address blocks under the root, no other entity could issue certificates for them accidentally (or otherwise).

An ISP may receive a certificate from an RIR, NIR/LIR, or even another ISP. An ISP will be issued more than one certificate if it has received address allocations from different sources. For example, an ISP may receive one allocation from a registry, and another allocation from another ISP, due to historical business relationships. In order to preserve the ability for a relying party to readily verify the validity of the address block and AS number extensions, a strict hierarchy must be maintained for path validation. Thus ISPs that have received allocations from multiple sources require multiple certificates. Note that different certificates issued to the same ISP to represent different allocations of address blocks (or AS numbers) from different organizations could all carry the same public key and the same subject name. They would have different issuer names and contain different extensions representing different allocations. The same holds true for subscribers who receive address or AS number allocations from multiple sources. Figure 3 includes an example of a subscriber of this sort (SUB_L).

When an organization receives an initial allocation of address blocks or AS numbers, it would typically receive one certificate. If the organization acquires additional address blocks (or AS numbers) over time, from the same issuer, a new certificate can be issued with the additional allocations included. However, an organization may want to treat the new allocation independently, e.g. because it will be managed by a different part of the organization. In this case the certificate issued to reflect the new allocation need not contain the old allocation. The new certificate may share the (subject) name and public key with the old certificate, or may use a new name and key.

If no address blocks or AS numbers are de-allocated when an organization receives a new allocation, there is no need to revoke the organization's old certificate, nor is there a need to change the private key. Keeping the key constant minimizes the impact on organizations further down the certification path, since it avoids the need to reissue certificates for those organizations. If an address block or AS number is no longer associated with an organization, then the certificate asserting that holding must be revoked when the resources are returned.

A registry generally would not return address blocks to a CA above it in the PKI, but it may transfer address blocks to another registry. In such cases, the registry transferring the allocation would issue a cross certificate to the registry receiving the allocation. On rare occasions, an ISP may return an address block to a registry (or ISP) from which it was acquired. The PKI accommodates returns or transfers of address space or AS numbers at any tier in the hierarchy, through revocation and reissuance of certificates, performed in the proper order.

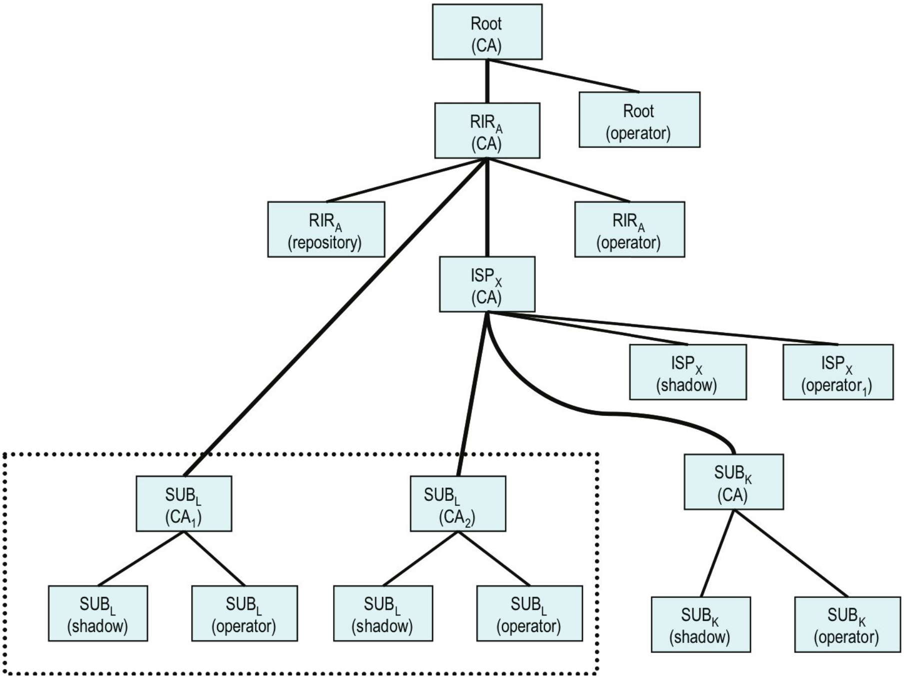


Fig. 3. CA, Shadow, and Operator Certificates Example

4.3 PKI Details

Each of the organizations in the address and AS number allocation hierarchy is represented in the PKI by one or more certificates, as noted above. All of these are CA certificates. It is obvious that the upper tier entities in the PKI must be CAs, because they issue certificates to lower tiers. However, even the leaves in the allocation hierarchy, small ISPs or multi-homed subscribers, need to be CAs as well. The motivation for this arises from considering the operational goals of the overall system, i.e., using the PKI to verify address space and AS number holder assertions.

The repositories that will hold certificates, CRLs, and ROAs require access controls to ensure that uploading data does not violate the inherent access control policy. All of this data is digitally signed, and thus tampering is detectable by those who download the data. However, overwriting repository entries with bogus data or out-of-date data would adversely affect operations, and that motivates controls on uploading data into repositories. A simple way to provide access controls is to require uploads to be authenticated using certificates issued under the PKI (see Section 7). A certificate used for this purpose would be issued to ISP or subscriber operations personnel, and thus would be traceable to the point in the PKI that the personnel represent. In that fashion, only someone authorized by the holder of a set of address blocks is authorized to upload certificates, CRLs, and ROAs for the address space in question. This argues for each address block holder to issue one or more end-entity

(EE) certificates to the operations staff responsible for managing the repository entries associated with those address blocks. The need to issue EE certificates for this purpose requires that each address block holder be represented by a CA certificate.

Another motivation for issuing (EE) certificates to operations personnel in support of repository access control, is that when such personnel leave, the issuing ISP can revoke their certificates, instead of having to change the key pair corresponding to the resource holder's (CA) certificate. Thus the indirection provided by using EE certificates for this purpose is beneficial from a security and operations perspective. Note that these operations personnel certificates do not need to be maintained in repositories or downloaded by ISPs, since the certificates are used in a local fashion for access control, e.g., they could be transmitted to a repository in conjunction with access via SSL (see Section 6).

Each address space holder also will need to issue one or more ROAs to each ISP authorized to originate routes for the address blocks in question. The ROA could be signed using the private key from the CA certificate that binds address blocks to the holder. However, it is preferable to not use a CA key to sign data objects other than certificates and CRLs. Thus we propose that each address space holder issue a "shadow" EE certificate under each CA certificate that it holds. The shadow certificate is used only for verifying ROAs. Using shadow certificates offers several security benefits. For example, using the shadow certificate to verify ROAs allows one to revoke the shadow certificate as a way of revoking the ROAs signed by it. This avoids the need to define a separate revocation mechanism for ROAs. Also, by introducing shadow certificates we may reduce the frequency with which CA private keys need to be accessed, which allows these keys to be better protected.

If a shadow certificate is issued with the "inherit" flag set in the 3779 address space extension, it inherits all of the address blocks associated with the issuing CA certificate -- we expect this to be the default. In this case, even if a new CA certificate is issued, e.g., to reflect additional address allocations, the shadow certificate need not be reissued. However, an address block holder can assign a subset of its address allocations to a shadow certificate by explicitly including these in the extension, thus providing fine-grained control over the ROAs that can be signed using the private key corresponding to the shadow certificate.

Figure 3 illustrates the use of CA, shadow, and operators certificates in a simplified sample PKI. Each entity in this sample PKI is represented by a CA. One subscriber (SUB_L) has received address allocations from two entities (RIR_A and ISP_X) and thus is represented by two CAs (CA_1 and CA_2). ISP_X, and SUB_K each have one shadow certificate, and SUB_L has two shadow certificates (one for each CA). ISPs use shadow certificates to verify the ROAs issued by other ISPs.[3]. The root, RIR_A, and each subscriber CA have one operator certificate, used to interact with the repository system for upload access control. ISP_X has two operator certificates, illustrating the ability to issue distinct certificates to different individuals fulfilling operator roles, if

[3] No shadow certificates are shown for the root or registries. These entities allocate addresses, but are not ISPs or subscribers in these roles, and thus do not sign ROAs. If these entities have address allocations that need to be represented via this PKI, e.g., not allocated via ISPs, the organizational entities representing the root and registries can appear as subscribers, divorced from their address allocation roles in the PKI.

desired. One additional certificate is shown here, a repository certificate under an RIR. Section 8 describes the use of this certificate.

5 Route Origination Authorizations

This PKI does not, by itself, allow an address block holder to express the notion of which ASes it authorizes to originate routes to its address blocks. To represent this binding, we introduce a digitally signed object called a route origination authorization (ROA). A ROA is analogous to the address attestation data structure defined in S-BGP, but there are differences, so we adopted a new name and syntax. To understand what data elements need to be in a ROA, it is important to examine how ROAs will be used in different contexts.

Usually an ISP holds address space and originates routes for that space. In this case, the ISP would issue a ROA to itself for the address space it holds. A subscriber may receive an allocation of address space from a registry, and then become the client of an ISP. In that case, even if the subscriber is singly-homed, the subscriber must issue a ROA identifying the ISP as the authorized route originator for the subscriber address space. If the subscriber contracts with multiple ISPs, then the subscriber could issue one ROA naming all of the ISPs as authorized route originators, or it might issue separate ROAs, one to each ISP. This latter approach makes sense if different parts of the address space are to be advertised by each ISP, but otherwise it is not necessary.

If a subscriber is singly-homed, and receives its address space from the ISP to which it is homed, there is no need for the subscriber to receive a certificate or to issue a ROA. The ISP is the address space holder and route originator and thus its issuance of a ROA to itself suffices. If a subscriber moves from ISP_A to ISP_B, and keeps the address space from ISP_A, the requirements change. ISP_A needs to issue a certificate to the subscriber and the subscriber needs to issue a ROA authorizing ISP_B to originate routes to the address space in question, since otherwise the address space is held only by ISP_A. A similar situation arises if a singly-homed subscriber with address space from ISP_A contracts with ISP_B to become multi-homed. Usually both ISPs will advertise routes to the same address space for the subscriber. This requires the subscriber to receive a certificate for the address space (from ISP_A) and to issue ROAs to both ISPs authorizing them to originate routes for that address space. This is analogous to the procedures followed today by ISPs, with regard to advertising more specific prefixes, to ensure that multi-homing works in such cases.

In some instances a subscriber may receive service from ISP_A but want the Internet to think that a bigger ISP, ISP_B, is the originator for routes to the address space in question. Today, this is accomplished by having ISP_B advertise itself as the origin, discarding ISP_A's advertisement, and manually installing routing table entries to ensure forwarding of traffic to the subscriber. This can be formalized under the current scheme by having the subscriber issue a ROA listing ISP_B as the originator, and having ISP_A ignore this ROA.

Sometimes an ISP (or subscriber) may hold addresses allocated from multiple sources, e.g., a registry and an upstream ISP. In these cases it would be attractive to allow the ISP to issue one ROA that authorizes an AS to originate routes for all of the address holdings. The ROA format enables this by allowing multiple signatures[4], where each signer (a shadow certificate subject under an ISP CA) is authorized to represent a different address block.

Based on this analysis, a ROA must contain several critical pieces of data:

1. It must include the prefix(es) for which the address block holder is authorizing one or more ISPs to originate routes.
2. It must identify each authorized route originator by AS number.
3. It should contain pointers to the certificate(s) used to verify the ROA.
4. It must have a validity interval (or expiration date) to limit the time that it is treated as valid, and to facilitate orderly changeover from one ROA to another.
5. It must be digitally signed using the private key(s) associated with the address block holder's shadow certificate(s).

The detailed format for a ROA is yet to be decided. Use of ASN.1 is potentially attractive, since it allows for reuse of the syntax defined for the same purposes in other signed data structures, e.g., address block and AS number lists from RFC 3779, validity interval and signature structures from RFC 3280, etc. However, ASCII text also might be attractive as it would enable ISPs and subscribers to construct ROAs using existing tools, e.g., SMIME for signing. Figure 4 illustrates the abstract format of a ROA.

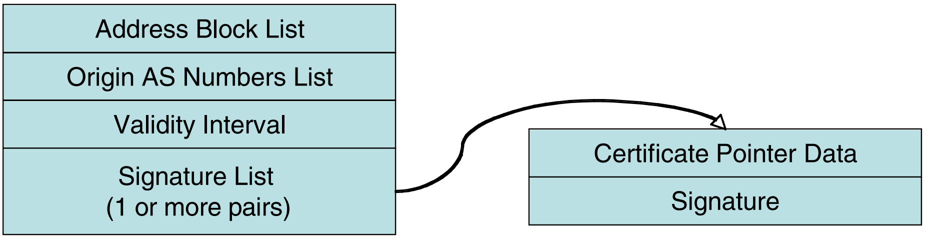


Fig. 4. ROA Abstract Format

As noted above, an ISP or subscriber may be represented by more than one CA, which motivates use of suitable back pointer(s) to the relevant shadow certificate(s). Many PKI-enabled application protocols identify a certificate by the combination of the certificate issuer's name and the certificate serial number. However, this combination of values may not always uniquely identify a certificate, e.g., if subject names are duplicated under different CAs. In X.509 certificates the subject key identifier (SKI), an extension required for CA certificates and recommended for EE certificates by RFC 3280 [13], can make use of the issuer name and serial number, or it may use another value, e.g., a truncated hash of a public key. We adopt this latter form of back pointer for ROAs. Thus a ROA includes an SKI value based on the hash of the public key from the shadow certificate that must be used to verify the signature on the ROA. A validity interval for a ROA could reuse the syntax from an X.509 certificate. If ASN.1 is employed, the signature would be represented using the syntax employed for certificates, CRLs, and many other digitally signed objects.

[4] The use of multiple signatures here precludes use of attribute certificates for this purpose.

6 A Repository System

Most contemporary PKIs make use of LDAP [16] as a repository for certificate and CRL distribution. However, these repositories are designed for an access model very different from what is required here. The certificates, CRLs, and ROAs described above must be retrieved by all network operators on a regular basis, e.g., daily. Thus the repository system for this PKI should be optimized for bulk downloading by a large user population (all ISPs), not for searching and retrieval of individual entries. The whois databases maintained by each RIR also are inappropriate in terms of their access models, e.g., they specifically prohibit bulk downloading.

However, the whois database model is relevant in that it seems to make sense for each RIR to offer a database containing the certificates and CRLs for all of its members, and for the entities to whom these members sub-allocate address blocks. (National or local registries under an RIR should also have their allocations represented here as well.) This would require that each ISP contact only five repositories in order to acquire all of the requisite PKI and ROA data, a manageable task for an ISP. The repository system need not be highly available in the sense of a DNS server, yet it should be robust. The ideal case would be a protocol that supports a query of the form "return all the entries that have changed since time X." Previous estimates [9] suggest that the database will be small enough so that even if each ISP downloaded all of the entries, the data transfers would be tolerable, at least for the near term.

Another distinct feature of this repository system is that all of the data stored in it is digitally signed. Thus tampering is detectable by operators when the data is downloaded. However, an access control system is still needed to prevent overwriting of data by an unauthorized party, and to prevent introduction of valid but superceded data, to minimize possible disruption of the repository service. We can make use of the structure of the data stored in the repository, plus the companion PKI, to enable automated access control management. Specifically, a repository can verify that data being uploaded by an ISP is consistent with the address space and AS number spaces allocated to the ISP, as reflected by the ISP operator's certificate. Each repository also can require use of SSL for operator access, to help reject nuisance attacks.

7 Using the Infrastructure

The goal of the infrastructure described above is to enable network operators to do a better job of securing routing using currently deployed routers. For example, ISPs can use the PKI and ROAs (and other signed objects yet to be defined in detail) to generate higher assurance inputs for generate of route filters, and to help thwart prefix hijacking via social engineering attacks against operations personnel.

We anticipate that a network operator would download certificates, CRLs, and ROAs from each RIR-operated repository, and verify each certificate path, using the procedure defined in RFC 3280, with the added requirement that the address blocks and AS numbers in the extensions of each certificate along the path must be a subset of the ones in the preceding certificate. This ensures that if any organization along the path (below the root) issues a certificate with address blocks or AS numbers that it was not allocated, the ISP will detect the error. Using the validated certificates, an ISP would verify the signature(s) on each ROA, and check that the address blocks listed in the ROA are held by the signer of the ROA.

The result of this procedure is a table that lists the address blocks for which each AS number is authorized to originate routes. Using this data, a network operator can construct route filters in an automated fashion, to detect and reject route advertisements with bogus route originations.

More generally, ISP_A might send a signed object to ISP_B, requesting ISP_B to accept a route from ISP_A even when ISP_A, does not originate the route. The signed object could include a nested, signed sequence of objects from ASes on the route that support the authorization of ISP_A to advertise the route. The signatures would be verified based on shadow certificates associated with the ISPs to attest to allocation of AS numbers. This is analogous to route attestation notion proposed in S-BGP. it would make use of out-of-band communication for route filter management, and thus would not be dynamic, but it also would not require changes to BGP.

Finally, a subscriber may attempt a social engineering attack in an effort to get an ISP to hijack address blocks unknowingly. To prevent this, an ISP could require a subscriber to digitally sign an electronic request for route origination, using the subscriber's operator certificate. An ISP could use the process described above to validate the operator certificate, and to check that the request for route origination is consistent with the address space held by the subscriber. This too can be an automated procedure, thus reducing operational costs for ISPs while also improving security.

8 Conclusions

This infrastructure offers a near term opportunity to improve routing security (better route filters, countermeasure against social engineering efforts to get ISPs to hijack address blocks, operator errors, etc.) without requiring changes to routers. It also sets the stage for more comprehensive BGP security initiatives in the future. The design of this infrastructure represents a departure from typical PKIs in many respects. If successful, this activity may encourage the creation of more PKIs where the CAs are authoritative for the data they certify, not merely "trusted" third parties who have inserted themselves into business practices.

Most of the proposals for BGP security say they would make use of the S-BGP PKI and all of them can make use of this infrastructure, e.g., S-BGP, SPV and recent work at Dartmouth [17]. psBGP [6] could make use of the AS number allocation portion of this PKI, but it calls for a different mechanism for issuing certificates that attest to address space allocation. soBGP [8] defines entity, policy, and authorization certificates, but only the entity certificate employs the X.509 format. soBGP also makes use of non-standard formats for revocation status data. Nonetheless, soBGP could make use of this PKI if it adopts standard certificate and CRL formats. Each ISP could configure a local set of trust anchors under this PKI to achieve the "web of trust" functionality specified by soBGP. However, a web of trust model does allow relying parties (e.g., ISPs) to check certificates against the allocation hierarchy to detect errors or attacks, as described in section 7, forgoing some of the security offered by the proposed PKI.

References

1. Y. Rekhter, T. Li, “A Border Gateway Protocol 4 (BGP-4), “ RFC 4271 (2006)
2. S. Murphy, “BGP Security Vulnerabilities Analysis,” RFC 4272, (2006)
3. S. Kent, C. Lynn, K. Seo, "Secure Border Gateway Protocol (S-BGP)", IEEE Journal on Selected Areas in Communications, Vol. 18, No. 4 (2000) 582-592
4. G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin, “Working Around BGP: An Incremental Approach to Improving Security and Accuracy for Interdomain Routing,” Network and Distributed System Security Symposium (2003) 75-85
5. Yih-Chun Hu, A. Perrig, and D. Johnson, “Efficient Security Mechanisms for Routing Protocols,” Network and Distributed System Security Symposium, (2003) 57-73
6. T. Wan, E. Kranakis, P.C. van Oorschot, “Pretty Secure BGP (psBGP),” Network and Distributed System Security Symposium (2005)
7. S. Kent, "Securing BGP: S-BGP ", The Internet Protocol Journal, Vol. 6 -3 (2003) 2-14
8. R. White, “Securing BGP: soBGP,” The Internet Protocol Journal, Vol. 6 - 3 (2003) 15-22
9. K. Seo, C. Lynn, S. Kent, "Public-Key Infrastructure for the Secure Border Gateway Protocol (S-BGP)", DARPA Information Survivability Conference and Exposition (2001)
10. “Recommendation for Key Management - Part 1: General,” NIST Special Publication 800-57 (2005)
11. J. Arkko, et al., “SEcure Neighbor Discovery (SEND),” RFC 3971, (2005)
12. S. Chokhani, W. Ford, R. Sabett, C. Merrill, S. Wu, “Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework”, RFC 3647 (2003)
13. R. Housley, W. Polk, W. Ford, D. Solo, ”Internet X.509 Public Key Infrastructure -- Certificate and Certificate Revocation List (CRL) Profile”, RFC 3280 (2002)
14. C. Lynn, S. Kent, K. Seo, “X.509 Extensions for IP Addresses and AS Identifiers”, RFC 3779 (2004)
15. FIPS Federal Information Processing Standards Publication 140-2 (FIPS PUB 140-2), “Security Requirements for Cryptographic Modules”, Information Technology Laboratory, National Institute of Standards and Technology (2001)
16. J. Hodges, R. Morgan, “Lightweight Directory Access Protocol (v3): Technical Specification,” RFC 3377 (2002)
17. M. Zhao, S. Smith, D. Nicol, “The Performance Impact of BGP Security,” IEEE Network Vol. 19 - 5, 42-48 (2005)

Fighting E-Mail Abuses: The EMPE Approach

Massimiliano Pala and Antonio Lioy

Politecnico di Torino
Dip. di Automatica e Informatica
Torino, Italy
{massimiliano.pala, lioy}@polito.it

Abstract. Electronic mail is one of the most used and abused service in today communication. While many efforts have been made to fight e-mail abuses, no effective solution has yet been developed. Furthermore new technologies (e.g. wireless roaming) and new user needs (e.g. mobility) completely break the existing e-mail authentication techniques based on network topology. In this paper we present the E-Mail Policy Enforcer system (EMPE) which provides a method to cryptographically bind the identity of the original sender of an e-mail to the message body by combining digital signatures and transport level authentication data.

1 Introduction

Electronic mail is used by millions of people for work, personal contact, or simply for any other activity that requires fast communication. Due to the importance it has acquired in business it is considered a critical and inestimable service by enterprises and professionals. Unfortunately it is also one of the most abused Internet services. Perhaps no problem plagues the Internet as deeply as that of unsolicited junk e-mail, or spam. The word spam comes from an old Monty Python skit [1] where some people are unable to have a conversation because of a noisy "Spam" song coming from the nearby table. The term became connected with computers in 1985 [2] when somebody annoyingly typed the word "spam" on a MUSH (Multi-User Shared Hallucination role playing game) on all the connected users' terminals.

Perhaps the first traced spam took place in May 1988 and it is named the "JJ incident" [3]. An even earlier example known as "the dinette set heard 'round the World" [4] consisted in only two posts sent to *net.general* which is seen everywhere.

Today the e-mail system is subject to a variety of abuses, this includes not only spam, but also viruses and worms. According to RFC-2505 [5], spam is the mass sending of unsolicited e-mail. However, it is not easy to establish what "unsolicited e-mail" is. In fact some e-mail addresses are meant to be public, e.g. addresses used to provide products support or help desk services, addresses of professionals or Public Administrations, and all of them should be able to receive mail from everyone.

Many attempts have been made to stop spammers by law. Many countries have laws that prohibit or regulate bulk sending of e-mail messages, but still

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 130–144, 2006.

no positive results have been achieved. In most cases laws address specifically commercial advertisements. To apply any existing law, however, it should be possible to trace back the real sender of the message. Unfortunately the biggest flaw in today e-mail system is that messages do not retain strong authentication information from the sending Mail Transfer Agent (MTA) to the receiving one. Consequently the electronic envelope is subject to the same problems as in traditional mail where the receiving post office has no means to verify if the sender printed on the envelope is real or forged. This problem is exploited by spammers to disguise their identities.

Another recently used technique, that is becoming ever more popular among spammers, is the so called *prefix hijacking* attack. By using the lack of security in the Internet routing protocol [6], spammers are able to impersonate whole sets of unallocated IP addresses as originating points when sending spam. In fact the Internet routing infrastructure is actually subject to different type of attacks (e.g. blackholling, redirection, subversion) and current counter measures are either generally ineffective (route filtering) or too heavyweight to deploy (S-BGP [7,8]). After sending unauthorised e-mails, then, attackers disappear by restoring original routes.

An important aspect to be analysed is the economics of e-mail abuses. Internet subscribers world-wide are unwittingly paying an estimated 10 billion euro a year in connection costs just to receive “junk” e-mails, according to a study undertaken for the European Commission in 2001 [9]. The high volume of messages exchanged because of spam and viruses raises costs for every subject involved in the e-mail delivery process, by requiring additional CPU power, disk storage and network bandwidth. Moreover this situation pushes users and enterprises to buy additional services and software, e.g. anti-spam or anti-virus products. The e-mail abuse has therefore opened a new market which is still growing.

The solution presented in this paper, i.e. *E-Mail Policy Enforcer* (EMPE), addresses the e-mail abuse problem by providing a tool to enforce strong authentication on message contents. The rest of the paper is organised as follows. In section 2 and 3 we analyse existing solutions and proposed standards. In section 4 we detail the EMPE architecture, while in section 5 we discuss issues related to EMPE deployment. Section 6 describes possible enhancements to our solution and future work.

2 Present Solutions

The Simple Mail Transfer Protocol (SMTP) [10] and the Internet Message Format [11], which represent the underlying communication standards for the e-mail system, were designed for open communication, and consequently authentication was not a priority during their design. Moreover these standards allow to configure a server to relay mail (i.e. by acting as an SMTP client to the next SMTP server after having accepted an e-mail from a user or another MTA), thus making it difficult to identify and track the original sender of a message. Therefore malicious users could forge the message contents with false or dangerous data.

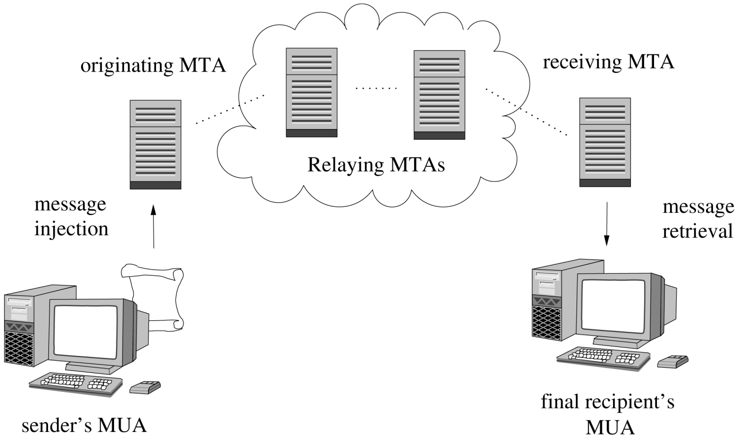


Fig. 1. E-Mail lifecycle, from its injection into the e-mail system to its retrieval by the final recipient

The lack of authentication has led to the possibility to perform attacks such as identity spoofing in which one user illegitimately assumes the identity of another user. Fig. 1 depicts the common life cycle of an email. It is possible to address the abuse problem at different stages of the e-mail message life cycle:(a) at message injection into the e-mail system, (b) at the receiving Post Office, or (c) at the user's mail agent (MUA).

The rest of this section focuses on the different solutions that have been designed at each stage to address the abuse problem when a new mail is injected into the system.

2.1 Message Injection

A first approach is to require and check authentication, i.e. to check if a message comes from an authorised user (or address). A solution which faces authentication is POP-before-SMTP. It is assumed that, by being able to authenticate herself for e-mail downloading, the user may be enabled to send messages as well. In POP-before-SMTP, as the first step, the user is required to download or at least to check her e-mails via the POP [12] protocol. Because the POP protocol requires the user to authenticate herself to the server, the MTA, upon success in POP access, assumes the user is "legitimate" and enables e-mail sending from the user's IP address within a small time frame.

The problem with this solution is that what it is really authenticated is the user's IP address, not the user herself. This allows an attacker to use spoofing techniques to send unauthenticated messages during the interval when the IP of the victim is enabled.

The AUTH [13] extension to SMTP has been introduced to authenticate the sender of an e-mail. It provides a way for one MTA to assert to another MTA that the former authenticated the sender. When the MTA successfully authenticates

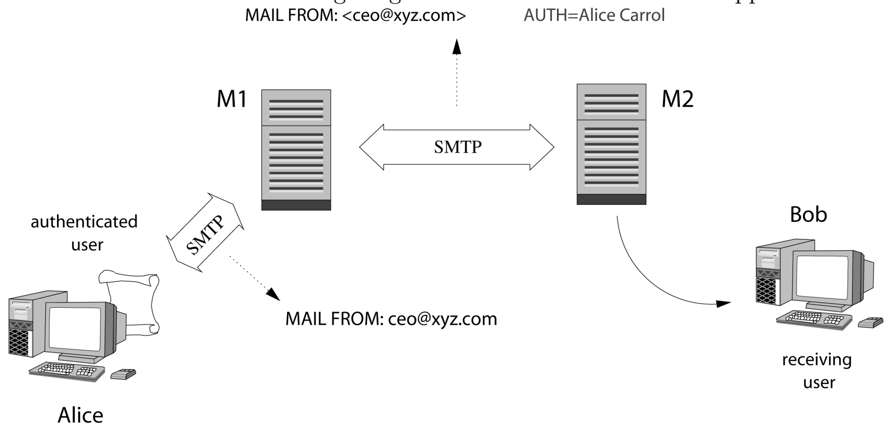


Fig. 2. SMTP AUTH extension example

the sender, it adds the "`AUTH=...`" keyword to the `MAIL FROM` command. In Fig. 2 Alice authenticates herself to M1 and sends a message. The originating server (M1) puts the authentication results in the AUTH content before sending the message to the receiving one (M2).

In our example Bob's receiving MTA has no way to know if the message was truly produced by Alice. In fact there is no way for M2 to automatically check the AUTH content in order to establish (a) if it has been altered/faked (as it is not cryptographically protected), (b) who has been actually authenticated by using the reported result string, and (c) if the originating MTA has been accurate in its assertion.

If M1 and M2 share a secure channel (SSL/TLS [14] or IPsec [15]), and some sort of trust agreement is present among them, then this authentication information could be propagated safely to the receiving server. This would solve point (a), but it would not address (b) and (c). Although it is possible to setup a secure channel between MTAs by using the STARTTLS [16] SMTP extension (which forces communication through a TLS channel). This merely offers protection against third parties, and definitely not against untrusted or misbehaving MTAs.

By using digital certificates to setup the communication channel between MTA and Mail User Agent (MUA) it could be possible to strongly authenticate the sender. Although most MTA's software support TLS client authentication, available MUAs do not, yet.

2.2 Receiving Post Office

Filtering messages on the receiving post office is one of the main anti-abuse current approaches. By mean of several techniques such as content-matching rules, content signatures or Bayesian filters, both clients and servers are capable to block suspect messages. The decision on what messages are to be filtered out is taken by analysing message headers, body and attachments.

This solution applies server-side filters to identify the spam before it gets to the user's mail box. Signature based filters work by comparing each incoming message to known spams. This works by generating a signature (e.g. by assigning a number to each character, then adding up all the numbers) and comparing it with known spam signatures. This kind of filters is not guaranteed to avoid false positives and, moreover, it suffers from the addition of random text to each new copy of a spam to generate a distict signature that easily fools the filter. This is the reason why sometimes random junk appears in the subject of a spam: it is there to trick signature-based filters. As soon as the filter developers figure out how to ignore one kind of random insertion, spammers switch to another one. This is why signature-based filters have never achieved very good performances.

Bayesian filters [17] are the latest in spam filtering technology. They recognise spam by looking at the tokens (e.g. punctuation and key words) they contain. A Bayesian filter starts with two collections of mail, one of spam and one of legitimate mail. For every word in these emails, it calculates a spam probability based on the proportion of spam occurrences.

One important issue of these techniques is that the user is not able to verify if the filter is actually filtering out legitimate e-mails. Because of the fallibility of this automated process, the message is usually only marked as spam (e.g. by modifying its subject) but it is not deleted from the user's mailbox.

2.3 Mail User Agent

A popular anti-spam technique is to run software on the user's computer to help her to filter out the spam. This allows the user to check if the filter is blocking true spam rather then important messages. Downside is that the user still have to download all of her e-mail, i.e. good *and* junk.

It is at MUA level that the usage of Bayesian filters provides best performances. This is true because spam has different characteristics from user to user. For example in a user's personal filter's database one word could have a spam probability of 97% because it mostly appears in what the user thinks to be junk while another one could have a spam probability of 48% because it occurs almost equally in spam and legitimate mail. When a new mail arrives, the filter collects the 15 or 20 words whose spam probabilities are furthest from a neutral 50% (in either direction), and calculates an overall probability that the mail is spam.

Bayesian filters are extremely accurate, and adapt automatically as spam evolves. They learn to distinguish spam from legitimate mail by looking at the actual mail sent to each user.

Bayesian filters, when trained by experienced users, have filtering rates that can reach up to 99% of spam and are particularly good at avoiding "false positives", i.e. legitimate mail misclassified as spam. The disadvantage of Bayesian filters is that they *need* to be trained: the user has to tell them whenever they misclassified a mail, therefore they might not be suitable for unexperienced users.

3 Proposals to Fight Spam at Transport Level

The IRTF Anti-Spam Research Group [18] is studying and evaluating different solutions to the message sender forgery problem. Three major contributions are currently being evaluated: the Sender Policy Framework (SPF) [19], the Designated Mailers Protocol (DMP) [20] and the Reverse Mail Exchange (RME) [21].

The basic idea behind SPF is to identify the servers allowed to send messages from a domain. It defines a method to publish domain policies in one DNS record. Upon receiving a message, the receiving MTA checks if the originating server is allowed to send e-mails from the exposed domain by checking informations published in the DNS. The allowed hosts may be specified in many ways. It is possible to indicate a single node, a whole domain, a single IP address or a whole network. While it is still possible to use TXT records to minimise the impact on existing systems, a new record type (SPF) is defined within the standard proposal. It is also possible to extend the SPF records content as new authentication methods are being developed. For example if the e-mail from a domain is supposed to use S/MIME [22] format, then it is possible to include that information by specifying the "smime" keyword within SPF records.

DMP uses TXT records to publish authorisation data into DNS. These records, published in a dedicated subdomain called "_smtp−client", associate an IP address or a group of IP addresses to the "allow" or "deny" directive. A single DNS query allows MTAs to retrieve the DMP records needed to authorise the sending MTA. If the retrieved record contains the "*allow*" directive, then the authentication process is successful and the receiving MTA accepts the incoming mail, otherwise if the retrieved record contains the "*deny*" directive, then the authorisation fails. If conflicting records (i.e. "*allow*" and "*deny*") are retrieved then the client has no valid DMP records in the DNS and the message should be rejected. This protocol also provides the possibility to completely bypass the DMP records checking if another authentication method is used, like the SMTP AUTH or if the receiving MTA's network address is in an "*allow relay*" list.

Focus of the RME work, instead, is to block a user to send a message using a sender address which he was not authorised to use. This proposal aims to provide a method to inform the recipient of a message that the sender is, eventually, not authorised to use that address. RME has two phases. The first phase is the retrieval of the IP address of the sending MTA from the TCP/IP connection, RME calls this step authentication. The second step consists in verifying if the `MAIL FROM` value is allowed to be used from the "*authenticated*" IP in the previous step. RME relies on the DNS to retrieve the authorisation information stored in RMX records that specify the host or networks allowed to send e-mails from a certain domain. The IP address of the originator and the envelope sender address are checked. By querying the authorisation records stored in the DNS, the MTA will establish if that particular sender is allowed to use that sender address. A message header is then added indicating the success of the authentication phase.

All these three approaches are very similar to each other in that all of them use DNS to store authorisation/authentication records. Still unresolved issues are present in these proposals that basically stem from the fact that DNS responses are unreliable. In fact poisoning attacks are possible and the lack of diffusion of DNSSEC [23] limits the trust it is possible to put in the DNS responses. Moreover these protocols are essentially based on network topology. The authentication they propose relies exclusively upon IP addresses. In fact MTAs provide their services to an IP, not to a user identity. Anyway, user mobility, that implies frequent address changes, introduces new issues. Another common problem is the possibility for an attacker to publish records in the DNS indicating the entire Internet is allowed to send e-mail for the domain. It is true that it is easy to ban a doman from sending junk e-mails, but it is at the same time true that new domain registrations are cheap enough for spammers to move from one domain to another and start mass mailing again. These protocols do not use any cryptographic techniques, and thus do not provide strong authentication. Moreover the creation and maintenance of DNS records that purport to identify authorised senders might be a non-trivial operational task.

Interesting work that uses public key cryptography has been carried out by Yahoo! Inc. in its DomainKeys [24] proposal. Recently IETF created a new working group, namely Domain Keys Identified Mail (DKIM) [25], that addresses the responsability of domains for having taken part in the transmission of an email message. The DKIM has recently publish a new Internet draft [26]. That envisages adding an header field that bears a digital signature of the entire message (headers and body). This signature is generated by the originating MTA when an authorised end-user within the domain sends a message. Upon receiving a message, the DomainKeys enabled MTA has to check the digital signature. Depending on the signature verification results the receiving email system applies local policies. All the signature related operations are handled by the involved MTAs. Thus no schanges or additional features are required on the MUAs. Revocation of signing keys is performed by deleting them from the DNS.

To verify the signature, the public-key of the originating servers must be made publicly available. Again this information is stored in DNS records by the domain owner. The corresponding private key is then shared between all the authorised outbound MTAs. Although the DomainKeys approach requires only small changes into existing MTAs software and it does not modify the SMTP protocol, the problem here is, once more, the unreliability of DNS responses. In table 1 a summary of the characteristics of the different approaches is reported.

Table 1. Existing solutions summary

Protocol	DNS records	Record Type	Crypto	Auth. Type	New Header	MTA
SPF	1	SPF/TXT	NO	IP-based	NO	Modified
DMP	≥ 1	TXT	NO	IP-based	NO	Modified
RME	≥ 1	RMX/TXT	NO	IP-based	YES	Modified
D-Keys	≥ 1	TXT	YES	RSA	YES	Modified

4 EMPE

The analysis of current solutions and proposals shows that an approach that provides a good tradeoff between ease of deployment and use of cryptographic techniques is the best solution today possible. The basic idea behind our proposed solution (EMPE) is to provide a method to check each message processed by the *originating MTA* to guarantee that:

- the sender is fully and uniquely identified (at least for mail originating from the local domains)
- the authentication informations obtained by the analysis of the communication channel and the message contents are securely sealed in the message body

In order to achieve these goals EMPE combines two different technologies: SMTP AUTH and S/MIME [22] features. EMPE analyses all the provided credentials, i.e. user credentials and SMTP envelope contents, and checks them to be congruent. This extends current systens where authentication at connection level is not checked against the SMTP envelope and the message headers. Then, if not already signed, the message is cryptographically signed by the system to securely bind authentication to the message body.

By implementing the described EMPE framework, then, only cryptographically authenticated messages are injected into the e-mail system.

4.1 Assumptions

To add real value to the e-mail system, a source of authentication data is needed. Therefore our work focuses its attention on messages that (a) have been received by the MTA on an authenticated channel (SSL/TLS + AUTH) or (b) bear authentication information within the message body (i.e. S/MIME signed).

It is important to point out that only one of the two authentication sources is needed as EMPE, by being capable of digitally signing messages on user's behalf, securely binds user's credentials to the message body. Another important consideration to be made is that our work is mainly focused on preventing e-mail abuses on *outgoing* messages, whilst current proposals focus on incoming messages; further study, as reported in sections 5 and 6, is needed to consider EMPE also for inbound e-mail traffic.

4.2 Design Considerations

One important requirement of our work was to provide a solution that would minimise the impact of its adoption on existing e-mail systems. To achieve our goal we analysed two important factors: where and how to operate changes. The chance to operate at client side has been discarded as this would require the development of different software depending on the operating system and the preferred MUA used, it would impact on the users habits in e-mail system usage, and it would prevent large-scale deployment in big environments.

Because the domain MTAs are directly controlled by the administrators, the deployment of EMPE directly on the domain MTAs is the most efficient solution: the number of MTAs (when compared to the number of e-mail clients) and the number of different installed OS (on MTAs) is quite small (and usually they are all aligned for ease of management).

The following section describes the EMPE system architecture at a suitable level of abstraction as to better explain its integration with existing MTAs. In section 4.4 the EMPE work flow is explained while the description of the implemented EMPE system is reported in section 4.5.

4.3 The EMPE Architecture

The EMPE system is formed by two main components: the EMPE core and the EMPE interface. The EMPE core is the main component of the proposed system. It handles all cryptographic operations over processed messages. Moreover it applies the defined security policies and decides if the message is to be accepted or not. As shown in Fig. 3, it is possible to break down the EMPE core into three different logical blocks. The first block is the Cryptographic Engine. This block includes all the needed functionalities to handle the S/MIME format and to validate (and apply) digital signatures. The second block, the Policy Enforcement Sub-System, is responsible for the enforcement of the configured security policies. Thanks to this sub system it is possible to configure a generic purpose policy which benefits from the enforced authentication information on the message. Actually only security oriented policies have been considered as reported in section 4.5, although the study and the definition of more generic policies is being analysed. The third and last logical block of the EMPE core is called E2I (EMPE to Interface). It takes care of message exchanging between the EMPE core and the EMPE interface.

The EMPE interface manages the communication between the EMPE system and the external environment. It is logically divided into two functional blocks: the interface implementation to the external environment – referred as I2X – and the interface to the EMPE core component – referred a I2E. The I2X handles all the communication with the external environment. This block, then, can be implemented as an SMTP interface for communication to the MTA, or it can be more tightly integrated into the system (e.g. implemented as a plug-in for the specific used MTA). The second block (I2E) is used to exchange messages with

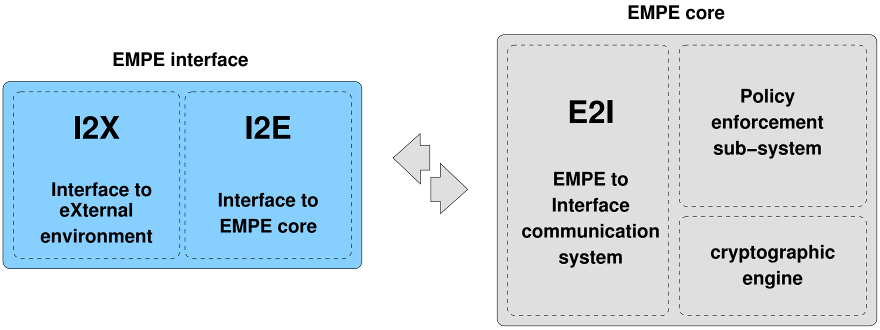

EMPE interface
I2X
Interface to eXternal environment
I2E
Interface to EMPE core
EMPE core
E2I
EMPE to Interface communication system
Policy enforcement sub–system
cryptographic engine

Fig. 3. The EMPE Core and EMPE Interface logical architecture

the EMPE core. No cryptographic operations or decisional steps are performed at the EMPE interface. On MTAs that provide a plugin interface, an advantage of this architecture is that no major changes are required on the mail server. Infact, as shown in section 4.5 the addition of EMPE does not require changes to already deployed servers but the adding of an interface to the EMPE system.

4.4 EMPE Workflow

Upon receiving a message originating from the local domain (i.e. from a local user), the MTA activates the I2X component of the EMPE interface. Then the EMPE interface sets up the communication channel with the EMPE core.

Communication between EMPE interface and EMPE core takes place in two different rounds. During the first phase the interface sends the SMTP envelope, the message headers and all the SMTP AUTH credentials – if any – to the EMPE core which analyses them and applies defined policies over the received data. If the first phase successfully completes, the interface starts the phase two and sends the entire message body to the core. This two-phases approach helps

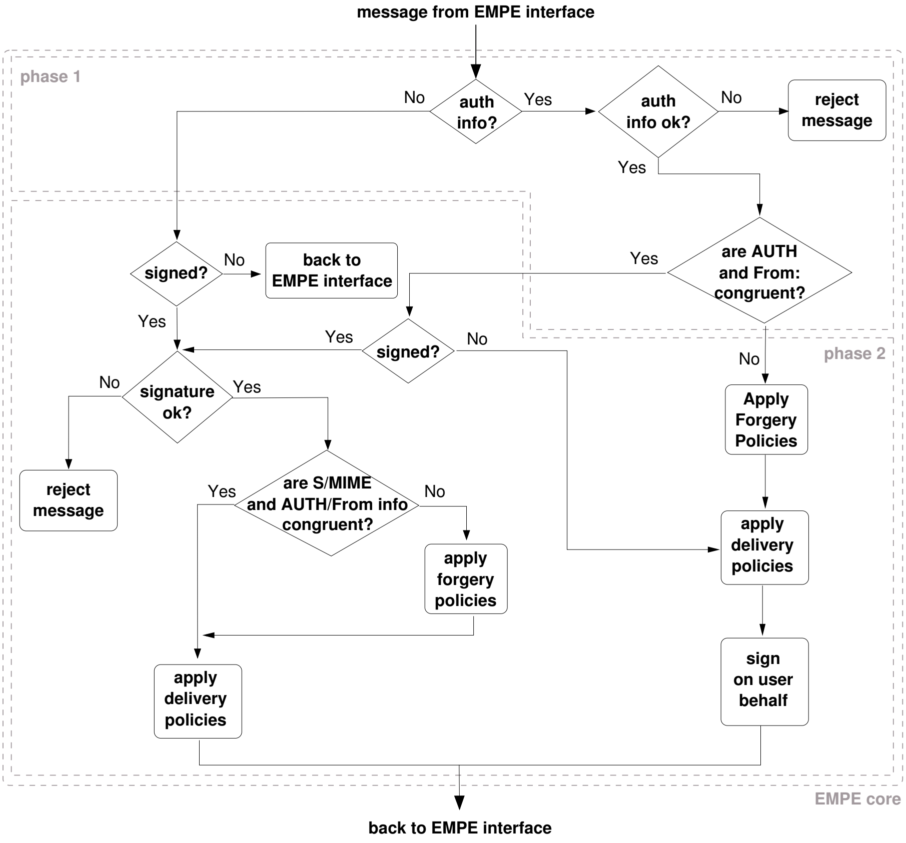


Fig. 4. EMPE core component work flow

in minimising the impact of very long messages (e.g. carrying big attachments) on the EMPE system. By adding a little communication overhead it is possible to save the message body transmission from the MTA to the EMPE core in case of failure during phase one. In fact if an error occurs during this phase, it is useless to send the whole message body to the EMPE core. The core work flow is shown in Fig. 4.

Phase One. In phase one EMPE checks if the user has presented any authentication credential to the MTA. These information are used to verify that no identity spoofing has been attempted by comparing them with the SMTP "`MAIL FROM:`" command and the "`From:`" header contents of the message. This step prevents the sender to use credentials which are different from the ones used when authenticating the connection or signing the message. Special tables are used to match multiple e-mail addresses to a single user. Moreover the same mechanism can be used for mailing lists by matching specific headers in the message.

If this congruency test fails or provided data are in contrast with the configured policies, then the EMPE core sends back the related error code to the EMPE Interface; otherwise the EMPE core sends back the proper result code to the EMPE Interface and enters phase two.

Phase Two. In phase two the whole message body is sent from the EMPE interface to the EMPE core. At this stage the Cryptographic Engine is used. If the message is digitally signed, EMPE proceeds by checking the S/MIME signature. In case of signature verification errors the message is rejected, otherwise the validity of the signer's certificate is checked.

If the signature is valid, and authentication data are provided, then EMPE checks them to be congruent. If this check succeeds, delivery policies are applied by the policy enforcement sub-system. Differently in case of failure, forgery policies are applied before the delivery ones.

If the message is not signed, EMPE signs the message on user behalf by using the server's key and encapsulating the original message into a S/MIME bag. This step is crucial as it preserves the original user's message and securely binds the identity of the sender to the message.

By using the EMPE interface, the message is sent back to the MTA and accepted for delivery. The control over the message, then, returns to the original MTA.

4.5 Our EMPE Implementation

After defining the general architecture of the EMPE system, we developed it for one of the most used MTA software available, i.e. Sendmail [27].

We developed two software components: the EMPE core – which implements the cryptographic engine, the E2I and the policy enforcement subsystem – as a standalone server listening on a TCP port, and the EMPE Interface – which implements the I2X and I2E logical blocks – as a plug-in for Sendmail by using the milter interface [28].

Communication between EMPE core and EMPE interface takes place by using TCP sockets. When activated, the sendmail plug-in connects to the EMPE core via socket and acts as the actual interface to the MTA. Indeed by using socket communication channels, the proposed solution envisages the possibility to have a single EMPE core serving more MTAs.

This implementation presents several advantages. First of all, by having a centralised EMPE core it is easier to manage policies for all the MTAs of a domain instead of having different configurations sparse on several servers. In addition, a serious burden on existing mail servers (which were not designed to handle such a load) would have to be managed because all cryptographic tasks are handled by the EMPE core. To avoid performance issues, the EMPE core component may be installed, if needed, on a dedicated server. This lets the EMPE burden not to additionally load the domain MTAs.

Moreover to better satisfy performance requirements, a cryptographic hardware accelerator may be used with the EMPE core component. The chosen design allows to share the cryptographic hardware accelerator among different MTAs. This results in a money saving for the organisation as this accelerators are usually quite expensive. We achieve hardware support by using OpenSSL [29] ENGINE extension. In fact our EMPE implementation may be used together with several cryptographic hardware accelerators out of the box[1].

The EMPE Interface implementation required special considerations during its design and actual implementation. The first and easiest considered approach was the development of an SMTP interface. In this case the server would forward the message to the EMPE System and receive back the results by using SMTP, thus no specific code for each MTA's software would be needed (i.e. any available software would use the same interface). Unfortunately the EMPE core needs to access all authentication data from the MTA server, this includes not only the message body or envelope, but authentication details as well (e.g. used authentication methods, presented digital certificates or connection derived data). Because some of these required information are derived from the client-server connection, it would not be possible to get them by using an I2X component based on SMTP. This design was then dropped in favour of MTAs specific plug-in adoption. In fact many MTA's software provide the possibility to write extensions to their functionalities by using plug-ins. Although each MTA software uses different methods to communicate with its specific EMPE plug-in component, each implemented plug-in can be made to export the same interface to communicate with the EMPE core. Therefore all MTA's dependent error codes and interface details can be directly handled by the EMPE plug-in (without requiring modifications to the EMPE core).

By including all the EMPE main code within the EMPE core component, the development of new plug-ins for different MTAs requires very few efforts. Actually EMPE plug-in components for QMail are under development,while Postfix extension is being considered as possible future work.

[1] Several hardware modules are directly supported by OpenSSL (CryptoSwift, nCipher, Nuron, UBSEC, Aep, Ibmca, SureWare, IBM 4758 CCA) while many vendors now provide their own extension for OpenSSL (e.g. ERACOM, Chrysalis, etc...).

5 EMPE Deployment Considerations

Deploying EMPE raises several issues to be considered.

Digital certificates and public-key technologies are used. One important aspect of PKI is the certificate valiadation and signature verify. This involves the downloading of CRLs or the usage of OCSP to retrieve the revocation status of a certificate. The validation process could add an high burden to the EMPE core as these operations can take up to several seconds per message if data is not locally available. It is to be considered, anyway, that EMPE is mostly applicable on outbound e-mail traffic, therefore it is reasonable that the number of digital certificates (in case users directly sign their messages) and the number of different PKIs is quite small. Caching mechanisms and pre-fetching of revocation data may be used to address this issue.

EMPE is mainly focused on outgoing messages. To apply EMPE policies on incoming messages as well some considerations are needed. As incoming messages are sent from other MTAs, authentication informations may be missing from the message (e.g. it may not be signed), however if the connecting MTA does provide credentials (e.g. by using SSL/TLS channel together with client authentication) system administrators could accept or deny messages by analysing those instead of checking only message contents.

EMPE needs to map authentication credentials to e-mail accounts. Some server configurations may need to be modified to provide a mapping system to support EMPE, e.g. in case a user is allowed to send e-mail from different e-mail accounts by using one single system account.

EMPE is most suitable in controlled domains and closed environments. Special considerations should be made for inter-domain environments and public networks as such a tool could violate the users' privacy or hurt particular ethic believes. For example, by being able to check S/MIME messages, the system is capable to prevent users from sending encrypted messages (or, if applied also on incoming e-mail, to receive such messages).

6 Conclusions and Future Work

In this work the most important issues which today affect the e-mail systems have been presented. Available solutions to the e-mail spoofing techniques and the spamming plague have also been analysed by discussing existing problems.

The solution introduced in this paper is presented by exposing some interesting scenarios where EMPE adds real value to the e-mail system. By using PKIs and cryptographic techniques, EMPE is capable of shifting connection oriented authentication information to the message contents. Because of the usage of cryptographic techniques it is possible to apply policies based on the sender's identity.

Future work will be oriented into two different directions. The first one is the study of EMPE integration with other available solutions. In fact, by using an EMPE-enabled mail system it is possible to completely stop spam originating from the controlled domain. However our work does not address different problems like *phishin* where spammers forge authorship of a message trying to get important data from the users or *joe-jobbing* where forged envelope senders or return-path cause messages with malicious contents to bounce to innocent users' mailboxes. Publishing authenticated domain informations may prevent these attacks to be effective. For example in SPF records it is possible to specify that a specific domain allows only S/MIME messages for outbound traffic: messages not signed may be automatically discarded.

The research work will then be be aimed toward the study of inter operable techniques to securely publish domain capabilities information. Usage of Trusted Computing [30] techniques will also be considered to establish trust relationships between the different actors involved in the e-mail management process. By combining remote attestation and attribute certificates [31], e-mail submission policy and authentication credentials may be securely propagated to the final recipient.

References

1. "Monty python's flying circus: final sketch of the 25th show." [Online]. Available: http://en.wikipedia.org/wiki/Spam_%28Monty_Python%29
2. M. Bilca, J. Lo, F. Kerrest, and D. Wytock, "The Ethics of SPAM." [Online]. Available: http://cse.stanford.edu/classes/cs201/projects-97-98/spam/
3. P. Linden, "Re: first case of spam." [Online]. Available: http://www.rahul.net/falk/jjspam.txt
4. net.general, "the dinette set heard 'round the world'." [Online]. Available: http://groups.google.com/groups?selm=3375%40drutx.UUCP
5. G. Lindberg, "Anti-Spam Recommendations for SMTP MTAs," RFC-2505, February 1999.
6. Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP 4)," RFC-4271, January 2006.
7. S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol (S-BGP)," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 582–592, 2000.
8. S. Kent, C. Lynn, J. Mikkelson, and K. Seo, "Secure Border Gateway Protocol (S-BGP) — Real World Performance and Deployment Issues," in *Proceedings of Network and Distributed Systems Security 2000.* Internet Society, February 2000.
9. S. Gauthronet and E. Drouard, "Unsolicited Commercial Communications and Data Protection," January 2001. [Online]. Available: http://europa.eu.int/comm/justice_home/fsj/privacy/studies/spam_en.htm
10. J. Klensin, "Simple Mail Transfer Protocol," RFC-2821, April 2001.
11. P. Resnick, "Internet Message Format," RFC-2822, April 2001.
12. J. Myers and M. Rose, "Post office protocol," RFC-1939, May 1996.
13. J. Mayers, "SMTP Service Extension for Authentication," RFC-2554, March 1999.
14. T. Dierks and C. Allen, "The TLS Protocol," RFC-2246, January 1999.
15. S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC-2401, November 1998.

16. P. Hoffman, "SMTP Service Extension for Secure SMTP over TLS," RFC-2487, January 1999.
17. M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian Approach to Filtering Junk E-Mail," in *Learning for Text Categorization: Papers from the 1998 Workshop*, July 1998.
18. "Anti-Spam Research Group HomePage." [Online]. Available: http://asrg.sp.am/
19. M. Lentczner and M. W. Wong, "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-MAIL," Internet draft, June 2005.
20. G. Fecyk, "Designated Mailers Protocol," Internet draft, May 2004.
21. H. Danisch, "The RMX DNS RR and method for lightweight SMTP sender authorization," Internet draft, May 2004.
22. B. Ramsdell, "Secure/Multipurpose Interet Mail Extensions (S/MIME) Version 3.1 Message Specification," RFC-3851, July 2004.
23. D. Eastlake, "Domain Name System Security Extensions," RFC-2535, March 1999.
24. M. Delany, "Domain–based Email Authentication Using Public–Keys Advertised in the DNS (DomainKeys)," Internet draft, September 2005.
25. "Domain Keys Identified Mail Working Group (DKIM)." [Online]. Available: http://www.ietf.org/html.charters/dkim-charter.html
26. E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, and M. Thomas, "DomainKeys Identified Mail Signatures (DKIM)," Internet draft, February 2006.
27. "Sendmail HomePage." [Online]. Available: http://www.sendmail.org
28. "Milter Community HomePage." [Online]. Available: http://www.milter.org
29. "OpenSSL Project HomePage." [Online]. Available: http://www.openssl.org
30. "Trusted Computing Working Group Homepage." [Online]. Available: https://www.trustedcomputinggroup.org
31. R. S.Farrel, "An Internet Attribute Certificate Profile for Authorization," RFC-3281, April 2002.

DomainKeys Identified Mail Demonstrates Good Reasons to Re-invent the Wheel

Stephen Farrell

Distributed Systems Group,
Department of Computer Science,
Trinity College, Dublin 2, Ireland
stephen.farrell@cs.tcd.ie
https://www.cs.tcd.ie/Stephen.Farrell/

Abstract. DomainKeys Identified Mail is an anti-spam proposal that involves mail servers digitally signing outbound email and verifying signatures on inbound email. The scheme makes no use of existing public key infrastructure or email security standards. This paper provides an outline of the scheme and discusses some reasons why re-use of existing standards is inappropriate in this context.

1 Introduction

Domain Keys Identified Mail (DKIM) [1] is an anti-spam approach that involves digitally signed email. The most basic rationale for DKIM is that it allows for better whitelist management since the digital signatures allow a verifier to more reliably detect that a message has originated from some mail domain. Even if it did nothing else, DKIM might be justified on this basis - that it is a real improvement over whitelists based on mail server IP addresses.

Typically however DKIM signature checking would form a part of a broader set of anti-spam measures, so a valid signature does not directly result in delivery of the message, but may rather be used to “turn down” the level of subsequent checking for that message, thus saving resources and allowing those released resources to be dedicated to checking unsigned email. In this way it is hoped that DKIM will result is more reliable delivery of genuine messages as well as better detection of certain types of spam.

DKIM also involves a second and separate mechanism allowing a domain to express a policy about its outbound email. In particular this policy allows a domain to state that it actually sends no email at all, which would be appropriate for some banking server domains. This mechanism, when combined with the signature mechanism, is aimed at reducing the ability of bad actors to create emails that appear to originate from domains where such strict policies actually apply. The basic idea here is to make some current phishing techniques somewhat less attractive, though recognizing that DKIM cannot “solve” phishing, or, more generally, spam.

From the above, we can see that DKIM will require some way to distribute public keys for signature verification – basically a public key infrastructure (PKI) or equivalent. There are at least three standard ways to do this, using the X.509 based PKIX

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 145 – 153, 2006.

approach [2], the OpenPGP based approach [3] or the XKMS approach [4]. In fact DKIM uses none of these. Similarly, DKIM must use some signature format, and again there are some standards in this area, primarily S/MIME [5] and XML Signature [6]. And once more, DKIM doesn't make use of these.

In the remainder of the paper we give a brief outline of DKIM, then examine why DKIM doesn't use a PKI, followed by consideration of why DKIM doesn't use one of the standard signature formats and lastly we offer some tentative conclusions.

2 DKIM Outlined

As stated DKIM consists of two parts – the first is the basic signature scheme [7] and the second describes the Sender Signing Policy (SSP) [8]. There is also a threat analysis document [9] that provides some additional background in terms of the threats that DKIM is intended to counter and also in terms of the new threats which come into play when a system like DKIM has been deployed. Since all of these documents are currently in draft form, we won't consider them in too much detail – detail that is still subject to change – but will rather take a somewhat abstract view of DKIM.

DKIM signatures are carried in a mail header field (DKIM-signature), placed there by a mail server, often called a Message Transfer Agent (MTA), and mostly not by a Mail User Agent (MUA). Similarly, the general intent is that DKIM signature verification is carried out by an MTA and not by a MUA. DKIM is therefore primarily a server-server protocol unlike more traditional email security protocols. While there have been suggestions that DKIM-enabled MUAs might be useful, the current IETF activity is not addressing this so we will therefore ignore DKIM-enabled MUAs in the remainder of the paper.

A DKIM signature can cover the body of the message as well as a number of mail header fields, in particular the "From:" header will often be signed. The DKIM-signature header field indicates which other parts of the message were signed, as well as the signing algorithm and other signature parameters.

The general model is that the public key to verify the signature is stored in the DNS entry of the signing/originating domain (which can sometimes differ!). A verifier therefore has to do a new DNS lookup to retrieve that key as part of signature verification. However, MTAs commonly do such lookups at the moment, e.g. to verify that a sender is not on a DNS based blacklist. Once the public key is retrieved then the signature can be checked and the message passed for further processing.

One important aspect of DKIM signature processing is that badly signed messages are to be treated as if they were unsigned. Practically, this is necessary because so many MTAs actually modify[1] messages in transit that it will be quite common for signatures not to verify for totally innocuous reasons.[2] So, in contrast to many signature applications, signature verification failure doesn't necessarily lead to a message processing failure.

[1] At the moment how multiple signatures might be placed on a message, e.g. by mail list agents is not well defined. So we omit consideration of such issues for the present.

[2] Hopefully, DKIM will start a move to only make signature-friendly changes to messages, but that's for the future.

Of course a DKIM verifier has a potential problem if an unsigned message arrives since there is the possibility that a bad actor may have stripped a signature from a message (or equivalently created a new, unsigned message). Since bad signatures are the equivalent of no signature the message will be processed. This is where SSP enters the picture, since it allows the verifier to check whether the message should have been signed. In fact SSP also serves some other purposes, for example allowing a domain to declare that it, in fact, never originates any email at all – this policy would be appropriate for many domains that are currently the subject of attempted phishing attacks. SSP policies are also published in the DNS.

SSP also handles another potential problem – handling 3rd party signatures. This is where the signer is not an MTA in the "From:" domain. There are some valid reasons why this can happen, for example, business email sent via a home ISP's server. An ISP who signs outbound email in this case might publish a policy (for the signing domain) saying that it signs everything, but that much of the mail it sends is not from its own domain. Similarly, the business (the domain matching the "From:") might publish a policy saying that some, but not all of its email is signed and that 3rd party signatures are acceptable. There are clearly some currently fairly common email practices that could be affected were domains to adopt some SSP policies – for example the above scenario might be prevented if the business publish a policy that all mail "From:" their domain must be 1st party signed. Note also, that the DKIM verifier currently only consults the business' policy, the ISPs policy is not consulted, even though it will presumably exist, given that the ISP signed the email.

SSP is a less mature proposal than the base DKIM and is more likely to change as the standards process proceeds. Partly this is due to the fact that policy assertions are inherently more complex than public key assertions – for evidence of that the reader is invited to compare X.509 based PKI against X.509 PMI [10] or XKMS [11] against SAML [12]. Partly this is perhaps also due to the fact that while we do know how to DKIM-sign and verify, we do not yet know the consequences this may have for email, and hence are not yet able to be authoritative about the scope of the policy language that SSP ought to allow.

3 Why Not PKI?

DKIM could have made use of a more standard PKI in which case there would seem to have been benefits in terms of the range of products that could be used to support DKIM, and the features they bring. For example, X.509 based PKIs [13] have very rich support for revocation related features and also for inclusion of policy information in certificates – such policy information could presumably be quite useful for reputation and/or accreditation services which are expected to be based on DKIM. Similarly, OpenPGP [14] based infrastructure might very easily allow interesting reputation services based on who signed which key. But DKIM gives up these seeming advantages, so let's examine reasons why this might be a good design choice.

The first and primary reason DKIM does not use standard PKI is that the overriding requirement for email is reachability. The ability of anyone to send email to anyone without the infrastructure imposing barriers is a fundamental tenet of email. The problem that arises is that at some point a verifier has to check an authority certificate or contact some key server.[3] There is no evidence that PKI key servers of the various types required can scale to the extent required for the email application. If such a key server became the bottleneck in mail processing, then DKIM would simply not be used – even if signatures were present they would not be checked for this reason, thus allowing insertion of messages with bogus signatures. In particular, if the verifier is in a large domain it's processing would be bounded by the signer's key server, and that signer might be in a small domain – the result is that the large domain runs at the speed of the slowest key server. Since DKIM uses the DNS for storing keys this is a non-issue as we already know how DNS performs and that it meets the mail application's performance requirements.

DKIM-with-PKI would also introduce a possibly large set of new failure-modes into mail processing in addition to the planned new signature-verifies-or-not addition that is the main feature of DKIM. Such additional failure modes introduce undesirable ambiguity ("the signature's ok, but I cannot contact the OCSP responder for the CA cert of the issuing CA") that might create either new opportunities to inject spam or else create new denial-of-service (DoS) vulnerabilities.

Were DKIM to use a PKI then one would imagine a design would involve inclusion of a certificate-like structure with the message. However, this assumes that the signer can select one of its potentially many certificates to include, which is in fact not really a tractable problem (at least without first undergoing a highly complex PKI profiling exercise). In fact, this problem is made even worse by the fact that a single message may have many recipients (and "bcc:" recipients too). On the receiver side, the signer-provided structure may in fact turn out to be misleading if a better path exists to the signer's public key, and detecting this is not a simple matter. The analogous problem with certificate based TLS client authentication is tackled (if not solved) by allowing the server to name some "trusted" CAs during the key negotiation. In email, we have no opportunity for such negotiation and so the certificate selection problem would remain. DKIM-as-is ensures that the signer-chosen key is as-available and as-ready-to-be-trusted as any other key.

So it makes sense that DKIM uses the DNS for storage of public keys or certificates or whatever structure is to be used. However, regardless of the structure used, DKIM has a dependency on the DNS minimally for availability but in fact also in that DNS poisoning [15] is a significant threat even to DKIM-with-PKI. To see this, consider that the reachability requirement, that overrides everything else, means that effectively the "trusted root" information has to also be present in the DNS – otherwise most signatures will not be verifiable at most recipients (or else we have to create yet another singly-rooted infrastructure), due to the lack of this information. And once the "trusted root" information is placed in DNS, then poisoning attacks allow spoofing.

If the alternative approach – to create a few new "worldwide DKIM roots" – were taken, then this is effectively replicating the work done by DNSSEC [16] and is in any case an approach that is not really consistent with the general thrust of PKIs.

Some PKIs do have highly sophisticated support for revocation. However the DNS itself, if considered a trusted database, can also support revocation via its caching mechanisms. In fact the DNS approach is far more efficient than would be the case with using an X.509 CRL, OCSP or XKMS based approach.

[3] LDAP certificate or CRL repositories, OCSP and/or XKMS responders, etc.

OpenPGP's revocation support however is arguably much better than simply deleting the DKM public key entry – since positive DNS caching has longer-lasting entries than negative DNS caches (i.e. you remember something you find longer than a "nothing there" answer). However, DKIM revocation calls for revoking keys not via DNS entry deletion, but rather by replacement with an "empty" entry so the DKIM scheme is as good as the OpenPGP one in terms of caching. In fact, some OpenPGP users may be reluctant to revoke keys since they are also relatively likely to be used to encrypt messages to that user and revocation removes that potential for future contact. DKIM keys should not be liable to this kind of overloading and so DKIM revocation has fewer side-effects than in the OpenPGP or PKIX cases.

The reasons for not using PKI with DKIM boil down to reachability and its consequences; scalability and the fact that the current DKIM proposal has equivalent functionality while being significantly simpler.

4 Why Not S/MIME Or XML Signature?

Were DKIM to use S/MIME [17] (actually CMS [18]) or XML signatures [19], then implementations would be able to use existing standard signature constructs which would appear to bring two major benefits – first there is plenty of source code out there that could be used when implementing DKIM and secondly, but more importantly, those constructs have undergone significant security review, so any problems with weaknesses or applicability issues are well understood. Again though, DKIM doesn't take this route so let's examine the reasoning behind this.

First though, we need to correct some potential confusion. DKIM signatures have to be carried in the mail headers and must not be carried in the mail body. This is, in fact the essential difference between DKIM and S/MIME – DKIM is basically an MTA-MTA protocol embodied in mail headers, whereas S/MIME is an MUA-MUA protocol that only involves changes to (and security of) the body of the message. Were DKIM to be recast as primarily an MTA-MUA protocol, or even an MUA-MUA protocol, then so many additional issues would be raised, that DKIM would effectively constitute a re-working of S/MIME at that stage.[4] There would also be a need to define some way to distinguish between "proper" MUA-MUA S/MIME signatures and "DKIM" MTA-MUA signatures and so all S/MIME aware MUA deployments would have to be updated in order to avoid this confusion (not a trivial task!).

There is also the important point that DKIM, as an MTA-MTA header protocol, does not interfere with the end-user experience. Changes to the body of the mail, would however inevitably do so – something that is correct for S/MIME but would be a major problem for DKIM. Having said that, one early proposal was apparently to use a MIME encapsulated signature in the message body, but to define an SMTP extension, such that whenever a mail signed in this way was to be forwarded, the forwarding MTA would ask the next MTA whether or not the next MTA was aware of this signing "trick". If the next MTA is so-aware, then the signed message is passed on. If not, the signature has to be stripped from the message before forwarding. In this fairly inelegant scheme, the SMTP extension is required in order to avoid the new

[4] To be fair, there do appear to be some who would like that work to be done, but any such work is outside the scope of the current IETF effort.

signature being accidentally seen by an MUA. Given the number of updates required this scheme was not pursued.

In any case, our real point of comparison is actually the use of CMS SignedData or an XML Signature element as the content of the "DKIM-Signature:" header versus the current proposal which is effectively a new signature construct designed specifically for DKIM. We'll start by considering the arguments against using CMS SignedData, and then briefly consider XML Signature.

Firstly, CMS SignedData is, by design, a highly complex and generic data structure designed to be usable in many application contexts. While such extensibility is beneficial, it does mean that each application using SignedData has to profile out those fields whose use is not allowed, or where the meaning might be ambiguous. In this case, the encapsulated content info field would be a potential source of confusion for example, since we are typically signing a combination of the message body (where there are existing rules for this field) and some headers (where no such rules exist). Similarly, a CMS SignedData allows for multiple signers in parallel, whereas in DKIM we almost certainly only require the ability to use multiple DKIM-Signature: headers (whether parallel or sequential). SignedData also allows for inclusion of CRLs which, given the above, would seem to make little sense with DKIM. There are a number of other examples of fields that would make little or no sense in the DKIM context.

So there is a minimum cost to adopting SignedData, namely, the necessity to profile a highly complex set of data structures. One should also note that the community that would be taking on this task are, in the main, the email community and hence many are not intimately familiar with S/MIME. So such a profiling task might be quite time consuming and, more seriously, quite error prone.

Secondly, CMS SignedData largely relies upon the use of an X.509 based PKI in order to identify the signer - at least in terms of the most widely supported and tested deployments – and as we've seen above we don't want to adopt a PKI for DKIM. Basically the options in the SignedData SignerInfo structure are to include the certificate issuer name and serial number from an X.509 certificate or else to include a key identifier which is an entirely unstructured octet string – this latter option is actually designed for symmetric cryptosystems and not really for signatures at all. For DKIM, neither option makes sense.

Even if we were to take the key identifier option and include a value that identifies the signer, then we would essentially be re-designing (for the N-th time!) yet another way to encode a DNS name – this time in an ASN.1 OCTET STRING. As it happens, there was a recent debate on exactly this topic in the PKIX working group and it turns out that the choices of how to do this can have complex and sometimes counterintuitive consequences.[5]

The CMS SignedData data structure allows the signer to identify the digest algorithm that they are using as part of the signature process. However, it does not allow specification of the canonicalization (c14n) algorithm, since this is part of the S/MIME message specification [17]. That algorithm is firstly fixed and not pluggable, which may not meet DKIM's requirements, and secondly only covers message bodies, and says nothing about c14n for headers. So DKIM has to do work to specify a c14n algorithm, or more likely a set of such algorithms, since the requirements for

[5] See for example threads related to: http://www.imc.org/ietf-pkix/mail-archive/msg02241.html

c14n will likely differ when signing interpersonal messages versus list-exploded messages. Now it may or may not make sense to adopt some of the work done in S/MIME c14n, but that work clearly is not sufficient for DKIM and so adoption of CMS doesn't help when it comes to c14n, which is in fact, one of the harder things to get right when designing a signature scheme.

If DKIM adopted CMS SignedData in order to carry signatures calculated over the message body and some (c14n'd) headers, then this would inevitably lead to additional data replication in signed messages. The reason is that SignedData, being an ASN.1 defined data structure has no real concept of a pointer and therefore any byte signed has to be represented somewhere in the ASN.1 structure. The most obvious way to do this would be to replicate that data inside the SignedData structure leading to the additional replication. The bad effect of such replication is that each instance creates a new way to go wrong for a verifier – if the signature is cryptographically correct, but the replicated data doesn't match what is in the message headers or body then what should the verifier do?

Alternatively, if DKIM defined a "phantom" data structure, with a rule as to how to represent a digest of that data as the SignedData plaintext, then DKIM would be essentially breaking the SignedData specification as we would be introducing a new digesting step but without that digest algorithm being represented in the SignedData itself.

Since SignedData is an ASN.1 data structure there are some "wasted" bytes required solely in order to indicate which of the many options supported we are using. Effectively this increases the size of each DKIM-Signature header. Now while message size is not a highly significant issue, MTA performance and storage is, especially for larger domains. Similarly, while mail archive storage is not in short supply, each additional byte will be stored for who knows how many years; so again, we should not waste such resources without good reason. The result is that minimizing the size of the DKIM-Signature: header is beneficial and SignedData is taking us in the wrong direction in that respect. There is a similar argument to be made with respect to processing complexity – SignedData requires more CPU cycles compared to the current DKIM proposal and again this can be significant for a busy MTA.

Many of the above points also apply for XML Signatures, however, it has to be said that the c14n situation, the PKI dependency and the data replication problems would all be much more easily handled were DKIM to adopt XML Signature. However, the profiling and efficiency considerations would arguably be worse, so overall there appears to no compelling reason for DKIM to adopt XML Signature.

In summary, when we compare the existing, more-or-less working DKIM-Signature: header proposal against the possible adoption of S/MIME or XML Signature then we see serious problems that would be created in terms of the additional profiling required, the lack of a PKI, data replication and efficiency.

5 Conclusions

The authors of the current DKIM proposal have chosen not to depend on standard PKI and signature schemes. We have presented arguments as to why these choices are reasonable and believe that those arguments are convincing, in particular when DKIM-as-proposed appears to work.

Whether this says something interesting about current PKI and signature scheme standards is an interesting question. The fact that PKI appears not to match an email signature application, when that was once considered one of the most compelling drivers for PKI is interesting to say the least. Perhaps additional attention paid to scaling issues or to a less prescriptive way to use PKI would be beneficial. Similarly, it is interesting that XML Signature appears to be a much closer match to the requirements for DKIM than S/MIME. While this may be simply due to the fact that XML Signature specification is newer and had no backwards compatibility issues to tackle, it may also be due to the flexibility that is inherent in XML based approaches and hard to achieve in ASN.1 based ones. However, the fact that XML Signature seems to be too inefficient for use here is also telling – that level of flexibility may have to come at a price which is unacceptable for a large scale application like email.

Acknowledgements

The author would like to acknowledge the work done by those who actually developed DKIM. A list of their names may be found in [7] – the author is not one of those. Thanks also to Jon Callas, Dave Crocker and Jim Fenton for insightful comments on an earlier version of this paper – all errors and omissions of course remain solely the author's responsibility.

References

1. IETF DKIM working group charter page:
   http://www.ietf.org/html.charters/dkim-charter.html
2. IETF PKIX working group charter page:
   http://www.ietf.org/html.charters/pkix-charter.html
3. IETF OpenPGP working group charter page:
   http://www.ietf.org/html.charters/openpgp-charter.html
4. W3C XML Key management system working group home page:
   http://www.w3.org/2001/XKMS/
5. IETF S/MIME working group charter page:
   http://www.ietf.org/html.charters/smime-charter.html
6. W3C XML Signature working group home page: http://www.w3.org/Signature/
7. Allman, E. et al, "DomainKeys Identified Mail Signatures (DKIM)", Internet draft, draft-ietf-dkim-base-00.txt, February 2006, work-in-progress.
   http://tools.ietf.org/wg/dkim/draft-ietf-dkim-base/
8. Allman, E.et al, "DKIM Sender Signing Policy", Internet draft, draft-allman-dkim-ssp-01, October 2005, work-in-progress. http://tools.ietf.org/wg/dkim/draft-allman-dkim-ssp-01.txt
9. Fenton, J., "Analysis of threats motivating DomainKeys Identified Mail (DKIM)", Internet draft, draft-ietf-dkim-threats-01.txt, March 2006, work-in-progress.
   http://tools.ietf.org/wg/dkim/draft-ietf-dkim-threats/
10. ITU-T Recommendation X.509: InformationTechnology - Open Systems Interconnection – The Directory: Public key and attribute certificate frameworks", August 2005.
11. Hallam-Baker, P, Mysore S., "XML Key Management Specification (XKMS 2.0)", W3C Recommendation, June 2005. http://www.w3.org/TR/xkms2/

12. Cantor, S. et al, “Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0” OASIS Standard, 15 March 2005. http://www.oasis-open.org/committees/download.php/11902/saml-2.0-os.zip
13. Housley, R. et al, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, RFC 3280, April 2002. http://www.ietf.org/rfc/rfc3280.txt
14. Callas, J. et al, “OpenOpenPGP Message Format”, RFC 2440, November 1998. http://www.ietf.org/rfc/rfc2440.txt
15. Householder, A., King, B. “Securing an Internet Name Server”, CERT Co-ordination center, August 2002. http://www.cert.org/archive/pdf/dns.pdf
16. IETF DNSSEC (concluded) working group charter page: http://www.ietf.org/html.charters/OLD/dnssec-charter.html
17. Ramsdell, B. “Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification”, RFC 3251, July 2004. http://www.ietf.org/rfc/rfc3851.txt
18. Housley, R. “Cryptographic Message Syntax (CMS)”, RFC 3252, July 2004. http://www.ietf.org/rfc/rfc3852.txt
19. Eastlake, D. et al, “XML Signature Syntax and Processing”, W3C Recommendation, February 2002. http://www.w3.org/TR/xmldsig-core/

Towards Secure Electronic Workflows

Sebastian Fritsch, Vangelis Karatsiolis, Marcus Lippert,
Alexander Wiesmaier, and Johannes Buchmann

Technische Universität Darmstadt,
Department of Computer Science,
Hochschulstraße 10, D-64289 Darmstadt, Germany
sfritsch@cdc.informatik.tu-darmstadt.de

Abstract. Despite the introduction of information technologies in governmental administrations, most bureaucratic processes are still paper-based. In this paper we present a framework to transfer conventional, paper-based processes to electronic workflows. Thereby, the transformation to e-Government applications has two challenges. First, to find an equivalent description for the single activities and their interaction for defining the entire process. Second, to ensure the security of the process. We identified four types of activities that can be used as basic components for the workflows considered in our work. The security aspects of the electronic representation are ensured by further framework components, for example authentication or authorization. Finally, we present how this framework can be used for other scenarios and discuss some details of our prototype implementation.

Keywords: Workflow Security, Digitize Workflows, Workflow Engine, XPDL, XACML.

1 Introduction

Even though IT systems were introduced in most administrations, bureaucratic processes are still mainly paper-based. Many papers are moved from one desktop to another. Even if an electronic form is used, it will be printed to send it to other workflow participants. Another problem is security issues that appear if sensitive data is affected. There is a need for e-Government applications which are able to handle complete workflows from the initiation to the last workflow step without any media discontinuity.

1.1 Motivation

In our university the appointment of a new professorship is a traditional paper-based workflow. The purpose of this workflow is to initiate an invitation to tender, discuss the possible candidates, and finally negotiate on the contract conditions of the new professor. In this workflow many papers are moved among a lot of people. The creation, distribution, and management of those papers is a time and resource consuming task. With every new appointment the same

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 154–168, 2006.

steps must be performed. Therefore we choose to digitize this workflow. Security considerations exist in this case since personal information is involved. Security must be preserved and the goals to achieve are confidentiality, authentication, integrity, and non-repudiation.

In the federal state of Lower Saxony in Germany about 130 million of paper pages are used for purposes of state administration every year.[1] There is the need to digitize the administration processes in order to make them easier and reduce the amount of paper. They employ a PKI for achieving this. PKI is also used in the JobCard context.[2] This project deals with enabling the employees and employers to administrate their certification documents. All these workflows are in the digitization process. Therefore we need to address this fact as well as the security challenges that occur.

1.2 Contribution

This paper shows how to transfer the traditional university workflow to an electronic form. This workflow consists of a sequence of steps. We point out the security aspects since these are of great importance for the complete workflow. We develop a generic framework for e-Government applications, which supports the reuse of parts of the implementation.

The paper is organized as follows: Section 2 introduces the term workflow and discusses concrete aspects of how to transfer workflows to an electronic representation. Section 3 gives an overview of the basic components we isolated and their relevance in the context of security. Section 4 shows the implementation details of the framework and the workflow components. We explain in Section 5 how our components can be used for transferring other workflows. Section 6 draws a conclusion and describes the future work.

2 Transferring Workflows

This section gives an introduction to the terminology of the workflow context, to workflow engines and to the standard of internal representation used in our system.

The Workflow Management Coalition (WfMC) is an organization that introduced a standard for workflow descriptions. It defines workflows as follows:

> The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to procedural rules. [1, Page 8]

Business processes are defined as linked procedures or activities. Each workflow consists of one or more processes. Processes consist of activities or workflow

[1] `http://www.izn.niedersachsen.de/master/C5252172_N5505837_L20_D0_I3654280.html` (date of access 06.04.2006).

[2] `http://www.itsg.de/download/BroschuereJobcard.pdf` (date of access06.04.2006).

steps that represent a piece of work. An activity requires human or machine interaction for execution. A workflow process has been completed if all its activities have been executed.

Digital workflows have several advantages over the paper-based ones. The first advantage is better process control. Second, auditing can be used. Third, the status of the workflows can be observed better. Fourth, enormous masses of paper can be avoided. These are general arguments that motivate electronic workflow management.

Applications and representation standards have been developed for handling and describing workflows. The applications are called workflow engines. They manage the sequence of workflow steps. Workflow engines can be described as run-time environments for processes. The workflow engine can be called from external applications to get or update the status of a workflow instance. Another way to work on active workflow instances are tool agents. A tool agent is an application that is called by the workflow engine directly. Mostly a tool agent has to solve one special problem, for example to send an email to all participants of a workflow. A tool agent can work on the given workflow data. Finally a tool agent can update the workflow state. Usually the associated workflow activity is completed when the tool agent has completed its task. For describing the processes, an XML based standard has been developed called XPDL [2].

Each XPDL description defines a package of workflow process definitions. Additionally, participants are defined, which are roles, persons, systems, resources or organization-units. A process defines the activities, for example to fill out a form. The activities are connected to each other by transitions. Further, route activities are used to realize decisions, branches, and merges in the process flow. Routing can be performed sequential or in parallel.

In each package, process or activity variables and attributes can be defined for characterizations or information storage. The workflow definition collected in these XPDL descriptions can finally be loaded into a workflow engine. In this engine the processes are instantiated.

Transferring a whole workflow is more than only transferring each step. It is not sufficient to transfer workflow steps into a web-application. Workflows can contain a lot of sensitive data. Therefore, the security properties are very important. The data's authenticity, non-repudiation, confidentiality, and integrity has to be provided for the whole workflow.

2.1 Related Work

The Electronic Circulation Folder (ECF) [6] has been proposed for realizing various e-Government applications. It is based on examining the way that typical processes in a bureau take place. This approach is based on the adoption of folders circulating among bureaus. These folders consist of two parts: the description and the content. The description part is used for describing a process as well as its status (for example at which office a document is found at a point in time). The content part contains all necessary data, like the documents needed in a process.

There are three important aspects that we can observe from the ECF concept. ECFs can contain all kind of data and documents. The flow or migration of an ECF defines the involved office workers and the set of steps that must be performed for a complete processing. Finally, the initiator of the workflow can define the migration of one individual ECF in a flexible way.

The step migration and the possibility to integrate the user in the definition of the processes was adopted in our work. The step migration is arranged by a workflow engine and the user integration by user modelling tools. But the ECF concept does not consider the security aspects which are the focus of our work.

In [5], Kandala and Sandhu present models for secure workflow management systems. They are based on roles and the RBAC framework. In our work we concentrate on the security of a concrete workflow. We introduce components to achieve security. Our goal is to reuse these components for securing other workflows as well.

3 Workflow Scenario and Components

This section introduces the workflow components. First we present the application scenario *Appointment of a new Professorship*. After this, the relationship between the components is described. Finally, we explain the different functionalities of the components.

3.1 Scenario: Appointment of a New Professorship

The benefit of transferring the scenario to an electronic workflow is to save the paper and shorten the time an instance of the workflow needs to be finished. Also the flow and lifetime of the process can be controlled more easily. These facts lead to the introduction of e-Government in the scenario's context. An overview of the whole scenario is given in Figure 1.

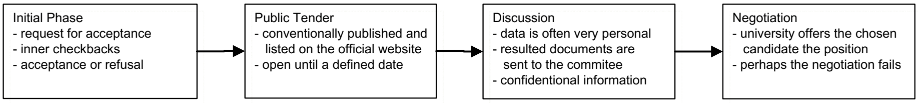

Initial Phase
- request for acceptance
- inner checkbacks
- acceptance or refusal
Public Tender
- conventionally published and listed on the official website
- open until a defined date
Discussion
- data is often very personal
- resulted documents are sent to the commitee
- confidentional information
Negotiation
- university offers the chosen candidate the position
- perhaps the negotiation fails

Fig. 1. Appointment of a new Professorship

The corresponding workflow can only be initiated by the request of a faculty's dean. The request includes detailed information of the intended professorship, the number of employees, the number of allocated rooms, and the period of time the advertisement will be open.

The request is answered by the university's president. The president's response depends on the answers of other administrative departments of the university. Authentication of all workflow participants and confidentiality of the workflow data is important at this stage of the workflow. After that, the dean is informed of the president's decision. In parallel the advertisement is initiated. It will be public for a defined period of time. While it is open, candidates are allowed to view a detailed description of the appointment.

Here availability must be supported since the defined time of acceptance has to be guaranteed. The candidates can send their application for evaluation. All information sent by the candidates has to be confident and unaltered. Additionally, the authenticity of the candidates has to be ensured.

After closing the advertisement, all applications are made available to all members of the appointment commission for internal discussion. These discussions are done on personal and electronic basis. Everything discussed in the commission is undisclosed and must be kept confidential. The discussion may include personal opinions of the commission members. At this stage of the workflow, confidentially is important and no information is allowed to appear in public.

The commission agrees on an ordered list of three candidates. This list is still closed, so it is only allowed to be read by the commission members and the university's president. Next, the president has to approve a candidate from the list, usually the first one. In the next step, the president has to negotiate with the chosen candidate on his conditions. This negotiation includes a lot of personal data. This forces to ensure confidentiality and integrity.

If the negotiation succeeds the workflow is finished. In case of failure, the president has to choose another candidate. To reduce the scenarios complexity, we assume that the entire workflow has to start over again. Thus, there is no need to observe all special cases in the workflow description.

3.2 Components

We use a top-down approach to design and implement the workflow. This also suggests a modular design which allows the reuse of basic components. We developed two different types of components. First, the activity components which are mapped to the activities that are used to build the workflow processes. Most e-Government applications are form based. Therefore, activity components are used for processing form data, inform affected workflow participants and handle branches in the logical process flow. Second, the technical components to realize security properties and to manage the entire workflow process. An overview of the relationship of the components is given in Figure 2.

3.3 Activity Components

As described above the activity components are used to represent most of the workflow activities. Four components have been developed.

Forms Component. In the scenario description some requests were considered, which are traditional paper form based. We represent them with the Forms component. Forms are composed of classical form elements like text fields or choice fields. This component can be suspended, its input can be frozen at any time. Later it can be resumed from storage. Offline usage is also possible. A workflow participant can export a form from the system, work on the data and send it back to the system.

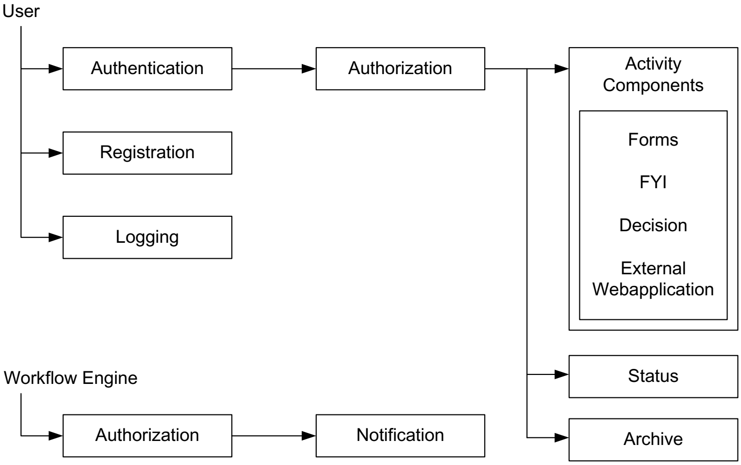


Fig. 2. Relationship of Components

FYI Component. FYI is the abbreviation of "for your information". For example the dean's request at the beginning of the workflow has to be shown to the president. This component presents some afore inserted information to a participant. This participant has to be informed only at this point of the process. After reading the information, the participant must commit having read it.

Because of using different routing strategies, the FYI component can be used as blocking component. That means the whole process has to wait until reading was committed, or it can be routed in parallel to the rest of the process.

Decision Component. In the scenario the president has to decide whether the dean's request is accepted or not. Later on, a candidate with whom the negotiations will start, must be chosen. These tasks are realized by introducing the Decision component. This component controls the process flow. Decisions from this component can be read in the following workflow steps and they can be used for defining which branch the process has to follow. To be concrete, the users are shown some kind of question that they have to answer and a decision therefore is met. The users' decision will be stored in a workflow variable.

External Webapplication Component. The advertisement and the electronic discussion of the appointment commission in the scenario are complex applications, which need special attention. They are linked to a process to provide important data, for example the applications of the candidates. We decided to introduce the External Webapplication Component. One external application can be linked to one process activity in a workflow.

The component mostly supports the input of data from external participants to a running workflow process. The data inserted to the web-application is available in the other workflow activities.

3.4 Technical Components

The technical components are used by the workflow system to support security properties and manage the workflow process.

Authentication Component. In the scenario's description a lot of authentication aspects have been introduced. Precisely, each request to the workflow system is based on an authentication.

The authentication is based on the user's knowledge or presentation of some information. Possible authentication mechanisms are, for example, username-password, PKI-based, or biometric authentication. Additionally, the physical presence of the user can also be supported as a traditional authentication mechanism.

The authentication ends with retrieving a list of groups in which the user is a member. The combination of both, that is the name and the group membership of the user, is called the user's identity. This identity is used in the next component, the Authorization component.

Authorization Component. This component restricts access to resources in the scenario, e.g. reading requests or access to the electronic discussion.

The authorization's decision takes place in a separated part of the application. The decisions must be enforced at the policy enforcement point. At this point the policy requests are generated and sent to the policy decision point.

If a user requests to perform a command on a resource, a policy request is generated. After sending a request the response returns a *deny* or *permit*. This is performed by the decision point.

Notification Component. The Notification component is used to inform participants for changes or news in the workflow proceedings. The component can be used for binding persons closer to the system when they use the system only sporadically, like the commission members do.

Security aspects have to be addressed in this component because information is sent out of the workflow system, and data can no longer be controlled. We decided to perform policy checks in this component, which are run by the authorization component.

Registration Component. The candidates are not members of the workflow system, but they have to send their applications to the system. Up to now our implementation is based on a smart card based public key infrastructure. Since it is not practical to integrate the candidates by registering them and providing them with a smart card, we have to find another solution.

Our solution is to provide temporarily valid certificates. The certificates and private keys are delivered in software. These certificates have to be mapped to the newly introduced external participants. These participants are handled as normal participants from the workflow system's point of view.

If a workflow process instance is finished, the corresponding certificates must be disabled. We chose not to revoke the certificates but to disable the authentication possibilities associated with them. Thus the access control is delegated to the Authentication component.

Status Component. If a participant has to make a decision based on some information added in previous workflow steps, this component supports the user to get an overview of the whole workflow and its attached documents. In this component policy checks have to be performed.

Archive Component. When introducing e-Government applications a central question is, how data, which is finally available only electronically, is archived. This is even more important in workflow systems since workflows are finished at some point of time. So there is a need to archive and later reconstruct the workflow's processing and the related data. These functions must be provided by an archive component.

The Archive component was designed to allow access to already finished workflow processes. The documents of these workflows are still available for all participants who have the authorization to view these files. An important aspect is to decide whether changes to the authorization settings have an impact on any archived data. If this shall be avoided the authorization settings must also be archived.

Logging Component. This component logs authentications, resource or policy requests, and changes to the workflow states. These auditing mechanisms allow to detect problematic authentications or policy decisions.

4 Implementing Components

We have seen the components that we need for realising the workflow. In this section we see their implementation.

4.1 The PKI Installation

In our university a campus-wide smart card is used. Every student possesses a smart card that contains a key pair and a corresponding certificate. This certificate is used for digital signature purposes. In the second phase of the project a second key pair can be written on the card. This will be used for encryption. The employees of the university will receive a smart card, too, that can be used for encryption and digital signature. The encryption keys will be backed up. The employees are the ones that are using the framework. The technical entities (like web servers) are also being certified for supporting services like SSL.

The PKI is used for securing our electronic workflow. It offers the authentication, confidentiality, and non-repudiation services. The authorization part of the workflow is organized by the use of XACML [8]. The authentication and authorization information are combined together in order to realize the access control mechanisms needed in the framework. Therefore we benefit from the existence of the current PKI installation.

4.2 Framework Concept and Implementation

Our implementation is based on a four-tier architecture. The whole framework concept is shown in Figure 3.

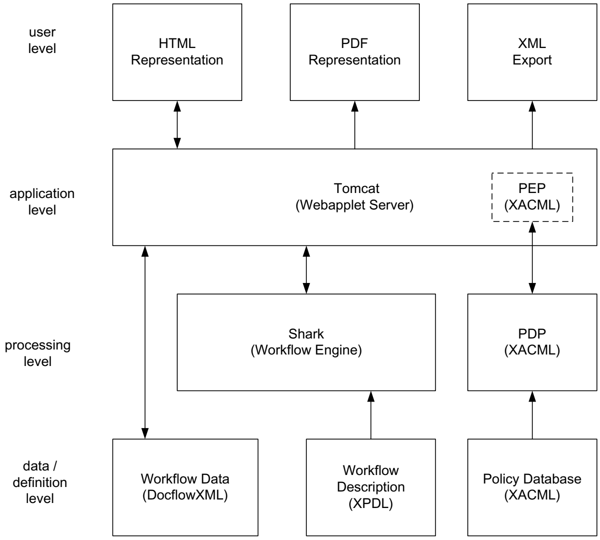


Fig. 3. Framework Overview

The main elements are the workflow engine and the web applet server. Our prototype is completely written in Java. Since Java offers platform independency, a flexible integration of our solution to existent systems is possible. We have chosen to use an Apache Tomcat web applet container for running the web application. Since we use XPDL as workflow description language we have chosen to use Enhydra Shark as the corresponding workflow engine. This workflow engine is queried by the central web application which is used by the workflow participants. The implementation of the components is done inside a servlet. The rest of this section will explain the implementation of the activity and the technical components.

4.3 Activity Components

The underlying software system, the workflow engine, enables to define attributes for each activity. The attributes are used as parameters to define the type and appearance of the different activity components described next.

All activity components are generally based on an HTML representation. In the XPDL definition can optionally be defined, if the inserted form data should be digitally signed. If this is enabled for an activity, the creation of the signature completes this activity.

Forms Component. The implementation of the Forms component is realized with HTML forms. The form structure is loaded from an external defined file similar to XForms.[3]

This component needs some attributes. First the XForms file which contains the structure of the form, and second the document's name to map the form data to the information storage place. The storage called Docflow[4] is document-centric organized.

If the workflow is suspended in this step, the data is saved to the Docflow file of the running instance. Later, the form elements input will be restored from that file. When completing the insertion, the data can optionally be digitally signed.

Another implementation that will be done as future work is a PDF based implementation. Therefore, the XForms information is read and a PDF document containing form elements is created.

FYI Component. This component shows some information to the assigned recipient. This information can be static or stored in the running workflow. The second case needs a reference to an internal workflow document. The participant has to commit that the information has been read. The blocking or non-blocking aspect is not affected by the implementation because this is determined by the process definition.

Decision Component. This component was introduced to control the process flow. Some type of question and possible answers is presented to the user. Finally, a workflow variable is set to the given answer.

This component needs the decision's question and answers. The name of the workflow variable must also be given. The value of this variable can be accessed in the next workflow processes.

External Webapplication Component. This flexible component is an extension of the workflow system implemented as a web-application. Complex workflow steps can be performed in such an external application.

To build such components, knowledge of the underlying workflow engine is needed. We have chosen to provide a lot of functionality in a library. This library was developed while building some example components. The most advanced example in our scenario is used for receiving applications to advertisements.

Integrating this component provided a lot of flexibility to our framework. Because of the web based structure there is no break in the representation to the user.

4.4 Technical Components

Authentication Component. Authentication has to be performed before a user is allowed to send a request to the system. Our implementation is mostly

[3] XForms is a standard to define form elements used in web-applications, see `http://www.w3.org/MarkUp/Forms/` (date of access 07.04.2006).

[4] As part of this work, Docflow was defined as a simple document management system which stores the workflow related data. We decided to implement this abstract definition with an XML document structure.

based on a smart card based public key infrastructure. Thus, digital certificates can be used for authentication. First, the webserver checks if a valid certificate is presented. If it succeeds, an HTTPS connection to the server is established.

Next it is checked if the certificate's distinguished name is allowed to log into the system. This information is stored in an XML file containing all valid users. If a user is allowed to log on, the user's group memberships are retrieved from an XML file as described in Section 3. In this file groups and subgroups are defined. It is possible to describe a complete hierarchy. After that, the identity of the user is known to the system.

If the authentication was not successful, a failure message is shown to the user. Further processing will not be done.

Authorization Component. We implement authorization by using an architecture based on XACML [8]. An XACML system is divided into three parts: the policy database, the Policy Enforcement Point (PEP), and the Policy Decision Point (PDP).

The policy database is an XACML file which contains the policies that the system has to enforce. The PEP is located on the application side and generates the policy requests which are sent to the PDP.

The PDP asks the policy database if a requested policy, a *(subject, object, command)* triplet, results in a *deny* or *permit* response. The PEP has to enforce the responded decisions. In the background an audit message is created which includes details about the requested resource and the response. Auditing is supported by this component. It can be enhanced by implementing the Logging component.

The workflow system has to determine the policy triplets. After collecting this information, the request is generated. If the response is *deny*, the system may fall back to request another command, for example if write access was denied, the system may ask in a second step to retrieve only read access.

We could have used XACML to accomplish the task of user authentication. The idea was introduced in [7]. Role Assignment Policies are used to determine a user's group memberships. The advantage of performing user authentication with XACML is the small number of different file formats. We did not use XACML for authentication purposes, because for determining the whole set of a user's group membership one request per available group is needed. This is very inefficient for a large number of groups.

Notification Component. This component has a wide range of implementation variants. We decided to use notifications based on email messages.

The content of these notifications can be static or dynamically filled with some document data inserted in the workflow process. We provide the possibility to enforce signing and encrypting of these notifications. This is no problem, if the infrastructure is certificate based, as our implementation which uses smart cards and X.509 certificates. Each user has an X.509 [3], [4] certificate, that enables the use of S/MIME [9]. This also includes the external workflow participants who are provided with a temporary valid certificate.

Policy checks have to be enforced, because information is sent out of the workflow system. Thus, the data leaves the security controlled system. The recipient of the notification is the subject that has to be policy checked.

Registration Component. An unknown user has to fill a web-based form in order to register to the system. Next, a new digital identity is created and a certificate is matched to his identity. The certificate and the corresponding private key is delivered to the user in a PKCS#12 [10] file by e-mail.

In the next step, the certificate must be installed in the user's browser. The PKCS#12 file can also be installed in an email client for securing the e-mail communication with a registered user. The created identity will be added to the user repository.

The further components, Status, Archive, and Logging have been designed but not implemented yet.

4.5 Conclusion

A prototype of the e-Government framework has been developed. We defined the configuration for the scenario *Appointment of a new Professorship*. This application was completely transferred to an e-Government application inside our framework.

However, since we developed a framework for transferring former paper-based workflows to e-Government applications, we have to show if and how other scenarios can be transferred. We decided to introduce two more scenarios and describe how the transformation process can be performed. This is discussed in detail in the next section.

To introduce new applications, three major steps have to be performed. The scenario or application has to be described in an XPDL workflow description. This can be done by using graphical tools. The paper forms used in the conventional workflow must be converted to an electronic representation, wherefore we use XForms. Finally, the security of the application must be defined. Documents introduced in the XPDL workflow definition are referenced in the policies. Additionally the participants and the corresponding actions have to be defined. After performing these three steps, new e-Government applications can be provided by our framework.

5 Applying the Framework to Other Scenarios

We have chosen two further applications from the university context to show how they can also be electronically transferred. The first scenario is called the *Request for Scholarship*. A student can request to receive a scholarship. The request is sent to the student office and is processed. The second scenario is the *Travel Expense Accounting* scenario. Employees request to refund the cost of their business journey. Both scenarios are explained in detail in the next two sections.

Scenario: Request for Scholarship

Two parties are involved in the Request for Scholarship scenario, a student and the student office. A student fills out a request for a scholarship and sends it to the university's student office. The affected office worker has to decide if further information is needed. If more information is needed, a request can be sent to the student who has to answer it.

Finally, the office worker (or a commission) has to decide if the student will receive the scholarship. The student will be informed about the decision. In this scenario the security targets are the authenticity of the student and the confidentiality of the workflow's data.

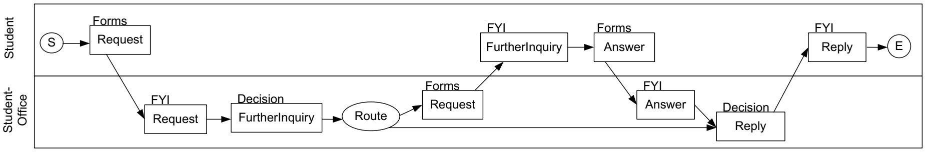


Fig. 4. Scenario Request for Scholarship

Scenario: Travel Expense Accounting

After a business journey, for example after visiting a conference, the employee is asked to fill out a travel expense accounting. This is given to the administrative office. The accounting is checked for completeness. If some information is missing the request is returned to the employee, who has to complete it. If the information is complete, the travel expense is accepted by the office, it is signed by the head of department and the workflow finishes.

In this scenario we have three involved roles: the employee, the office worker, and the head of department. The task to check the completeness and the task of acceptance is bound to the administrative office. All data concerning this workflow may only be read by these three roles. In addition only the employee is allowed to fill out such a form and only the department head is allowed to finally sign it.

We succeed in transferring both scenarios by using our framework. We show in Figure 4 what components have been (re)used to implement the Scholarship

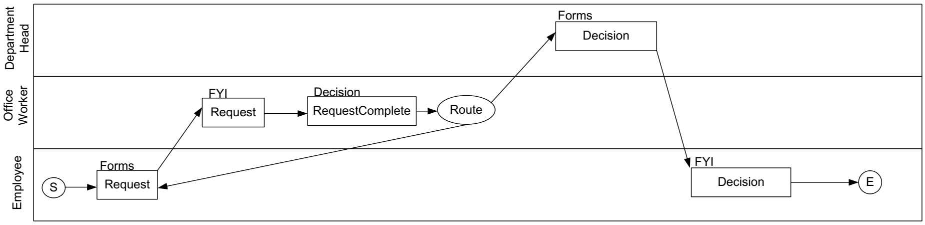


Fig. 5. Scenario Travel Expense Accounting

scenario. The implementation contains one routing branch. Which branch will be activated is decided in the *FurtherInquiry* element, which is an instance of the Decision component. A simple yes-no question is asked. In addition or alternatively the Notification component could be used to inform the student about the decision of the student office. The realization of the travel expense scenario is shown in Figure 5.

6 Conclusion and Future Work

We introduced a framework for transferring traditional workflows to electronic ones. First, we gave an introduction to the terminology. Then we showed the scenario *Appointment of a new Professorship* and how our framework fulfills its requirements by introducing the different components. Their concept and implementation have been developed in a top-down approach. Finally, we briefly showed how to transfer other workflows by using the same components.

Our studied workflows can be extended by a retrace functionality. This is important, for example, if a participant has not completely filled out a form and this step should be reassigned again. We call this a retrograde step migration. We must test whether the used workflow engine supports this functionality.

We can also enhance the scope of our framework to be used in other contexts. Arbitrary binary data can not be efficiently integrated in our current implementation, since forms are the basis of our data representation. Such data is contained for example, in a workflow that processes construction plans. An efficient representation of this data is required. Moreover, this data has special security properties that we must also consider.

References

1. Workflow Mangement Coalition. WfMC Terminology & Glossary, Document Number WFMC-TC-1011. Avalaible at `http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf` (06 Apr. 2006), February 1999.
2. Workflow Mangement Coalition. XML Process Definition Language, Document Number WFMC-TC-1025. Avalaible at `http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf` (06 Apr. 2006), October 2002.
3. R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure - Certificate and Certificate Revocation List (CRL) Profile. *IETF Request For Comments*, 3280, April 2002.
4. Recommendation X.509 ITU-T. Information Technology – Open Systems Interconnection – The Directory: Authentication Framework, August 1997.
5. S. Kandala and R. Sandhu. Secure Role-Based Workflow Models. In *Database and Application Security XV, IFIP TC11/WG11.3 Fifteenth Annual Working Conference on Database and Application Security*, volume 215 of *IFIP Conference Proceedings*, pages 45–58. Kluwer, 2001.
6. B. Karbe, N. Ramsperger, and P. Weiss. Support of Cooperative Work by Electronic Circulation Folders. In *Proceedings of the ACM SIGOIS and IEEE CS TC-OA conference on Office information systems*, pages 109–117, April 1990.

7. G. López, O. Cánovas, and A. F. Gómez-Skarmeta. Use of XACML Policies for a Network Access Control Service. In *The 4th International Workshop for Applied PKI, IWAP 2005*, pages 111–122, September 2005.
8. Organization for the Advancement of Structured Information Standards (OASIS). XACML 2.0 - OASIS Standard Specification Set. Avalaible at `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml` (06 Apr. 2006).
9. B. Ramsdell. Secure / Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification. *IETF Request For Comments*, 3851, July 2004.
10. Laboratories RSA. PKCS#12 v1.0: Personal Information Exchange Syntax. Avalaible at `http://www.rsasecurity.com/rsalabs/node.asp?id=2124` (06 Apr. 2006), June 1999.

An Access Control System for Multimedia Content Distribution

Manuel Sánchez[1], Gabriel López[1], Óscar Cánovas[2], Juan A. Sánchez[1], and Antonio F. Gómez-Skarmeta[1]

[1] Department of Information Engineering and Communications
[2] Department of Computer Engineering
University of Murcia, Spain
{msc, gabilm, jlaguna, skarmeta}@dif.um.es,
ocanovas@ditec.um.es

Abstract. Multimedia content distribution has appeared as a new growth market offered by network providers, defining resource access infrastructures able to support both wired and wireless accesses. Although these infrastructures have been widely studied in the last years, the main aim of those works has been focused more on the distribution process than on a suitable security infrastructure to protect that content. Therefore, the study of security systems able to offer authentication, authorization and other security-related requirements for those kinds of scenarios is still an open research field. In this paper, we propose a new scheme which takes advantage of a previously existing underlying authorization infrastructure among the involved organizations, the NAS-SAML system, to build a multimedia content distribution with an advanced and extensible authorization mechanism. The target scenario is the one proposed by the VIDIOS project, which defines an architecture for multimedia transmissions across error prone networks such as Internet backbones and mobile access networks.

1 Introduction

Wireless and wired broadband accesses are a strategic growth market covered by almost all European network providers. European Internet Service Providers (ISP) identified multimedia content distribution as a potential means to create significant revenue above pure infrastructure business. In fact, video streaming is regarded as a short term emerging service with several different opportunities, ranging from personal video conferencing to video on demand.

One of the main concerns of a multimedia distribution system is to protect the distribution process against malicious users. First, it is necessary to ensure that only users which have paid the fees can access to the system. Second, the system must ensure that the protected content is only obtained by those users with the appropriate access level, that is, only by authorized users. Finally, the confidentiality and integrity of the multimedia streaming must be protected from passive and active attacks.

It is worth noting that in these scenarios, where multimedia contents are transported from providers to customers through open data networks, it is possible to find inter-domain scenarios, for example when the domain providing the multimedia content and the costumer's ISP domain are different. Moreover, users can access to the content

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 169–183, 2006.

provider from different ISPs. This involves an explicit agreement among the involved domains in order to exchange the information needed to perform access control functions, as well as the QoS enforcement.

Although access control is a key feature in content distribution, this is not an exclusive topic of this field. Traditionally, organizations have protected critical resources, for example the communication network. In fact, the AAA architecture [15] was designed to solve this last problem, using different mechanisms to identify end users, such as login/password or identity certificates. Therefore, one of the most common access control mechanisms used by network providers is the one based on the AAA architecture.

In this paper, we present an access control architecture developed for the VIDIOS project [10], an international consortium composed by eight institutions (T-Systems International, FH Mannheim, Quix, Satec, Scopus, Telefonica, University of Goettingen and University of Murcia). Among the main aims of this project is the design and validation of an architecture for delivering multimedia content, especially MPEG-4 encoded video, over a Multi Protocol Label Switching (MPLS) backbone. Due to the similarities we can find regarding other existing access control architectures, it would be desirable to reuse most of the ideas and contributions included in the existing proposals to define the access control architecture for VIDIOS. In fact, a successfully tested system such as NAS-SAML [20] will be used as the starting point of the authentication, authorization and QoS enforcement scenarios. As we will see, NAS-SAML makes use of XML-based standards to manage the authentication and authorization data and to express the access control policies in an extensible and distributed way.

The rest of this paper is structured as follows. Section 2 defines the VIDIOS project. Then, Section 3 establishes the main requirements of the access control architecture once we have analyzed the main goals of VIDIOS. Section 4 introduces the NAS-SAML system, which will be used as the starting point to define the access control architecture. Next, Section 5 presents the main elements of that architecture and Section 6 details the way the authentication, authorization and QoS enforcement is finally performed. Section 7 shows some details about the implementation. Section 8 describes the related work that informed our research. Finally, we conclude the paper with our remarks and future directions.

2 VIDIOS

VIDIOS designs and validates an architecture which delivers end-to-end Quality of Service for multimedia transmissions across error prone networks such as Internet backbones and mobile access networks. MPEG-4 Advanced Video Coding, combined with robust and scalable coding techniques are applied by VIDIOS for robust transport over a MPLS backbone. The MPLS backbone further protects the application stream by QoS based on already implemented standards.

Figure 1 shows a basic overview of the VIDIOS functionality. This image shows the content provider (CP) where the user is registered and two different ISP domains. Users can access to the service through any of the ISPs, as long as the ISP has an agreement with the CP. This agreement must specify the QoS that the ISP has to provide to its customers, which is directly related to the access level the user has subscribed with the content provider. Regarding to users, it is necessary that they provide some kind of identification when accessing the system. Moreover, the access level determines the quality of the content they can watch. Finally, it can be seen how the multimedia stream is protected to ensure the content privacy.

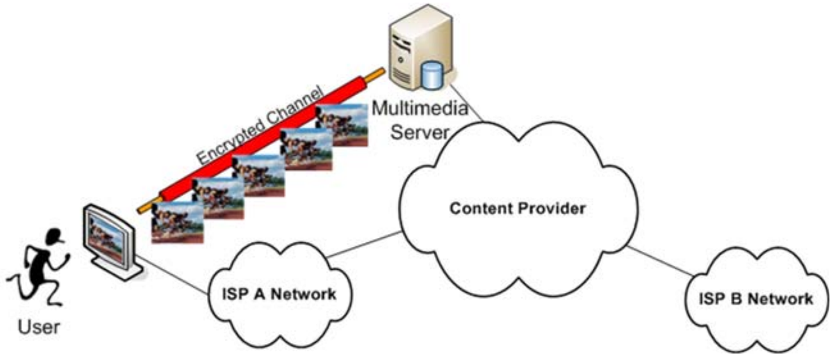


Fig. 1. VIDIOS overview

The system should define two kind of users with different functionalities: administrators and regular users or clients. On one hand, administrators can manage multimedia contents and access levels. The management of multimedia contents include creating and deleting them, and starting multicast sessions. The management of access levels include the definition, modification and assignment of those levels to each user. Moreover, administrators manage the encryption keys used to protect the multimedia content delivery. On the other hand, clients can only access to authorized contents.

Access to multimedia contents should be done via unicast or multicast, depending on the kind of content being transmitted. A live content is more suitable to be transmitted over multicast than a recorded film. This architecture must be flexible enough to allow both kinds of transport in a transparent way for the users.

3 Requirements of the Access Control Architecture in VIDIOS

To provide the functionality described above it is necessary that the access control architecture fulfills, at least, a set of requirements related to multimedia streaming, support for interdomain scenarios, security issues and QoS enforcement. It is important to emphasize that the architecture developed in VIDIOS should not introduce new protocols or policy languages, so it must be based on existing and contrasted proposals.

3.1 Multimedia Streaming Requirements

It is necessary a streaming protocol which addresses all the needs described in the previous section. That is, support for both unicast and multicast streaming and extensibility to include key management. Furthermore, for on-demand contents, the end user must be able to control the video stream, e.g. play, pause or stop the video. RTSP [25] is the best choice to manage the different streaming services that VIDIOS provides.

The Real Time Streaming Protocol (RTSP), is an application-level protocol for control over the delivery of data with real-time properties. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. Sources of data can include both live data feeds and stored clips. This protocol is intended to control multiple data delivery sessions, provide a means for choosing delivery channels such as UDP, multicast UDP and TCP, and provides a means for choosing delivery mechanisms based upon RTP. It provides 'VCR-style' remote control functionality for audio and video streams, like pause, fast forward, reverse, and absolute positioning.

3.2 Security Requirements

The two main security requirements to be fulfilled are authentication and authorization. In fact, VIDIOS defines two independent services, the first one to identify the user and the second one to check whether he is able to watch a particular content. As we detail below, they will be related by means of the use of a security token:

- *Authentication*. When a user tries to watch multimedia contents, the first step is to authenticate the identity during session initiation. Some kind of Single Sign On (SSO) functionality should be provided, so the user has only to log in the system once. Therefore, after the user has been authenticated, some kind of token should be generated, which will be used afterward during the authorization check, that is, every time the user is willing to watch a particular content.
- *Authorization*. It should be possible to define different access levels. Every multimedia content has to be classified, belonging to one access level. This is the way to characterize the contents that can be accessed from each access level. Finally, every user has to be assigned to an access level.

Besides, two new requirements arise when considering the media stream protection. First, it is necessary to encrypt all the multimedia content streamed to the client to prevent not registered clients from decoding the content without paying for it. But also, as a mean of protection against old users who can have the keys after they have deregistered from the service, it is recommended that encryption keys should be regenerated.

The stream protection entails the need for distributing the encryption keys to authorized users in a secure way. Therefore, it is necessary to define a way to share those keys between the RTSP server and authorized clients. In this scenario, MIKEY [11] can be used as key management scheme for the distribution of the encryption keys. The extension of RTSP to include the use of MIKEY in order to secure the media stream is described in [12].

3.3 QoS Requirements

Since the user is paying for the service, it is necessary to ensure the user satisfaction in terms of QoS. Therefore a third independent service can be introduced in the content delivery process. This one must be responsible for checking the availability of bandwidth between the user and the multimedia server, and for enforcing the suitable

QoS level of the multimedia stream according to the user's access level. Following the SSO scheme, this service will also make use of the token to identify the corresponding user.

The first requirement can be addressed by using the Priority Promotion Scheme (PPS) [22]. PPS offers a mechanism to perform an end-to-end measurement of the availability of network resources. The main idea is that an application measures whether sufficient bandwidth is available by sending application-like measurement traffic before it starts to send the real application traffic. The measurement and application data packets get different treatment regarding the QoS parameters provided by the network, and therefore the PPS must be supported by the network. PPS can be integrated in RTSP, in such a way that only if enough bandwidth is available, the RTSP protocol can continue properly.

The second requirement implies an interdomain communication, since it is necessary that the ISP recovers some user information from the CP to determine the QoS level to apply to the user. Once the ISP obtains the user's access level, it is necessary to translate it into an adequate QoS level according to the service level agreement with the content provider. Then, the QoS is enforced by means of a DiffServ [13] schema. In this way, the RTSP server marks the packets belonging to the multimedia stream and the ISP network processes them with the suitable QoS.

3.4 Interdomain Requirements

The existence of different domains in the system implies the need for some kind of service level agreement (SLA), where the CPs and ISPs specify the different types of users and the QoS which the ISPs have to enforce for each one. This SLA requires the exchange of some information between domains to determine, on one hand, the kind of access the user obtains in the CP domain, and on the other hand the level of QoS that the ISP has to provide to that user. This can be accomplished by means of the deployment of an AAA infrastructure [15] among the different domains to enable the exchange of data. This kind of infrastructure involves the existence of an AAA server in each domain which centralizes the authentication and authorization tasks.

4 SAML-Based Network Access Control Architecture

During the last years, how to control the users that are making use of computers networks has become an increasing concern for network administrators. As a direct consequence, several security technologies have recently emerged in order to provide access control mechanisms based on the authentication of users [18], [17].

Traditionally, network access systems have been based on login/password mechanism. Other systems, following a more advanced approach for mutual authentication, are based on X.509 identity certificates. These systems are especially useful for organizations which are concerned about the real identity of the requester. There are other organizations where the different users are classified according to their administrative tasks, the type of service obtained, or some others internal requirements. In those previous scenarios, the user's identity could not be enough to grant the access to the resource being controlled, since we should know the role being played by the user in order to offer the right service. Therefore, a system able to grant to the different users the set of attributes specifying those privileges or roles is needed. This kind of systems is usually designed following the principles of the Role Based Access Control (RBAC) model [16].

In [20], a network access control approach based on X.509 identity certificates and authorization attributes is presented. This proposal is based on the SAML and the XACML standards, which will be used for expressing access control policies based on attributes, authorization statements, and authorization protocols. Authorization is mainly based on the definition of access control policies [19] including the sets of users pertaining to different subject domains which will be able to be assigned to different roles in order to gain access to the network of a service provider, under specific circumstances.

The system operates as follows. Every end user belongs to a home domain, where he was given a set of attributes stating the roles he plays. When the user requests a network connection in a particular domain by means of an 802.1X connection, the request is captured by a AAA (Authentication, Authorization and Accouting)[15] server located in the target domain, and it makes a query to obtain the attributes linked to the user from an authority responsible for managing them, located in the user's home domain. Alternatively, following a push approach, the user can present itself its attributes instead of let the AAA server to recover them. The communication between different domains is carried out using the DIAMETER protocol. Finally, the AAA server sends an authorization decision query to a local PDP (Policy Decision Point) entity, and that element provides an answer indicating whether the attributes satisfy the resource access control policy. Furthermore, that policy can also establish the set of obligations derived from that decision, for example some QoS parameters, security options, etc. This general scheme works both in single and inter-domain scenarios.

NAS-SAML has been also integrated with other authorization systems, such as PERMIS [21], by means of a credential conversion service [14] used to translate authorization credentials from one source domain to a target one, and also has been integrated with other high level applications, such as Grid Computing [24], in order to provide the required authorization process.

NAS-SAML was defined to solve the authorization problems in a network access control environment, defining how networking and authorization entities should interact and the type of security information they should exchange. Once this scenario is defined and authorization entities are established, it can easily be adapted to be used by any high level service or application. One example of those high level applications is the multimedia distribution content over multi-domain scenarios, where ISPs and CPs need to establish authorization agreements in order to define the multimedia content distribution, access levels or QoS properties. Those domains can take advantage of a previously established NAS-SAML infrastructure in order to define how the user authentication and authorization process for protecting the multimedia content and the exchange of QoS information will be done. The following sections detail the proposed scenario.

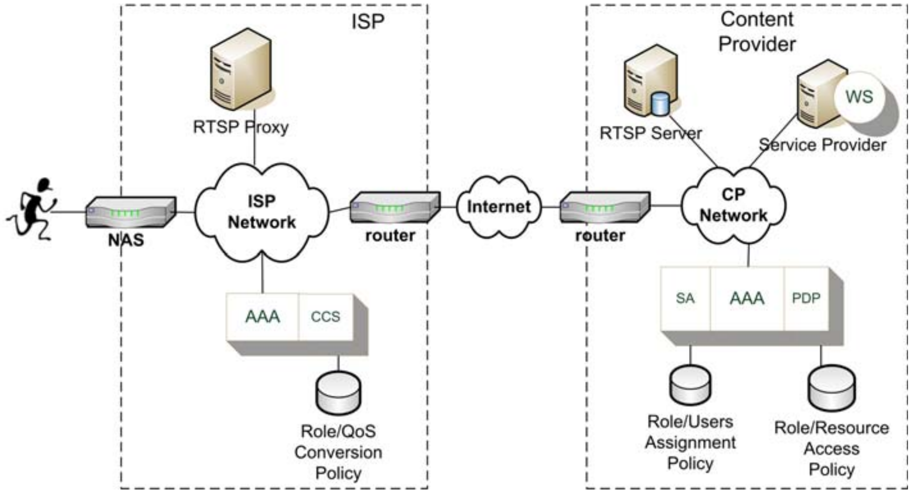


Fig. 2. Architectural Elements

5 Proposed Access Control Architecture

This section describes elements conforming the authentication, authorization and QoS enforcement processes. As Figure 2 shows, this architecture might be used when two or more organizations share an AAA infrastructure with NAS-SAML support.

From the point of view of the NAS-SAML system, each organization has its own AAA server, the key element in this scenario since it is responsible for performing the authorization process in every domain. On one hand, the Content Provider domain, where users are going to be authenticated and authorized, needs to define two modules in order to help the AAA server to perform the authorization tasks. First, the module for producing authorization attributes. In this scenario the authorization attributes represent the access level assigned to the user by the Content Provider. Second, the module to generate authorization decisions based on those access level attributes.

On the other hand, the Internet Service Provider, where QoS properties are enforced, needs a new module in order to help the system to translate the user's access level attributes into QoS parameters.

In order to define the appropriate authentication, authorization and QoS processes, these entities need to make use of a set of policies which lead their behavior. The Content Provider domain needs a policy to assign access levels to end users, and another policy to define the access rights regarding the user access level. Moreover, the ISP domain needs a policy where the expected access level attributes from a CP domain need to be translated into the appropriate QoS parameters.

From the point of view of the multimedia content distribution system, both domains need to define how the existing entities will take advantage of the NAS-SAML infrastructure. First, the Content Provider, which makes use of a Service Provider to show the set of available services, will be responsible for the authentication process. Then, the RTSP Server will guide the user authorization process, making also use of the

NAS-SAML infrastructure. Finally, the ISP domain defines a RTSP proxy entity, which acts as an intermediary element between the end user and the RTSP server during the authorization process, and which will be responsible for the establishment of the QoS properties once the authorization has been performed.

Next, a brief overview of the different components is provided.

- *End User*: Entity requesting access to the multimedia content. The end user pays for a specific access level in the Content Provider domain, and based on this access level he is able to view a set of selected multimedia contents. The access level also ensures a specific QoS in the ISP provider.
- *Service Provider*: This entity is the entry point to all the services in the CP domain. It is in charge of obtaining the user's login and password, and showing the list of available video resources.
- *Content Provider AAA Server*: This AAA server is used to manage the authentication, authorization and attribute requests. It makes use of the DIAMETER protocol as transport mechanism between domains.
- *Source Authority (SA)*: This module manages the assignment of access levels to users. The SA will receive requests, always through the AAA server, and will be guided by a *access level assignment policy*, defined in XACML.
- *RTSP Proxy*: This element must be present in every ISP domain which has subscribed a SLA with a Content Provider domain. This service is responsible for requesting the selected multimedia content and for performing the QoS related tasks.
- *Content Provider Policy Decision Point (PDP)*: This module is the entity responsible for generating the statements related to authorization decisions. Moreover, this element interacts with the policy repository, where a XACML *resource access policy* is stored. The PDP has to obtain the user's access level, since the access control policy is expressed in terms of these access levels. Finally, the PDP will generate an authorization decision statement regarding all the collected evidences.
- *Credential Conversion Service (CCS)*: This service [14] is leaded by a *credential conversion policy*, written in XACML, which establishes the relationship between access levels and QoS properties.
- *Network Access Router (NAS)*: This element is the network device which connects the user to the ISP network, and where some of the QoS properties must be enforced.

Once we have depicted the needed elements for the access control system, the next section will provide the details concerning to how the authentication, authorization and QoS enforcement processes are actually performed.

6 Design of the Access Control System

Interactions among the different components described in the previous section will depend on the requirements already explained. The access control process is performed in three different steps, which must be successfully accomplished in order to proceed with the next stage. First, the system uses a Single Sign On (SSO) mechanism in order to authenticate the identity of the user who is going to request the access to the protected

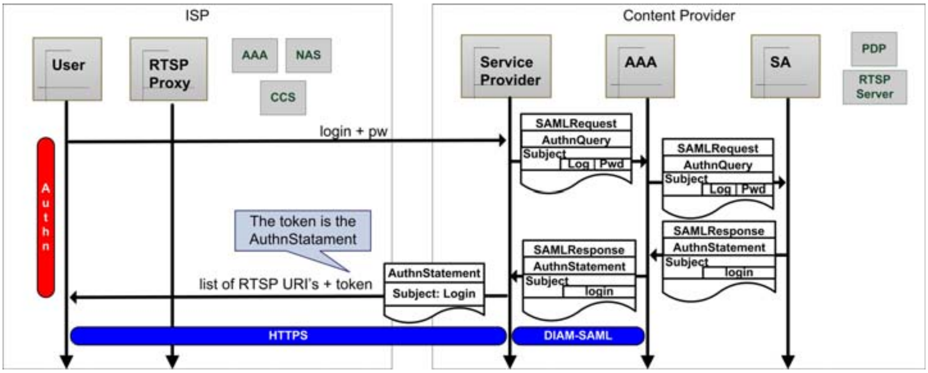


Fig. 3. User authentication process

videos. As a result of this authentication, a security token will be generated to indicate that the user was indeed validated and to express the digital identity of the user inside our system. Moreover, they also retrieve a list of the multimedia content that can be distributed from the service provider. Once users have obtained this information, they can select one of those videos and request for its distribution. During this step, using the security information previously generated, the service provider will check whether the access level assigned to the end user enables the distribution of the requested content. This process will be performed using a pull approach, that is, the acquisition and validation of user attributes will be performed by internal entities, with no user intervention. Finally, when requests are approved, the system must initiate the last process, that is, the enforcement of the QoS properties that will be necessary to watch the video in a proper way.

It is out of the scope of this paper the way the users are registered in the multimedia service provider and how they are assigned to a particular access level. Hereinafter we will assume that the end users have already signed in for a particular service provider, and therefore they are authorized to access some of the protected contents.

The following subsections will present the details of each step of the access control system, paying special attention to the different protocols, messages and pieces of information used to perform each process.

6.1 User Authentication

The first step to watch a particular video is to log in the system (Figure 3). This SSO process is accomplished via Web, using a protected HTTP connection. End users provide their username and password pairs, for example using a HTML form, that must be checked by the service provider. This validation process is actually performed by an AAA server, using some mechanism such as a database query, a LDAP connection, etc. Therefore, the service provider must exchange the login/password pair with the AAA server using some protocol suitable for this kind of communication. We decided to use the DIAMETER-SAML protocol (a DIAMETER encapsulation of SAML messages) since it is the mechanism defined in the NAS-SAML system. Consequently, a *SAM-*

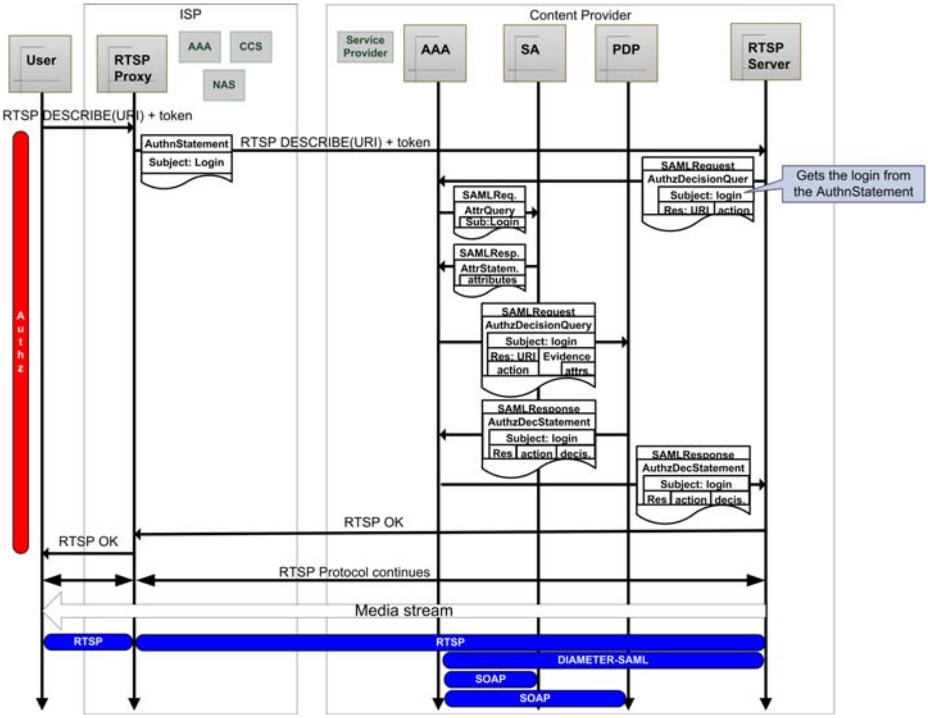


Fig. 4. User authorization process

LAuthenticationQuery must be created in order to insert the login/password pair, as we can see in Figure 3. Once the authentication has been successfully performed by the SA (the authentication module of the AAA server), a security token is generated to indicate that the user signed in, and that there will be no need to reauthenticate the user in this system during a determined period of time. This security token is a digitally-signed *SAML Authentication Statement* which contains a locally unique identifier of the user. This identifier will then be used to obtain user attributes during the authorization step.

6.2 Authorization

Once the user has been authenticated, he is able to request some of the contents included in the list of URIs obtained in the previous step. Using a RTSP client, the user provides the URI and the security token to the RTSP server (this transmission is actually performed through a RTSP proxy that will be explained in the next section).

Prior to initiate the distribution of the multimedia content, the RTSP server must validate that the user has been assigned to an access level that is in accordance with the content being requested (Figure 4). This validation is also performed by the local AAA server, using again the DIAMETER-SAML protocol. Now, the server has to build a *SAMLAuthorizationDecisionQuery* containing the user login and the requested resource. Since the authorization process is performed using the access levels defined

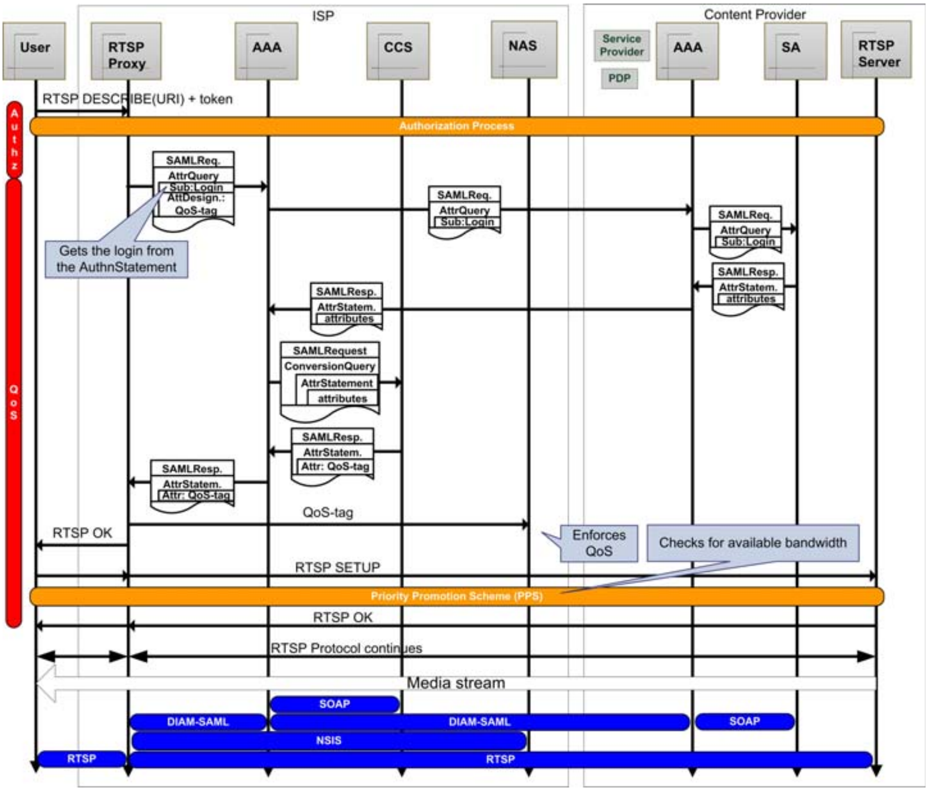


Fig. 5. QoS enforcement process

in the system, the next step is to obtain the user attributes from the SA, that is, the access levels assigned to the user. These attributes will be expressed as SAML sentences and are obtained as explained in [20]. Finally, the attributes are checked against the resource access policy by the PDP and a *SAMLAuthorizationDecisionStatment* is sent to the RTSP server indicating whether the action was approved. When the distribution of the video has been authorized, a RTSP OK message is sent back to the RTSP client in order to go on with the media streaming.

Next, the last step is an optional process that we call *QoS enforcement*. During that step it has to be validated whether the user's ISP can assure the required QoS for watching the video.

6.3 QoS Enforcement

Despite it could be thought that the QoS enforcement is not part of an access control system, we have included this process as an optional part of our proposal. The main idea behind this mechanism is to provide a way to assure that the user will obtain from the ISP the required QoS to watch the video. Moreover, that QoS will be established according to the access level that has been assigned to the user by the service provider.

Therefore, we need a way to obtain the access level in the ISP domain and to translate that level into a particular QoS profile. When this mechanism is used, the reception of the demanded video will be canceled whenever the QoS cannot be fulfilled.

Figure 5 depicts the process. First, using the login included in the security token, the RTSP proxy builds a *SAMLAttributeQuery* in order to obtain the QoS level to enforce. The *AttributeDesignator* field of the attribute query is set to *QoS-tag* to specify the kind of attribute needed. In this way, after recovering the user access level from the CP domain, the AAA server translates this value into the particular QoS profile using the CCS service and the corresponding Conversion Policy. Then, this QoS level is enforced in the NAS to apply the suitable priority to the multimedia stream. Finally, using the priority promotion scheme (PPS), the system must check if enough bandwidth is available to deliver the multimedia content.

7 Implementation Details

This section shows the most important implementation details from the prototype of the VIDIOS sytem presented in the Celtic Event 2006 [2]. This prototype includes initial versions of the service provider, and the RTSP client and server. The service provider authenticates the user and generates the security token. The RTSP server authorizes the user to access to the selected resource and generates a encrypted multimedia streaming. Finally, the RTSP client adds the token to the request to access to the multimedia contenent and receives the encryption key from the RTSP server by means of MIKEY to decrypt the streaming. Besides, the authentication and authorization processes delegate in the NAS-SAML system developed at University of Murcia. The only feature not included still in the system prototype is the QoS enforcement.

Related to the software used to build the VIDIOS prototype, the service provider is implemented using the LAMP framework (Linux + Apache + MySQL + PHP). RTSP/RTP support is obtained from Live555 Streaming Media [4] and MIKEY from the MiniSIP project [5]. About NAS-SAML, the Source Authority, Policy Decision Point and Credential Conversion Service are servlets running in a Tomcat server [1]. SAML and XACML functionality is provided by OpenSAML [7] and SunXACML [8]

Table 1. Software used for testing purposes

Application	Version	Use
Apache	2.0.54	HTTP server
MySQL	4.0.24	Data base
PHP	4.4.0	Dinamic HTML generation
livemedia (RTSP/RTP)	2005.07.14	Multimedia streaming
libmikey (MIKEY)	0.4.1	Key distribution
Tomcat	4.1.30	Servlet container
OpenDIAMETER	1.0.7	AAA infrastructure
OpenSAML	1.0 (C++ and Java)	SAML support
SunXACML	1.2	XACML policies

libraries respectively. And finally, the AAA infrastructure is built using the OpenDIAMETER [6] implementation. Detailed information about specific software versions is provided in Table 1.

8 Related Work

This section describes other works which have informed this proposal. Although the multimedia distribution content has been widely studied in the last years, the main aim of those works has been focused more on the proper distribution media content than on a suitable security infrastructure to protect contents.

The ENTHRONE project [3] proposes an integrated management solution, which covers an entire audio-visual service distribution chain including content generation and protection, distribution across networks and reception at user terminals. The aim is not to unify or impose a strategy on each individual entity of the chain, but to harmonize their functionality, in order to support an end-to-end QoS architecture over heterogeneous networks, applicable to a variety of audio-visual services, which are delivered at the various user terminals. To meet its objectives, the project will rely on an efficient, distributed and open management architecture for the end-to-end delivery chain. The availability and access to resources will be clearly identified, described and controlled all the way along the content distribution chain. The MPEG-21 data model will be used to provide the common support for implementing and managing the functionalities of the resources.

In this system, the user has a set-top box which contains its public ID and an unique secret. This information is used to identify himself in the system and to establish secure channels with other entities to transmit sensible data. When the user wants to access to a specific content, the system recovers the content's license specifying its access control restrictions and checks from the user profile if he has the needed rights to access to this content. Finally, the license, which contains the content decryption key, is transmitted to the set-top box through a secure channel, and the multimedia streaming starts. In this way, the access control is bound with the information contained in the set-top box, limiting the user to only access to the service through this hardware element, and therefore not allowing the user to start two simultaneous multimedia streams, for example from the TV and the computer, which limits the user mobility.

Such as the ENTHRONE project, many other ones related with multimedia streaming, for example TIRAMISU [9], include a full DRM [23] system instead of only access control. The reason why VIDIOS does not include a DRM specification is that this specification is specially oriented to contents which can be downloaded to a device and shared between users, whereas VIDIOS is a streaming service oriented. Anyway, the elements in the VIDIOS architecture can be mapped to a DRM system. That is, in streming media the multimedia contents and the metadata, such as price or quality, are stored in servers, and they are protected by means of encryption in the moment of the delivery to the user. Since we are in a streaming solution, the only right is to play the media streaming, therefore it is not necessary to code it in any DRM language because it is controlled by the Service Provider. The resposible of issuing this right is the administrator when assigns a content to an access level as well as when he assigns users to an access level. Likewise, licenses in this system only express the right to play the contents and contain the encryption key, so it is not necessary this kind of docuement since the encryption key itself can act as license. Thus, the issue of a license would be equivalent to the key distribution process. Finally, since the RTSP client in VIDIOS must be extended to support RTSP and MIKEY, this part of the media player represents the DRM agent which is responsible for reproducing the protected content from a "license" that contains the rights and the encryption key.

9 Conclusions and Future Work

The multimedia distribution content over communication networks requires a service infrastructure able to distribute multimedia resources between end users and Content Providers. One of these proposed infrastructures is the VIDIOS project. We have depicted the main entities of this solution and analyzed the requirements to ensure a secure and trusted communication channel between the involved domains, which could be easily applied to any generic multimedia content distribution system.

Consequently, we propose the NAS-SAML infrastructure in order to address those requirements. In this way, the solution defines how the authentication, authorization and QoS enforcement processes can be defined taking advantage of this previously defined underlying authorization infrastructure. Therefore, we define the communication interfaces and protocols between VIDIOS entities and NAS-SAML services.

It is important to note that no new security protocols, services or entities need to be defined, since NAS-SAML provides an easy way to be extended for high level services.

As a statement of direction, we are working on the definition of NAS-SAML based on the SAML version 2.0, recently accepted as standard version. Moreover, we are planning the integration of other XML technologies such as XKMS.

Acknowledgments

This work has been partially supported by VIDIOS FIT-330220-2005-74.

References

1. Apache tomcat project home page. `http://tomcat.apache.org`.
2. Celtic event 2006 home page. `http://www.celtic-initiative.org/Events/Celtic-Event06/welcome.asp`.
3. *End-to-End QoS through Integrated Management of Content, Networks and Terminals (ENTHRONE)*. `http://www.enthrone.org`. Funded under 5th FWP.
4. Live networks home page. `http://www.live555.com`.
5. MiniSIP project home page. `http://www.minisip.org`.
6. OpenDIAMETER project home page. `http://www.opendiameter.org`.
7. OpenSAML project home page. `http://www.opensaml.org`.
8. SunXACML project home page. `http://sunxacml.sourceforge.net`.
9. *The Innovative Rights and Access Management Inter-platform SolUtion (TIRAMISU)*. `http://www.tiramisu-project.org`. Funded under 6th FWP.

10. *VIdeo DIstribution Over MPLS networks supporting heterogeneous format environments (VIDIOS)*. `http://projects.celtic-initiative.org/vidios`.
11. J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman. *MIKEY: Multimedia Internet KEYing*, August 2004. RFC 3830.
12. J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman. *Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)*, June 2005. IETF Draft.
13. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *An architecture for Differentiated Services*, December 1998. RFC 2475.
14. O. Cánovas, G. Lopez, and A.F. Gómez-Skarmeta. A credential conversion service for saml-based scenarios. In *Proceedings First European PKI Workshop*, volume 3093 of *Lecture Notes in Computer Science*, pages 297–305. Springer, 2004.
15. C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, and D. Spence. *Generic AAA Architecture*, August 2000. RFC 2903.
16. D. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, and R. Chandramouli. Proposed nist standard for role-based access control. *ACM Transaction on Information and System Security*, 4(3), 2001.
17. P. Jayarama, R. López, Y. Ohba, M. Parthasarathy, and A. Yegin. *PANA Framework*, 2005. IETF Draft.
18. LAN MAN Standards Committee of the IEEE Computer Society. *IEEE Draft P802.1X/D11: Standard for Port based Network Access Control*, March 2001.
19. G. López, O. Cánovas, and A. F. Gómez. Use of xacml policies for a network access control service. In *Proceedings 4th International Workshop for Applied PKI, IWAP 05*, pages 111–122. IOS Press, 2005.
20. G. López, O. Cánovas, A. F. Gómez, J. D. Jimenez, and R. Marín. A network access control approach based on the aaa architecture and authorzation attributes. *Journal of Network and Computer Applications JNCA*, 2006. To be published.
21. Gabriel López, Óscar Cánovas, Antonio F. Gómez-Skarmeta, Sassa Otenko, and David Chadwick. A heterogeneos network access service based on permis and saml. In *Proceedings 2nd European PKI Workshop*, volume 3545 of *Lecture Notes in Computer Science*, pages 55–72. Springer, 2005.
22. N. Morita and G. Karlsson. *Framework of Priority Promotion Scheme*, October 2003. IETF Draft.
23. Open Mobile Alliance. *DRM specification*, April 2004. Draft Version 2.0.
24. M. Sanchez, G. Lopez, O. Cánovas, and A.F. Gómez-Skarmeta. *Grid Authorization Based on Existing AAA Architectures*, 2006. Submitted to The Fourth International Workshop on Security In Information Systems WOSIS-2006.
25. H. Schulzrinne, A. Rao, and R. Lanphier. *Real Time Streaming Protocol (RTSP)*, April 1998. RFC 2326.

Efficient Conjunctive Keyword Search on Encrypted Data Storage System[*]

Jin Wook Byun, Dong Hoon Lee, and Jongin Lim

Center for Information Security Technologies (CIST),
Korea University, Anam Dong, Sungbuk Gu, Seoul, Korea
{byunstar, donghlee, jilim}@korea.ac.kr

Abstract. We study *conjunctive* keyword search scheme allowing for remote search of data containing each of several keywords on encrypted data storage system. A data supplier first uploads encrypted data on a storage system, and then a user of the storage system searches data containing keywords over encrypted data hence insider (such as an administrator of the storage system) and outsider attackers do not learn anything else about the data. Recently, Golle *et al.* first suggested conjunctive keyword search scheme, but the communication and storage costs linearly depend on the number of stored data in the database, hence it is not really suitable for a large scale database.

In this paper, we propose an efficient conjunctive keyword search scheme over encrypted data in aspects of communication and storage costs. Concretely, we reduce the storage cost of a user and the communication cost between a user and a data supplier to the constant amounts. We formally define security model for a conjunctive keyword search scheme and prove that the proposed scheme is secure under the decisional bilinear Diffie-Hellman (DBDH) assumption in the random oracle model.

Keywords: Conjunctive keyword search over encrypted data, database security and privacy.

1 Introduction

As the amount of information to be stored and managed on the Internet rapidly increases, protecting data in a database from outsider/insider attackers has been hot issues in a secure database management system. The most simple solution to prevent theft and misuse of data from outsider/insider attackers is that a user of storage system simply encrypts personal data with his own private key, and stores the encrypted results on the storage system. The user should also manage his encryption key securely without revealing it to the outsider/insider attackers. However, secure encryption makes data look random, and unreadable to anyone

[*] This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 184–196, 2006.

other than the users holding the encryption keys, hence the server is unable to determine which encrypted data contain specific keywords. And then, how can original documents be efficiently searched including the user specific keywords over the encrypted documents? Intuitively, one may think a trivial search process that the user downloads all documents and decrypt them with his secret key, and searches documents containing specific keywords on the user's machine. As one can easily see, this process is very inefficient and would impose massive burdens on the user side as stored documents rapidly increase. To resolve this problem, there has been much research on efficient and secure keyword search over the encrypted documents based on the various scenarios [1, 3, 4, 5, 7, 8, 10, 11, 12, 13].

1.1 Related Works and Our Contributions

In this paper, we consider a *conjunctive keyword search* [7, 12] which finds data containing each of several keywords by asking one query. One may argue that a conjunctive keyword search scheme can be built from the multiple executions of any single keyword search scheme. In this case, however, the server should find all data containing each keyword by using the single keyword search, check the intersection set of all data, then return the results to the user. This approach requires high computation cost and redundancy to the server due to duplicated comparisons and search.

A conjunctive keyword search scheme over encrypted data consists of three entities: a data supplier, a storage system such as a database, and a user of storage system. A data supplier uploads encrypted data on a storage system, and then a user of the storage system searches data containing keywords. Let's suppose an untrusted web-based personal storage (PS) system in which a user of personal storage system oneself may store encrypted data over the server and search data containing appropriate keywords on the encrypted data. Many schemes [13, 6, 8, 7] have been suggested in this setting by using only symmetric cryptography such as block, stream cipher, and Bloom filter. Song *et al.* suggested an efficient and provably secure keyword search scheme by using stream and block cipher [13]. In [8], Goh suggested a secure search scheme using a Bloom filter [2]. Very recently, Chang and Mitzenmacher also suggested a more practical keyword search protocol in terms of communication and storage overheads. However, these schemes are not appropriate for fully conjunctive keyword search. As pointed out in [6, 8], the design of conjunctive keyword search scheme using only symmetric cryptography still remains as a challenging open problem. Recently, to provide a conjunctive keyword search in this setting, Golle *et al.* applied public key cryptography to the keyword search scheme, and first proposed two secure conjunctive keyword search protocols over encrypted data.

However, the one of schemes is very inefficient in aspect of communication and storage costs, as analyzed in Table 1. That is, the costs linearly depend on the number of stored data, and the scheme requires huge communication and computation costs in case of a large scale database. For example, MS SQL Server™ 2005 Edition has at most 1,048,516 TBytes size [9], and if we suppose that one record requires about 10 MBytes then the server has at most 104,851,600,000

Table 1. Comparison with Golle *et al.*'s scheme

Scheme	TNR	TSC	NS	SM	CON
Golle *et al.* I [7]	1	$\boxed{(n+1)\lvert q\rvert+\log m}$	$(n+2)\lvert q\rvert$	RO	Y
Golle *et al.* II [7]	1	$3\lvert q\rvert+\log m$	$2\lvert q\rvert$	$\boxed{\mathsf{NST}}$	Y
ECKS-PS	1	$2\lvert q\rvert+\log m$	$2\lvert q\rvert$	RO	Y

$*$ The square box means shortcoming of Golle *et al*'s scheme.
$*$ m : The number of fields in database, n : The number of rows in database
$*$ TNR : The total number of rounds
$*$ TSC : The total size of communication between user and database
$*$ NS : The number of secret keys for a user
$*$ SM : Security model used (RO : Random oracle model, ST : Standard model)
$*$ NST : Nonstandard assumption.
$*$ CON : Ability to provide conjunctive keyword search (Y : Yes, N : No)
$*$ q : A large prime order of a group $\mathbb{G}_1$

records. In this case, the communication size of [7] for a one conjunctive keyword search is surprisingly about $104,851,600,000 * 1024$ bits $= 13,421,004,800,000$ Bytes $\simeq$ 13 TBytes where the number of fields are supposed to be 100.[1] The storage amounts for a user is also about 13 TBytes. No companies would like to take this solution to preserve privacy of their storage systems.[2] In addition, the other scheme of Golle *et al.* is based on a new assumption which is not proved by standard assumption such as computational Diffie-Hellman or decisional Diffie-Helmman. Therefore we cannot convince that the scheme is really secure or not. Indeed, one can design many schemes relying on unverifiable assumptions as many as one wants. First of all, difficulties of design of a secure protocol come from that we should design cryptographic protocols under the standard assumptions or show that reasonable relationship between new assumption and the well-known standard assumptions. We have the following contributions.

- **Constant communication and storage overheads.** In this paper, we design an efficient conjunctive keyword search in the personal storage system such as web-hard where users themselves manage their own storage in the server (for short, we call the scheme ECKS-PS). Surprisingly, the storage cost of a user and the communication cost between a user and a server in the scheme are constant. It means that our scheme does not linearly depend on the size of the stored data. For instance, although the value of n is grower up to GBytes or TBytes, our scheme only requires at most 2,052 bits to perform a one conjunctive keyword search where q is 1,024 bits and m is 10. The storage amount is only 2,048 bits.

[1] As shown in Table 1, we note that the number of fields are not important factor of total communication costs. Even if we have 100 fields in a table, the costs for fields is just log100 $\simeq$ 10 bits.

[2] Even if we assume that one record has 100 MBytes (this is the rare case), the scheme of [7] needs about 1 TBytes to query a one query.

- **Formal Security Proof.** We support the proposed $\mathsf{ECKS\text{-}PS}$ scheme with formal security model and security proof. Our security relies on a new multi decisional bilinear Diffie-Hellman (MDBDH) assumption. We also prove that the MDBDH assumption is an equivalent version of the well-known decisional bilinear Diffie-Hellman (DBDH) assumption.

1.2 Organization

In Section 2, we define security model of conjunctive keyword search scheme and its security definition. In Section 3, we present an efficient conjunctive keyword search scheme in the personal storage system ($\mathsf{ECKS\text{-}PS}$) and prove that $\mathsf{ECKS\text{-}PS}$ is secure in the random oracle model. Lastly, we conclude in Section 4.

2 Conjunctive Keyword Search Scheme and Its Security Definition

We consider a simple database which has several records, each of which contains fields. This database can be viewed as a two-dimensional table where a record is a row and each row has several fields. We assume that there are m keyword fields for each encrypted document. In this paper, we assume the followings as in [7]: the same keyword never appears in two different keyword fields and every keyword field is defined for every document.

Let n be the total number of documents, and we have n rows in the database. For each row R_i ($1 \leq i \leq n$), we define the i-th document by $D_i = \{W_{i,1}, ..., W_{i,m}\}$ where $W_{i,j}$ ($1 \leq j \leq m$) is the j-th keyword of document D_i. The row R_i consists of an encrypted document and conjunctive searchable information (CSI) CSI_i where $\mathsf{CSI}_i = \{I_i, \mathsf{CSI}_{i,1}(W_{i,1}), ..., \mathsf{CSI}_{i,m}(W_{i,m})\}$ for m keyword fields. I_i is an additional information needed for conjunctive keyword search. For $1 \leq j \leq m$, the $\mathsf{CSI}_{i,j}(W_{i,j})$ is the corresponding searchable information of the $W_{i,j}$ and it is stored on the j-th keyword field of the i-th row.

2.1 Conjunctive Keyword Search Scheme

[**Personal Storage System**]. In this setting, users of the server (data supplier) upload their sensitive data in an encrypted form, and retrieve the encrypted data containing the specific conjunctive keywords. We assume that the data are encrypted by a standard symmetric encryption algorithm $E_K(\cdot)$ where K is a secret key. A conjunctive keyword search in the personal storage system consists of the following four polynomial time algorithms.

- Key generation algorithm $\mathsf{KeyGen}(1^k)$: It takes as an input a security parameter k, and outputs a private/public key pair (prk, pk) for a user of server.
- Conjunctive searchable information (CSI) algorithm $\mathsf{CSI}(prk,\ pk,\ D_i)$: It takes as inputs a user's private key prk, a public key pk, and a data $D_i = \{W_{i,1}, ..., W_{i,m}\}$, and outputs conjunctive searchable information $\mathsf{CSI}_i = \{I_i(prk, pk), \mathsf{CSI}_{i,1}(W_{i,1}, pk), ..., \mathsf{CSI}_{i,m}(W_{i,m}, pk)\}$ where I_i is an additional information needed for conjunctive keyword search, and $\mathsf{CSI}_{i,j}(W_{i,j}, pk)$ is the corresponding searchable information of $W_{i,j}$ for $1 \leq j \leq m$.

- Trapdoor generation algorithm for conjunctive queries $\mathsf{TCK}(prk,\ pk,\ p_1, ..., p_l,\ Q_l)$: For $1 \leq l \leq m$, it takes as inputs a private key prk, a list of names of target keyword fields in the database, and the corresponding l conjunctive keywords $Q_l = \{W_{p_1}, ..., W_{p_l}\}$. It outputs a trapdoor T_l for Q_l.
- Test algorithm $\mathsf{Test}(\mathsf{CSI}_i,\ T_l)$: It takes as inputs the conjunctive searchable information $\mathsf{CSI}_i = \{I_i(prk, pk), \mathsf{CSI}_{i,1}(W_{i,1}, pk), .., \mathsf{CSI}_{i,m}(W_{i,m}, pk)\}$ and the trapdoor T_l=$\mathsf{TCK}(prk, p_1, ..., p_l,\ Q_l = \{W_{p_1}, ..., W_{p_l}\})$. It outputs '`Yes`' if the condition $(W_{i,p_1} = W_{p_1}) \wedge ... \wedge (W_{i,p_l} = W_{p_l})$ holds, and '`No`' otherwise.

We illustrate the framework of conjunctive keyword search scheme in Figure 1.

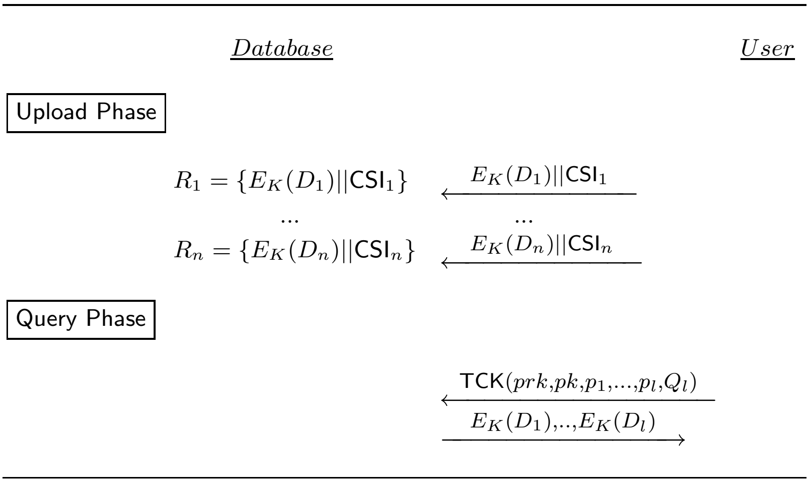

Database
User
Upload Phase
$R_1 = \{E_K(D_1)||\mathsf{CSI}_1\}$
$E_K(D_1)||\mathsf{CSI}_1$
...
...
$R_n = \{E_K(D_n)||\mathsf{CSI}_n\}$
$E_K(D_n)||\mathsf{CSI}_n$
Query Phase
$\mathsf{TCK}(prk,pk,p_1,...,p_l,Q_l)$
$E_K(D_1),..,E_K(D_l)$

Fig. 1. A CKS protocol in the personal storage system

Definition 2.1 [SS-CTA Security in the PS Setting]. Let ECKS-PS = ($\mathsf{KeyGen}(1^k)$, $\mathsf{CSI}(prk,pk,D_i)$, $\mathsf{TCK}(prk,pk,p_1, ..., p_l,\ Q_l)$ $\mathsf{Test}(\mathsf{CSI}_i,\ T_l)$) be an efficient conjunctive keyword search scheme in the PS setting and $\mathcal{A}$ be an adversary of ECKS-PS. We imagine an adversary $\mathcal{A}$ run in an experiment. During the experiment, $\mathcal{A}$ may access trapdoor oracle $\mathcal{O}_T(\cdot)$ and CSI oracle $\mathcal{O}_C(\cdot)$, adaptively. That is, $\mathcal{A}$ can adaptively ask for trapdoor and CSI of his choice with the restriction that $\mathcal{A}$ may not ask for trapdoor and CSI distinguishing two target documents. The adversary chooses two documents D_0 and D_1, and then gets a challenge CSI_b of D_b for a random bit b. It also retains some state information s. Note that the adversary is not allowed to ask for a trapdoor that is distinguishing D_0 and D_1. At some point, the adversary must say which CSI_b was chosen. That is, the main goal of the adversary is to distinguish between CSIs of two documents D_0 and D_1. The adversary wins if she correctly identifies which data goes with CSI_b. The notations of $\mathcal{A}^{\mathcal{O}_T(\cdot)}$ and $\mathcal{A}^{\mathcal{O}_C(\cdot)}$ denote the adversary $\mathcal{A}$ can access trapdoor oracle $\mathcal{O}_T(\cdot)$ and CSI oracle

$\mathcal{O}_C(\cdot)$, respectively. With this knowledge, we now formally define a security notion of semantic security against chosen trapdoor attack (for short, SS-CTA).

Experiment $\mathbf{Exp}_{\mathcal{A}}^{cta}(k)$

1. $(prk, pk) \overset{R}{\leftarrow} \mathsf{KeyGen}(1^k)$
2. $\{D_0 = \{W_{0,1}, .., W_{0,m}\}, D_1 = \{W_{1,1}, .., W_{1,m}\},\ s\} \leftarrow \mathcal{A}^{\mathcal{O}_T(\cdot)}(pk)$
3. $b \overset{R}{\leftarrow} \{0, 1\}$
4. $d \leftarrow \mathcal{A}^{\mathcal{O}_T(\cdot), \mathcal{O}_C(\cdot)}(\mathsf{CSI}_b,\ s)$
   where $\mathsf{CSI}_b = (I_b(prk, pk), \mathsf{CSI}_{b,1}(W_{b,1}, pk), ..., \mathsf{CSI}_{b,m}(W_{b,m}, pk))$
5. If $b = d$ then return 1. Otherwise, return 0.

We define a *ss-cta* advantage of $\mathcal{A}$ as follows.

$$\mathbf{Adv}_{\mathcal{A}}^{cta}(k, q_T, q_C) = |Pr[\mathbf{Exp}_{\mathcal{A}}^{cta}(k) = 1 | b = 1] - Pr[\mathbf{Exp}_{\mathcal{A}}^{cta}(k) = 1 | b = 0]|.$$

We say that a ECKS-PS scheme is SS-CTA secure if the above advantage is negligible for any polynomial time adversary $\mathcal{A}$.

Definition 2.2 [Admissible Bilinear Map]. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of a large prime order q and g is a generator of $\mathbb{G}_1$. We call $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ an *admissible bilinear* map if it satisfies the following properties: (1) Bilinear : $e(u^a, v^b) = e(u, v)^{ab}$ where $u, v \in \mathbb{G}_1$ and $a, b \in Z_q^*$; (2) Non-degenerate : e does not send all pairs of points in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$. If g is a generator of $\mathbb{G}_1$ then $e(g, g)$ is a generator of $\mathbb{G}_2$; (3) Computable : for all $u, v \in \mathbb{G}_1$, the map $e(u, v)$ is efficiently computable.

Definition 2.3 [Decisional Bilinear Diffie-Hellman (DBDH) Assumption]. We first define DBDH parameter generator as follows.

- *DBDH Parameter Generator.* A DBDH parameter generator $\mathcal{IG}_{DBDH}$ is a probabilistic polynomial time (PPT) algorithm that takes a security parameter k, runs in polynomial time, and outputs the description of two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same order q and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$.
- *DBDH Problem.* The DBDH problem is to distinguish between tuples of the form $(g, g^a, g^b, g^c, e(g, g)^{abc})$ and $(g, g^a, g^b, g^c, e(g, g)^d)$ for a random $g \in \mathbb{G}_1$, and $a, b, c, d \in \mathbb{Z}_q^*$. Let $\mathcal{A}_{\mathcal{D}}$ be a DBDH adversary trying to distinguish the above two distributions within polynomial time $T_{\mathcal{D}}$. More formally, let's consider the experiments.

Experiment $\mathbf{Exp}_{\mathcal{A}_{\mathcal{D}}}^{real}(k)$	Experiment $\mathbf{Exp}_{\mathcal{A}_{\mathcal{D}}}^{rand}(k)$
$(\mathbb{G}_1, \mathbb{G}_2, e) \leftarrow \mathcal{IG}_{DBDH}(k)$	$(\mathbb{G}_1, \mathbb{G}_2, e) \leftarrow \mathcal{IG}_{DBDH}(k)$
$g \leftarrow \mathbb{G}_1; a, b, c \leftarrow \mathbb{Z}_q^*$	$g \leftarrow \mathbb{G}_1; a, b, c, d \leftarrow \mathbb{Z}_q^*$
$W = g^a, X = g^b, Y = g^c$	$W = g^a, X = g^b, Y = g^c$
$Z = e(g, g)^{abc}$	$Z = e(g, g)^d$
$b \leftarrow \mathcal{A}_{\mathcal{D}}(W, X, Y, Z)$	$b \leftarrow \mathcal{A}_{\mathcal{D}}(W, X, Y, Z)$
return b	return b

We define an advantage of $\mathcal{A}_\mathcal{D}$ in solving the **DBDH** problem as follows.

$$\mathbf{Adv}^{dbdh}_{\mathcal{A}_\mathcal{D}}(T_\mathcal{D}, k) = |Pr[\mathbf{Exp}^{real}_{\mathcal{A}_\mathcal{D}}(k) = 1] - Pr[\mathbf{Exp}^{rand}_{\mathcal{A}_\mathcal{D}}(k) = 1]|.$$

- *DBDH Assumption.* We say that the DBDH assumption holds in $\mathbb{G}_1$ if no polynomial time algorithm has negligible advantage ϵ in solving the DBDH problem in $\mathbb{G}_1$.

Definition 2.4 [Multi-Decisional Bilinear Diffie-Hellman (MDBDH) Assumption]. A MDBDH parameter generator is same to the DBDH parameter generator.

- *MDBDH Problem.* MDBDH problem is to distinguish between following MDBDH tuples where a generator $g \in \mathbb{G}_1$, and $a, b, c_1, ..., c_m, r_1, ..., r_m \in \mathbb{Z}_q^*$

$$\mathbf{MDBDH}_{real} = (g, g^a, g^b, g^{c_1}, ..., g^{c_m}, e(g, g)^{abc_1}, ..., e(g, g)^{abc_m})$$
$$\mathbf{MDBDH}_{rand} = (g, g^a, g^b, g^{c_1}, ..., g^{c_m}, e(g, g)^{r_1}, ..., e(g, g)^{r_m})$$

Let $\mathcal{A}^M_\mathcal{D}$ be a **MDBDH** adversary trying to distinguish the above two distributions within polynomial time $T_\mathcal{D}$. More formally, let's consider the experiments.

```
Experiment $\mathbf{Exp}^{real}_{\mathcal{A}^M_\mathcal{D}}(k)$
$(\mathbb{G}_1, \mathbb{G}_2, e) \leftarrow \mathcal{IG}_{MDBDH}(k)$
$b \leftarrow \mathcal{A}^M_\mathcal{D}(\mathbf{MDBDH}_{real})$
return $b$
```

```
Experiment $\mathbf{Exp}^{rand}_{\mathcal{A}^M_\mathcal{D}}(k)$
$(\mathbb{G}_1, \mathbb{G}_2, e) \leftarrow \mathcal{IG}_{MDBDH}(k)$
$b \leftarrow \mathcal{A}^M_\mathcal{D}(\mathbf{MDBDH}_{rand})$
return $b$
```

We define an advantage of $\mathcal{A}^M_\mathcal{D}$ in solving the **MDBDH** problem as follows.

$$\mathbf{Adv}^{mdbdh}_{\mathcal{A}^M_\mathcal{D}}(T^M_\mathcal{D}, k) = |Pr[\mathbf{Exp}^{real}_{\mathcal{A}^M_\mathcal{D}}(k) = 1] - Pr[\mathbf{Exp}^{rand}_{\mathcal{A}^M_\mathcal{D}}(k) = 1]|.$$

- *MDBDH Assumption.* We say that the MDBDH assumption holds in $\mathbb{G}_1$ if no polynomial time algorithm has negligible advantage ϵ in solving the MDBDH problem in $\mathbb{G}_1$.

Next, we show that the MDBDH assumption is an equivalent assumption of the DBDH assumption.

Lemma 2.1 For any integer m and common parameters $(\mathbb{G}_1, \mathbb{G}_2, e, q, g)$,

$$(1)\ \mathbf{Adv}^{mdbdh}_{\mathcal{A}^M_\mathcal{D}}(T^M_\mathcal{D}, k) \leq (m-1)\mathbf{Adv}^{dbdh}_{\mathcal{A}_\mathcal{D}}(T^M_\mathcal{D} + 2mT_{\mathbb{G}_1}, k)$$
$$(2)\ \mathbf{Adv}^{dbdh}_{\mathcal{A}_\mathcal{D}}(T_\mathcal{D}, k) \leq \mathbf{Adv}^{mdbdh}_{\mathcal{A}^M_\mathcal{D}}(T_\mathcal{D} + 2mT_{\mathbb{G}_1}, k)$$

where $T_{\mathbb{G}_1}$ is the computational time for an exponentiation in $\mathbb{G}_1$.

Proof. It is straightforward to derive the first result that MDBDH problem implies DBDH problem by using a hybrid argument. For the second result, we construct MDBDH breaker algorithm Δ_{mdbdh} from DBDH breaker algorithm

Δ_{dbdh} by using random self reducibility. Δ_{mdbdh} first takes input as a tuple $(u_1 = g^a, u_2 = g^b, u_{3,1} = g^{c_1}, ..., u_{3,m} = g^{c_m}, u_{4,1} = e(g,g)^{abr_1}, ..., u_{4,m} = e(g,g)^{abr_m})$ where all values of r_i are different (namely, MDBDH_{rand}) or same to c_i, respectively, (namely, MDBDH_{real}) for $1 \leq i \leq m$. Δ_{mdbdh} chooses random numbers $v_1, v_2, ..., v_m$, and computes inputs of Δ_{dbdh} as follows.

$$\begin{aligned}
&\alpha_m = u_1, \quad \beta_m = u_2 \\
&\gamma_m = (u_{3,1})^{v_1} \times ... \times (u_{3,m})^{v_m} = g^{c_1v_1 + c_2v_2 + ... + c_mv_m} = g^{A_m} \\
&\delta_m = (u_{4,1})^{v_1} \times ... \times (u_{4,m})^{v_m} = e(g,g)^{abr_1v_1 + ... + abr_mv_m} \\
&\quad\; = e(g,g)^{abA_m + (r_1 - c_1)abv_1 + ... + (r_m - c_m)abv_m}
\end{aligned}$$

Δ_{mdbdh} outputs b according to the output bit of Δ_{dbdh}. For all $1 \leq i \leq m$, if $r_i = c_i$, then the generated tuple $(\alpha_m, \beta_m, \gamma_m, \delta_m)$ is a real DBDH tuple, otherwise, it is a random DBDH tuple. Hence, Δ_{mdbdh} inherits the success probability of Δ_{dbdh} with time cost for $2m$ exponentiations. □

3 Efficient Conjunctive Keyword Search Protocol

In this section, we present an efficient conjunctive keyword search protocol in the personal storage system (ECKS-PS). ECKS-PS is a SS-CTA secure protocol relying on the random oracle assumption.

3.1 A SS-CTA Secure ECKS-PS Based on Random Oracle Assumption

We efficiently design conjunctive keyword search by applying two private keys and bilinear maps to generate conjunctive searchable information and trapdoor query. Remarkably, our ECKS-PS provides conjunctive keyword search requiring only constant communication size per a trapdoor query. We use an ideal hash function $H : \{0,1\}^* \to \mathbb{G}_1$. The ECKS-PS protocol works as follows.

- $\mathsf{KeyGen}(1^k)$: It takes a security parameter k, and determines two groups $\mathbb{G}_1$ and $\mathbb{G}_2$. It chooses $\alpha, \theta \in Z_q^*$ and a generator g of $\mathbb{G}_1$, then outputs a public key $pk = (g, y = g^\alpha)$ and private key $prk = (\theta, \alpha) \in Z_q^*$.
- $\mathsf{CSI}(prk, pk, D_i)$: It takes as inputs a private key, a public key and a data $D_i = \{W_{i,1}, ..., W_{i,m}\}$. It first chooses a random value $a_i \in Z_q^*$ and outputs CSI_i as follows.

$$\mathsf{CSI}_i = \begin{cases} I_{i,1}(prk, pk) = g^{a_i\theta}, I_{i,2}(prk, pk) = g^{a_i} \\ \mathsf{CSI}_{i,1}(W_{i,1}, pk) = e(y, H(W_{i,1})^{a_i}), .., \\ \mathsf{CSI}_{i,m}(W_{i,m}, pk) = e(y, H(W_{i,m})^{a_i}). \end{cases}$$

- $\mathsf{Trapdoor}(prk, pk, p_1, .., p_l, Q_l)$: For $1 \leq l \leq m$, it outputs $T_l = [A, B, p_1, .., p_l]$ where $A = (H(W_{i,p_1}) \times ... \times H(W_{i,p_l}))^\alpha \cdot g^{\theta r}$ and $B = g^r$ using private key α, θ, a random value $r \in Z_q^*$, the list of names of keyword fields, and conjunctive keywords $Q_l = \{W_{i,p_1}, .., W_{i,p_l}\}$.

- $\mathsf{Test}(\mathsf{CSI}_i, T_l)$: For all $1 \leq i \leq n$, it tests if
$$\frac{e(I_{i,2}, A)}{\mathsf{CSI}_{i,p_1}(W_{i,p_1}, pk) \times .... \times \mathsf{CSI}_{i,p_l}(W_{i,p_l}, pk)} = e(I_{i,1}, B).$$
If so, it outputs 'Yes'. Otherwise, it outputs 'No'.

3.2 Security Proof of ECKS-PS

The following theorem shows that the security of ECKS-PS is reduced to the problem of DBDH under the random oracle assumption.

Theorem 3.1 *Let $\mathcal{A}$ be a polynomial time adversary which tries to break the semantic security of the proposed ECKS-PS using a chosen trapdoor attack, adaptively. $\mathcal{A}$ asks at most q_h hash queries, q_T trapdoor queries, q_C CSI queries within a polynomial time bound T. Suppose that MDBDH problem is hard for $\mathbb{G}_1$, then the proposed ECKS-PS is SS-CTA secure against $\mathcal{A}$. Concretely, the advantage for $\mathcal{A}$ is*

$$\mathbf{Adv}^{cta}_{\mathcal{A}}(k, T, q_T, q_C, q_h) \leq 2e^m \cdot (q_T + 1)^{2q_C+1} \cdot \mathbf{Adv}^{mdbdh}_{\Delta_{mdbdh}}(k, T^M_{\mathcal{D}}) + \frac{2}{q^m}$$

where $T^M_{\mathcal{D}} \geq T + (q_h + q_T + q_C)mT_{\mathbb{G}_1}$. $T_{\mathbb{G}_1}$ denotes the computing time for an exponentiation in $\mathbb{G}_1$. The constant e is base of the natural logarithm.

Proof. Suppose that $\mathcal{A}$ breaks the proposed ECKS-PS with non-negligible probability ϵ. By using $\mathcal{A}$ as a subroutine, we construct an algorithm Δ_{mdbdh} that solves MDBDH problem with probability at least $\epsilon' = \frac{1}{2e^m(q_T+1)^{2q_C+1}}(\epsilon - \frac{2}{q^m})$. Δ_{mdbdh} starts by taking input as a tuple $(g, u_1 = g^s, u_{2,1} = g^{t_1}, .., u_{2,m} = g^{t_m}, u_3 = g^u, u_{4,1} = e(g,g)^{s \cdot t_1 \cdot u}, .., u_{4,m} = e(g,g)^{s \cdot t_m \cdot u}))$.

Algorithm Δ_{mdbdh}

- KeyGen: Δ_{mdbdh} sets $\mathcal{A}$'s public key to be $pk = [g, u_1]$. Δ_{mdbdh} selects a secret value $\theta \in Z^*_q$ and set one of $\mathcal{A}'s$ private key to be $prk = \theta$.
- Hash queries: In order to respond hash queries of H, it is necessary to maintain a list of tuples $<W_{i,j}, h_{i,j}, a_{i,j}, c_{i,j}>$. We call this list as H-list. The hash oracle H is controlled by Δ_{mdbdh} as follows.
  – If the adversary $\mathcal{A}$ asks a hash query $W_{i,j}$ such that it appears on the H-list in a tuple $<W_{i,j}, h_{i,j}, a_{i,j}, c_{i,j}>$, then Δ_{mdbdh} returns $H(W_{i,j}) = h_{i,j}$.
  – Otherwise, Δ_{mdbdh} picks a random coin $c_{i,j}$ satisfying $Pr[c_{i,j} = 0] = \frac{1}{q_T+1}$. If $c_{i,j} = 0$, then Δ_{mdbdh} makes $h_i \leftarrow (u_3)^{a_{i,j}} \in \mathbb{G}_1$ for a random number $a_{i,j} \in Z^*_q$. If $c_{i,j} = 1$, then Δ_{mdbdh} makes $h_i \leftarrow g^{a_{i,j}} \in \mathbb{G}_1$.
  – Δ_{mdbdh} adds the tuple $<W_{i,j}, h_{i,j}, a_{i,j}, c_{i,j}>$ to the H-list. When $\mathcal{A}$ later asks the same query $W_{i,j}$, then Δ_{mdbdh} responds with $H(W_{i,j}) = h_{i,j}$.
- Trapdoor queries: When $\mathcal{A}$ asks a trapdoor query of $Q_{i,l} = \{W_{i,p_1}, ..., W_{i,p_l}\}$ for $1 \leq l \leq m$, Δ_{mdbdh} outputs a trapdoor $T_{i,l}$ for $Q_{i,l}$ as follows.

- Δ_{mdbdh} first obtains $H(W_{i,p_j}) = h_{i,p_j}$ for $1 \leq j \leq l$ by executing the above hash simulation. Let $<W_{i,p_j}, h_{i,p_j}, a_{i,p_j}, c_{i,p_j}>$ be the corresponding tuple on the H-list. If all c_{i,p_j} $(1 \leq j \leq l)$ on the tuples are not 1, then Δ_{mdbdh} reports failure and terminates.
- If $c_{i,p_j} = 1$, Δ_{mdbdh} defines $\xi_{i,p_j} = u_1^{a_{i,p_j}}$. We note that $\xi_{i,p_j} = H(W_{i,p_j})^s$. Δ_{mdbdh} selects a random value $r \in \mathbb{G}_1$, and computes $A = (\xi_{i,p_1} \times ..... \times \xi_{i,p_l}) \cdot g^{\theta r} = (H(W_{i,p_1}) \times ..... \times H(W_{i,p_l}))^s \cdot g^{\theta r}$, $B = g^r$ by using the private key prk. Since s is a secret key (which is correspondent with a private key α unknown to $\mathcal{A}$), we can check that A and B are valid trapdoors for a some row i, as follows.

$$\frac{e(g^{a_i}, (H(W_{i,p_1}) \times ... \times H(W_{i,p_l}))^s \cdot g^{\theta r})}{e(g^s, H(W_{i,p_1})^{a_i}) \times ... \times e(g^s, H(W_{i,p_l})^{a_i})} = e(g^{\theta a_i}, g^r).$$

Δ_{mdbdh} answers A and B for the trapdoor query of $Q_{i,l}$.

- Challenge queries: $\mathcal{A}$ creates two challenging data $D_0 = \{W_{0,1}, .., W_{0,m}\}$ and $D_1 = \{W_{1,1}, .., W_{1,m}\}$.
  - For a selected data $D_b = \{W_{b,1}, .., W_{b,m}\}$, Δ_{mdbdh} obtains $H(W_{b,j}) = h_{b,j}$ for $1 \leq j \leq m$ by executing the above hash simulation. Let $<W_{b,j}, h_{b,j},$
    $a_{b,j}, c_{b,j}>$ be the corresponding tuple on the H-list. If $c_{0,j} = 1$ or $c_{1,j} = 1$ for $1 \leq j \leq m$, then Δ_{mdbdh} reports failure and aborts. Thus, at least one of $c_{0,j}$, $c_{1,j}$ are equal to 0 for all $1 \leq j \leq m$. If all $c_{0,j}$ and $c_{1,j}$ are equal to 0 for all $1 \leq j \leq m$, then Δ_{mdbdh} randomly selects a bit b such that $c_{b,j} = 0$. Otherwise if all $c_{b,j} = 0$ for a specific bit b then no random selection is needed.
  - Δ_{mdbdh} responds with $\mathsf{CSI}_b = [I_{i,1}(prk, pk) = (u_3)^\theta, I_{i,2}(prk, pk) = u_3$, $u_{4,1} = e(u_1, h_{b,1})$,..,$u_{4,m} = e(u_1, h_{b,m})]$ for a private key θ. We note that if $u_{4,1}, .., u_{4,m}$ are $e(g,g)^{s \cdot t_1 \cdot u}$,..,$e(g,g)^{s \cdot t_m \cdot u}$, respectively, then the challenge CSI_b is a real MDBDH tuple. On the other hand, if $u_{4,1}, .., u_{4,m}$ are uniform and independent in $\mathbb{G}_1$, the challenge CSI_b is a random MDBDH tuple.
- CSI queries: The answers for CSI queries are simple. When asked to compute CSI of $D = \{W_{i,1}, .., W_{i,l}\}$, Δ_{dbdh} first computes hash value of each keyword $H(W_{i,j}) = h_{i,j}$ for $1 \leq j \leq l$. Let $<W_{i,j}, h_{i,j}, a_{i,j}, c_{i,j}>$ be the corresponding tuple on the H-list. If all $c_{i,j}$ $(1 \leq j \leq l)$ on the tuples are not 1, then Δ_{mdbdh} reports failure and terminates. Δ_{dbdh} responds with $\mathsf{CSI} = [I_{i,1} = g^{a_i \theta}$, $I_{i,2} = g^{a_i}$, $\mathsf{CSI}_{i,1}(W_1, pk) = e(u_1, h_{i,1}^{a_i}), ..., \mathsf{CSI}_{i,l}(W_l, pk) = e(u_1, h_{i,l}^{a_i})]$ for a random $a_i \in Z_q^*$
- More Trapdoor queries and CSI queries: $\mathcal{A}$ can issue more trapdoor and CSI queries, and then Δ_{mdbdh} responds as before. The only restriction that $\mathcal{A}$ cannot ask for the queries distinguishing two target documents.
- Output: $\mathcal{A}$ outputs its guess d. If $d = b$ outputs 1. Otherwise, it outputs 0.

Let's consider the probability that Δ_{mdbdh} does not fail during the execution of algorithm. As described in the above algorithm, Δ_{mdbdh} may fail in the trapdoor query and challenge query phases. We define three events and compute its probabilities as follows.

- NFT: We define an event that Δ_{mdbdh} does not fail during the trapdoor queries phase by NFT. $\mathcal{A}$ can ask trapdoor queries q_T at most, and $Pr[\mathsf{NFT}] = (1 - \frac{1}{q_T+1})^{2q_T \cdot m} \geq 1/e^m$ for sufficiently large q_T.
- NFC: We define an event that Δ_{mdbdh} does not fail during the challenge queries phase by NFC.

$$Pr[(c_{0,j} = 1) \vee (c_{1,j} = 1) | 1 \leq j \leq m] = (1 - (\frac{1}{q_T + 1}))^{2m} < 1 - (\frac{1}{q_T + 1}).$$

  Therefore we have $Pr[\mathsf{NFC}] \geq \frac{1}{(q_T+1)}$.
- NFS: We define an event that Δ_{mdbdh} does not fail during the CSI queries phase by NFS. $\mathcal{A}$ can ask CSI queries q_C at most, and $Pr[\mathsf{NFS}] = (1 - \frac{1}{q_T+1})^{2q_C \cdot m} \geq (\frac{1}{q_T+1})^{2q_C \cdot m}$.
- NF= ($\mathsf{NFT} \wedge \mathsf{NFC} \wedge \mathsf{NFS}$): By the above probabilities we have $Pr[\mathsf{NF}] \geq \frac{1}{e^m (q_T+1)^{2q_C+1}}$.

Now we consider the success probability of Δ_{mdbdh}. In the case of $\mathbf{Exp}^{real}_{\Delta_{mdbdh}}(k)$, the input tuple $(g, u_1, u_{2,1}, .., u_{2,m}, u_3, u_{4,1}, .., u_{4,m})$ is a real MDBDH tuple, and the success probability of Δ_{mdbdh} depends on the advantage of $\mathcal{A}$.

$$\begin{aligned} Pr[\mathbf{Exp}^{real}_{\Delta_{mdbdh}}(k) = 1] &= Pr[\mathbf{Exp}^{real}_{\Delta_{mdbdh}}(k) = 1 | b = 1] Pr[b = 1] \\ &+ Pr[\mathbf{Exp}^{real}_{\Delta_{mdbdh}}(k) = 0 | b = 0] Pr[b = 0] \\ &= \frac{1}{2} \cdot Pr[\mathbf{Exp}^{cta}_{\mathcal{A}}(k) = 1 | b = 1] \\ &+ \frac{1}{2} \cdot (1 - Pr[\mathbf{Exp}^{cta}_{\mathcal{A}}(k) = 1 | b = 0]) \\ &= \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Adv}^{cta}_{\mathcal{A}}(T, q_T, q_h) \end{aligned} \tag{1}$$

In the case of $\mathbf{Exp}^{rand}_{\Delta_{mdbdh}}(k)$, the input tuple $(g, u_1, u_{2,1}, .., u_{2,m}, u_3, u_{4,1}, .., u_{4,m})$ is a random tuple. It means that the challenge CSI_b is uniformly and independently distributed over $\mathbb{G}_1$. Therefore, CSI_b gives no information about bit b to Δ_{mdbdh}. However, we note that $u_{4,1}, .., u_{4,m}$ are generated by uniformly and independently from $\mathbb{G}_1$. So we consider probability that the random triple happens to be a valid MDBDH triple, occasionally. The last term of (2) indicates this maximum probability. We have

$$Pr[\mathbf{Exp}^{rand}_{\Delta_{mdbdh}} = 1] \leq \frac{1}{2} + \frac{1}{q^m}. \tag{2}$$

Consequently, Theorem 3.1 follows by the equations (1), (2), and $Pr[\mathsf{NF}]$. □

4 Concluding Remarks

In recent years, efficient and secure search of data using keywords have received a lot of attentions in the literature. Recently, the work of [7] first considered an operation of conjunctive keyword search, but the communication and storage costs linearly depends on the size of the stored data. Hence the work of [7] is not suitable for a large scale database. In this paper, we proposed an efficient conjunctive keyword search scheme ECKS-PS only requiring constant communication and storage costs. We also showed that its security is reduced to the well-known computational assumption. Our ECKS-PS is the first provably secure and efficient conjunctive keyword search scheme requring only constant communication and storage costs.

In this paper, we only focused on a design of SS-CTA secure ECKS-PS scheme in the random oracle model. To the best of our knowledge, it is never an easy problem to design a SS-CTA secure and efficient ECKS-PS in the standard model (not use a random oracle assumption), still keeping constant costs of communication and storage. It may be a good future work to remove a random oracle assumption from the ECKS-PS scheme while keeping the constant costs.

Acknowledgement

The authors would like to thank Philippe Golle for his helpful comments on early proposals of schemes. The authors also would like to thank anonymous referees for their valuable comments.

References

1. M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, “Encryption with keyword search, revisited: consistency conditions, relations to anonymous IBE, and extensions”, *In Proceedings of Crypto ’05*, LNCS Vol. 3621, pp. 205-222, Springer-Verlag, 2005.
2. B. Bloom, “Space/time trade-offs in hash coding with allowable errors”, *Communications of the ACM*, 13(7):422-426, 1970.
3. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public Key Encryption with Keyword Search”, *In Proceedings of Eurocrypt ’04*, LNCS Vol. 3089, pp. 31-45, Springer-Verlag, 2004.
4. B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, “Private Information Retrieval”, *In Proceedings of 29th STOC*, 1997.
5. G. Di. Crescenzo, Y. Ishai, and R. Ostrovsky, “Universal Servie-providers for Dtabase Private Information Retrieval”, *In Proceedings of 17th PODC*, 1998.
6. Y. Chang and M. Mitzenmacher, “Privacy preserving keyword searches on remote encrypted data”, *In Proceedings of ACNS ’05* LNCS Vol. 3531, pp. 442-455, Springer-Verlag, 2005, An early version of this paper is appeared on Cryptology ePrint Archieve. Availabe at `http://eprint.iacr.org/2004/051`
7. P. Golle, J. Staddon, and B. Waters, “Secure Conjunctive keyword search over encrytped data”, *In Proceedings of ACNS ’04*, LNCS Vol. 3089, pp. 31-45, Springer-Verlag, 2004.
8. E. Goh, “Secure Indexes”, In Cryptology ePrint Archieve on March 16, 2004, This paper is availabe at `http://eprint.iacr.org/2003/216`
9. Microsoft Developer Network (MSDN), in the part of *Maximum Capacity Specifications for SQL Server 2005*.
   Refer to `http://msdn2.microsoft.com/en-us/library/ms143432(SQL.90).aspx`

10. R. Ostrovsky and W. Skeith, “Private keyword search on streaming data”, *This paper will be appear in Crypto05.*
11. W. Ogata and K. Kurosawa, “Oblivious keyword search” *Journal of Complexity* Vol. 20, Issues 2-3, pp. 356-371, 2004.
12. D. J. Park, K. Kim, and P. J. Lee, “Public Key Encryption with Conjunctive Field Keyword Search”, *In Proceedings of WISA ’04*, LNCS Vol. 3325, pp. 73-86, Springer-Verlag, 2004.
13. D. Song, D. Wagner, and A. Perrig, “Practical Techniques for Searches on Encrypted Data”, *In Proceedings of IEEE sysmposium on Security and Privacy*, 2000.

Enhanced Forward-Secure User Authentication Scheme with Smart Cards

Eun-Jun Yoon and Kee-Young Yoo*

Department of Computer Engineering,
Kyungpook National University,
Daegu 702-701, South Korea
Tel.: +82-53-950-5553; Fax: +82-53-957-4846
ejyoon@infosec.knu.ac.kr, yook@knu.ac.kr

Abstract. In 2006, Wang-Li proposed a new user authentication scheme using smart cards which can offer forward secrecy. However, this paper will demonstrate that Wang-Li's scheme is vulnerable to parallel session attack and reflection attack. Furthermore, the current paper presents a more efficient and secure scheme that not only resolves such problems, but also involves fewer computations and communications than Wang-Li's scheme.

Keywords: Network security, Secure protocol, Smart card, Authentication, Password.

1 Introduction

User authentication is an important aspect of security, along with confidentiality and integrity, for systems that allow remote access over untrustworthy networks, like the Internet. As such, a remote password authentication scheme authenticates the legitimacy of users over an insecure channel, where the password is often regarded as a secret shared between the remote system and user. Based on knowledge of the password, a user can use it to create and send a valid login message to a remote system to gain the right to access the system. Meanwhile, the remote system also uses the shared password to check the validity of the login message and to authenticate the user.

In 1981, Lamport [1] proposed a password authentication scheme for insecure communication that scheme requires the remote server to maintain a password table for purpose of verification. In 2000, Hwang and Li [2] proposed a new scheme using smart cards. The advantage of the Hwang-Li's scheme is that it does not need any pass- word table. Subsequently, Yoon et al. [3] proposed a mutual authentication scheme based on generalized ElGamal signature scheme, which is more efficient than Hwang and Li's scheme in terms of computation and communication cost. In addition, the Yoon-Ryu-Yoo's scheme provides the function of key exchange. However, in 2005, Wang-Li [4] pointed out a security

* Corresponding author.

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 197–206, 2006.

leak of the Yoon-Ryu-Yoo's scheme in that an intruder is able to reveal previous session keys by means of disclosed secret parameters.

The current discussion will demonstrate that Wang-Li's scheme is vulnerable to parallel session attack [5] and that an attacker without knowing a user's password can masquerade as the legal user by creating a valid login message from an eavesdropped communication between authentication server and the user. Additionally, we will point out that Wang-Li's scheme is vulnerable to reflection attack [6] in which an attacker can masquerade as the legal authentication server by creating a valid response message from an eavesdropped communication between authentication server and the user. The current paper presents a more efficient and secure scheme that not only resolves such problems, but also involves fewer computations and communications than Wang-Li's scheme.

This paper is organized as follows: Section 2 briefly reviews Wang-Li's forward-secure remote user authentication scheme with smart cards, then Section 3 discusses its weaknesses. The proposed scheme is presented in Section 4, while Section 5 discuss the security and efficiency of the proposed scheme. Our conclusions are presented in Section 6.

2 Review of Wang-Li's Scheme

There are three phases in Wang-Li's scheme [4]: registration, login, and authentication. In addition, their scheme has a password change phase that allows users to update their passwords freely without the help of a remote system. Fig. 1 illustrates Wang-Li's remote user authentication scheme.

Registration: User U_i submits his or her identifier ID_i and PW_i to the remote system, where PW_i is the chosen password. Initially, the remote system performs the following steps:

(1) Chooses a secure one-way function $h(\cdot)$, p, q, and g, where p is a large prime number with bit size 1024, q is a prime divisor of $p-1$ with bit size 160, and g is an element of order q in the finite field $GF(p)$. The bit size of the output of $h(\cdot)$ is $|q|$;
(2) Computes $R_i = h(ID_i||x_s)$, $X_i = R_i \oplus h(ID_i||PW_i)$, where $||$ denotes a concatenation operation;
(3) Writes ID_i, R_i, X_i, $h(\cdot)$, p, q, g to the memory of the smart card and issue the card to U_i. Note that $h(\cdot)$, p, q and g are the public parameters, while R_i and X are the kept secret.

Login: If user U_i wants to log in to a remote system, he or she must insert his or her smart card into a card reader and key in his or her identifier ID_i and password PW_i. Then the smart card performs the following steps:

(1) Generates a random number $r \in Z_q^*$;
(2) Computes $t = g^r \bmod p$;

(3) Computes $V_i = X_i \oplus h(ID_i||PW_i)$. Then the smart card computes $W_i = h(V_i \oplus T)$, where T is the current time-stamp;
(4) Computes $s = h(t||W_i)$;
(5) Sends a message $C_1 = \{ID_i, t, s, T\}$ to the remote system.

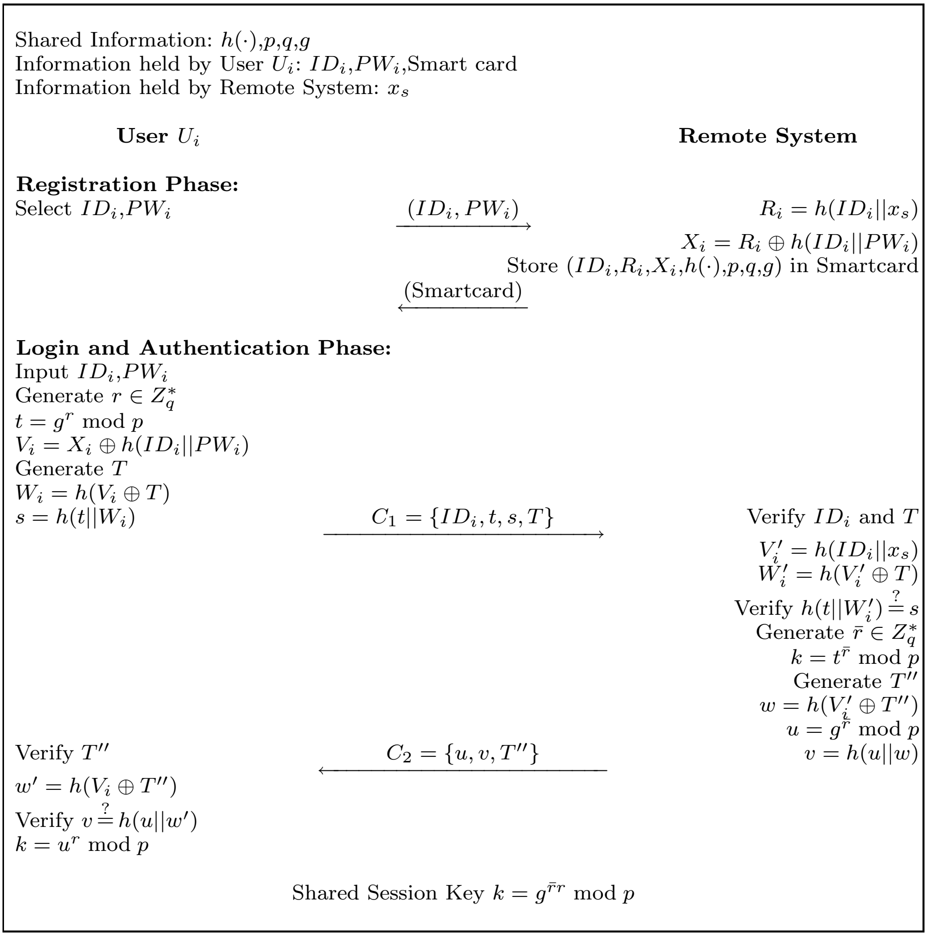

Shared Information: $h(\cdot)$,p,q,g
Information held by User U_i: ID_i,PW_i,Smart card
Information held by Remote System: x_s
User U_i
Remote System
Registration Phase:
Select ID_i,PW_i
$\xrightarrow{(ID_i, PW_i)}$
$R_i = h(ID_i||x_s)$
$X_i = R_i \oplus h(ID_i||PW_i)$
Store (ID_i,R_i,X_i,$h(\cdot)$,p,q,g) in Smartcard
$\xleftarrow{\text{(Smartcard)}}$
Login and Authentication Phase:
Input ID_i,PW_i
Generate $r \in Z_q^*$
$t = g^r \bmod p$
$V_i = X_i \oplus h(ID_i||PW_i)$
Generate T
$W_i = h(V_i \oplus T)$
$s = h(t||W_i)$
$\xrightarrow{C_1 = \{ID_i, t, s, T\}}$
Verify ID_i and T
$V_i' = h(ID_i||x_s)$
$W_i' = h(V_i' \oplus T)$
Verify $h(t||W_i') \overset{?}{=} s$
Generate $\bar{r} \in Z_q^*$
$k = t^{\bar{r}} \bmod p$
Generate T''
$w = h(V_i' \oplus T'')$
$u = g^{\bar{r}} \bmod p$
$v = h(u||w)$
Verify T''
$\xleftarrow{C_2 = \{u, v, T''\}}$
$w' = h(V_i \oplus T'')$
Verify $v \overset{?}{=} h(u||w')$
$k = u^r \bmod p$
Shared Session Key $k = g^{\bar{r}r} \bmod p$

Fig. 1. Wang-Li's a forward-secure user authentication scheme with smart cards

Authentication: Upon receiving the authentication request message C_1, the remote system and the smart card will perform the following steps for mutual authentication between the user and the remote system:

(1) The remote system verifies that ID_i is correct. If not, the login request is rejected.
(2) Let T' be the time that the system receives C_1. The system compares T and T'. If the difference between T and T' is within a valid time interval ΔT, C_1 is considered as a valid message.

(3) The system computes $V_i' = h(ID_i||x_s)$ as well as $W_i' = h(V_i' \oplus T)$.
(4) The system compares $h(t||W_i')$ with s. If they are equal, then the system accepts the login request and proceeds to the next step; otherwise. it rejects the login request.
(5) The system picks a random number $\bar{r} \in Z_q^*$ and computes the session key $k = t^{\bar{r}} \bmod p$.
(6) The system acquires the current time-stamp T'' and computes $w = h(V_i' \oplus T'')$, $u = g^{\bar{r}} \bmod p$, $v = h(u||w)$. The system sends back the message $C_2 = \{u, v, T''\}$ to U_i.
(7) Upon receiving the message $\{u, v, T''\}$, the smart card verifies the validity of the time interval between T'' and the current time-stamp T''', and then computes $w' = h(V_i \oplus T'')$. If $v = h(u||w')$, the mutual authentication is complete. Then $k = g^{\bar{r}r} \bmod p$ is used as the session key between the user U_i and the remote system.

Wang-Li's scheme also enables users to change their password freely and securely using following steps:

Password Change: If the user U_i wants to change his or her password from PW_i to PW_i', he or she should insert his smart card into a card reader and keys in his or her identifier ID_i and password PW_i. Then the smart card performs the following steps:

(1) Computes $V_i = X_i \oplus h(ID_i||PW_i)$ and compares V_i with R_i. If they are equal, then the smart card proceeds to the next step; otherwise, it rejects the password change request.
(2) The user U_i keys in a new password PW_i'.
(3) The smart card computes $X_i' = V_i \oplus h(ID_i||PW_i')$ and stores X_i' in place of X_i.

3 Cryptanalysis of Wang-Li's Scheme

In this section, we will show that Wang-Li's scheme has the following security flaws:

Parallel Session Attack: Consider a scenario of a parallel session attack [5] in which an attacker U_a without knowing users' passwords wants to masquerade as a legal user U_i by creating a valid login message from the eavesdropped communication between the remote system and U_i. The parallel session attack performs as follows:

(1) When U_i wants to login the remote system, U_i sends the login message $C_1 = \{ID_i, t, s, T\}$ to the remote system, where T is the current time stamp. If $C_1 = \{ID_i, t, s, T\}$ is valid, the identification of U_i is authenticated and the remote system responds $C_2 = \{u, v, T''\}$ to U_i, where T'' is the current time stamp.

(2) Once an attacker U_a intercepts this message, he or she masquerades as the legal user U_i to start a new session with the remote system by sending $C_1^* = \{ID_i, t^*, s^*, T^*\}$ back to the remote system, where $t^* = u$, $s^* = v$ and $T^* = T''$.

(3) The login message $C_1^* = \{ID_i, t^*, s^*, T^*\}$ will pass the user authentication of Wang-Li's scheme [4] due to the fact that $s^* = v = h(u||w) = h(u||h(V_i' \oplus T^*))$, where $V_i = V_i'$.

(4) Finally, the remote system responds to the message $C_2^* = \{u^*, v^*, T'''\}$ to U_i, where T''' is the current timestamp. The attacker U_a intercepts and drops this message. Fig. 2 depicts the message transmission of the parallel session attack.

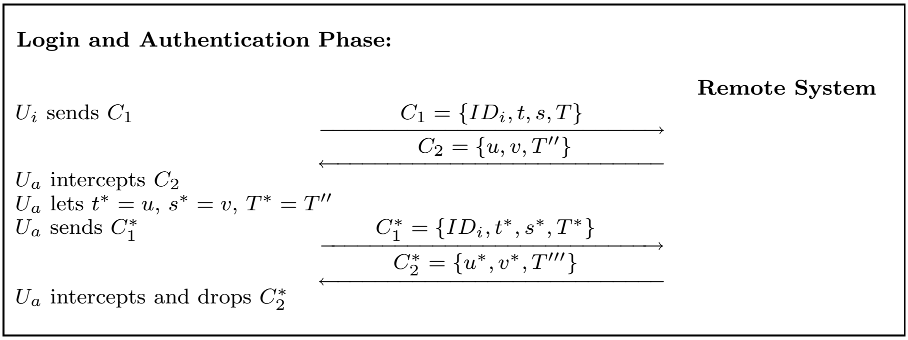

Login and Authentication Phase:
Remote System
U_i sends C_1
$C_1 = \{ID_i, t, s, T\}$
$C_2 = \{u, v, T''\}$
U_a intercepts C_2
U_a lets $t^* = u$, $s^* = v$, $T^* = T''$
U_a sends C_1^*
$C_1^* = \{ID_i, t^*, s^*, T^*\}$
$C_2^* = \{u^*, v^*, T'''\}$
U_a intercepts and drops C_2^*

Fig. 2. Parallel session attack on Wang-Li's scheme

Furthermore, if U_a does not drop the message C_2^* and continually performs the above mentioned parallel session attack, then the remote system is subject to a Denial-of-Service (DoS) attack because the system resources of the remote system are consumed rapidly by the attack. We consider DoS attacks against a remote system's memory, which makes it store useless information, and computational power, which makes it calculate useless process. By using this DoS attack, an attacker can send a great deal of useless requests to a remote system. This results in the remote system being unable to deal with a proper user's authentication request which transacting on attacker's requests. Fig. 3 depicts the message transmission of the Denial-of-Service attack based on a parallel session attack.

Reflection Attack: Consider the scenario of a reflection attack [6]. In the login phase, if attacker U_a has intercepted and blocked a message transmitted in Step (5), i.e., $C_1 = \{ID_i, t, s, T\}$, he or she can impersonate the remote system and send $C_2 = \{u, v, T''\}$ to U_i in step (7) of the authentication phase, where $u = t$, $v = s$ and $T'' = T$ is the current timestamp. Upon receiving the first item of the received message, i.e., T'', U_i will compute $w' = h(h(V_i \oplus T''))$. Note that steps (1)-(6) of the authentication phase are skipped by attacker U_a. Since the computed result equals the second item of the received message, i.e.,

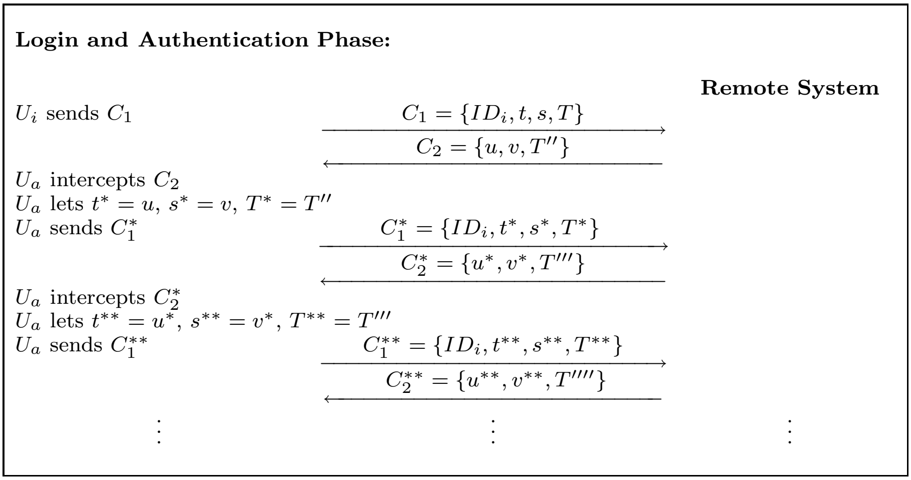

Login and Authentication Phase:
Remote System
U_i sends C_1
$C_1 = \{ID_i, t, s, T\}$
$C_2 = \{u, v, T''\}$
U_a intercepts C_2
U_a lets $t^* = u$, $s^* = v$, $T^* = T''$
U_a sends C_1^*
$C_1^* = \{ID_i, t^*, s^*, T^*\}$
$C_2^* = \{u^*, v^*, T'''\}$
U_a intercepts C_2^*
U_a lets $t^{**} = u^*$, $s^{**} = v^*$, $T^{**} = T'''$
U_a sends C_1^{**}
$C_1^{**} = \{ID_i, t^{**}, s^{**}, T^{**}\}$
$C_2^{**} = \{u^{**}, v^{**}, T''''\}$
$\vdots$
$\vdots$
$\vdots$

Fig. 3. Denial-of-Service attack based on parallel session attack on Wang-Li's scheme

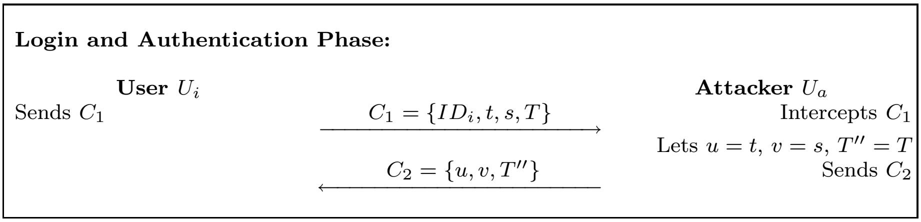

Login and Authentication Phase:
User U_i
Attacker U_a
Sends C_1
$C_1 = \{ID_i, t, s, T\}$
Intercepts C_1
Lets $u = t$, $v = s$, $T'' = T$
$C_2 = \{u, v, T''\}$
Sends C_2

Fig. 4. Reflection attack on Wang-Li's scheme

v, U_i will be fooled into believing that the attacker is the legal remote system. Since U_i cannot actually authenticate the remote system's identity, Wang-Li's authentication scheme fails to provide mutual authentication as the authors claim. Fig. 4 depicts the message transmission of the reflection attack.

4 Countermeasure and Efficiently Optimized Scheme

In this section, we propose an enhancement to Wang-Li's scheme that can withstand the security flaws described in the previous sections. Fig. 5 illustrates the proposed remote user authentication scheme. The registration is same as Wang-Li's scheme. The only difference between the proposed scheme and Wang-Li's scheme is in the login and authentication phases. To resist such attacks, the proposed login and authentication phases are performed as follows:

Login: If user U_i wants to log in to a remote system, he or she must insert his or her smart card into a card reader and key in his or her identifier ID_i and password PW_i. Then the smart card performs the following steps:

(1) Generates a random number $r \in Z_q^*$;
(2) Computes $t = g^r \bmod p$;
(3) Computes $V_i = X_i \oplus h(ID_i||PW_i)$. Then the smart card computes $s = h(t||V_i||T)$, where T is the current time-stamp.
(4) Sends a message $C_1 = \{ID_i, t, s, T\}$ to the remote system.

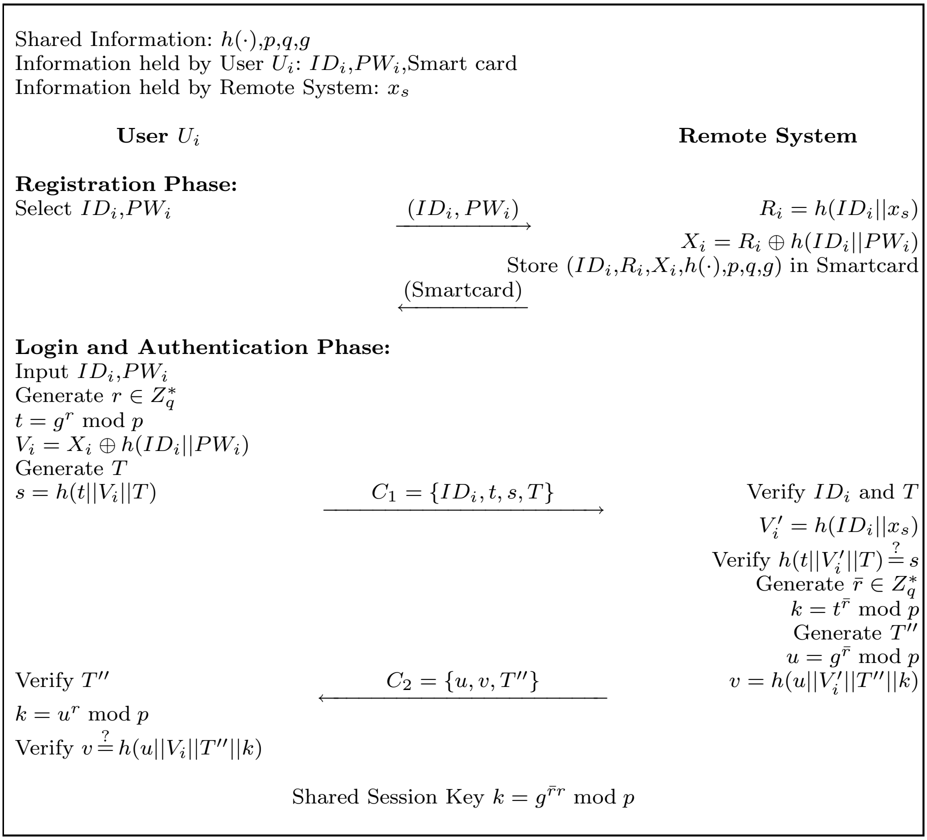

Shared Information: $h(\cdot)$,p,q,g
Information held by User U_i: ID_i,PW_i,Smart card
Information held by Remote System: x_s
User U_i
Remote System
Registration Phase:
Select ID_i,PW_i
$\xrightarrow{(ID_i, PW_i)}$
$R_i = h(ID_i||x_s)$
$X_i = R_i \oplus h(ID_i||PW_i)$
Store (ID_i,R_i,X_i,$h(\cdot)$,p,q,g) in Smartcard
$\xleftarrow{\text{(Smartcard)}}$
Login and Authentication Phase:
Input ID_i,PW_i
Generate $r \in Z_q^*$
$t = g^r \bmod p$
$V_i = X_i \oplus h(ID_i||PW_i)$
Generate T
$s = h(t||V_i||T)$
$\xrightarrow{C_1 = \{ID_i, t, s, T\}}$
Verify ID_i and T
$V_i' = h(ID_i||x_s)$
Verify $h(t||V_i'||T) \overset{?}{=} s$
Generate $\bar{r} \in Z_q^*$
$k = t^{\bar{r}} \bmod p$
Generate T''
$u = g^{\bar{r}} \bmod p$
$v = h(u||V_i'||T''||k)$
$\xleftarrow{C_2 = \{u, v, T''\}}$
Verify T''
$k = u^r \bmod p$
Verify $v \overset{?}{=} h(u||V_i||T''||k)$
Shared Session Key $k = g^{\bar{r}r} \bmod p$

Fig. 5. Proposed forward-secure user authentication scheme with smart cards

Authentication: Upon receiving the authentication request message C_1, the remote system and the smart card will perform the following steps for mutual authentication between the user and the remote system.

(1) The remote system verifies that ID_i is correct. If not, the login request is rejected.
(2) Let T' be the time that the system receives C_1. The system compares T and T'. If the difference between T and T' is within a valid time interval ΔT, C_1 is considered as a valid message.
(3) The system computes $V_i' = h(ID_i||x_s)$.

(4) The system compares $h(t||V_i'||T)$ with s. If they are equal, then the system accepts the login request and proceeds to the next step; otherwise, it rejects the login request.

(5) The system picks a random number $\bar{r} \in Z_q^*$ and computes the session key $k = t^{\bar{r}} \bmod p$.

(6) The system acquires the current time-stamp T'' and computes $u = g^{\bar{r}} \bmod p$, $v = h(u||V_i'||T''||k)$. The system sends back the message $C_2 = \{u, v, T''\}$ to U_i.

(7) Upon receiving the message $\{u, v, T''\}$, the smart card verifies the validity of the time interval between T'' and the current time-stamp T''', and then computes the session key $k = g^{\bar{r}r} \bmod p$. If $v = h(u||V_i||T''||k)$, the mutual authentication is complete. Then $k = g^{\bar{r}r} \bmod p$ is used as the session key between the user U_i and the remote system.

5 Security and Efficiency Analysis

In this section, we will only discuss the enhanced security and efficiency features of the proposed scheme. The other features are the same as original Wang-Li's scheme [4]. The security properties of Wang-Li's scheme and the proposed scheme are summarized in Table 1. In contrast with Wang-Li's scheme, the proposed scheme is more secure.

Table 1. Comparison of security properties

	Wang-Li's Scheme	Proposed Scheme
Parallel session attack	Insecure	Secure
Reflection attack	Insecure	Secure
Replay attack	Secure	Secure
Mutual authentication	Provide	Provide
Session key confirmation	Implicit	Explicit
Perfect forward secrecy	Provide	Provide

Theorem 1. *The proposed scheme prevents the parallel session attack and the reflection attack in Wang-Li's scheme.*

Proof. The parallel session attack and the reflection attack on Wang-Li's scheme can succeed due to the symmetric structure of the messages (e.g. $s = h(t||W_i)$ and $v = h(u||w)$) exchanged between the user U_i and the remote system. However, the proposed scheme can prevent a parallel session attack and a reflection attack as in Wang-Li's scheme because of the different message structure between $s = h(t||W_i)$ and $v = h(w||w||k)$. Thus, the proposed scheme prevents the parallel session attack and the reflection attack in Wang-Li's scheme.

Theorem 2. *The proposed scheme is more efficient than Wang-Li's scheme.*

Proof. The security of the proposed scheme and Wang-Li's scheme is based on a one-way hash function and a discrete logarithm problem [7]. The hash computation costs of Wang-Li's scheme and the proposed scheme in registration, login, and authentication phases are summarized in Table 2. In the login and authentication phases, Wang-Li's scheme requires a total of 10 times hash operations for mutual authentication, but the proposed scheme requires only 6 times hash operations. Thus, the proposed scheme is more efficient than Wang-Li's scheme.

Table 2. Comparisons of computational costs

	Wang-Li's Scheme		Proposed Scheme	
	User	Server	User	Server
Registration Phase	·	2Hash + 1Xor	·	2Hash + 1Xor
Login Phase	1Exp + 3Hash + 1Xor	·	1Exp + 2Hash + 1Xor	·
Authentication Phase	1Exp + 2Hash	2Exp + 5Hash	1Exp + 1Hash	2Exp + 3Hash
Communication Costs	$\approx 2 * (1024 + 160)$ bits		$\approx 2 * (1024 + 160)$ bits	

Exp: Exponentiation operations; Hash: Cryptographic hash operations;
Xor: Bitwise exclusive or ($\oplus$) operations.

6 Conclusions

The current study has demonstrated that Wang-Li's scheme is vulnerable to parallel session attack and reflection attack. The current paper presents a more efficient and secure scheme that not only resolves such problems, but also involves fewer computations and communications than Wang-Li's scheme.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments in improving our manuscript. This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

References

1. Lamport, L.: Password Authentication with Insecure Communication. Communications of the ACM. Vol. 24. No. 11. (1981) 770-772.
2. Hwang, M.S., Li, L.H.: A New Remote User Authentication Scheme Using Smart Cards. IEEE Trans. On Consumer Electronics. Vol. 46. No. 1. (2000) 28-30
3. Yoon, E.J., Ryu, E.K., Yoo, K.Y.: Efficient Remote User Authentication Scheme based on Generalized ElGamal Signature Scheme. IEEE Transactions on Consumer Electronics. Vol. 50. No. 2. (2004) 568-570

4. Wang. B., Li, Z.Q.: A Forward-Secure User Authentication Scheme with Smart Cards. International Journal of Network Security, Vol. 3, No. 2, (Sept. 2006) (http://isrc.nchu.edu.tw/ijns/) 108-111
5. Hsu, C.L.: Security of Chien et al.'s Remote User Authentication Scheme Using Smart Cards. Computer Standards & Interfaces. Vol. 26. No. 3. (May 2004) 167-169
6. Yoon, E.J., Yoo, K.Y.: More Efficient and Secure Remote User Authentication Scheme using Smart Cards. Proceedings of the 2005 11th International Conference on Parallel and Distributed Systems (ICPADS'05). Vol. 2. (July 2005) 73-77
7. Menezes, A.J., Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptograph. CRC Press. New York. (1997)

Pseudonymous PKI for Ubiquitous Computing

Ke Zeng

NEC Laboratories, China
zengke@research.nec.com.cn

Abstract. Conventional PKI is the most effective and efficient solution to non-repudiation. But, it also puts user privacy in danger because the user's activities could be tracked via the unique public-key and certificate he presents in multiple transactions. Pseudonymous PKI (PPKI) solution achieves non-repudiation as well as privacy protection at the same time by providing Pseudonymous Public-Key (PPK) and Pseudonymous Certificate (PCert) that are computed by the user without CA intervention. PPK is as effective as conventional public-key in terms of non-repudiation. Furthermore, the PPKI solution is very efficient in terms of the size of PPK and PCert, and is scalable in terms of certification authority overhead. Therefore PPKI is particularly suitable for ubiquitous computing environments where authenticity, non-repudiation, privacy protection, efficiency, and scalability are key requirements.

1 Introduction

In its simplest form, conventional Public-Key Infrastructure (PKI) consists of the Certification Authority (CA) and registered certificate users of CA. Each PKI user has his own public-key and private-key pair. The basic task of CA is to issue certificate to each user's public-key. The CA also maintains and applies Certificate Revocation List (CRL) to revoke the certificates of misbehaving users. For decades, conventional PKI performs well in many kinds of traditional businesses e.g. secure email and access control that entail various security guarantees.

With the proliferation of smart gadgets, appliances, mobile devices, PDAs, and sensors, ubiquitous computing environments may be constructed of such interconnected devices and services, which promise seamless integration of digital infrastructure into our everyday lives [1]. Conventional PKI can be applied to emerging ubiquitous computing environments to resolve security issues, e.g. non-repudiation [2]. But, it may incur side effects on user privacy. Consider two peers (which may be users or devices) working with each other in a ubiquitous computing environment. For instance, a mobile phone approaches an Access Point (AP) of wireless LAN and tries to surf the wireless LAN. For another instance, a PDA approaches a printer and tries to print a document. For a third instance, a laptop approaches a TV set and tries to render a film on the TV. In all these cases, it may be necessary for the peers to authenticate each message received. At first glance, if each peer has a CA certified certificate, it suffices for conventional PKI to fulfill security requirements. Unfortunately, the single public-key embedded in a conventional certificate is effectively a unique identifier of the

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 207 – 222, 2006.

key holder and hence can jeopardize the key holder's privacy. In aforementioned examples, the single public-key makes it easy for the AP, the printer or the TV set to identify which key holder is on-site, and consequently infer the interest and the behavior pattern of the key holder. Conventional PKI was designed in an era when privacy was not an issue for businesses that needed it for security reasons hence privacy protection was not taken into consideration in conventional PKI design.

Conventional PKI is so far the most effective and efficient solution for non-repudiation message transferring mandated by many ubiquitous computing applications [3, 4]. In order to keep enjoying the merits of conventional PKI for non-repudiation while resolve the intrinsic privacy issue caused by the unique public-key, solutions such as anonymous public-key [5, 6] and incomparable public-key [7] have been proposed. These solutions are suitable for DLP [8] based public-key cryptosystems. Given generators g , h_1 , and h_2 of some finite cyclic group and the private-key x , let $(y_1 = g, y_2 = g^x)$ be the root public-key. Two anonymous public-keys could be generated as $(y_1^{r_1}, y_2^{r_1})$ and $(y_1^{r_2}, y_2^{r_2})$ [5], where r_1 and r_2 are different random integers, while two incomparable public-keys could be generated as (h_1, h_1^x) and (h_2, h_2^x) [7].

It's not explained in Waters et al. [7] how to generate certificates for incomparable public-keys. While in Oishi et al. [5], the anonymous public-keys are supposed to be generated by CA and naturally CA will issue disjoint certificates to different anonymous public-keys. In ubiquitous computing environments, particularly in large-scale mobile computing environments, the proposal of [5] will cause huge amounts of peer requests for anonymous public-keys and certificates, and huge amounts of queries against the CRL. This can result in heavily loaded CA and slow peer computation speed. Hence, the proposal of [5] cannot scale and is evidently inappropriate for ubiquitous computing environments.

Most recently, Ateniese et al. [9] proposed another solution that partially resolves the certificate issue of the anonymous public-keys. Simply speaking, the proposal of [9] enables each peer to compute anonymous public-key as well as the corresponding certificate, all by the peer itself. This apparently distributed solution is much more efficient than that of [5] because the CA is totally free from generating anonymous public-key and corresponding certificate for its users. Hence the proposal of [9] is scalable and should be more applicable in ubiquitous computing environment. However, the proposal of [9] doesn't provide a full solution for ubiquitous computing environments because it lacks tracing and revocation capabilities against misbehaving peers. Finally, a solution that satisfies security and privacy requirements with the smallest possible certificate size and fastest possible message authentication rate is highly desirable for ubiquitous computing environments.

In the following, Pseudonymous PKI (PPKI) solution for ubiquitous computing environments is presented. The key advantages of PPKI are three. First, each peer could generate distinct Pseudonymous Public-Key (PPK) and corresponding Pseudonymous Certificate (PCert) by itself without any involvement of CA. Second, the CA is equipped with efficient tracing and revocation mechanisms. Third, PPKI is very efficient in terms of the size of PPK and PCert and the time for message authentication.

2 Preliminaries

Before presenting the PPKI solution, let's first define some notations and review a few number-theoretic preliminaries.

2.1 Notation

Basically, notations introduced in Camenisch et al. [10] are reused. If S is a finite set, $x \overset{R}{\leftarrow} S$ as well as $x \in_R S$ denotes that x is chosen from S uniformly at random. $y \leftarrow A(x) : b(y)$ denotes that $b(y)$ is true after y was obtained by running algorithm $A(\cdot)$ on input x.

$A^{O_X}(\cdot)$ denotes an oracle O and a Turing Machine A that makes query to oracle O, where X is their common input. $Q \leftarrow A^{O}(x)$ denotes the contents of the Turing Machine's query tape when it terminates interaction with O on input x. Obviously, Q contains both the queries issued to the oracle O and the answers received from the oracle.

A function $v(k)$ is said negligible if for every positive polynomial $p(\cdot)$ and for sufficiently large k, $v(k) < 1/p(k)$.

Sometimes, (non-interactive zero-) knowledge proof technique is used for validity of statements regarding discrete logarithms. For instance, a signature of knowledge proof that signs message m along with the knowledge proofs for discrete logarithms u_1 and u_2 such that $y_1 = g_1^{u_1} \cdot h_1^{u_2}$ and $y_2 = g_2^{u_2}$, where g_1, h_1, g_2 are generators of finite cyclic groups $G_1 = \langle g_1 \rangle = \langle h_1 \rangle$ and $G_2 = \langle g_2 \rangle$, is denoted by

$$SKP\{(x_1, x_2) : \; y_1 = g_1^{x_1} \cdot h_1^{x_2} \wedge y_2 = g_2^{x_2}\}(m)$$

Knowledge proof for discrete logarithms has been extensively studied in the past two decades [11, 12, 13, 14, 15]. Such studies produced many efficient techniques.

2.2 Number-Theoretic Preliminaries

Throughout this paper, traditional multiplicative group notation is used, instead of the additive notation often used in elliptic curve settings.

Let $\mathbb{G}_1 = \langle g_1 \rangle$ and $\mathbb{G}_2 = \langle g_2 \rangle$ be two (multiplicative) finite cyclic groups with additional group $\mathcal{G} = \langle g \rangle$ such that $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathcal{G}| = p$ where p is some large prime. Bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathcal{G}$ is a function, such that [10]

- Bilinear: for all $h_1 \in \mathbb{G}_1$, $h_2 \in \mathbb{G}_2$, for all $a, b \in \mathbb{Z}_p$, $e(h_1^{a}, h_2^{b}) = e(h_1, h_2)^{ab}$
- Non-degenerate: $\exists h_1 \in \mathbb{G}_1$, $\exists h_2 \in \mathbb{G}_2$ such that $e(h_1, h_2) \neq I$ where I is the identity of $\mathcal{G}$
- Computable: there exists an efficient algorithm for computing e

It's supposed that there is a setup algorithm $Setup$ that on input security parameter 1^k, outputs above settings of bilinear map and the algorithm $Setup$ is written as:

$$(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e) \leftarrow Setup(1^k)$$

Since $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathcal{G}$ are of the same prime order p, according to Bilinear property and Non-degenerate property it's easy to see that $e(g_1, g_2) = \boldsymbol{g}$.

Co-CDH Assumption. Suppose that $\mathbb{G}_1 = \langle g_1 \rangle$ and $\mathbb{G}_2 = \langle g_2 \rangle$ are groups chosen by the setup algorithm $Setup$. Then for all Probabilistic Polynomial Time (P.P.T.) adversaries $\mathcal{A}$, $v(k)$ defined as follows is a negligible function:

$$(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e) \leftarrow Setup(1^k); a \overset{R}{\leftarrow} \mathbb{Z}_p; h \overset{R}{\leftarrow} \mathbb{G}_1;$$

$$\Pr\left[y \leftarrow \mathcal{A}(g_2, g_2^{\ a}, h) : y = h^a\right] = v(k)$$

Co-DDH Assumption. Suppose that $\mathbb{G}_1 = \langle g_1 \rangle$ and $\mathbb{G}_2 = \langle g_2 \rangle$ are groups chosen by the setup algorithm $Setup$. Then for all P.P.T. adversaries $\mathcal{A}$, $v(k)$ defined as follows is a negligible function:

$$(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e) \leftarrow Setup(1^k); a \overset{R}{\leftarrow} \mathbb{Z}_p; b \overset{R}{\leftarrow} \mathbb{Z}_p;$$

$$\left|\Pr\left[y \leftarrow \mathcal{A}(g_1, g_1^{\ a}, g_2, g_2^{\ a}) : y = \text{"}true\text{"}\right] - \Pr\left[y \leftarrow \mathcal{A}(g_1, g_1^{\ a}, g_2, g_2^{\ b}) : y = \text{"}true\text{"}\right]\right| = v(k)$$

In the presence of an efficiently computable bilinear map e between $\mathbb{G}_1$ and $\mathbb{G}_2$, so far the Co-CDH Assumption is believed to hold. Unfortunately, since verifying $e(g_1^{\ a}, g_2) = e(g_1, g_2^{\ a}) \neq e(g_1, g_2^{\ b})$ is efficient, the Co-DDH Assumption is broken. Nevertheless, the standard DDH assumption is hopefully to hold on $\mathbb{G}_1$. Below XDH Assumption [9, 16, 17] formalizes this belief on DDH-hard $\mathbb{G}_1$.

XDH Assumption. Suppose that $\mathbb{G}_1 = \langle g_1 \rangle$ and $\mathbb{G}_2 = \langle g_2 \rangle$ are groups chosen by the setup algorithm $Setup$. Then for all P.P.T. adversaries $\mathcal{A}$, $v(k)$ defined as follows is a negligible function:

$$(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e) \leftarrow Setup(1^k); a \overset{R}{\leftarrow} \mathbb{Z}_p; b \overset{R}{\leftarrow} \mathbb{Z}_p; c \overset{R}{\leftarrow} \mathbb{Z}_p;$$

$$\left|\Pr\left[y \leftarrow \mathcal{A}(g_1, g_1^{\ a}, g_1^{\ b}, g_1^{\ ab}) : y = \text{"}true\text{"}\right] - \Pr\left[y \leftarrow \mathcal{A}(g_1, g_1^{\ a}, g_1^{\ b}, g_1^{\ c}) : y = \text{"}true\text{"}\right]\right| = v(k)$$

It's notable that pairing doesn't serve as sufficient condition for deciding DDH instances in $\mathbb{G}_1$. Instead, DDH-easy $\mathbb{G}_1$ is obtained via isomorphism $\psi : \mathbb{G}_1 \rightarrow \mathbb{G}_2$ (also called distortion map). The XDH Assumption places DDH-hard condition on $\mathbb{G}_1$ only. Even though $\mathbb{G}_2$ is DDH-easy, as long as isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ is computationally one-way, DDH-hard requirement may still hold in $\mathbb{G}_1$.

Although efficient distortion maps always exist for pairing implementations over the popular supersingular curves, they are not present in all pairing groups, in particular those recently discovered MNT curves [18] which are non-supersingular elliptic curves with efficiently computable pairings.

q-SDH Assumption. Suppose that $\mathbb{G}_1 = \langle g_1 \rangle$ and $\mathbb{G}_2 = \langle g_2 \rangle$ are groups chosen by the setup algorithm $Setup$. Then for all P.P.T. adversaries $\mathcal{A}$, $v(k)$ defined as follows is a negligible function:

$$(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e) \leftarrow Setup(1^k); a \overset{R}{\leftarrow} \mathbb{Z}_p;$$

$$\Pr\left[(x, y) \leftarrow \mathcal{A}({g_2}^a, {g_2}^{a^2}, \cdots, {g_2}^{a^q}) : x \in \mathbb{Z}_p \wedge y = {g_1}^{1/(a+x)}\right] = v(k)$$

The lower bound on the computational complexity of breaking the q-SDH Assumption was given in Boneh et al. [19]. The q-SDH Assumption has been used to construct traitor tracing system [20], short signature scheme [19], and short group signatures [21]. Below BB Theorem is proved in [19] resulted in a secure signature scheme in the sense of existential unforgeability.

BB Theorem [19]**.** Suppose q-SDH Assumption holds in $\mathbb{G}_1 = \langle g_1 \rangle$ and $\mathbb{G}_2 = \langle g_2 \rangle$ chosen by the setup algorithm $Setup$. Let $A = {g_2}^a \in \mathbb{G}_2$. Let $O_A(\cdot)$ be an oracle that on input $x \in \mathbb{Z}_p$, outputs two-tuple (t, x) such that $e(t, A \cdot {g_2}^x) = e(g_1, g_2)$. Then for all P.P.T. adversaries $\mathcal{A}^?$, $v(k)$ defined as follows is a negligible function:

$$(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e) \leftarrow Setup(1^k); a \overset{R}{\leftarrow} \mathbb{Z}_p; A = {g_2}^a;$$

$$Q \leftarrow \mathcal{A}^{O_A}(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e);$$

$$\Pr\left[(t, x) \leftarrow \mathcal{A}(Q, p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e, A) : x \notin Q \wedge t^{a+x} = g_1\right] = v(k)$$

The BB Theorem was proven secure against a weak chosen message attack where the adversary must submit all of his queries in advance of the public-key generation.

The BB Theorem could be extended to more general case where the adversary gets more from the oracle but only needs to output forgery in the basic form.

General Theorem. Suppose q-SDH Assumption holds in $\mathbb{G}_1 = \langle g_1 \rangle$ and $\mathbb{G}_2 = \langle g_2 \rangle$ chosen by the setup algorithm $Setup$. Let $(h_1 = g_1, h_2, \ldots, h_k) \in {\mathbb{G}_1}^k$, $A = {g_2}^a \in \mathbb{G}_2$. Let $O_A(\cdot)$ be an oracle that on input $x \in \mathbb{Z}_p$, outputs a set of two-tuples (t_j, x) , $j = 1, 2, \ldots, k$, such that $e(t_j, A \cdot {g_2}^x) = e(h_j, g_2)$. Then for all P.P.T. adversaries $\mathcal{A}^?$, $v(k)$ defined as follows is a negligible function:

$$(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e) \leftarrow Setup(1^k); h_1 = g_1; (h_2, h_3, \ldots, h_k) \overset{R}{\leftarrow} {\mathbb{G}_1}^{k-1}; a \overset{R}{\leftarrow} \mathbb{Z}_p; A = {g_2}^a;$$

$$Q \leftarrow \mathcal{A}^{O_A}(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_2, e, h_1, h_2, \ldots, h_k);$$

$$\Pr\left[(t, x) \leftarrow \mathcal{A}(Q, p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_2, e, h_1, h_2, \ldots, h_k, A) : x \notin Q \wedge t^{a+x} = \prod_{j=1}^{k} h_j\right] = v(k)$$

3 Pseudonymous PKI Solution

Suppose two peers in ubiquitous computing environments need to communicate with each other for certain purposes, e.g. a series of service requests and service responses. The most important requirements for their communications are authenticity, non-repudiation and privacy protection. The PPKI solution, directly on the basis of General Theorem and XDH Assumption, fulfills the requirements for ubiquitous computing. The PPKI solution includes five phases, namely CA Setup, Peer Registration, Peer-to-Peer Authentic Communication, Tracing, and Revocation.

3.1 CA Setup

At time the system initializes, CA needs to determine security strength it desires, select underlying algebra, generate its private-key, and publish the corresponding public-key. Below steps describe the procedures for CA to setup.

1. CA defines the security parameter and calls $Setup$
2. CA chooses a secure one-way hash function $Hash(\cdot) : \{0,1\}^* \to \{0,1\}^p$
3. CA chooses integer $a \in_R \mathbb{Z}_p$ as private-key and computes $A = g_2^{\,a} \in \mathbb{G}_2$
4. CA chooses $(h_1, h) \in_R \mathbb{G}_1^{\,2}$

CA publishes public-key $(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e, h_1, h, A)$ and $Hash(\cdot)$. In addition, CA initializes and publishes a string Ver as version of the public-key, and an empty revocation list RL.

3.2 Peer Registration

Given CA public-key $(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e, h_1, h, A)$ of version Ver, in order to register with CA, a peer that has private-key $x \in_R \mathbb{Z}_p$, root public-key $(h, y = h^x) \in \mathbb{G}_1^{\,2}$, and peer identifier ID carries out the following with CA.

1. Peer computes $y' = g_2^{\,x} \in \mathbb{G}_2$ and sends ID , y , y' to CA
2. Peer interacts with CA and generates proof s_{id} for ownership of its identifier ID and the valid binding between its root public-key (h, y) and ID
3. CA verifies that $e(y, g_2) = e(h, y')$ holds
4. CA selects $\xi \in_R \mathbb{Z}_p$ and computes $z = Hash(ID \mid y \mid s_{id} \mid \xi) \in \mathbb{Z}_p$, where $\mid$ denotes concatenation
5. CA computes $(t_g = g_1^{\,1/(a+z)}, t_h = (h_1 \cdot h^x)^{1/(a+z)}) \in \mathbb{G}_1^{\,2}$
6. CA stores (ID, y, s_{id}, y', ξ) in its database
7. CA sends (t_g, t_h, z) to peer as root certificate
8. Peer verifies that $e(t_g, A \cdot g_2^{\,z}) = e(g_1, g_2)$ and $e(t_h, A \cdot g_2^{\,z}) = e(h_1 \cdot h^x, g_2)$
9. Peer stores CA public-key $(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e, h_1, h, A)$ and its version Ver

10. Peer computes $(v_1 = e(g_1 \cdot h_1, g_2), v_2 = e(t_g \cdot t_h, g_2{}^{-1}), v_3 = e(h, g_2)) \in \mathcal{G}^3$
11. Peer stores (t_g, t_h, z) and accelerators (v_1, v_2, v_3) in accordance with Ver

When a peer registers with CA, it's necessary for CA to ascertain that the peer who alleges holding identifier ID really owns the ID . Otherwise malicious peer may claim a different identifier resulting in denial of tracing (as will be seen in later tracing section). Detail analysis on proving ID ownership is out of the scope of this paper.

In terms of (ID, y, s_{id}, y', ξ) stored by the CA for the peer, it's notable that y' must not be published otherwise privacy protection for the peer is broken. On the other hand, (ID, y, s_{id}, ξ) , and z implicitly, may be published to the public as e.g. the CA's registered peer database (as will be shown in later sections).

3.3 Peer-to-Peer Authentic Communication

Below steps describe the procedures for a peer, namely the prover peer, to send authentic messages to another peer, namely the verifier peer. The messages are for instance service requests or service responses.

Given CA public-key $(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e, h_1, h, A)$ of version Ver , prover peer that has root public-key (h, h^x) and root certificate (t_g, t_h, z) , first makes decision either to reuse a set of pre-computed PPK plus PCert, or to generate a new set online.

1. If prover peer decides to generate a new set
   a). Prover peer selects integer $r \in_R \mathbb{Z}_p$
   b). Prover peer computes PPK $(t = (t_g \cdot t_h)^r, t_y = t^x) \in \mathbb{G}_1{}^2$
   c). Prover peer computes signature of knowledge proof

$$s = SKP\{(x_1, x_2, x_3): \; e(t, A) = v_1{}^{x_1} \cdot (v_2{}^r)^{x_2} \cdot v_3{}^{x_3} \wedge 1 = (t^{-1})^{x_3} \cdot t_y{}^{x_1}\}(Ver) \in \mathbb{Z}_p{}^4$$

   d). Prover peer stores PPK (t, t_y) and PCert (Ver, s) in its local database in accordance with Ver
2. For a set of PPK (t, t_y) and PCert (Ver, s) , let $m = (TS \mid M)$ where TS denotes time stamp and M denotes the message, prover peer computes signature on m as

$$s_m = SKP\{(x_1): \; t_y = t^{x_1}\}(m) \in \mathbb{Z}_p{}^2$$

3. Prover peer sends PPK (t, t_y) , PCert (Ver, s) , m , and s_m to the verifier peer

Above steps illustrate that on the basis of root public-key and root certificate, the prover peer can generate the PPK and corresponding PCert without CA involvement.

Given CA public-key $(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e, h_1, h, A)$ of version Ver , on receiving purported PPK (t, t_y) , PCert (Ver, s) , m and s_m from prover peer, the verifier peer executes the following.

1. Verifier peer compares the highest version of CA public-key it stored with Ver received from the prover peer. If they are different, according to the mechanism to

be described in following revocation section, the peer that is using lower version CA public-key may get necessary revocation data from another peer or from CA directly, and update to higher version CA public-key

2. Verifier peer searches its local database for stored (t, t_y), if it doesn't find a match

   a). Verifier peer checks that s is valid signature of knowledge proof with respect to

$$s = SKP\{(x_1, x_2, x_3):\ e(t, A) = e(g_1 \cdot h_1, g_2)^{x_1} \cdot e(t, g_2^{-1})^{x_2} \cdot e(h, g_2)^{x_3} \wedge 1 = (t^{-1})^{x_3} \cdot t_y^{x_1}\}(Ver)$$

   b). Verifier peer stores (t, t_y) and (Ver, s) in its local database as accepted PPK and PCert pair in accordance with Ver

3. Verifier peer checks that s_m is valid signature on m with respect to PPK (t, t_y)

As shown above, the PPK could be verified as CA certified based on the PCert. Similar to conventional PKI system, before investigating message authenticity, the verifier peer can make sure that the PPK is really certified by CA thru checking validity of the PCert, although the PCert is indeed self-computed by the prover peer without CA intervention. Furthermore, if the PPK has once been verified as valid and accepted by the verifier peer, hence has been stored in his local database, to verify subsequent messages from the prover peer, it suffices for the verifier peer to validate signature s_m only. There is henceforth no need for the verifier peer to check the validity of PCert at all. Therefore, the prover peer may at least omit PCert from transmission of subsequent messages.

What the signature of knowledge proofs s and s_m guarantee, without revealing r , z , and x , are that

$$\begin{aligned} e(t, A) &= e(g_1 \cdot h_1, g_2)^r \cdot e(t, g_2^{-1})^z \cdot e(h, g_2)^{rx} \\ e(t^a, g_2) &= e((g_1 \cdot h_1 \cdot h^x)^r \cdot t^{-z}, g_2) \\ t_y = t^x \text{ and } \qquad t^a &= (g_1 \cdot h_1 \cdot h^x)^r \cdot t^{-z} \\ t^{a+z} &= (g_1 \cdot h_1 \cdot h^x)^r \\ (t^{1/r})^{a+z} &= g_1 \cdot h_1 \cdot h^x \end{aligned}$$

Given sound Honest Verifier Zero-Knowledge (HVZK) proof technique in use, informally speaking, success probability of a dishonest P.P.T. prover peer is negligible, then the prover peer indeed has $(t^{1/r}, z, x)$ such that $(t^{1/r})^{a+z} = g_1 \cdot h_1 \cdot h^x$ holds. According to General Theorem, $(t^{1/r}, z)$ must be issued by CA to certain registered peer who has root public-key (h, h^x) .

Theorem 1. On the basis of General Theorem and sound HVZK proof technique, success probability for P.P.T. attackers to fabricate PCert is negligible.

Given a set of PPK and PCert of certain prover peer, it's notable that no other peer can impersonate the prover peer, even CA that has (ID, y, s_{id}, y', ξ) and (t_g, t_h, z) of all its registered peers is unable to, because generating a valid signature s_m that could be verified by the PPK needs knowledge of private-key x which is only known to the prover peer.

Apparently, the PCert (Ver, s), which is the output of HVZK proof, never hazards privacy of the prover peer. Since only PPK (t, t_y) and PCert (Ver, s) are of privacy concern (regardless of the application specific message m), the last hope of malicious verifier peers are, on receiving multiple instances of PPKs, trying to figure out whether the PPKs belong to the same prover peer or not. But this attack is hopeless because PPK of the prover peer could be interpreted as ElGamal cipher-text of plaintext 1 using public-key $(t_g \cdot t_h, (t_g \cdot t_h)^x)$. Equality between indistinguishability of these cipher-texts and DDH assumption has been proved in Tsiounis et al. [22]. So, XDH Assumption that the PPKI solution relies on assures indistinguishability of all the PPKs. Finally, the claim in previous section that it's safe, in terms of peer privacy, for the CA to publish (ID, y, s_{id}, ξ) as its registered peer database is justified.

Theorem 2. On the basis of XDH Assumption and sound HVZK proof technique, success probability for P.P.T. attackers to break pseudonymity of PPK and PCert is negligible.

Suppose Schnorr scheme [13] is utilized to compute and validate signature of knowledge proofs s and s_m. In order to generate s, the prover peer first selects $(\alpha, \beta, \gamma) \in_R \mathbb{Z}_p^{\ 3}$ then computes

$$R_A = v_1^{\ \alpha} \cdot v_2^{\ r\beta} \cdot v_3^{\ \gamma} \in \mathcal{G}\,, R_x = t^{-\gamma} \cdot t_y^{\ \alpha} \in \mathbb{G}_1\,, c_s = Hash(R_A \mid R_x \mid Ver) \in \mathbb{Z}_p$$

$$\text{and}\,(s_1 = \alpha - c_s \cdot r, s_2 = \beta - c_s \cdot z, s_3 = \gamma - c_s \cdot rx) \in \mathbb{Z}_p^{\ 3}$$

In the sequel, the signature of knowledge proof s is $s = (c_s, s_1, s_2, s_3) \in \mathbb{Z}_p^{\ 4}$.

In order to generate s_m, the prover peer selects $\lambda \in_R \mathbb{Z}_p$ and computes

$$R_t = t^\lambda \in \mathbb{G}_1\,, c_{sm} = Hash(R_t \mid m) \in \mathbb{Z}_p\,,\ \text{and}\, s_{sm} = \lambda - c_{sm} \cdot x \in \mathbb{Z}_p$$

Finally, the signature of knowledge proof s_m is $s_m = (c_{sm}, s_{sm}) \in \mathbb{Z}_p^{\ 2}$.

In terms of signature verification, similar to the prover peer, it's possible for the verifier peer to pre-compute and store $v_1 = e(g_1 \cdot h_1, g_2)$ and $v_3 = e(h, g_2)$ as well. In order to verify s, the verifier peer computes $R_A = v_1^{\ s_1} \cdot e(t, g_2^{\ -s_2} \cdot A^{c_s}) \cdot v_3^{\ s_3} \in \mathcal{G}$, $R_x = t^{-s_3} \cdot t_y^{\ s_1} \in \mathbb{G}_1$ and $c_s' = Hash(R_A \mid R_x \mid Ver) \in \mathbb{Z}_p$. The verifier peer accepts s iff $c_s = c_s'$.

In order to verify s_m, the verifier peer computes $R_t = t^{s_{sm}} \cdot t_y^{\ c_{sm}} \in \mathbb{G}_1$ and $c_{sm}' = Hash(R_t \mid m) \in \mathbb{Z}_p$. The verifier peer accepts s_m iff $c_{sm} = c_{sm}'$.

Now it's clear that to generate PPK and PCert, the prover peer spends 4 scalar multiplications in $\mathbb{G}_1$ and 3 modular exponentiations in $\mathcal{G}$. To generate signature s_m, the prover peer spends 1 scalar multiplication in $\mathbb{G}_1$. Benefit from the accelerators, online pairing evaluations are totally avoided. Pre-computation technique can further help the prover peer be free from all online scalar multiplications and online modular exponentiations as well.

In terms of data transferred from the prover peer to the verifier peer, the PPK and PCert consist of 2 elements of $\mathbb{G}_1$ and 4 elements of $\mathbb{Z}_p$ while the signature s_m consists of 2 elements of $\mathbb{Z}_p$. As reference, one can take p to be a 170 bits prime and use a group $\mathbb{G}_1$ where each element is 171 bits. With these parameters, security strength is approximately the same as standard 1024 bits RSA signature scheme [23]. In such setting, the PPK is 342 bits long, the PCert takes only 680 bits, while the signature s_m is 340 bits in length.

At the verifier side, to validate the PCert, it costs the verifier peer 1 online pairing computations, 2 scalar multiplications in $\mathbb{G}_1$, 2 scalar multiplications in $\mathbb{G}_2$, and 2 modular exponentiations in $\mathcal{G}$. To verify the signature s_m, it costs the verifier merely 2 scalar multiplications in $\mathbb{G}_1$.

It's notable that signature of knowledge proofs s and s_m could be combined into one knowledge proof s' as

$$s' = SKP\{(x_1, x_2, x_3):\ e(t, A) = e(g_1 \cdot h_1, g_2)^{x_1} \cdot e(t, g_2{}^{-1})^{x_2} \cdot e(h, g_2)^{x_3} \wedge 1 = (t^{-1})^{x_3} \cdot t_y{}^{x_1}\}(Ver \mid m)$$

And authentic message delivery from the prover peer to the verifier peer could be achieved as well. But comparing with PPK plus PCert model, utilizing s' is quite inefficient in ubiquitous computing environments because it takes much more time to generate, transmit, and verify s' for each authentic message.

3.4 Tracing

Given a valid set of PPK (t, t_y), PCert (Ver, s), m and s_m, released by a misbehaving peer, under lawful conditions CA can execute the following to uncover the peer.

1. For all y_i' stored in its database, CA evaluates $e(t, y_i') \overset{?}{=} e(t_y, g_2)$ which eventually reveals y', and consequently (ID, y, s_{id}, y', ξ), of the peer being traced
2. CA selects $\sigma \in_R \mathbb{Z}_p$, computes $y_\sigma = h^\sigma \in \mathbb{G}_1$ and $y_\sigma' = y' \cdot g_2{}^\sigma \in \mathbb{G}_2$
3. CA computes signature of knowledge proof
$$\overline{s} = SKP\{(x_1):\ e(t, y_\sigma') \,/\, e(t_y, g_2) = e(t, g_2)^{x_1}\}$$
4. CA publishes $(ID, y, s_{id}, y_\sigma, y_\sigma', \overline{s})$ as the tracing result
5. Everyone can verify the correct binding between ID and y based on validity of s_{id}
6. Everyone can verify that $y \neq y_\sigma{}^{-1}$ and $e(y \cdot y_\sigma, g_2) = e(h, y_\sigma')$
7. Everyone can verify that $\log_t{}^{t_y} = \log_h{}^y$ based on the validity of $\overline{s}$

Note that XDH Assumption holds implies non-existence of distortion map $\psi : \mathbb{G}_1 \to \mathbb{G}_2$. Therefore $y_\sigma' = y' \cdot g_2{}^\sigma$ unconditionally hides y' [24].

It's easy to see that by simply publishing y' everyone can verify the equality of discrete logarithm $\log_t{}^{t_y} = \log_h{}^y$ and ascertain that ID is the identifier of the PPK's holder. But if CA does publish y', active attackers can use y' to test ownership of any other PPK $(t', t'_y = t'^x)$ generated by the same peer because for any t', t'', and $\widehat{x} \neq x$, it's easy to check that $e(t', y') = e(t'_y, g_2)$ and $e(t'', y') \neq e(t''^{\widehat{x}}, g_2)$. Therefore privacy protection is no longer dependable if y' is published.

To prove equality of discrete logarithm $\log_t^{t_y} = \log_h^{y}$, conventional knowledge proof scheme requires that the prover actually knows the discrete logarithm. However above tracing procedures enable CA to prove $\log_t^{t_y} = \log_h^{y}$ although CA doesn't know the discrete logarithm at all. Note that this discrete logarithm is indeed the private-key of the peer being traced and obviously must not be known to the CA.

Theorem 3. On the basis of sound HVZK proof technique and the hardness of DLP, success probability for P.P.T. CA to generate valid signature of knowledge proof $\overline{s} = SKP\{(x_1) : \; e(t, y'_\sigma) / e(t_y, g_2) = e(t, g_2)^{x_1}\}$, in case $\log_t^{t_y} \neq \log_h^{y}$ and $\log_h^{y}$ is unknown to CA, is negligible.

Define perfect forward anonymity as linking a PPK to a peer will not reveal previous as well as future transactions by the peer, even if all past and future behaviors under all past and future PPKs of the peer are logged [25]. When $(ID, y, s_{id}, y_\sigma, y'_\sigma, \overline{s})$ is published as tracing result and the peer is exposed, perfect forward anonymity of the peer is protected. This is true since (ID, y, s_{id}) in any case could be published, and $(y_\sigma, y'_\sigma, \overline{s})$ reveals nothing regarding y' on the basis of XDH Assumption and HVZK proof technique in use. In other words, the tracing result $(ID, y, s_{id}, y_\sigma, y'_\sigma, \overline{s})$ discloses no more information for P.P.T. active attackers to distinguish the peer's any other set of PPK and PCert, either previous sets or those in the future.

3.5 Revocation

Given a peer identified by $(\widehat{ID}, \widehat{y}, \widehat{s}_{id}, \widehat{y}', \widehat{\xi})$, if revocation is necessary in case e.g. the peer behaves illegally or reports key compromise, below steps are effective to revoke $\widehat{z}$ held by the peer.

1. CA re-computes $\widehat{z} = Hash(\widehat{ID} \mid \widehat{y} \mid \widehat{s}_{id} \mid \widehat{\xi}) \in \mathbb{Z}_p$
2. CA computes $\tilde{g}_1 = g_1^{1/(a+\widehat{z})}$, $\tilde{g}_2 = g_2^{1/(a+\widehat{z})}$, and $\tilde{A} = g_2 \cdot \tilde{g}_2^{-\widehat{z}}$
3. CA increases version Ver and publishes public-key $(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, \tilde{g}_1, \tilde{g}_2, e, h_1, h, \tilde{A})$ of version Ver
4. CA adds revocation data $(Ver, \tilde{g}_1, \tilde{g}_2, \widehat{z})$ into RL
5. CA publishes $(\widehat{ID}, \widehat{y}, \widehat{s}_{id}, \widehat{\xi})$ in addition

Except the revoked peer, any peer that has root certificate (t_g, t_h, z), on receiving revocation data $(Ver, \tilde{g}_1, \tilde{g}_2, \widehat{z})$ where Ver is just higher than the version of CA public-key he holds, can do the following to update its root certificate by himself.

1. Peer computes $\tilde{A} = g_2 \cdot \tilde{g}_2^{-\hat{z}}$
2. Peer stores new CA public-key $(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, \tilde{g}_1, \tilde{g}_2, e, h_1, h, \tilde{A})$ of version Ver
3. Peer stores revocation data $(Ver, \tilde{g}_1, \tilde{g}_2, \hat{z})$
4. Peer computes $\tilde{t}_g = (\tilde{g}_1 / t_g)^{1/(z-\hat{z})}$
5. Peer computes $v_1 = e(\tilde{g}_1 \cdot h_1, \tilde{g}_2)$, $v_2 = e(\tilde{t}_g \cdot t_h, \tilde{g}_2)$, and $v_3 = e(h, \tilde{g}_2)$
6. Peer stores new root certificate $(\tilde{t}_g, t_h, z)$ and updated accelerators (v_1, v_2, v_3) in accordance with Ver

By $(\widehat{ID}, \hat{y}, \hat{s}_{id}, \hat{\xi})$, everyone can verify that $\hat{z} = Hash(\widehat{ID} \mid \hat{y} \mid \hat{s}_{id} \mid \hat{\xi})$ really belongs to the peer being revoked. It could be regarded as a proof that CA does not incorrectly revoke some other innocent peer.

It's easy to verify that $\tilde{t}_g^{\,a+z} = g_1^{1/(a+\hat{z})} = \tilde{g}_1$ and $\tilde{g}_2^{\,a} = g_2 \cdot \tilde{g}_2^{-\hat{z}} = \tilde{A}$ hold. Therefore $e(\tilde{t}_g, \tilde{A}\tilde{g}_2^{\,z}) = e(\tilde{g}_1, \tilde{g}_2)$ holds. Note that $e(t_h, \tilde{A} \cdot \tilde{g}_2^{\,z}) = e(h_1 \cdot h^x, \tilde{g}_2)$ holds for $\tilde{g}_2$ and $\tilde{A}$ as well, congruence $e(\tilde{t}_g \cdot t_h, \tilde{A}\tilde{g}_2^{\,z}) = e(\tilde{g}_1 \cdot h_1 \cdot h^x, \tilde{g}_2)$ will hold for the CA public-key of higher version.

The basic revocation scheme is similar to that proposed in Boneh et al. [21]. Based on General Theorem, here the root certificate is securely divided into two portions (t_g, z) and (t_h, z) . Revocation procedures need be applied to (t_g, z) portion only resulting in shortest revocation data per revoked peer. The revocation data $(Ver, \tilde{g}_1, \tilde{g}_2, \hat{z})$ contains one element of $\mathbb{G}_1$, one element of $\mathbb{G}_2$, and one element of $\mathbb{Z}_p$. The original scheme of [21] needs one more element of $\mathbb{G}_1$ for each revocation data (indeed this element is $\tilde{h} = h^{1/(a+\hat{z})}$ as illustrated in Furukawa et al. [26]).

Define forward anonymity, a weak version of perfect forward anonymity, as that linking a PPK to a peer will not reveal previous transactions by the peer, even if all past behaviors under all PPKs of the peer are logged [27]. It's notable that when a peer is revoked, forward anonymity of the revoked peer is protected. This is true since $(\widehat{ID}, \hat{y}, \hat{s}_{id}, \hat{\xi})$ in any case could be published hence the revocation data reveals no more information for P.P.T. active attackers to distinguish the peer's previous sets of PPK and PCert.

4 Conclusion

The shift to the ubiquitous computing paradigm brings forth new challenges to security and privacy. A solution that satisfies authenticity, non-repudiation and privacy protection requirements with the smallest possible data transmitted and fastest possible message authentication rate is highly desirable for ubiquitous computing environments.

The PPKI solution fully satisfies the security and privacy requirements at the cost of 342 bits PPK, shortest 680 bits PCert, mere 340 bits message signature, and only 2 scalar multiplications to verify the message signature. With these settings, the security strength is approximately the same as standard 1024 bits RSA signature scheme.

Besides its efficiency, with tracing and revocation capabilities, the PPKI solution is full functional and scalable makes it particularly suitable for large-scale highly dynamic ubiquitous computing environments where authenticity, non-repudiation, privacy protection, efficiency and scalability are to be satisfied simultaneously.

Acknowledgements

The author would like to thank Min-Yu Hsueh, Tomoyuki Fujita, Kazue Sako, and Jun Furukawa for pointing to this research and for inspiring discussions during the development of the ideas herein presented. The author would also like to thank the anonymous reviewers, whose comments and suggestions help to improve the quality of this paper and stimulate new thoughts.

References

[1] J. A. Muhtadi, A. Ranganathan, R. Campbell, M. D. Mickunas, A Flexible, Privacy-Preserving Authentication Framework for Ubiquitous Computing Environments. In Proc. Of 22nd International Conference on Distributed Computing Systems Workshops (ICDC'02), pp. 771~776, 2002

[2] R. K. Thomas, R. Sandhu, Models, Protocols, and Architectures for Secure Pervasive Computing: Challenges and Research Directions. In Proc. Of 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW'04), pp. 164~170, 2004

[3] L. Bussard, Y. Roudier. Authentication in Ubiquitous Computing. In Proc. Of Workshop on Security in Ubiquitous Computing (Ubicom'02), pp. 1~5, 2002

[4] R. Campbell, J. Al-Muhtadi, P. Naldurg, G. Sampemane, M. D. Mickunas, Towards Security and Privacy for Pervasive Computing. In Proc. of International Symposium on Software Security, pp. 1~15, 2002

[5] K. Oishi, M. Mambo and E. Okamoto, Anonymous Public Key Certificates and Their Applications. IEICE Trans. Fundam. Electron. Commun. Comput. Sci., E81-A(1): 56~64, 1998

[6] P. Golle, M. Jakobsson, A. Juels, P. Syverson. Universal Re-encryption for Mixnets. In Proc. Of RSA Conference Cryptographers' Track'04, pp. 163~178, 2004

[7] B. R. Waters, E. W. Felten, A. Sahai, Receiver Anonymity via Incomparable Public Keys, In Proc. Of 10th ACM Conference on Computer Communication Security (CCS'03), pp. 112~121, 2003

[8] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography. CRC Press, 1996

[9] G. Ateniese, J. Camenisch, B. Medeiros, Untraceable RFID Tags via Insubvertible Encryption. In Proc. Of 12th ACM Conference on Computer Communication Security (CCS'05), pp. 92~101, 2005

[10] J. Camenisch, A. Lysyanskaya, Signature Schemes and Anonymous Credentials from Bilinear Maps. In Proc. Of CRYPTO'04, pp. 56~72, 2004

[11] A. Fiat, A. Shamir, How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In Proc. Of CRYPTO'86, pp. 186~194, 1986

[12] D. Chaum, J. H. Evertse, J. van de Graaf, An Improved Protocol for Demonstrating Possession of Discrete Logarithms and Some Generalizations. In Proc. Of EUROCRYPTO'87, pp. 127~141, 1987

[13] C. P. Schnorr, Efficient Identification and Signatures for Smart Cards, In Proc. Of CRYPTO’89, pp. 239~252, 1989

[14] T. Okamoto, An Efficient Divisible Electronic Cash Scheme. In Proc. Of CRYPTO’95, pp. 438~451, 1995

[15] A. Chan, Y. Frankel, Y. Tsiounis, Easy Come - Easy Go Divisible Cash. In Proc. Of EUROCRYPT’98, pp. 561~575, 1998

[16] L. Ballard, M. Green, B. Medeiros, F. Monrose, Correlation-Resistant Storage via Keyword-Searchable Encryption. Available online at http://eprint.iacr.org/2005/417.pdf

[17] G. Ateniese, J. Camenisch, S. Hohenberger, B. Medeiros, Practical Group Signatures without Random Oracles. Available online at http://eprint.iacr.org/2005/385.pdf

[18] A. Miyaji, M. Nakabayashi, S. Takano, New Explicit Conditions of Elliptic Curves for FR-reduction. IEICE Trans. Fundamentals, E84-A(5): 1234~1243, 2001

[19] D. Boneh, X.Boyen, Short Signatures Without Random Oracles. In Proc. Of Eurocrypt’04, pp. 56~73, 2004

[20] S. Mitsunari, R. Sakai, M. Kasahara, A New Traitor Tracing. IEICE Trans. Fundamentals, E85-A(2): 481~484, 2002

[21] D. Boneh, X.Boyen, H. Shacham, Short Group Signatures. In Proc. Of Crypto’04, pp. 41~55, 2004

[22] Y. Tsiounis, M. Yung, On the Security of ElGamal Based Encryption. In Proc. Of International Workshop on Practice and Theory in Public Key Cryptography (PKC'98), pp. 117~134, 1998

[23] D. Boneh, B. Lynn, H. Shacham, Short Signatures from the Weil Pairing. In Proc. Of Asiacrypt’01, pp. 514~532, 2001

[24] T. P. Pedersen, Non-interactive and Information-Theoretic Secure Verifiable Secret Sharing. In Proc. Of CRYPTO’91, pp. 129~140, 1991

[25] S. Schechter, T. Parnell, A. Hartemink, Anonymous Authentication of Membership in Dynamic Groups. In Proc. Of 3rd International Conference on Financial Cryptography (FC’99), pp. 184~195, 1999

[26] J. Furukawa, H. Imai, An Efficient Group Signature Scheme from Bilinear Maps. In Proc. Of ACISP’05, pp. 455~467, 2005

[27] R. Dingledine, N. Mathewson, P. Syverson, Reputation in P2P Anonymity Systems. In Proc. Of Workshop on Economics of P2P Systems, pp. 57~62, 2003

Appendix A

Here a sketch of proof for General Theorem is given. Assume $\mathcal{A}$ is a P.P.T. forger that breaks the General Theorem. P.P.T. algorithm $\mathcal{B}$ is constructed that interacts with $\mathcal{A}$ and eventually breaks q-SDH Assumption. The construction is similar to the proof of Lemma 3.2 in Boneh et al. [19].

It begins by giving $\mathcal{B}$ $(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, g_1, g_2, e)$, $(T_0 = g_2, T_1 = {g_2}^{a}, \cdots, T_q = {g_2}^{a^q})$, and isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ for unknown $a \in \mathbb{Z}_p$. The goal of algorithm $\mathcal{B}$ is to produce a solution $t^{a+x} = g_1$ for some $x \in \mathbb{Z}_p$. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows.

Query: $\mathcal{A}$ outputs a list of $q-1$ queries $x_1, x_2, \ldots, x_{q-1}$.

Response: $\mathcal{B}$ computes $f(\Upsilon) = \prod_{i=1}^{q-1}(\Upsilon + x_i) = \sum_{i=0}^{q-1} \alpha_i \Upsilon^i$ and

$$\widehat{g}_2 = \prod\nolimits_{i=0}^{q-1} T_i^{\alpha_i} = g_2^{\,f(a)},\ w = \prod\nolimits_{i=1}^{q} T_i^{\alpha_{i-1}} = g_2^{\,a\cdot f(a)} = \widehat{g}_2^{\,a}$$

Let $h_1 = \widehat{g}_1 = \psi(\widehat{g}_2) = g_1^{\,f(a)}$, $h_j = \widehat{g}_1^{\,r_j}$ for $r_j \in_R \mathbb{Z}_p$, $j = 2,3,\ldots,k$. Note that $(\widehat{g}_2, h_1, h_2, \ldots, h_k)$ has the correct probabilistic distribution, $\mathcal{A}$ will be fed with w and $(p, \mathbb{G}_1, \mathbb{G}_2, \mathcal{G}, \widehat{g}_2, e, h_1, h_2, \ldots, h_k)$. On receiving correct responses to its queries $x_1, x_2, \ldots, x_{q-1}$, $\mathcal{A}$ will output $t'^{a+x} = \prod\nolimits_{j=1}^{k} h_j$ for some different $x \in \mathbb{Z}_p$ at non-negligible probability.

For each $l = 1,2,\ldots,q-1$, $\mathcal{B}$ computes $f_l(\Upsilon) = f(\Upsilon)/(\Upsilon + x_l) = \sum\nolimits_{i=0}^{q-2} \beta_i \Upsilon^i$. Therefore $\mathcal{B}$ has

$$t_l' = \prod\nolimits_{i=0}^{q-2} T_i^{\beta_i} = g_2^{\,f_l(a)} = g_2^{\,f(a)/(a+x_l)} = \widehat{g}_2^{\,1/(a+x_l)} \text{ and } t_l = \psi(t_l')$$

Let $r_1 = 1$, $\mathcal{B}$ gives $t_l^{r_j}$ to $\mathcal{A}$ as response to query x_l , where $l = 1,2,\ldots,q-1$, $j = 1,2,\ldots,k$, because $(t_l^{r_j})^{(a+x_l)} = \widehat{g}_1^{\,r_j} = h_j$ holds, i.e. $e(t_l^{r_j}, w \cdot \widehat{g}_2^{\,x_l}) = e(h_j, \widehat{g}_2)$.

Output: $\mathcal{A}$ returns a forgery (t', x) that $t'^{a+x} = \prod\nolimits_{j=1}^{k} h_j$ where $x \notin \{x_1, x_2, \ldots, x_{q-1}\}$. Since $h_j = \widehat{g}_1^{\,r_j}$, $j = 1,2,\ldots,k$, $\mathcal{B}$ has $t'^{a+x} = \widehat{g}_1^{\,\sum_{j=1}^{k} r_j}$. Let $R = \sum\nolimits_{j=1}^{k} r_j$, congruence $t''^{a+x} = \widehat{g}_1 = g_1^{\,f(a)}$ holds where $t'' = t'^{1/R}$.

Let $f(\Upsilon) = (\Upsilon + x)\cdot \tilde{f}(\Upsilon) + \eta = (\Upsilon + x)\cdot \sum\nolimits_{i=0}^{q-2} \rho_i \Upsilon^i + \eta$ for $\eta \in \mathbb{Z}_p$ and $\eta \neq 0$. Note that $x \notin \{x_1, x_2, \ldots, x_{q-1}\}$, $\mathcal{B}$ computes $t = (t'' / \prod\nolimits_{i=0}^{q-2} \psi(T_i)^{\rho_i})^{1/\eta}$ and outputs (t, x) as its solution.

It's easy to verity that

$$\begin{aligned} t^{a+x} &= (t'' / \prod\nolimits_{i=0}^{q-2} \psi(T_i)^{\rho_i})^{(a+x)/\eta} \\ &= (t'' / g_1^{\,\tilde{f}(a)})^{(a+x)/\eta} \\ &= (t''^{(a+x)} / g_1^{\,(a+x)\cdot \tilde{f}(a)})^{1/\eta} \\ &= (g_1^{\,f(a)} / g_1^{\,f(a)-\eta})^{1/\eta} \\ &= g_1 \end{aligned}$$

□

Appendix B

Here a sketch of proof for Theorem 3 is given. First, let $y = h^x$ and $y_\sigma = h^\sigma$, congruence $e(y \cdot y_\sigma, g_2) = e(h, y_\sigma')$ actually shows that $y_\sigma' = g_2^{\,x+\sigma}$ where x is unknown to CA. If for some $t_y = t^{\widehat{x}}$, CA generates valid signature of knowledge proof $\overline{s}$, then

$$
\begin{aligned}
e(t, {g_2}^{x+\sigma}) &= e(t^{\widehat{x}}, g_2) \cdot e(t, g_2)^{\widehat{\sigma}} \\
e(t, {g_2}^{x+\sigma}) &= e(t, {g_2}^{\widehat{x}+\widehat{\sigma}}) \\
e(t, g_2)^{x+\sigma} &= e(t, g_2)^{\widehat{x}+\widehat{\sigma}} \\
x + \sigma &= \widehat{x} + \widehat{\sigma}
\end{aligned}
$$

where σ and $\widehat{\sigma}$ are known to CA.

If $x \neq \widehat{x}$, then $\log_h{}^y = \widehat{x} + \widehat{\sigma} - \sigma$. Note that $\widehat{x}$ may be known to CA in case CA masquerades as a normal peer, or $\widehat{x}$ may be known to some peer that colludes with CA. Therefore, CA or CA colluding with some peer is capable of solving discrete logarithm problem which contradicts the hardness of DLP in the setting of P.P.T. CA. On the other hand, if $x = \widehat{x}$ then $\sigma = \widehat{\sigma}$ which means that by selecting $\widehat{\sigma} = \sigma$ CA can prove that $x = \widehat{x}$ although CA indeed has no knowledge of x as well as $\widehat{x}$.

Finally, it should be noted that soundness of HVZK proof technique means success probability of a dishonest P.P.T. CA to fabricate $\overline{s}$ is negligible, which concludes the proof. □

An Efficient POP Protocol Based on the Signcryption Scheme for the WAP PKI

Sungduk Kim[1], Kyungjin Kim[2], Jaedong Jung[3], and Dongho Won[1,⋆,⋆⋆]

[1] Information Security Group, Sungkyunkwan University, Korea
netsec@naver.com, dhwon@security.re.kr
[2] Mobile Communication Division, Samsung Electronics, Korea
kjkim.kim@samsung.com
[3] IT Infra. Business Division, KOSCOM Corporate, Korea
jjd@koscom.co.kr

Abstract. WAP Forum recommends to use WTLS handshake protocol and signText() function to certify the POP (proof of possession) of authentication key and signing key. However, it causes plenty of computation and communication overload to mobile devices with low computation and communication power. In this paper, we propose an efficient POP confirmation protocol based on the signcryption scheme, which requires less computation and communication cost. It would be useful for the wireless and wired PKI. The proposed protocol is based on Zheng's signcryption scheme, because it is the first and only signcryption scheme submitted to the international standard institute(IEEE p1363).

Keywords: POP, proof of possession, signcryption.

1 Introduction

POP (proof of possession) confirmation is a cryptographic procedure that CA or RA certifies whether a certificate applicant possesses a proper private key accordance with a public key sent by a certificate applicant. Therefore, the detail steps may differ from the selected algorithm and the purpose of keys [1, 2, 10].

The well known standards relating to POP confirmation are RFC 4210 CMP (Certificate Management Protocols), and RFC 4211 (CRMF : Certificate Request Message Format) of IETF[1, 2, 3, 4]. However, such standards are too complicated to apply to small mobile devices such as cellular phones. There is a specific standard procedure for small devices which was released by the WAP Forum[10, 11].

In conformance with WAP specifications, the POP of authentication key is validated through the PKI portal which is based on a successful WTLS handshake which makes session key such as ECDH key. The POP of a signing key is a signature for the least part of the information (containing a challenge from the

⋆ This work was supported by the University IT Research Center Project funded by the Korean Ministry of Information and Communication.

⋆⋆ Corresponding author.

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 223–232, 2006.

PKI portal) and it is generated by the signText() function. Signature may be passed to the PKI portal in the form of signedContents. It consists of a signature and a self-signed certificate(or original signing certificate)[10, 11, 12].

Although the performance of mobile devices has improved rapidly, this approach causes much computation and communication cost and problems. Therefore, we propose an efficient POP confirmation protocol based on the signcryption scheme which is original and only signcryption scheme submitted to the international standard institute(IEEE p1363)[9].

This paper assumes the following situations :

- ECDSA and ECDH scheme are used in the mobile device.
- The mobile device has no user certificate.
- The user requests two certificates simultaneously, applies ECDH for the authentication key, and ECDSA for the signing key.

In this paper, we briefly introduce relating works in section 2(the POP procedure of WAP) and section 3(Signcryption scheme). We propose an efficient POP confirmation protocol based on signcryption scheme in section 4, and analyze the efficiency and security of the proposed protocol in section 5. Finally we make conclusions in section 6.

The following notations will be used in this paper, it is based on the ANSI X9.62[6].

- $X \parallel Y$: Concatenation of two strings X and Y
- n : The order of the base point G
- G : A distinguished point on an elliptic curve of large prime order n, called the base point or generating point
- d_{X_Y} : An elliptic curve private key, X means a owner, and Y means the purpose of key; S is signing, and KM is key management
- Q_{X_Y} : An elliptic curve public key, X means a owner, and Y means the purpose of key; S is signing key, and KM is key management key
- $Cert_{X_Y}$: An elliptic curve public key certificate, X means a owner, and Y means the purpose of key; S is signing key, and KM is key management key
- H : a one-way hash function
- KH_X : A keyed one way hash function. X is a key value
- $[x, y]$: The interval of integers between and including x and y
- $barx$: Convert the field element x to an integer
- ID/PW : one time ID and password issued by PKI portal

2 The POP Confirmation Procedure of WAP

According to the definition of WAP PKI, the PKI portal considers that the user obtain POP confirmation of authentication key when one executes the WTLS protocol and successfully logs in to the PKI portal. The PKI portal considers that the user obtain a POP confirmation for a signing key when a signature of signedContent which is generated by the signText() function is verified by the PKI portal.

If the user wants to submit both POPs, one must follow the WTLS protocol with a self-signed certificate then generates a signature with a self-signed certificate.

WAP PKI specification requires the certificate form to transfer an applicant's public key. Therefore applicant should make two self-signed certificates.

The POP validation procedure for ECDH key and ECDSA key are as follows [5, 6, 10, 11].

Table 1. The POP Confirmation of WAP for Authentication Key and Signing Key

Client(Applicant)	Communication	PKI Portal(CA)
[Client Hello]	$\rightarrow R_A \rightarrow$	[ServerHello]
$R_A \in_R [1, n-1]$		$R1_{CA} \in_R [1, n-1]$
		[Certificate]
		[CertificateRequest]
	$\leftarrow R1_{CA}, Cert_{CA_{km}} \leftarrow$	[ServerHelloDone]
$[Certificate]$		
$[CertificateVerify]$		
$K_P = d_{A_{km}} Q_{CA_{km}}$		
$K_m = H(K_p, R_A, R_{CA})$		
[ChangeCipherSpec]		
[Finished]	$\rightarrow SelfCert_{A_{km}} \rightarrow$	$K_P = d_{CA_{km}} Q_{A_{km}}$
		$K_m = H(K_p, R_A, R_{CA})$
		[ChangeCipherSpec]
[ApplicationData]	$\leftarrow WTLS\ Message \leftarrow$	[Finished][ApplicationData]
	$\rightarrow start\ signal \rightarrow$	
	$\leftarrow R2_{CA} \leftarrow$	$R2_{CA} \in_R [1, n-1]$
$k \in_R [1, n-1]$		
$Q_k = (x_k, y_k) = kG$		
$r = \overline{x_k}\ (mod\ n)$		
$C = R2_{CA} \| ID \| PW$		
$c = H(C)$		
$s = (c + r \cdot d_{A_S})k^{-1}\ (mod\ n)$	$\rightarrow C,\ r,\ s \rightarrow$	Check : $R2_{CA}$
	$SelfCert_{A_S}$	$u = c \cdot s^{-1}$
		$Q_k = u \cdot G + r \cdot s^{-1} Q_{A_S}$
		$Check : r = \overline{x_k}\ (mod\ n)$
		$MakeCertificate :$
	$\leftarrow Cert(Q_{A_S}) \leftarrow$	$Cert(Q_{A_S}), Cert(Q_{A_{km}})$
	$Cert(Q_{A_{km}})$	

2.1 POP Validation for ECDH Key

The client inputs his/her ID and password(PW) then starts the WTLS handshake protocol by sending the [ClientHello] message which includes a random number of the client(R_A) to the PKI portal. Then the PKI portal consecutively sends a [ServerHello] message which includes a random number of the PKI portal($R1_{CA}$), a [Certificate] message which holds the certificate of the PKI portal($Cert_{CA_{km}}$), a [CertificateRequest] message, and a [ServerHelloDone] message to the client. [CertificateRequest] message refers to the PKI portal requests the client's certificate.

After the client receives these messages, the client verifies the PKI portal's certificate, makes his own self-signed authentication certificate($SelfCert_{A_{km}}$) and sends it to the PKI portal through [Certificate] message. The PKI portal verifies a client's selfsigned certificate.

The client and PKI portal generates the same WTLS session key(K_m) based on the ECDH key(K_p), their public keys($Q_{A_{km}}$, $Q_{CA_{km}}$), and random numbers (R_A, $R1_{CA}$).

The [ChangeCipherSpec] message is sent by the client to notify the PKI portal where subsequent records will be encrypted with the negotiated session key(K_m). The [Finished] is the first encrypted message with the negotiated session key. The PKI portal then decrypts its contents. If there are no problems, the PKI portal considers the client has a secret key and the capability of processing a key agreement procedure based on the ECDH key.

2.2 POP Validation for ECDSA Key

POP validation procedures are conducted on the previously settled WTLS session. If the PKI portal sends a one time random number($R2_{CA}$) to the client, it generates a signature for $R2_{CA}$+":"+Name+":"+ID+":"+Password with a signText() function of WAP PKI specification and issues a self-signed public key certificate for the signing key. Then the client converts the result in the form of signedContent and sends it to the PKI portal through the WTLS session.

Request=Crypto.signText($R2_{CA}$+":"+Name+":"+ID+":"+Password, 5, 0);

The PKI portal verifies the client's signature with the public key of the self-signed certificate within the signedContent. If there is no error, the PKI portal convinces that the client has a proper secret key for the signing.

The POP confirmation method of WAP uses WTLS handshake protocol and signedText() function defined in the WAP PKI specification. It is necessary that the client makes self-signed certificates for each key. The WAP Forum does not define the process nor the function for making a self-signed certificate, and it is also unnecessary after POP confirmation. Although it is a temporary function, the mobile device vendor should develop a module according to the certificate type.

3 Signcryption Scheme

The signcryption scheme was proposed by Zheng in 1997[7, 8, 9]. It simultaneously provides confidentiality, authentication, and non-repudiation(digital signature) with low communication and computation cost. This scheme converges a key agreement and a digital signature to a single process.

[Table 2] describes detail steps of Zheng's signcryption scheme for the elliptic curve cryptography.

Table 2. Zheng's Signcryption Scheme

Alice(Signer)	Communication	Bob(Verifier)
$v \ \in_R [1, n-1]$ $(k_1,\ k_2) \ = \ H(v \cdot Q_B)$ $c \ = \ E_{k_1}(M)$ $r \ = \ KH_{k_2}(M \Vert BindInfo)$ $[SECDSS1]$ $s \ = \ v(r + d_A)^{-1} \ (mod\ n)$ $[SECDSS2]$ $s \ = \ v(1 + r \cdot d_A)^{-1} \ (mod\ n)$	$\rightarrow (c,\ r,\ s) \rightarrow$	$u \ = \ s \cdot d_B \ (mod\ n)$ $[SECDSS1]$ $T \ = \ uQ_A + urG \ (mod\ n)$ $[SECDSS2]$ $T \ = \ uG + urQ_A \ (mod\ n)$ $(k_1,\ k_2) \ = \ H(T)$ $M \ = \ D_{k_1}(c)$ $Check:$ $r \ = \ KH_{k_2}(M \Vert BindInfo)$

Upon a successful signcryption scheme, only the designated recipient can decrypt the encrypted message with his/her private key. After Zheng's contribution, various signcryption schemes have been proposed. Some signcryption schemes have a property that signcryption can be publicly verified without a recipient's private key[13].

Zheng's signcryption scheme is not a publicly verifiable scheme. Recipient must execute zero knowledge interactive protocol to prove the validation of signcryption without revealing his/her private key. However, publicly verifiable signcryption schemes is not necessary for the PKI portal's POP confirmation, because there is no case that the PKI portal proves the applicant's POP to the third party. Generally, a non publicly verifiable signcryption scheme is more efficient than a publicly verifiable one[13].

4 The POP Protocol Based on the Signcryption Scheme

4.1 Motivation

In general, signcryption schemes are based on DLP or ECC using a random number(v) selected by the sender for each session. The sender keeps it secure and the recipient can not find any more information from signcryption except g^v or $v \cdot G$. Our major motivation is using the sender's authentication private key as a random number for the POP of signing key.

The following describes Zheng's signcryption scheme for ECC.

$[SECDSS1] \ T \ = \ u \cdot Q_A + u \cdot r \cdot G \ = \ v \cdot Q_B$

$[SECDSS2] \ T \ = \ u \cdot G + u \cdot r \cdot Q_A \ = \ v \cdot Q_B$

If we replace value v with sender's authentication private key($d_{A_{km}}$), sender and recipient can generate a same value $Q_{AB}(= \ d_{A_{km}} \cdot d_{B_{km}})$.

$[SECDSS1] \ T \ = \ u \cdot Q_{A_S} + u \cdot r \cdot G \ = \ d_{A_{km}} \cdot d_{B_{km}} G$

$[SECDSS2] \ T \ = \ u \cdot G + u \cdot r \cdot Q_{A_S} \ = \ d_{A_{km}} \cdot d_{B_{km}} G$

Q_{AB} is well-known ECDH value and where the recipient confirms that sender has a proper private key and capability for processing ECDH key.

Of course, the Zheng's signcryption scheme is one of the digital signature scheme which has the property that the designated recipient is able to decrypt

Table 3. POP Confirmation Protocol Based on Signcryption Scheme

Applicant	Communication	PKI portal
$d_{A_S}, d_{A_{km}} \in_R [1, n-1]$		
$Q_{A_S} = d_{A_S}G, Q_{A_{km}} = d_{A_{km}}G$	$\rightarrow$ start signal $\rightarrow$	
Verify CA Certificate	$\leftarrow Cert_{CA_{km}} \leftarrow$	$R_{CA} \in_R [1, n-1]$
$(k_1, k_2) = H(d_{A_{km}}Q_{CA_{km}} \Vert R_{CA})$	R_{CA}	
$M = ID \Vert Q_{A_{km}}$		
$EM = E_{k_1}(M)$		
$c = Q_{A_S} \Vert EM$		
$r = KH_{k_2}(PW \Vert c)$		
$[SECDSS1]$		
$s = d_{A_{km}}(r + d_{A_S})^{-1}\ (mod\ n)$		
$[SECDSS2]$		
$s = d_{A_{km}}(1 + r \cdot d_{A_S})^{-1}(mod\ n)$	$\rightarrow (c,\ r,\ s) \rightarrow$	$u = s \cdot d_{CA_{km}}(mod\ n)$
		Separate Q_{A_S} and EM from c
		$[SECDSS1]T = (uQ_{A_S} + urG)$
		$[SECDSS2]T = (uG + urQ_{A_S})$
		$(k_1,\ k_2) = H(T \Vert R_{CA})$
		$(ID \Vert Q_{A_{km}}) = D_{k_1}(EM)$
		Retrieve PW from DB
		Check : $r = KH_{k_2}(PW \Vert c)$
		Check : $T = d_{CA_{km}}Q_{A_{km}}$
		Make Certificate :
	$\leftarrow Cert(Q_{A_S}) \leftarrow$	$Cert(Q_{A_S}), Cert(Q_{A_{km}})$
	$Cert(Q_{A_{km}})$	

the signature and verify the result. Therefore, the PKI portal can regard a signcryption as a POP for signing key.

4.2 POP Generation

The following steps are POP generation procedures for the certificate applicant [Table 3].

1. Register to CA(RA) and gets an ID and password pair.
2. Randomly choose two numbers $(d_{A_S}, d_{A_{km}} \in_R [1, n-1])$: signing private key(d_{A_S}) and authentication private key($d_{A_{km}}$)
3. Generate public keys : signing public key($Q_{A_S} = d_{A_S}G$) and authentication public key($Q_{A_{km}} = d_{A_{km}}G$)
4. Send start signal then receive a random number(R_{CA}) and authentication public key certificate from the PKI portal.
5. Verify the PKI portal's certificate
6. Generate session keys(k_1, k_2) from the hash value of scalar multiplication using the applicant's authentication private key($d_{A_{km}}$), the PKI portal's public key($Q_{CA_{km}}$), and a random number from the PKI portal(R_{CA}).
7. Encrypt ID and $Q_{A_{km}}$) with k_1
8. Concatenate encrypted value and Q_{A_S}.
9. Hash (password, Q_{A_S}, ID, R_{CA}, $Q_{A_{km}}$) with key(k_2) and set the result as r.

10. Make signcryption according to the Zheng's signcryption scheme : SECDSS1 or SECDSS2.
11. Send a signcryption(c, r, s) to the PKI portal.

Step 6 is procedure for generating an ECDH key and secondary secure keys (k_1, k_2).

Step 9 produces a secret key(k_2) based password authentication code. Only proper certificate applicant who has password and private key calculates it.

In step 10, applicant generates a sign(r, s) with a signing key. This sign is verified by the PKI portal for POP confirmation.

4.3 POP Confirmation

The PKI portal confirms the applicant's POP by the followings [Table 3].

1. Receive a start signal, make a random number(R_{CA}) and send it to the client.
2. Receive a signcryption and compute u by multiplying the PKI portal's authentication private key($d_{CA_{km}}$) by signcryption s.
3. Separate applicant's signing public key(Q_{A_S}) and encrypted message(EM) from signcryption c.
4. Make T according to Zheng's signcryption scheme and hash it with random number(R_{CA}). Then set it (k_1, k_2).
5. Decrypt c with k_1, and set the result as ID, $Q_{A_{km}}$.
6. Retrieve a password from database according to the ID.
7. Compute $KH_{k_2}(PW||c)$ and check whether the result is the same as r received from applicant.
8. Compute $d_{CA_{km}}Q_{A_{km}}$ and verify the result is the same as T calculated above.
9. Send the signing key certificate($Cert(Q_{A_S})$) and key agrement key certificate $(Cert(Q_{A_{km}}))$, to the applicant.

The PKI portal performs above procedures except step 9 to confirm possession of signing key. It is a verification process of Zheng's signcryption.

Computation process for certifying possession of authentication key is step 4 and 8. Through step 2, 3, and 4, the PKI portal calculates an ECDH key.

$[SECDSS1]T = (uQ_{A_S} + urG) = u(d_{A_S} + r)G = d_{CA_{km}}d_{A_{km}}(d_{A_S} + r)^{-1}$ $(d_{A_S} + r)G = ECDH$

$[SECDSS2]T = (uG + urQ_{A_S}) = u(1 + rd_{A_S})G = d_{CA_{km}}d_{A_{km}}(1 + rd_{A_S})^{-1}$ $(1 + rd_{A_S})G = ECDH$

Then the PKI portal computes an ECDH key with his authentication private key and the client's authentication public key, then compares it with T.

5 Properties

5.1 Computation Cost

In conformance with WAP, the client should generate three ECDSA signs (1 for POP and 2 for self-signed certificate) and one ECDH key, and verifies one

ECDSA sign for the PKI portal's authentication key certificate. However, on the proposed method, the client generates a ECDH key and a signature simulataneously through the signcryption scheme without making self-signed certificates and verifies one ECDSA sign for the PKI portal's authentication key certificate.

In the ECC, most of the time consumption process is the scalar multiply operation. ECDSA sign and ECDH key generation consists of one scalar multiplication in each. However, ECDSA verification consists of two scalar multiplications. On the 80MHz ARM7TDMI processor with 192 bit ECC, one scalar multiplication takes 38ms[14].

If a mobile device follows the WAP PKI, the client should carry out six scalar multiplications which takes about 228ms. However, the proposed protocol requires three scalar multiplications and takes about 114ms. The proposed protocol reduces the client's computation cost to 50%.

WAP requests the PKI portal performs seven scalar multiplications(three ECDSA verification and one ECDH key generation). However, on the proposed method suggested that the PKI portal carries out three scalar multiplications in verifying of a signcryption. Thus the proposed protocol reduces the server's computation cost to 43%.

Table 4. Computation Cost Comparison

Computation Cost		WAP	Proposed Protocol
Client	Scalar Multiplication	2	1
	Hash	2	2
	Inverse	1	1
	Multiplication	2	1
	Making Self-signed Cert	2	0
	Verifying Cert	1	1
Server	Scalar Multiplication	3	3
	Hash	2	2
	Inverse	1	0
	Multiplication	2	2
	Verifying Self-signed Cert	2	0

5.2 Communication Cost

Let $|a|$ the length of message a. In conformance with WAP, the client and PKI portal transmits the $5|n| + 3|Certificate| + |H| + |ID| + |PW|$ length messages without the client's certificates issued by the PKI portal and extra WTLS messages. The POP processes consist of 4 times interactions(send and receive) between the client and PKI portal.

However, with the proposed method, the client and PKI portal communicate with one random number, one certificate, and one signcryption. Thus, the client and the PKI portal send $2|n| + |Certificate| + |H| + |ID| + 2|Q|$ length message. And the POP processes consist of 2 times interaction between the PKI portal and client.

Therefore the proposed protocol reduces $3|n| + 2|Certificate| + |PW|$ length and increases only $2|Q|$ length.

5.3 Security

The security assessment and known limitation of Zheng's signcryption scheme are described in [9], and the proposed protocol is based on it.

The proposed protocol will be identical to Zheng's EC signcryption scheme, if we regard the authentication private key($d_{A_{km}}$) as the random number(v) of Zheng's signcryption scheme, while excluding the PKI portal's random number(R_{CA}) from protocol.

On the Zheng's signcryption scheme, it is infeasible for both the verifier and the adversary to find and/or forge the random number(v) signer selected. Therefore, we can use the random number(v) as a authentication private key($d_{A_{km}}$) without any risk.

The PKI portal's random number(R_{CA}) in the proposed protocol is just used to prevent the adversary's replay attack, and applied to the input value of hash function. The adversary should know the applicant's authentication private key to forge the applicant's signcryption, even if he can predict or capture the PKI portal's random number. Therefore, the PKI portal's random number(R_{CA}) does not decrease the strength of proposed scheme's security.

If the Zheng's signcryption scheme is secure, the proposed scheme is infeasible for both the PKI portal and the adversary to find and/or forge the authentication private key($d_{A_{km}}$) and signing private key.

5.4 Availability of Signcryption Scheme

Usually, mobile phones have low computation power. Therefore, the less computation cost required, the more efficiency can be received. Mobile phone must have a function for generating a self-signed certificate to achieve the POP confirmation according to the WAP specification. However there are no specified functions or procedures within the WAP PKI. Thus, each vendor has to generate a temporary certificate proper for a mobile phone environment. We expect that this function will not be used frequently except POP procedure. However, the signcryption will be very useful to mobile phone because it needs small computation and communication cost. If we define the signcryption generation function in signText() or signcryptionText() function separately, it will be very useful for POP as well as other services.

6 Conclusions

In this paper, we propose an efficient POP protocol based on the signcryption scheme. It reduces the computation cost to half of its level, and decreases communication overload in comparison with WAP PKI. Even though the performance of small wireless device has improved rapidly, computation and communication overload still causes several problems. Therefore, the proposed protocol will be very useful for mobile devices to process the certificate request message. Naturally, the proposed protocol can be applied to wired PKI with powerful devices.

The proposed scheme based on Zheng's signcryption scheme. After Zheng proposed signcryption scheme, various signcryption scheme has been proposed.

Recent signcryption schemes can be found on [13]. If the signcryption scheme uses a random number except a private key and keeps it secure, you can be applied that to POP protocol. If you want to add some special properties to POP protocol, simply select a proper signcryption scheme.

References

1. C. Adams, S. Farrell, T. Kause, T. Mononen"Internet X.509 Public Key Infrastructure Certificate Management Protocols", IETF RFC 4210, September 2005.
2. J. Schaad "Internet X.509 Certificate Request Message Format", IETF RFC 4211, September 2005.
3. Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed Hashing for Message Authentication", RFC 2104, February 1997.
4. Cheng, P. and R. Glenn, "Test Cases for HMAC-MD5 and HMAC-SHA-1", RFC 2202, September 1997.
5. FIPS PUB 186-2 Digital Signature Standard, 2000 January 27
6. ANSI X9.62 "Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm(ECDSA)
7. Y. Zheng. Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature) + cost (encryption). In: CRYPTO'97, LNCS 1294, pages 165-179. Springer-Verlag, 1997.
8. Y. Zheng. Signcryption and its application in efficient public key solution. In: Information Security Workshop (ISW'97), LNCS 1397, pages 291-312. Springer-Verlag, 1998.
9. IEEE P1363a : Statndard Specifications for Public-Key Cryptography : Additional Techniques, "Shortened Digital Signature, Signcryption and Compact and Unforgeable Key Agreement Schemes", Yuliang Zheng, 1998.
10. Wireless Application Protocol Public Key Infrastructure Definition, WAP-217-WPKI Version 24-Apr-2001
11. Wireless Application Protocol WMLScript Crypto Library, WAP-161-WMLScriptCrypto-20010620-a Version 20-Jun-2001
12. Dong Jin Kwak, Jae Cheol Ha, Hoon Jae Lee, Hwan Koo Kim, Sang Jae Moon, "A WTLS Handshake Protocol with User Anonymity and Forward Secrecy", Mobile Communications: 7th CDMA International Conference, CIC 2002, Seoul, Korea, October 29 - November 1, 2002. Revised Papers, LNCS 2524, pp.219-230.
13. Signcryption Central, http://www.signcryption.net/publications/
14. MIRACL(Multiprecision Integer and Rational Arithmetic C/C++ Library) of Shamus Software Ltd, http://indigo.ie/ mscott/

On the Resilience of Key Agreement Protocols to Key Compromise Impersonation

Maurizio Adriano Strangio

University of Rome "Tor Vergata", Rome, Italy
strangio@disp.uniroma2.it

Abstract. Key agreement protocols are a fundamental building block for ensuring authenticated and private communications between two parties over an insecure network. This paper focuses on key agreement protocols in the asymmetric trust model, wherein parties hold a public/private key pair. In particular, we consider a type of known key attack called key compromise impersonation that may occur once the adversary has obtained the private key of an honest party. This attack represents a subtle threat that is often underestimated and difficult to counter. Several protocols are shown vulnerable to this attack despite their authors claiming the opposite. We also consider in more detail how three formal (complexity-theoretic based) models of distributed computing found in the literature cover such attacks.

Keywords: key compromise impersonation, key agreement protocols.

1 Introduction

Key agreement protocols are a fundamental building block for ensuring authenticated and private communications between two parties over an insecure network. We consider the asymmetric trust model wherein parties hold a long-term private/public key pair (and thus can establish each others true identity by exchanging digital certificates issued by a trusted Certification Authority).

The private key must be handled with care (e.g. stored on a tamper-proof storage token) to prevent it from falling into the hands of a malicious party since it is used across all communication sessions. On the other hand, ephemeral session-specific data is used once and is therefore simply discarded when (or even before) a session terminates.

A (two-party) key agreement protocol should provide some form of assurance regarding the authenticity of the session key as well as ensuring that the key is known only by the two intended participants at the end of the protocol execution. This includes the requirement that the protocol must also provide some form of entity authentication (e.g. by adding the digital certificates to the message flows).

The notions of "authenticated key agreement" considered within the compass of this article are the following: *Authenticated Key* (AK) agreement whereby a party A is assured that no one aside from the identified principal B (the intended partner of A in the communication) can possibly learn the value of the session key and, *AK with key Confirmation* (AKC), whereby party A is assured that B (and/or vice versa) has actually computed (or knows how to compute) the session key (see [8, 5] for further discussion).

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 233–247, 2006.

We will consider only implicitly authenticated AK protocols, i.e. authentication is complete when both principals have successfully concluded a run of the protocol and have then proved knowledge of the session key in subsequent communication.

As usual, we designate the two generic parties participating in a protocol run as Alice and Bob. Suppose an adversary (say Eve) was able to obtain the private key of Alice either by compromising the machine running an instance of the protocol (e.g. when the key is stored in conventional memory as part of the current state) or perhaps by cloning Alice's smart card while she inadvertently left it unattended. Eve may now be able to mount the following attacks against the protocol:

1. impersonate Alice in a protocol run;
2. impersonate a different party (e.g. Bob) in a protocol run with Alice;
3. obtain previously generated session keys established in honest-party runs of the protocol.

In case 1. Eve can send messages on behalf of Alice and these will be accepted as authentic, in case 2. Eve could establish a session with Alice while masquerading as another party; this is known as Key Compromise Impersonation (KCI) and seems to appear for the first time in [17]. For example, Eve could impersonate a banking system and cause Alice to accept a predetermined session key and then obtain her credit card number over the resulting private communication link. In case 3. Eve may be able to decrypt the data exchanged by Alice and Bob in previous runs of the protocol (provided the transcripts are known).

The discussion above demonstrates that long-term key compromise can lead to undesirable consequences (at least until the involved principal discovers that his key was compromised). However, protocol designers are often concerned with forward secrecy and seem to ignore key compromise impersonation.

The main thesis of this paper is that key compromise impersonation is not less important than forward secrecy; one should require that a secure key agreement protocol be also KCI-resilient since this security attribute is also related to party corruption.

In Section 4 we show that several implicitly authenticated key agreement protocols found in the literature do not withstand KCI attacks despite the authors claims. In order to offer a simplified and uniform treatment, all the protocols considered are specified in an elliptic curve setting since most of them were originally conceived in EC-based groups. In Section 5 we consider in more detail how three formal (complexity-theoretic based) models of distributed computing cover such attacks.

2 Notation and Mathematical Background

Given two strings s_1, s_2, the symbol $s_1 \| s_2$ denotes string concatenation. If X is a finite set then $x \overset{R}{\leftarrow} X$ denotes the sampling of an element uniformly at random from X. If α is neither an algorithm nor a set $x \leftarrow \alpha$ represents a simple assignment statement. The hash function $\mathcal{H}$ is used as a key derivation function (see [15] for practical KDFs).

The protocols we consider are based on EC cryptosystems. Let $\mathbb{F}_q$ denote the finite field containing q elements, where q is a prime power ($q = p$ or $q = 2^m$). An elliptic curve $\mathrm{E}(\mathbb{F}_q)$ over the field $\mathbb{F}_q$ (for simplicity we let $q = p$ with p a prime greater 3) is the set of points $P \equiv (P.x, P.y)$ that satisfy an (Weierstrass) equation of the form:

$$y^2 = x^3 + ax + b, \quad 4a^3 + 27b^2 \neq 0$$

where $a, b \in \mathbb{F}_q$. The set $\mathrm{E}(\mathbb{F}_q)$ with the operation of *point addition* $Q = P + R$ defined according to a *chord-and-tangent* rule and the point at infinity P_∞ serving as the identity element forms an (abelian) group structure. Repeated addition of a point P to itself x times is known as *scalar multiplication* $Q = xP$, this operation often dominates the execution time of elliptic curve based cryptographic schemes. The public elliptic curve domain parameters over $\mathbb{F}_q$ are specified by an 8-tuple:

$$\Phi_{EC} \equiv (q, FR, S, a, b, P, n, h)$$

where q is the underlying field order, FR (*field representation*) is an indication of the method used to represent field elements in $\mathbb{F}_q$, the *seed* S is for randomly generated elliptic curves, the two *coefficients* $a, b \in \mathbb{F}_q$ define the equation of the elliptic curve E over $\mathbb{F}_q$, the *base point* $P = (P.x, P.y)$ in $\mathrm{E}(\mathbb{F}_q)$, the prime order n of P and the cofactor $h = \sharp\mathrm{E}(\mathbb{F}_q)/n$. There are several well known algorithms for validating the parameters Φ_{EC} (see [12, 13]).

In an elliptic curve public-key cryptosystem user A is assigned a key pair (w_A, W_A) which is compatible with the set of domain parameters Φ_{EC}. In practice, the *private key* is a randomly selected value w_A in $[1, n-1]$ and the corresponding *public key* is the elliptic curve point $W_A{=}w_A P$.

There exist elliptic curve groups where the discrete logarithm assumption is known to hold: given the generator P and a random element $X \in \mathrm{E}(\mathbb{F}_q)$ it is computationally hard to compute $\log_P X$. More formally,

Assumption 1 (ECDL). *The EC group* $\mathrm{E}(\mathbb{F}_q)$ *satisfies the discrete logarithm assumption if for all PPT algorithms* $\mathcal{A}$ *we have:*

$$\Pr\left[x \xleftarrow{R} \mathbb{F}_q^*; X \leftarrow xP : \mathcal{A}(\Phi_{EC}, X){=}x\right] < \epsilon$$

where the probability is taken over the coin tosses of $\mathcal{A}$ *(and random choice of* x*) and* ϵ *is a negligible function.*

The elliptic curve computational Diffie-Hellman (ECCDH) assumption holds in the group $\mathrm{E}(\mathbb{F}_q)$ if for random elements $X, Y \in \mathrm{E}(\mathbb{F}_q)$ it is computationally hard to compute $\log_P X \log_P YP$ (i.e. if $X = xP$ and $Y = yP$ then the output should be xyP).

Assumption 2 (ECCDH). *The EC group* $\mathrm{E}(\mathbb{F}_q)$ *satisfies the computational Diffie-Hellman assumption if for all PPT algorithms we have:*

$$\Pr\left[x \xleftarrow{R} \mathbb{F}_q^*; y \xleftarrow{R} \mathbb{F}_q^*; X \leftarrow xP; Y \leftarrow yP : \mathcal{A}(\Phi_{EC}, X, Y){=}xyP\right] < \epsilon$$

where the probability is taken over the coin tosses of $\mathcal{A}$ *(and random choices of* x, y*) and* ϵ *is a negligible function.*

3 A Closer Look at Key Compromise Impersonation

A KCI attack involves an adversary that has obtained the private key of an honest party. The adversarial goal is then to impersonate a different user and try to establish a valid session key with the "corrupted" party. This attack represents a serious and subtle threat since a user may not even be aware that his computer was "hijacked" and that a malicious party has obtained his private key. Set out below is a formal definition of KCI resilience:

Definition 1 (KCI-resilience). *A key agreement protocol is* KCI-resilient *if compromise of the long-term key of a specific principal does not allow the adversary to establish a session key with that principal by masquerading as a different principal.*

In the real world, a KCI attack is carried through as a man-in-the-middle attack. Let us consider the Unified Model [2, 14] *AK* protocol described in Figure 1. Suppose adversary E knows w_A. Message Q_A is delivered to its intended recipient B without any interference from the adversary. Now, E intercepts B's response Q_B and substitutes it with $Q_E = r_E P$. As a result, E causes A to accept the session key $\mathcal{H}(w_A W_B \| r_A Q_E)$ and is able to compute the same key as $\mathcal{H}(w_A W_B \| r_E Q_A)$. Furthermore, for this protocol, the attack works in exactly the same way if the adversary corrupts B.

Although the above example seems a trivial one, it is useful because it draws our attention on at least two important points: (1) many protocols are designed without considering KCI resilience as a security goal; (2) the corrupted party may not be able to detect the attack since a message received by the adversary (impersonating a

$$
\begin{aligned}
A &: r_A \overset{R}{\leftarrow} [1, n-1] \\
&\quad Q_A \leftarrow r_A P \\
A \rightarrow B &: Q_A \\
B &: r_B \overset{R}{\leftarrow} [1, n-1] \\
&\quad Q_B \leftarrow r_B P \\
B \rightarrow A &: Q_B \\
A &: sk_A \leftarrow w_A W_B \| r_A Q_B \\
B &: sk_B \leftarrow w_B W_A \| r_B Q_A
\end{aligned}
$$

Fig. 1. Protocol UM

$$
\begin{aligned}
A &: r_A \overset{R}{\leftarrow} [1, n-1] \\
&\quad Q_A \leftarrow r_A P \\
A \rightarrow B &: Q_A \\
B &: r_B \overset{R}{\leftarrow} [1, n-1] \\
&\quad Q_B \leftarrow r_B P \\
B \rightarrow A &: Q_B \\
A &: sk_A \leftarrow \mathcal{H}(r_A W_B + w_A Q_B) \\
B &: sk_B \leftarrow \mathcal{H}(r_B W_A + w_B Q_A)
\end{aligned}
$$

Fig. 2. Protocol MTI/A0

$$\begin{aligned}
A :&\ r_A \overset{R}{\leftarrow} [1, n-1] \\
&\ Q_A \leftarrow r_A P \\
A \rightarrow B :&\ Q_A \\
B :&\ r_B \overset{R}{\leftarrow} [1, n-1] \\
&\ Q_B \leftarrow r_B P \\
B \rightarrow A :&\ Q_B \\
A :&\ s_A \leftarrow r_A + \overline{Q}_A w_A \\
&\ sk_A \leftarrow s_A(Q_B + \overline{Q}_B W_B) \\
B :&\ s_B \leftarrow r_B + \overline{Q}_B w_B \\
&\ sk_B \leftarrow s_B(Q_A + \overline{Q}_A W_A)
\end{aligned}$$

Fig. 3. Protocol MQV

legitimate user and not constrained to follow the protocol specification exactly) is perfectly indistinguishable (e.g. message Q_E above) from one received by an honest party.

We now turn to examine the MTI/A0 (Figure 2, [21]) and the MQV protocols (Figure 3, [19]) since they are apparently immune to KCI attacks. For both these implicitly authenticated key agreement protocols it appears to be infeasible to setup an attack (that exploits the algebraic group structure), similar to those presented in Section 4, with the only information known by the adversary being the long-term private key of a party. Indeed, for the MTI/A0 protocol Eve should be able to find a value Q_E such that the session key computed by A as $r_A W_B + w_A Q_E$ can also be calculated by an adversary knowing w_A. However, this does not seem possible unless either one of r_A or w_B are available to Eve. Similar reasoning also applies to the MQV protocol where the use of a non standard function[1] destroys the algebraic structure of the group. To be honest, resistance to KCI attacks per se was not a design goal of the MQV protocol (and perhaps neither for the MTI/A0 protocol). In fact, the authors claim that the computation of $\overline{Q}$ (for a group element Q) was introduced for increased efficiency.

Notice that the protocols are easily broken if the adversary obtains the ephemeral data used by A, B (e.g. r_A, r_B and any other session-specific information stored in the current state); for this to occur, either the adversary is able to solve an instance of the discrete logarithm problem in an EC group (see Section 2) or she is given the capability of compromising a principals' machine (therefore obtaining the states of all running protocol instances at that time). The later case amounts to a stronger corruption model (which is also harder to put into practice) than the one we consider in this paper.

4 Cryptanalysis of KCI-Resilient AK Protocols

In this section we illustrate successful KCI attacks brought against several implicitly authenticated AK key agreement protocols despite their authors have claimed resilience against such attacks. These are efficient one-round protocols with desirable communication and computational complexity.

[1] Recall that if Q ($\neq P_\infty$) is an elliptic curve point and $\overline{x}$ denotes the integer obtained from the binary representation of $Q.x$, then $\overline{Q}$ is the integer given by ($\overline{x}$ mod $2^{\lceil f/2 \rceil}$) + $2^{\lceil f/2 \rceil}$ where $f = \lfloor \log_2 n \rfloor + 1$.

$$\begin{array}{rl}
A : & r_A \overset{R}{\leftarrow} [1, n-1] \\
& Q_A \leftarrow r_A W_B \\
A \rightarrow B : & Q_A \\
B : & r_B \overset{R}{\leftarrow} [1, n-1] \\
& R_B \leftarrow r_B W_A \\
& T_B \leftarrow Q_A + R_B \\
& Q_B \leftarrow R_B + r_B w_B^{-1} Q_A \\
B \rightarrow A : & Q_B \\
A : & T_A \leftarrow Q_A + w_A (w_A + r_A)^{-1} Q_B \\
& sk_A \leftarrow T_A \\
B : & sk_B \leftarrow T_B
\end{array}$$

Fig. 4. Protocol LLK

$$\begin{array}{rl}
A : & r_A \overset{R}{\leftarrow} [1, n-1] \\
& Q_A \leftarrow r_A P \\
A \rightarrow B : & Q_A \\
B : & r_B \overset{R}{\leftarrow} [1, n-1] \\
& Q_B \leftarrow r_B P \\
B \rightarrow A : & Q_B \\
A : & sk_A \leftarrow r_A W_B + (w_A + r_A) Q_B \\
B : & sk_B \leftarrow r_B W_A + (w_B + r_B) Q_A
\end{array}$$

Fig. 5. Protocol SK

4.1 LLK Protocol

The LLK key agreement protocol of Figure 4 is due to Lee *et al.* [20]. The session key is computed from the expression $(r_A w_B + r_B w_A)P$. Although the authors have conjectured that a KCI attack against the protocol is unfeasible, the following proves the opposite. Adversary E, impersonating B with knowledge of w_A, computes the EC point $Q_E \leftarrow r_E w_A^{-1}(Q_A + w_A W_B)$ and sends it to A in a run of the protocol. It is easily verified that A and E both accept with the key $\mathcal{H}(Q_A + r_E W_B)$.

4.2 SK Protocol

The SK key agreement protocol of Figure 5 was proposed by Song *et al.* [24]. The session key is derived from the expression $(r_A w_B + r_B w_A + r_A r_B)P$. The authors claim resistance against KCI attacks. However, we now describe a successful KCI attack. Adversary E, impersonating B with knowledge of w_A, computes the EC point $Q_E \leftarrow r_E P - W_B$ and sends it to A in a run of the protocol. It is easily verified that A and E both accept with the key $\mathcal{H}(r_E(W_A + Q_A) - w_A W_B)$.

$$
\begin{aligned}
A &: r_A \overset{R}{\leftarrow} [1, n-1] \\
&\quad Q_A \leftarrow r_A W_B \\
A \rightarrow B &: Q_A \\
B &: r_B \overset{R}{\leftarrow} [1, n-1] \\
&\quad Q_B \leftarrow r_B W_A \\
B \rightarrow A &: Q_B \\
A &: sk_A \leftarrow r_A w_A^{-1} Q_B + w_A W_B \\
B &: sk_B \leftarrow r_B w_B^{-1} Q_A + w_B W_A
\end{aligned}
$$

Fig. 6. Protocol SSEB

$$
\begin{aligned}
A &: r_A \overset{R}{\leftarrow} [1, n-1] \\
&\quad Q_A \leftarrow r_A P \\
&\quad e_A \leftarrow \mathcal{H}(Q_A.x, K_s.x) \\
A \rightarrow B &: Q_A, e_A \\
B &: r_B \overset{R}{\leftarrow} [1, n-1] \\
&\quad Q_B \leftarrow r_B P \\
&\quad e_B \leftarrow \mathcal{H}(Q_B.x, K_s.x) \\
B \rightarrow A &: Q_B, e_B \\
A &: e_B \overset{?}{=} \mathcal{H}(Q_B.x, K_s.x) \\
&\quad \text{if false then abort else } sk_A \leftarrow -r_A Q_B \\
B &: e_A \overset{?}{=} \mathcal{H}(Q_A.x, K_s.x) \\
&\quad \text{if false then abort else } sk_B \leftarrow -r_B Q_A
\end{aligned}
$$

Fig. 7. Protocol PU

4.3 SSEB Protocol

The SSEB key agreement protocol of Figure 6 was designed by Al-Sultan *et al.* [1]. The session key is derived from the expression $(r_A r_B + w_A w_B)P$. The conjectured security attributes include KCI resilience. To prove that this claim is false it suffices for an adversary E, impersonating B with knowledge of w_A, to send the EC point $Q_E \leftarrow r_E w_A W_B$ to A in a run of the protocol. Once again it is easily verified that A and E both accept with the key $\mathcal{H}(r_E Q_A + w_A W_B)$.

4.4 PU Protocol

The PU key agreement protocol of Figure 7 was proposed by Popescu [22]. The conjectured security attributes include KCI resilience. The private/public key pair of a principal X is $(w_X, W_X = -w_X P)$ while the map $\mathcal{H}(\cdot)$ is a hash function. To communicate, principals A, B must share a static secret key $K_s = -w_A W_B = -w_B W_A = w_A w_B P$. A trivial attack shows that the protocol is not secure against KCI attacks. Indeed, an adversary E impersonating B with knowledge of w_A (and therefore can easily compute K_s), needs to send the EC point $Q_E \leftarrow r_E P$ to A in a run of the protocol. Clearly, both E and A will accept with the key $-r_E Q_A = -r_A Q_E$.

4.5 Discussion

The LLK, SSEB, SK, MTI protocols use a closely related approach to compute the session key. The intuition is that, in a protocol run, the established (fresh) session key should be the result of an inextricable combination of information proving the identity of a principal (e.g. private keys w_A, w_B) and of session-specific data (e.g. r_A, r_B), in such a way to to avoid any danger of one of A or B being fooled into associating the key with the wrong principal. This is done by exploiting the algebraic structure of the underlying group. However, although this approach guarantees some important security properties (e.g. key independence, forward secrecy), it makes possible for the adversary to mount the attacks illustrated in Section 4. As already pointed out before, to prevent these attacks, either the session key should be computed in such a way that the adversary is unable to cause the corrupted principal (say A) to accept a particular session key without knowing the private key of the principal she is impersonating (w_B) and/or the session-specific data of A (r_A), or by following the approach used by the MQV protocol where such attacks are avoided by destroying the algebraic structure of the group.

The UM protocol (and, for example, the three protocols presented in [16]) are vulnerable to key compromise impersonation because they use a static shared key ($w_A W_B = w_B W_A$) to compute the session key. To obtain this key, it is sufficient for the adversary to corrupt either one of the principals A or B.

Apparently, public key cryptography seems the only way to obtain KCI-resilient key agreement protocols.

5 KCI vs. Provable Security

In general, it is difficult to formally prove that a protocol is KCI-resilient. Indeed, it easier to find some specific message that demonstrates that the property does not hold (as shown in Section 4). To this end, it is worthwhile considering in more detail how the formal models of distributed computing found in the literature cover such attacks. We explore this issue in the present section. The focus is on three models:

- The model due to Bellare and Rogaway [5] was originally conceived for two-party protocols in the private-key trust model. It was later extended to the password-based [4] and public-key settings [10, 9, 8]. This model introduces the notion of matching conversations as a means of entity authentication (and the more general idea of partnering functions [6]). Secrecy of a session key is modeled by its indistinguishability from a random string;
- The approach followed by Shoup [23], to model the security of key agreement protocols in the asymmetric trust model, is completely different since it stems from the notion of simulatability (which is extensively present in the cryptographic literature as the basis for zero knowledge proofs);
- The initial version of the model of Bellare *et al.* [3] was also founded on the notion of simulatability. Later on, Canetti and Krawczyk [11] published an improved version with a re-formulation of the security definitions in terms of the indistinguishability paradigm (a-la [5]).

Generally speaking, one expects that a formal security proof in the above models implies resilience to a whole range of attacks. However, although the adversarial models usually considered are almost (polynomially) all powerful (e.g. the adversary can relay messages out of order or to unintended recipients, concurrently interact with multiple instances of the protocol, corrupt legitimate users, acquire session keys, etc) a protocol is at best proven secure with respect to impersonation, known-key (Denning-Sacco) attacks and perhaps to exposure of long-term keys (but only as far as forward secrecy is concerned).

5.1 KCI in the Bellare-Rogaway Model

We briefly recall the main concepts of the Bellare-Rogaway [5] model in the asymmetric trust model. A (two-party) key agreement protocol is defined as a pair $\Sigma = (\mathcal{G}, \Pi)$ of poly-time computable functions where $\mathcal{G}$ generates the long-term (private-public) keys assigned to a principal P_i (suppose there is a finite number N of principals) and Π specifies the protocol actions and message formats. The symbol Π_i^r denotes the r-th protocol instance (oracle) run by principal P_i. Oracle Π_i^r has an intended communication partner (say P_j) denoted by pid_i^r. In honest runs of the protocol there exists a unique oracle Π_j^s (having $\mathsf{pid}_j^s = P_i$) which is *partnered* to Π_i^r, i.e. with $\mathsf{sid}_i^r = \mathsf{sid}_j^s$. The session identifier sid_i^r (resp. sid_j^s) is defined as the concatenation of incoming and outgoing messages for the instance Π_i^r (resp. Π_j^s) .

The adversary can initiate and interact with any (polynomial in ℓ — the security parameter) number of protocol instances (oracles) by asking the following queries:

- (init,i, j): this query sets $\mathsf{pid}_i^r = P_j$ and activates (the r-th) instance Π_i^r of the protocol. As a result, oracle Π_i^r enters the idle state;
- (send,i, r, M): the adversary sends message M to instance Π_i^r masquerading as pid_i^r. The answer is computed exactly as specified by the protocol specification (unless P_i is corrupted) with instance Π_i^r entering an expecting state. When $M \equiv \mathsf{start}$ an instance Π_i^r in the idle state is prompted to send the first message according to the protocol specification;
- ($\mathsf{execute}$,i, j): this query models a passive adversary eavesdropping on a run of the protocol between honest principals P_i, P_j. The resulting transcript is given to the adversary. In principle, an $\mathsf{execute}$ query can be simulated by send and init queries. However, in an $\mathsf{execute}$ query the parties and oracle instances strictly adhere to the protocol specification. This is opposed to send queries wherein the adversary can specify messages of her own choice (which if indistinguishable from valid ones and may cause the recipient oracle to accept);
- (reveal,i, r): this query models exposure of the session key of the instance Π_i^r due, for example, to improper erasure after its use, hijacking of the machine running the protocol or perhaps cryptanalysis. It is applicable only to instances that have accepted;
- ($\mathsf{corrupt}$,i): we work in the *weak corruption model* wherein a $\mathsf{corrupt}$ query only exposes the long-term private key of a principal P_i, as opposed to the *strong corruption model* wherein the adversary also obtains the internal state of the instances run by P_i. The adversary can use the compromised private key to impersonate P_i

with send queries. We stress that the adversary does not obtain the session key as the result of a corrupt query on a instance Π_i^r that has accepted;

- (test,i, r): when the adversary $\mathcal{A}$ asks this query an unbiased coin b is flipped and K_b is returned. If $b = 0$ then $\mathsf{K}_0 \leftarrow \mathsf{sk}_i^r$ otherwise $\mathsf{K}_1 \overset{R}{\leftarrow} \{0,1\}^{\ell_1}$ (ℓ_1 is a secondary security parameter related to ℓ). The adversary $\mathcal{A}$ must distinguish which one.

The security of protocol Σ is defined in the context of the following game between a challenger $\mathcal{C}$ and the adversary $\mathcal{A}$:

(a) Setup: The challenger $\mathcal{C}$ runs algorithm $\mathcal{G}(1^\ell)$ to generate a private-public key pair (SK_i, PK_i) for every principal P_i. The adversary is given the set $\{PK_i | i \in N\}$;
(b) Queries: Adversary $\mathcal{A}$ adaptively asks (a polynomial in ℓ number of) oracle queries (a single test query is allowed). If required, both the challenger and the adversary can access a (public) random oracle modeling a hash function;
(c) Output: The adversary attempts to distinguish whether a key obtained from the test query is a real session key or a random one (or equivalently the adversary must output a correct guess b' of the bit b chosen by the challenger when answering the test query).

At the end of the above game the advantage of the adversary must be negligible for the protocol to be secure. In a concrete analysis this advantage is expressed as a function of the resource expenditure required to win the game.

Any meaningful notion of security depends on the adversarial capabilities, which are expressed in terms of the types of queries the adversary is allowed to ask during the game. For example, (a weak form of) *forward secrecy*, captures the inability of obtaining information on already generated session keys even for an adversary that has corrupted the principals and has eavesdropped on several protocol runs. This is modeled by considering an FS-game wherein the adversary can ask init, send, execute, reveal, corrupt queries. To win the game the adversary must try to guess (using the test query) the session key of a FS-fresh oracle, i.e. an oracle that (at the end of the game) has not been the target of a reveal query (neither has its partner oracle) and no send queries were asked to that oracle and to its partner.

The advantage of the adversary is defined as $\mathsf{Adv}_{\Sigma}^{\mathsf{FS}}(\ell) = |2 \cdot \Pr[b' = b] - 1|$ and the protocol is FS-secure if the following inequality holds:

$$\mathsf{Adv}_{\Sigma}^{\mathsf{FS}}(\ell, t) = \max_{\mathcal{A}}\{\mathsf{Adv}_{\Sigma}^{\mathsf{FS}}(\ell)\} \leq \epsilon(\ell)$$

for negligible $\epsilon(\ell)$ and where the maximum is evaluated with respect to all adversaries running in polynomial time t (i.e. t is a polynomial in ℓ).

As it is, the model of Bellare-Rogaway offers no formalisation of KCI resilience. In order to provide for such a possibility, we present for consideration the following definition of a KCI-fresh oracle. The adversary can ask the test query to a KCI-fresh oracle in the game defined above while being able to ask init, send, reveal, corrupt queries.

Definition 2 (KCI-fresh oracle). *An oracle Π_i^r (with* $\mathsf{pid}_i^r = P_j$*), for some r, is* KCI-fresh *if the following conditions hold at the end of the game:*

1. $\mathsf{acc}_i^r = \mathtt{TRUE}$*;*
2. (reveal, i, r) *has not been asked by the adversary;*
3. *if the adversary has queried* $(\mathsf{corrupt}, i, r)$*, then no* (send, j, s, M) *query was asked where* M *is a message that depends on the private key of* P_i *and* $\mathsf{pid}_j^s = P_i$*.*

The advantage of the adversary is defined as $\mathsf{Adv}_{\Sigma}^{\mathsf{KCI}}(\ell, t) = |\Pr[b' = b] - \frac{1}{2}|$ and it must be negligible for the protocol to be KCI-secure.

The above definition of a KCI-fresh oracle (obviously) requires that if message M depends on the private key of the corrupted principal P_i then it was never used by the adversary to impersonate P_i to other principals (after P_i is corrupted). However, notice that message M may not necessarily depend on the private key of P_i. Observe also that Π_i^r may not terminate with a partnered oracle (indeed, we are not concerned whether Π_j^s accepts or not).

Unfortunately, key compromise impersonation resilience must be established on its own since there appears to be no relationship, for example, with forward secrecy (which, on the other hand, almost always implies key independence).

We now prove the following theorem.

Theorem 1. *Given the* EC *parameters* Φ_{EC}*, the* MTI/A0 *protocol (Figure 2) is a* KCI-*resilient protocol assuming the group* $E(\mathbb{F}_q)$ *satisfies the* ECCDH *assumption and the hash function* $\mathcal{H}$ *is modeled as a random oracle. Concretely, we have*

$$\mathsf{Adv}_{\text{MTI/A0}}^{\mathsf{KCI\text{-}R}}(\ell, t, q_h, q_{re}, q_{co}, q_{se}) \leq 1/N^2 \cdot 1/q_h \cdot \epsilon,$$

where t *is the total running time of the game played by the adversary (including its execution time),* ℓ *the security parameter and* $q_h, q_{co}, q_{re}, q_{se}$*, respectively, the number of random oracle,* corrupt*,* reveal *and* send *queries and* N *is the number of principals.*

Proof. Given $X = xP, Y = yP$ the symbol DH(X, Y) denotes the Diffie-Hellman secret xyP. The proof is by a reduction technique; if an adversary $\mathcal{A}$ is able to break KCI-resilience then we may construct an adversary $\mathcal{F}$ that uses $\mathcal{A}$ as a subroutine and succeeds in solving the computational Diffie-Hellman problem (CDHP) in the underlying elliptic curve group E$(\mathbb{F}_q)$. Algorithm $\mathcal{F}$ simulates the game played by $\mathcal{A}$ (against the challenger $\mathcal{C}$ — see above) in such a way that $\mathcal{A}$'s view is indistinguishable from the real game. A description of $\mathcal{F}$ follows:

1. $\mathcal{F}$ receives in input ($X = xP, Y = yP$), chooses i^*, j^* guessing that i^* will be the principal corrupted by $\mathcal{A}$ in its game and that j^* is the principal impersonated by $\mathcal{A}$ in the attack;
2. $\mathcal{F}$ generates keys (w_i,W_i) for all principals P_i except for P_i^* for which she sets $W_{j^*} = Y$;
3. $\mathcal{F}$ runs $\mathcal{A}$ as a subroutine answering its queries as follows:
   - For (send,i, r, M) queries with Π_i^r in the idle state the answer is given by choosing random r_i and outputting r_iP. If $i = i^*$ and $\mathsf{pid}_{i^*}^r = P_j*$ then the response is $aP + X$, for random a (and oracle Π_i^r moves into an expecting state);
   - For (send,i, r, M) queries with Π_i^r in the expecting state the answer is given by setting the session key sk_i^r equal to a random element in $\{0, 1\}^{\ell}$; if $i = i^*$, $\mathsf{pid}_{i^*}^r = P_j^*$ and a (corrupt,i^*) query was asked $\mathcal{F}$ stores the record (aP+X,M) in the list L1;

  - For random oracle queries $\mathcal{H}(i, j, U, V, W)$ the response is a random element sampled from $\{0, 1\}^\ell$ (or eventually the same value output before); if $i = i^*$ and $\mathsf{pid}_{i^*}^r = P_j^*$ then $\mathcal{F}$ finds the record $(aP + X, M)$ in L1 s.t. $U = aP + X$ and $V = M$ and writes $(aP + X, M, W)$ to list L2;
  - send, reveal, test queries are answered normally;
  - (corrupt,i) queries are answered as usual except that if $i = j^*$ then $\mathcal{F}$ aborts.

4. When $\mathcal{F}$ terminates (exactly when $\mathcal{A}$ does) it chooses a random element in list L2 and outputs DH(X,Y)=W-aY-$w_i M$ (W=$(a + x)$Y+$w_i M$). Observe that oracles $\Pi_{i^*}^u$, for any u, are KCI-fresh according to the simulation (and therefore any test query that $\mathcal{A}$ asks of these oracles can be correctly answered by $\mathcal{F}$).

It is straightforward to verify that the success probability of $\mathcal{F}$ is bounded from above by $1/N^2 \cdot 1/q_h \cdot \epsilon$. □

5.2 KCI in the Canetti-Krawczyk Model

Recently, Krawczyk [18] has attempted to formally define KCI attacks in the model of Canetti-Krawczyk [11]. The formalism is introduced to prove the resilience of a hash-based version of the MQV protocol (HMQV). In this model security of a key agreement protocol is modeled along the lines of indistinguishability of the session key from a random value, as in [5]. Two communication models exist; the first one is the simplified authenticated-links (AM) model wherein the communication links are assumed to be authenticated, the second one is the unauthenticated-links (UM) model wherein the network is totally controlled by the adversary. In the latter model the adversary is given capabilities which allow different levels of information exposure of a session or principal (the adversary may ask queries session-state reveal, party-corruption, session-key query, session expiration, test-session).

A secure key exchange protocol is formalised ([11], definition 4) in a context similar to the game of Section 5.1 by requiring that (1) if two uncorrupted principals complete matching sessions then they both output the same key and (2) the probability of success of the adversary in distinguishing the session key from a random one is no greater than 1/2 plus a negligible function.

In the model it is hypothesized that in real world implementations long-term secret keys are often granted better protection (e.g. by using cryptographic modules) than session-specific data; this is reflected in the attackers' capabilities by considering separate party corruption and session state reveal operations. The authors speculate that whenever this is not a realistic assumption one could weaken the model by omitting the session-state reveal operation. However, in practice almost *all* computations can take place in a cryptographic module (e.g. those involving the generation of ephemeral Diffie-Hellman public keys) thus making session-specific information leakage more difficult. Furthermore, hardware-specific attacks (e.g. power analysis) are completely ignored.

The basic definition of a secure protocol does not consider the case of corrupted principals, therefore, in [18] a new notion is introduced into the model to account for KCI attacks, namely, that of a clean session. The goal is to capture the situations wherein the adversary has learned the long-term private key of a principal but has not actively controlled the session (nor has impersonated the principal) during a run of the protocol.

A key agreement protocol is considered resilient to KCI attacks if the adversary is unable to distinguish the real session key (from a random one) of a complete session, being this session clean and the partner session at an uncorrupted principal also clean. Under this definition ([18], Definition 20) it is shown that the HMQV protocol is secure in the model of [11].

Although the definition seems consistent there are still issues that are not clear in the security proof of the HMQV protocol ([18], Lemma 21 which refers to Section 5.2). In particular, (1) it is not shown how the proof extends to the case that principal $\widehat{B}$ is corrupted; (2) it is not exactly specified how party corruption actually occurs since, if the adversary $\mathcal{M}$ corrupts $\widehat{A}$ then, in order for the sessions initiated at $\widehat{A}$ to be unexposed and, therefore candidate test-sessions, a party corruption query must be scheduled only when the sessions have expired (or were never initiated) at $\widehat{A}$. The latter remark is a more general one since it is concerned with the relationship between clean and exposed sessions at a corrupted principal.

5.3 KCI in the Shoup Model

In the formal model of Shoup [23] security is defined via simulation. There is an ideal world wherein the service offered by the key agreement protocol is defined, and a real world which describes how protocol participants communicate. An ideal world adversary is essentially constrained to be benign. A security proof shows that the adversary in the real world is essentially "simulatable" by the adversary in the ideal world and therefore one deduces that the protocol is secure in the real world. Again, the simulation takes place in the context of a game similar to those defined in the preceding models. Three classes of adversaries are defined, according to their capability of obtaining private information held by users (either static or ephemeral data), that give rise to static corruptions, adaptive corruptions and strong adaptive corruptions.

Let us examine how KCI attacks can be viewed in the adaptive corruptions model. We use the notation and terminology of [23]. Consider an instance I_{ij} engaging in the key agreement protocol (e.g. the two pass AK LLK protocol) with a *compatible* instance $I_{i'j'}$. Suppose that after the first message M_1 (e.g. $Q_i = r_i W_{i'}$ in protocol LLK) is delivered, $I_{i'j'}$ accepts the session key $K_{i'j'}$. The adversary now corrupts user U_i. Instance $I_{i'j'}$ responds with a message M_2 (e.g. $Q_{i'} = r_{i'} W_i$ in protocol LLK), the adversary intercepts it and instead delivers message $\overline{M}_2$ (e.g. $\overline{M}_2 \equiv \overline{Q}_{i'} = \overline{r}_{i'} w_i^{-1}(Q_i + w_i W_{i'})$ in protocol LLK). At this point I_{ij} will accept a session key K_{ij} known by the adversary and different from $K_{i'j'}$ (which the adversary ignores). Now, in the ideal world, when $I_{i'j'}$ generated its session key, it was not corrupted so the only connection assignment possible for $I_{i'j'}$ is create. On the other hand, the only possible connection assignment for I_{ij}, being U_i corrupted, is compromise. However, I_{ij} and $I_{i'j'}$ are *compatible*, hence I_{ij} cannot be compromised without breaking the rules of the game since $PID_{ij} = ID_{i'}$ is assigned to user $U_{i'}$. Moreover, a connect is also not possible between I_{ij} and $I_{i'j'}$ since this would imply $K_{ij} = K_{i'j'}$. We must conclude that the simulation is not possible since it would lead to inconsistent real world and ideal world transcripts. Note that we have used the *liberal compromise rule* as defined in [23] (the simulation is still not be possible under the *conservative compromise rule*).

6 Conclusions and Future Work

In this paper we discussed key compromise impersonation resilience for key agreement protocols in the asymmetric trust model. Several protocols, whose authors have mistakenly claimed resilience to KCI, are proven vulnerable to such attacks. For these protocols, explicit key confirmation (e.g. using the compilers of [8, 7]) may provide an effective countermeasure since the parties involved (A, B) accept different session keys. However, this is achieved at the expense of increased computational and round complexity.

It appears that protocol designers do not always pay attention to key compromise impersonation. Instead, forward secrecy, which is indeed another harmful threat related to party corruption, is usually considered more important. However, our thesis is that the security analysis of key agreement protocols is incomplete with a corruption model that considers only forward secrecy.

Although there is a constant debate in the research community concerning formal (complexity-theoretic based) security models, they undoubtedly constitute a valuable approach to achieve proactively secure key agreement protocols. Surprisingly, however, three of the most significant models found in the literature do not have a satisfactory approach (besides having one at all) to KCI. We have attempted to incorporate a reasonable notion of KCI resilience into the model of Bellare-Rogaway. Future work includes formulating an appropriate notion of resilience to KCI into the formal security model of Shoup.

Acknowledgements

The author is grateful to the anonymous reviewers for their valuable comments and suggestions.

References

1. K. Al-Sultan, M. Saeb, M. Elmessiery, and U.A.Badawi. A new two-pass key agreement protocol. *Proceedings of the IEEE Midwest 2003 Symp. on Circuits, Systems and Computers*, 2003.
2. R. Ankney, D. Hohnson, and M. Matyas. The Unified Model. *Contribution to X9F1*, 1995.
3. M. Bellare, R. Canetti, and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. *In 30th Symposium on Theory of Computing*, pages 419–428, 1998.
4. M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attack. *In Proceedings of EUROCRYPT 2000*, LNCS 1807:139–155, 2000.
5. M. Bellare and P. Rogaway. Entity authentication and key distribution. *In Proceedings of CRYPTO 1993*, LNCS 773:232–249, 1993.
6. M. Bellare and P. Rogaway. Provably secure session key distribution - the three party case. *In Proceedings of 27th ACM Symposium on the Theory of Computing 1995*, 1995.
7. M. Bellare and P. Rogaway. The AuthA protocol for password-based authenticated key exchange. *Contribution to IEEE P1363*, 2000.

8. S. Blake-Wilson, D. Johnson, and A. Menezes. Key agreement protocols and their security analysis. *In Proceedings of the 6th IMA Int.l Conf on Cryptography and Coding*, LNCS 1355:30–45, 1997.
9. S. Blake-Wilson and A. Menezes. Entity authentication and authenticated key transport protocols employing asymmetric techniques. *Security Protocols - 5th International Workshop*, LNCS 1361:137–158, 1998.
10. S. Blake-Wilson and A. Menezes. Authenticated Diffie-Hellmann key agreement protocols. *Selected Areas in Cryptography - 5th International Workshop*, LNCS 1556:339–361, 1999.
11. R. Canetti and H. Krawczyk. Analysis of key exchange protocols and their use for building secure channels. *Advances in Cryptology-EUROCRYPT 2001*, LNCS 2045:453–474, 2001.
12. FIPS-PUB-186-2. Digital Signature Standard. National Institute of Standards and Technology, 2000.
13. D. Hankerson, A. Menezes, and S. Vanstone. *Guide to Elliptic Curve Cryptography*. Springer Professional Computing, New York, 2004.
14. IEEE-P1363-2000. Standard specifications for public key cryptography. Institute of Electrical and Electronics Engineers, 2000.
15. IEEE-P1363.2/D15. Standard specifications for password-based public key cryptographic techniques. Institute of Electrical and Electronics Engineers, 2004.
16. I. Jeong, J. Katz, and D. Lee. One-Round Protocols for Two-Party Authenticated Key Exchange. *Applied Cryptography and Network Security 2004*, 2004.
17. M. Just and S. Vaudenay. Authenticated Multi-Party Key Agreement. *Advances in Cryptology-ASIACRYPT 1996*, LNCS 1163:36–49, 1996.
18. H. Krawczyk. HMQV: A high-performance secure Diffie-Hellmann protocol. *http://eprint.iacr.org/2005/176*, 2005.
19. L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone. An efficient protocol for authenticated key agreement. *Designs, Codes and Cryptography*, 28:119–134, 2003.
20. C. Lee, J. Lim, and J. Kim. An efficient and secure key agreement. *IEEE p1363a draft*, 1998.
21. T. Matsumoto, Y. Takashima, and H. Imai. On seeking smart public-key distribution systems. *Transactions of IEICE*, VolE69:99–106, 1986.
22. C. Popescu. A Secure Authenticated Key Agreement Protocol. *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference*, 2004.
23. V. Shoup. On Formal Models for Secure Key Exchange. Technical Report RZ 3120, IBM Research, 1999.
24. B. Song and K. Kim. Two-pass authenticated key agreement protocol with key confirmation. *Progress in Cryptology - Indocrypt 2000*, LNCS 1977:237–249, 2000.

A PKI System for Detecting the Exposure of a User's Secret Key

Younggyo Lee[1], Jeonghee Ahn[2], Seungjoo Kim[1],
and Dongho Won[1,⋆,⋆⋆]

[1] Information Security Group,
Sungkyunkwan University, Korea
{yglee, skim, dhwon}@security.re.kr
[2] Department of Computer Science,
Doowon Technical College, Korea
jhpro@doowon.ac.kr

Abstract. Dodis *et al* proposed a key-insulated signature scheme in 2003. The scheme can minimize the damage caused by the secret key's exposure but can not protect the user from the secret key's exposure perfectly. We propose a PKI system which can detect immediately even a single illegitimate signature due to the exposure of a user's secret key. The system uses the one-time hash chain based on NOVOMODO and can prevent the users from compromising the secret key more perfectly and effectively than the key-insulated signature scheme.

Keywords: key-insulated signature, one-time hash chain, NOVOMODO.

1 Introduction

Dodis *et al* proposed a key-insulated signature scheme at 2003 in [2, 3]. In the scheme, the master secret key is stored in the physically secure device and not used for signing directly. Total lifetime of the master secret key is divided into time periods and the different secret keys refreshed by the master key are used for each time period. Therefore the scheme can minimize the damage caused by the secret key's exposure but can not protect the user from the secret key's exposure in a time period perfectly. Just a single illegitimate signature by the exposure of a user's secret key can give extensive damage to the user in E-business or E-commerce. In this paper, we propose a PKI system that can immediately detect even a single illegitimate signature caused by the exposure of a user's secret key. The system uses the one-time hash chain based on NOVOMODO [1] and can prevent users from compromising the secret key more perfectly and effectively than the key-insulated signature scheme.

[⋆] This work was supported by the University IT Research Center Project funded by the Korean Ministry of Information and Communication.

[⋆⋆] Corresponding author.

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 248–250, 2006.

2 A PKI System Detecting the Exposure of a User's Secret Key

Initial procedure: The user's certificate issuance by CA

1. The user A computes Z_0 by 10,000 hashing operations from random Z_K.

$$Z_K \xrightarrow{h} Z_{K-1} \xrightarrow{h} \ldots Z_i \ldots \xrightarrow{h} Z_1 \xrightarrow{h} Z_0$$

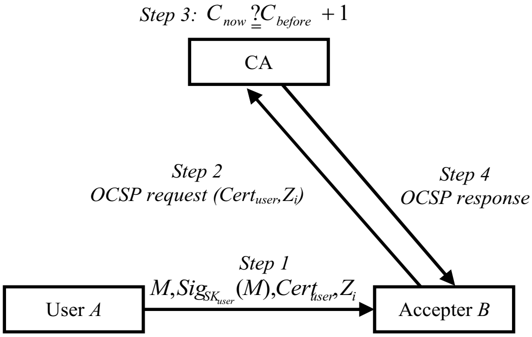
2. The user A sends the own public key, the own identification information, and Z_0 safely to CA for the request of own certificate issuance.

3. The CA issues the user A's certificate $Cert_{user}$ Z_0 is also included as follows. And the CA sets the counter C_{before} for user A to *0*. The counter C_{before} is stored and managed in CA.

$$Cert_{user} = Sig_{SK_{CA}}(PK_{user}, SN, I, S, V, Z_0), \; C_{before} \leftarrow 0$$

4. The user A stores his own secret key, the input value and all intermediate values in step 1 securely.

Service procedure: Signature and certificate status validation


Step 3: $C_{now} \overset{?}{=} C_{before} + 1$
CA
Step 2
OCSP request ($Cert_{user}$,Z_i)
Step 4
OCSP response
Step 1
M,$Sig_{SK_{user}}(M)$,$Cert_{user}$,Z_i
User A
Accepter B

1. When the user A signs, he sends his own certificate and the hash value Z_i with the signed message M to the acceptor B. The Z_i is delivered in the order of $Z_1, Z_2, Z_3, ..., Z_K$.

$$M, Sig_{SK_{user}}(M), Cert_{user}, Z_i$$

2. After receiving the signed message from user A, the acceptor B requests the user A's certificate status information to CA.

$$OCSP\ request, Cert_{user}, Z_i$$

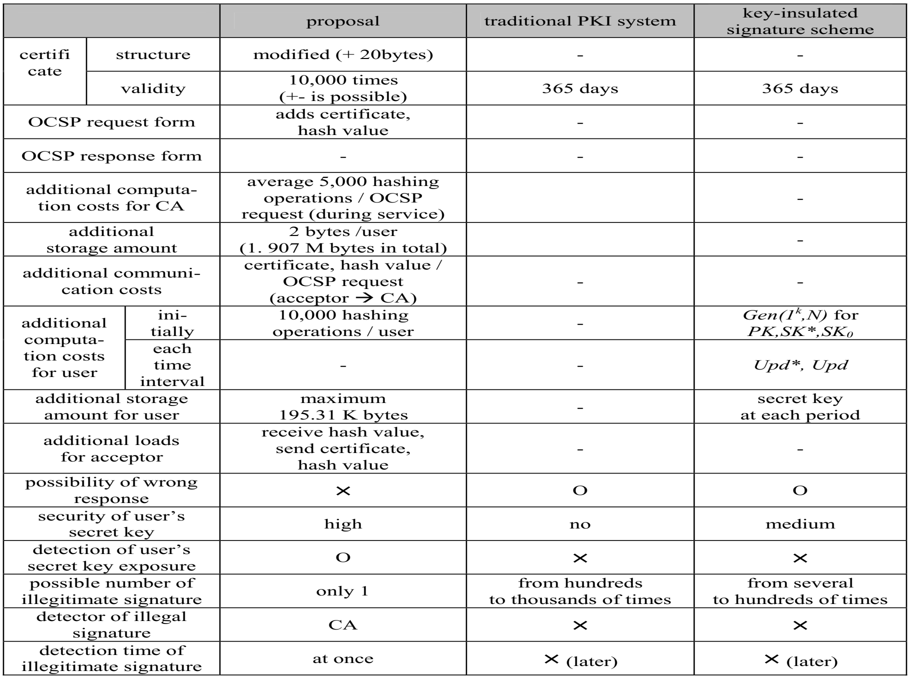
3. If these conditions are satisfied, then the CA confirms that the user A's secret key was not used in an illegitimate signature.

$$h(Z_i)^i \overset{?}{=} Z_0, \quad C_{now} \overset{?}{=} C_{before} + 1$$

4. The CA delivers the corresponding OCSP response including the user A's certificate status information (*"good"*) to the acceptor B.

$$OCSP\ response,\ \ C_{before} \leftarrow C_{before} + 1,\ C_{now} \leftarrow 0$$

Comparisons



		proposal	traditional PKI system	key-insulated signature scheme
certificate	structure	modified (+ 20bytes)	-	-
	validity	10,000 times (+- is possible)	365 days	365 days
OCSP request form		adds certificate, hash value	-	-
OCSP response form		-	-	-
additional computation costs for CA		average 5,000 hashing operations / OCSP request (during service)		-
additional storage amount		2 bytes /user (1. 907 M bytes in total)		-
additional communication costs		certificate, hash value / OCSP request (acceptor → CA)	-	-
additional computation costs for user	initially	10,000 hashing operations / user	-	$Gen(1^k,N)$ for PK,SK^*,SK_0
	each time interval	-	-	Upd^*, Upd
additional storage amount for user		maximum 195.31 K bytes	-	secret key at each period
additional loads for acceptor		receive hash value, send certificate, hash value	-	-
possibility of wrong response		×	O	O
security of user's secret key		high	no	medium
detection of user's secret key exposure		O	×	×
possible number of illegitimate signature		only 1	from hundreds to thousands of times	from several to hundreds of times
detector of illegal signature		CA	×	×
detection time of illegitimate signature		at once	× (later)	× (later)

References

1. Silvio Micali.:NOVOMODO ; Scable Certificate Validation And Simplified PKI Management, 1st Annual PKI Research Workshop Preproceedings, pp.15-25, 2002.
2. Yevgeniy Dodis, Jonathan Katz, Shouhuai Xu, and Moti Yung.: Key-Insulated Public Key Crytosystems, EUROCRYPT 2002, LNCS 2332, pp. 65-82, 2002.
3. Yevgeniy Dodis, Jonathan Katz, Shouhuai Xu, and Moti Yung.: Strong Key-Insulated Signature Schemes, PKC 2003, LNCS 2567, pp. 130-1442, 2003.
4. Younggyo Lee, Injung Kim, Seungjoo Kim, and Dongho Won.: A Method for Detecting the Exposure of an OCSP Responder's Session Private Key in D-OCSP-KIS, Euro PKI 2005, LNCS 3545, pp. 215-226, 2005.
5. Younggyo Lee, Jeonghee Ahn, Seungjoo Kim, and Dongho Won.: A Method for Detecting the Exposure of an OCSP Responder's Private Key using One-Time Hash Value, IJCSNS International Journal of Computer Science and Network Security, VOL. 5 No.8, pp. 179-186, August 2005.

A Guide to the Nightmares of the Certification Service Provider

Lenka Kadlčáková

ICZ a.s.,
Prague, Czech Republic
lenka.kadlcakova@i.cz

Abstract. Czech Republic electronic signature scheme is based on the EU Directive 1999/93 EC. Currently there are three public certification service providers that passed the process of voluntary accreditation (Directive 1999/93/EC, article 2) and were granted by the regulation body of Czech Republic to provide the services as the accredited certification service provider according to Czech law. The paper is written from the perspective of the PKI supplier that participates strongly on the certification service provider support and also on the support of its customers.

1 Introduction

The Electronic Signature Act of Czech Republic is based on the EU Directive 1999/93 EC [1]. It covers the area of providing the qualified certification services, namely the issuance of qualified certificates as defined by the Directive, qualified timestamps and qualified system certificates for the electronic stamp verification. Electronic stamp was defined in the Czech Electronic Signature Act novel in July 2004. It is the analogy of electronic signature based on qualified certificate, intended to be used by the automatic process or application. (The electronic stamp is created automatically by the process; it is not supposed that the document was checked by natural person before it was stamped.)

There are three public certification service providers that were granted the accreditation by the Czech regulator to provide the qualified certificate services on the Czech market. All of them issue the qualified and qualified system certificates, one of them also the qualified timestamps since the beginning of this year. Among them, Czech Post is a bit exceptional. It is a huge company, which does not have the electronic services as its core business, though it is changing nowadays. The size and the fact that the Czech Post service spectrum is truly wide caused considerable complications during the Certification Authority implementation and causes difficulties also during its operation.

2 Czech Post PKI Model and Basic Philosophy

Czech Post PKI is based on the hierarchy of certification authorities. Czech Post Root CA is common root that issued the certificate to the subordinate Qualified

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 251–253, 2006.

Certification Authority that issues the qualified certificates and qualified system certificates to public. Currently, Czech Post operates 75 Registration Authorities countrywide; the plan is to operate up to 200 Registration Authorities in 2007.

The Qualified Certification Authority of Czech Post went to the full operation on 1st September 2005. Since, it has issued about 4000 end user's certificates, roughly 2% of this amount were the qualified system certificates. The number of issued certificates increased slowly month by month, significant increase was realized during December 2005, when the regularly issued number doubled to 800. Czech Post sells the qualified certificate for € 7 and the qualified system certificate for € 96; all the end user certificates have the validity of one year.

3 First Months of Operation Experience

The first thing anyone always talks about after the first month of the full operation of whatever service is the support underestimation. So, we will not mention here the ordinary underestimation, like the lack of trained people. Rather, we would like to mention the facts that surprised us somehow, and that caused the change of the supposed support philosophy. Czech Post was the second accredited certification service provider in Czech Republic; the first one has been on the market since 2002. Mainly because of this fact, after Czech Post went operational, mostly the questions from the professionals were expected. There were many, of course, but even more questions were really basic, like how to sign an email, or what shall I do with the certificate. Of course, the end user guides were prepared before going operational, but they described mainly the tools offered by Czech Post, certificate request registration, certificate issuance and revocation. After the first few weeks of the full operation, it turned out that a guide describing the elementary possibilities of public key certificate usage is necessary in the case Czech Post wants to fully support all its clients.

The spectrum of the end user's queries was another thing that taxed and still taxes the support heavily. From the just mentioned elementary tasks, they went through the OS configuration and user rights settings, key material backup and recovery to some really special commands in OpenSSL. What is also very wide is the technological background of the end users. Crowd of the Win98 users is approximately of the same size as the crowd of the Unix-based OS users and those who don't have the floppy disk mechanics in their Laptop or PC. This all generates a lot of different technological demands on the Certification Authority system (OK, we will define the precautions and processes to accept the certificate requests on USB flash disk) and also on the support application (for example the optional tool for the key pairs generation supports now four OS and also the detailed description of how to generate and manage the key pair using OpenSSL was prepared).

It has to be mentioned here that the main problem does not reside in the support people knowledge or in their capacity. The question to decide is whether the service provider will invest the significant amount of money to solve the problems that are not really at its side or with its product, which is public key certificate in this case and eventually some application delivered to the end users.

Interesting discussion opened also on the allowed usage of qualified certificates. It is defined in European standards [2] and in certification policies. However, from the point of view of typical end user that was enforced to buy the qualified certificate by some of his business partners, why should he buy another one, moreover in year out, when he already has the qualified one? The most demanded usage is the authentication. And to make the situation of the certification services provider that is convinced that the qualified certificates and corresponding key pairs should not be used for authentication more difficult, some of the widely used web servers and portals accept the qualified certificates for the authentication. So, another business for support that has to explain the situation over and over again.

In average, the support solves about 280 queries per month. In the table below, the distribution of the queries can be seen.

Basic unfamiliarity with the computer (everything must be explained)	20%
Application Errors (during the key pair generation or certificate import, failed to sign email, . . .)	60%
Mistake or misinformation caused by the CA operators	5%
Skilled or special technical questions, remarks or requirements	15%

About 29% of all support queries comes via email, the rest are phone calls. About 40% of phone calls last up to 15 minutes, 50% 15–30 minutes, and 10% more than 30 minutes.

After going operational there were only two employees to solve the support queries, beside their normal duties. It turned out that the support taxed them more than full time. Now, five more people were trained to provide the Certification authority support and the Registration Authorities operators were trained to act as the first support line.

4 Conclusion

It is the well known fact that the PKI implementation is expensive and also that running PKI is expensive. But the price of running PKI is influenced, beside other factors, by the price of the support. It should be decided, before the PKI goes operational, what level of support will be offered to the customers and design the relevant application and processes depending on this decision.

References

1. DIRECTIVE 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures.
2. Electronic Signatures and Infrastructures (ESI); Policy requirements for certification authorities issuing qualified certificates; ETSI TS 101 456 V1.4.1 (2006-01).

A High-Level 3G Wireless PKI Solution for Secure Healthcare Communications

Chung-Ming Ou[1] and C.R. Ou[2]

[1] Department of Information Management, Kainan University
Luchu 338, Taiwan
cou077@mail.knu.edu.tw

[2] Department of Electrical Engineering, Hsiuping Institute of Technology
Taichung 412, Taiwan
crou@mail.hit.edu.tw

Abstract. We propose a new wireless PKI security scheme suitable for future mobile communication such as the third generation (3G). This scheme is based on two major elements, one is a trusted server, and the other is the dual public-key cryptosystems, to provide end-to-end security between mobile clients and the Healthcare Information System (HIS). Applications and services based on this WPKI scheme are also proposed.

1 Introduction

The third generation (3G) provides mobile Internet services with high network bandwidth. For a mobile equipment (ME) and the healthcare information systems (HIS), the ***end-to-end security*** means that ME and HIS authenticate each other first, and then share a common symmetric key. It establishes secure tunnels between MEs and HIS.

Trusted Third Parties (TTPs) are operationally connected through certificate paths. The scope of WPKI-based TTPs such as Certificate Authorities (CA) is to provide the authentication and non-repudiation service within the end-to-end security. The roles of a TTS are also introduced in [3].

2 The Architecture for WPKI-Based HIS

A basic 3G system architecture is composed of the following entities, namely a 3G mobile client, a base station with a WAP proxy and/or a SMSC, and varied application servers (for example, the HIS). A 3G mobile client is consist of a mobile equipment (ME, such as a 3G handset) with a USIM card installed in it (3GME+USIM). The ME utilizes the installed USIM card to store subscriber's WPKI components; 3GME is capable of verifying typical X.509 digital signatures in order to authenticate HIS.

We deploy a TTP based on [1] called the Mobile Authentication Server (MAS) to assist WAP proxy to authenticate MEs and HIS. We also design a middleware, which is

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 254 – 256, 2006.

named the PKI End-to-End Secure Server (PESS), dedicated to PKI operations for HIS, see Figure 1. Each ME has two public-key/private-key pairs; one is used for encryption/decryption, the other is used for digital signature generation and verification.

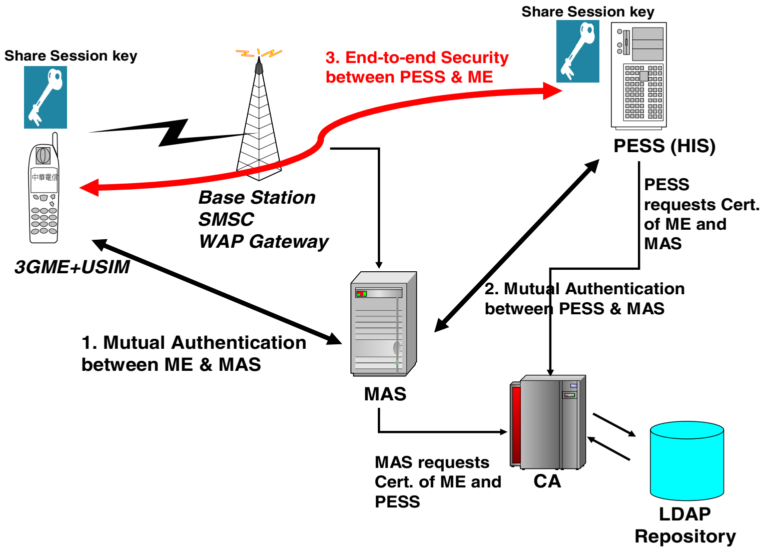

Share Session key
Share Session key
3. End-to-end Security between PESS & ME
PESS (HIS)
PESS requests Cert. of ME and MAS
Base Station
SMSC
WAP Gateway
3GME+USIM
2. Mutual Authentication between PESS & MAS
1. Mutual Authentication between ME & MAS
MAS
MAS requests Cert. of ME and PESS
CA
LDAP Repository

Fig. 1. End-to-end security scheme between ME and HIS

Once the mutual authentications between ME and MAS have been achieved, MAS is basically assisting ME to authenticate HIS. Therefore ME and HIS can reach the end-to-end security and share a common session key.

3 WPKI Services and Applications in Healthcare

WPKI provides certificate-based services such as confidentiality, integrity, authentication, non-repudiation and authorization. Examples of WPKI certificate usages within the HIS are as follows; (1) data encryption ; (2) digital signature ; (3) secure e-mail ; (4) secure web-server, and (5) time-stamping. Based on [2], we list general WPKI applications in eHealthcare, while the corresponding WPKI services are listed in the parenthesis by the above numbers (from 1 to 5).

- Mutual authentications of both the HIS service and the customers (1,2,4)
- Non-repudiation of the transaction/document (2,5)
- Authorized access to health-related patient data systems (2,4)
- Confidentiality and the integrity of stored medical data (1,2,5)
- Network security and confidentiality of transmitted patient information (1,2)
- Electronic prescriptions (1,2,3,4,5)

4 Conclusion

We propose a three-tier security scheme suitable for 3G mobile transaction. Security of this scheme is based on the dual-key cryptosystem, also this end-to-end security scheme can help a mobile client and the HIS establish secure channel. We propose basic WPKI services for major WPKI applications in eHealthcare.

References

1. Chanson S.,Cheung T.-W., Design and implementation of a PKI-based end-to-end secure infrastructure for mobile e-commerce. World Wide Web, 4:235-253 (2001).
2. Jelekainen P., GSN-PKI solution enabling secure mobile communications, Int. J. Medical Informatics (2004) 73, 317-320.
3. D. Lekkas, S. Gritzalis, S. Katsikas, Quality assured trusted third parties for deploying secure Internet-based healthcare applications, Int. J. Medical Informatics 65 (2002) 79-96

Identity-Based Strong Multi-Designated Verifiers Signatures

Sherman S.M. Chow

Department of Computer Science
Courant Institute of Mathematical Sciences
New York University, NY 10012, USA
schow@cs.nyu.edu

Abstract. Designated verifier signatures are privacy-oriented signatures that provide message authenticity only to a specified verifier but nobody else. We consider strong multi-designated verifiers such that knowledge of either one of designated verifiers' private keys is required to verify the signature. We propose the first identity-based construction.

1 Introduction

Designated verifier signatures (DVS), introduced by Chaum and Jakobsson *et al.* [5] independently, convince only a specific verifier about the validity of the signature. Like other privacy-oriented signatures scheme (e.g. undeniable signature, ring signature), the "loss" of the non-repudiation property of traditional signature makes it useful in various commercial cryptographic applications.

Application. We briefly talk about one of its applications. Suppose an organization initiates a call for tenders, asking some companies to propose their own prices for offering certain goods or services. The organization wants authenticity of the tender such that the selected company cannot later repudiate what they agreed to after. They can sign on the tender using traditional scheme, but such signature can be subsequently shown to others (e.g. by the tender-caller) such that other competing parties can prepare a "tailor-made" tender accordingly.

Working Principle. The working mechanism of DVS is that it consists of a proof showing either "the signer has signed on a message" or "the signer has the verifier's secret key" is true. The designated verifier, being confident that his/her private key is kept in secret, get convinced that the signer has signed on a message. No other third party can be convinced by this signature since the designated verifier can always generate such proof with his/her private key.

Strong and Multiple. Yet, this level of *signer ambiguity* (or source-hiding property) is not enough in scenario where one can certain that the verifier has not generated such proof. Consider when the signature is captured before reaching the verifier, the eavesdropper knows who the real signer is as there are only two possibilities. To address this problem, we need a *strong* DVS (termed in [5] and

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 257–259, 2006.

formalized in [7]) such that the verifier needs to use his/her private key to verify the signature. This property is referred as *signer's privacy*, such that given a DVS and two potential signing public keys, it is computationally infeasible to determine under which of the two corresponding signing key is used.

At CRYPTO 03's rump session, Desmedt [3] asked for a multi-designated verifiers signature scheme such that there are more than one designated verifier. Such scheme can help in multi-party activities like distributed contract signing.

Related Work. A generic MDVS construction, from any discrete logarithm based ring signature (e.g. [2]) and any secure multi-party computation protocol, was proposed in [6]. The authors suggested the use of an additional layer of encryption that is indistinguishable under adaptive chosen-ciphertext-attack (CCA2) to remedy the weaker notion of signer privacy. By exploiting the bilinearity of pairings on elliptic curve, strong 2DVS was proposed in the same paper. Generic construction of identity-based (ID-based) scheme was proposed in [8], followed by a recent proposal of strong DVS schemes with short signature length (both PKI-based and ID-based) [4]. These schemes satisfy the strong notion of signer privacy, but only single designated verifier is considered.

Our Contribution. This paper proposes a strong multi-designated verifies signature scheme (SM-DVS). Under traditional public key infrastructure, signer can generate SM-DVS only after *all* of the designated verifiers have obtained the certification. Motivated by the above problem, we consider ID-based keys (for both signer and verifiers), i.e. the public key is derived from a string denoting the identity of the user and there exists a trusted key generation centre (KGC) who generates the corresponding private keys on request.

2 Strong Multi-designated Verifier Signatures (SM-DVS)

Let $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, \cdot)$ be two cyclic groups of prime order q. The bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a map that $\forall P, Q, R \in \mathbb{G}_1$, $\hat{e}(P + Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$, and $\hat{e}(P, Q + R) = \hat{e}(P, Q)\hat{e}(P, R)$; and $\exists P, Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq 1$.

`Setup`: The KGC randomly chooses $s \in_R \mathbb{Z}_q^*$ as the master secret. System parameter is $\{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub} = sP, Q, H_1 : \{0,1\}^* \to \mathbb{G}_1, H_2 : \{0,1\}^* \to \mathbb{Z}_q^*\}$.

`Extract`: The user with identity $ID \in \{0,1\}^*$ submits ID to the KGC. The user's public key Q_{ID} is equal to $H_1(ID) \in \mathbb{G}$, The KGC computes the user's private key S_{ID} by $S_{ID} = sQ_{ID}$, where $s \in \mathbb{Z}_q^*$ is the master secret.

`Sign`: Let $L = \{ID_S, ID_{V_1}, \cdots, ID_{V_n}\}$ be the set of all identities of these $n + 1$ parties. For the signer ID_S to sign on the message m that can be verified by the group of n verifiers $\{ID_{V_i}\}$, he follows the steps below.

1. Computes $P_V = \sum_{i=1}^{n} \{H_1(ID_{V_i})\}$.
2. Randomly choose l from $\mathbb{Z}_q^*$, computes $Y = lP$ and $k = \hat{e}(lQ, P_{pub})$.
3. For $i \in \{1, 2, \cdots, n\}$, computes $Z_i = lH_1(ID_{V_i}) + lQ$.
4. Randomly chooses $U_2 \in \mathbb{G}_1$ and computes $h_2 = H_2(m||L||U_2||k)$.

5. Chooses $r'_1 \in_R \mathbb{Z}_q^*$, computes $U_1 = r'_1 H_1(ID_S) - U_2 - h_2 P_V$.
6. Computes $h_1 = H_2(m||L||U_1||k)$ and $V = (h_1 + r'_1)S_{ID_S}$.
7. Outputs the signature $\sigma = \{U_1, U_2, V, Y, Z_1, Z_2, \cdots, Z_n\}$.

`Verify`: The verifier ID_{V_i} performs the following steps to verify a SM-DVS.

1. Computes $P_V = \sum_{i=1}^{n} \{H_1(ID_{V_i})\}$ and $k' = \hat{e}(P_{pub}, Z_i)/\hat{e}(Y, S_{ID_{V_i}})$.
2. Computes $h_1 = H_2(m||L||U_1||k')$ and $h_2 = H_2(m||L||U_2||k')$.
3. Return $\top$ if $\hat{e}(P_{pub}, U_1 + h_1 H_1(ID_S) + U_2 + h_2 P_V) = \hat{e}(P, V)$, $\perp$ otherwise.

Efficiency. Only a constant number of pairings are required (sign:1, verify:4).

Security. The security model is basically the same as that in [6], with additional private key extraction query capturing the insider security of ID-based system and a natural extension from 2 verifiers to n verifiers. The scheme's unforgeability and signer ambiguity are directly related to the 1-out-of-n-groups ID-based ring signature in [2]. The signer-privacy can be proven using the idea of the proof of the multi-recipient ID-based encryption scheme against chosen-plaintext-attack (CPA) in [1], and that of the ID-based strong-DVS scheme in [4], yet the signing query of the challenge message can be supported. Thanks to the random oracle model and the bilinear pairing, we do not need decryption oracle from CCA2 security to answer verification queries. CPA security is sufficient since verification can be done by checking whether there exists an input-output tuple in the random oracle simulation satisfy some correct relationship among the signature's components by using pairing.

References

1. Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. Efficient Multi-receiver Identity-Based Encryption and Its Application to Broadcast Encryption. *Public Key Cryptography - PKC 2005*, *LNCS* 3386, pp. 380–397, 2005.
2. Sherman S.M. Chow, S.M. Yiu, and Lucas C.K. Hui. Efficient Identity Based Ring Signature . *Applied Cryptography and Network Security, ACNS 2005*, *LNCS* 3531, pp. 499–512. Also available at Cryptology ePrint Archive, Report 2004/327.
3. Yvo Desmedt. Verifier-designated Signatures. Available at http://www.cs.fsu.edu/~desmedt/lectures/verifier-designated-signatures.pdf.
4. Xinyi Huang, Willy Susilo, Yi Mu, and Futai Zhang. Short (Identity-Based) Strong Designated Verifier Signature Schemes. *Information Security Practice and Experience, ISPEC 2006*, *LNCS* 3903, pp. 214–225.
5. Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated Verifier Proofs and Their Applications. *EUROCRYPT '96*, *LNCS* 1070, pp. 143–154.
6. Fabien Laguillaumie and Damien Vergnaud. Multi-designated Verifiers Signatures. *Information and Communications Security, ICICS 2004*, *LNCS* 3269, pp. 495–507.
7. Shahrokh Saeednia, Steve Kremer, and Olivier Markowitch. An Efficient Strong Designated Verifier Signature Scheme. *Information Security and Cryptology - ICISC 2003*, *LNCS* 2971, pp. 40–54.
8. Yi Mu Willy Susilo, Fangguo Zhang. Identity-Based Strong Designated Verifier Signature Schemes. In *Australasian Conference on Information Security and Privacy, ACISP 2004*, volume 3108 of *LNCS* 3108, pp. 313–324.

Author Index