2007

SEFAP: An Email System for Anti-Phishing

Qoing Ren
University of Wollongong, qr02@uow.edu.au

Yi Mu
University of Wollongong, ymu@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Publication Details

This conference paper was originally published as Qoing Ren, Yi Mu, Susilo, W., SEFAP: An Email System for Anti-Phishing, 6th IEEE/ACIS International Conference on Computer and Information Science ICIS 2007, 11-13 Jul, 782-787.

SEFAP: An Email System for Anti-Phishing

Abstract
More and more users are suffering from email-based phishing attacks over the past years. Despite the use of various technologies for anti-phishing, phishing is still one of most serious attacks against Internet users. Email phishing attacks fabricate the email's origin. Unfortunately, current email server systems can not authenticate the genuineness of in-coming emails. In this paper, we present a novel antiphishing mechanism: Signed Email for Anti-Phishing (SEFAP), designed to automatically identify an email's origin to mitigate email phishing attacks. The SEFAP system is an extendable secure cryptographic system that accommodates multiple signature schemes. SEFAP can adopt any signature scheme which has two properties: Identity-based and repudiability. Identity-based property removes the unrealistic full PKI infrastructure deployment requirement and the repudiability property protects sender's privacy. We describe how to integrate the SEFAP system into a standard SMTP server. We also proposed an efficient implementation based on a novel ID-based ring signature scheme.

Keywords
Anti-Phishing, Malicious Email, Digital Signature

Disciplines
Physical Sciences and Mathematics

Publication Details
This conference paper was originally published as Qoing Ren, Yi Mu, Susilo, W., SEFAP: An Email System for Anti-Phishing, 6th IEEE/ACIS International Conference on Computer and Information Science ICIS 2007, 11-13 Jul, 782-787.

SEFAP: An Email System for Anti-Phishing

Qiong Ren, Yi Mu, Willy Susilo
Centre for Computer and Information Security Research,
School of Computer Science and Software Engineering,
University of Wollongong,
Northfields Avenue Wollongong, NSW 2522, Australia
{qr02,ymu,wsusilo}@uow.edu.au

Abstract

More and more users are suffering from email-based phishing attacks over the past years. Despite the use of various technologies for anti-phishing, phishing is still one of most serious attacks against Internet users. Email phishing attacks fabricate the email's origin. Unfortunately, current email server systems can not authenticate the genuineness of in-coming emails. In this paper, we present a novel anti-phishing mechanism: Signed Email for Anti-Phishing (SEFAP), designed to automatically identify an email's origin to mitigate email phishing attacks. The SEFAP system is an extendable secure cryptographic system that accommodates multiple signature schemes. SEFAP can adopt any signature scheme which has two properties: Identity-based and repudiability. Identity-based property removes the unrealistic full PKI infrastructure deployment requirement and the repudiability property protects sender's privacy. We describe how to integrate the SEFAP system into a standard SMTP server. We also proposed an efficient implementation based on a novel ID-based ring signature scheme.

Keywords: Anti-Phishing, Malicious Email, Digital Signature.

1 Introduction

Email is an essential component of the Internet infrastructure. Email has become a vital part of our life with its convenient and fast communication service. However, email is not what it was designed to be. More than 75% emails on the Internet are unsolicited emails such as Spam, imposter email and so on [1]. Compared with Spam, phishing email is a kind of malicious attacks to aim to steal person's private information such as bank account passwords. Email has become an abused and powerful tool for imposter to launch malicious attacks. The number of these attacks and the cost caused by these attacks are increased each year [3]. Email-based phishing attacks become more sophisticated recently.

Email-based phishing attacks are one kind of malicious imposter emails because the email origin of this kind of emails is fraudulent. For instance, a phisher can easily launch an email phishing attack by sending an email claimed from anyone he wants to impersonate because the email server (SMTP: the Simple Mail Transfer Protocol [2]) never authenticates the reality of the email origin. In other words, the sender's real email address could not be the same email address as it claimed in the "from field" of the email header. The fraudulent sender's email address and the official looking email contents make the huge success of those phishing attacks which waste the significant amount of human and machine resources.

Mechanism called MAUDE, Multiserver Authentic User Detection, dealing with malicious imposter emails with attachment, was explored in [4] in 2005. However, it depends on the interaction between the incoming email server and the outgoing email server. It also requires the email servers have the established trust (i.e., sharing a common secret), which requires the unrealistic, widespread common key exchange and deployment. Mechanism, CAMEL (ContAining Malicious Emails Locally), dealing with the malicious imposter email including the case that the claimed email senders could have been compromised by the adversary, was explored in [6] in 2006. The mechanism was designed to block outgoing malicious imposter emails to avoid the malicious imposter email attacks. However, first, email header (e.g., the From and Subject fields) can be forged easily. Second, the adversary does not have to use ContAining or MAUDE system to launch attacks since he can easily build his own email server. Thus, both ContAining and MAUDE can not block such kind of malicious imposter email attacks.

The drawback of authentication with SMTP protocol is partially responsible for philshing. In this work, we propose a secure authenticated mechanism that allows the email ori-

6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)
0-7695-2841-4/07 $25.00 © 2007 IEEE

gin to be authenticated for mitigating phishing attacks. We present our SEFAP system to provide the authentication of sender. SEFAP is designed to deal with both signed and unsigned email messages. Those email messages could come from the imposter's email servers which do not adopt any SEFAP system. In the case of signed incoming email, SEFAP verifies the genuineness of sender and outputs true or false to indicate email server's action. In the case of unsigned incoming email, SEFAP gives a warning to user and indicates email server to put it into unidentifiable email box. SEFAP consists of six system components. The first one is dispatcher, which is an optimization algorithm to balance the best time to verify the incoming emails. The second is PKG, Private key Generator, generating email server domain system parameters and publishing public parameters. The third is Extractor, generating its legal email user's private key and delivering it to its user within a secure channel. The fourth is S-Generator, Signature Generator, generating signature on the email content based on the selected signature scheme. The fifth is S-Verifier, Signature Verifier, verifying the signature on the chosen signature scheme and outputting true or false. The sixth is system management module to configure which signature scheme is adopted by the email server. All of these components are integrated with SMTP server, email server and clients. The SEFAP system can adopt any signature scheme if it is identity-based and have the property of repudiability.

The rest of this paper is organized as follows. In Section 2, we revisit current email systems. In Section 3, we describe the SEFAP system and show how to integrate the SEFAP system into a standard SMTP server. In Section 4, we discuss some cryptographic primitives and describe an efficient ID-based ring signature scheme, which can be applied to the SEFAP system. The final concluding remarks are given in Section 5.

2 Email Model

A current email system network consists of multiple email servers which have their own unique domain name. Email servers and DNS (Domain Name Service) provide convenient and efficient email service for users around the world. Each email system transmits email messages based on SMTP and provides web page client (Graphic User Interface) for email user to manage his email including compose new email, read incoming email, delete email and so on. Each email uses a domain address and depends on translation service (DNS) to translate from domain name to IP address. Outgoing email server sends email and incoming email server receives email from other email servers.

We assume that an email message consists of three parts: header (From field, To field, Subject, CC field, Reply field), a non-attachment content and an attachment. The message, which is signed, includes these three parts.

Definition 1 *A phishing email is an email sent to a recipient R with ($whitelist_R$, $Filter_R$) such that*

- *Pr[sender(Email) $\in whitelist_R$]=1, meaning that the sender's email address is on R's whitelist. Thus, in the original email system, no action has been taken at such email.*
- *Pr[$Filter_R$(Email)]=1, meaning that the non-attachment content looks perfect and legitimate but it's actually impersonated.*

3 The SEFAP Mechanism

The SEFAP system is an extendable, signature-based, and secure email system. The SEFAP consists of three layers: presentation layer, business layer and database information layer. Only the presentation layer is accessible by users and the other two lower layers are protected and accessible to system administrators only.

SEFAP consists of SEFAP client, SEFAP server, and database server which communicate through a secure channel. SEFAP authenticates the origin of incoming email and takes appropriate actions to suspicious phishing email in order to mitigate phishing attacks. SEFAP is designed to adopt signature schemes. Since each original email server already has its unique domain name on the Internet, domain level system parameters are designed to be generated by the SEFAP server located in each physical email server. The SEFAP server provides six sub-services: system management module, signature-scheme based parameters setup module, private key extraction module, verification module, synchronization module and dispatcher module. The system management module specifies a signature scheme for the current email server system. It can also add a new signature scheme into the SEFAP system and delete an old signature scheme from the SEFAP system. Thus, SEFAP can be updated for a new signature scheme though uploading the new signature-scheme component to the SEFAP system. The signature-scheme-based parameter setup module generates domain system parameters under a selected scheme. The private key extraction module generates its user's private key and delivers it to its user with a secure channel. The verification module provides the signature verification service even if the outgoing email server uses a different signature scheme from the incoming email server, and instructs the email server to take appropriate actions to unidentified emails. The synchronization module deals with the domain parameter synchronization operation and publication. The dispatcher module provides the most efficient process schedule to verify incoming email.

The SEFAP client located at the sender's machine is responsible for signing email when the user instructs the email server to send an email message. The SEFAP client also is in charge of system parameters license synchronization including checking the parameter version, expiration, and signature scheme identification using an efficient synchronization algorithm.

3.1 The Protocol

The SEFAP system (Fig. 1) consists of system management, Private Key Generator (PKG), Extractor, S-Generator, S-Verifier, synchronization and dispatcher. In the initialization of the email system, the SEFAP Server runs PKG for the system parameter generation, then publishes public parameters. It also calls Extractor to generate the private key for each new email user after his registration has been approved by the email server. When a user sends its outgoing email, SEFAP call S-Generator to sign it. When a user requires to read its incoming email, SEFAP implements S-Verifier and S-Verifier outputs true or false which implies the action of the email server. If it is true, SEFAP informs the email server to process the email as in the original email system; otherwise, the email is suspended and an appropriate action is taken (e.g., SEFAP blocks the

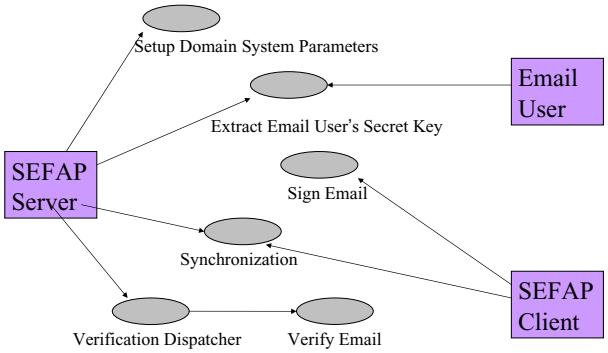

Figure 1. USE CASE OF SEFAP.

The SEFAP server is deployed on each email server an the SEFAP client runs in a PC terminal. The SEFAP i based on Identity-based signature schemes that consists c the following algorithms.

- PKG: A private key generation algorithm, on input a security parameter κ, Signature Scheme $\mathcal{SN}$, generates email domain system parameters: public parameters including the system public key, the system private key S kept in SEFAP.
- $Extractor$: A user private key extract algorithm, on input user's ID, the public parameters W, the system private key S, and the signing scheme SN, outputs user's private key S_{ID}.
- S-Generator, A PPT signature generation algorithm, on input a message M, the private signing key S_{ID}, and the signing scheme $\mathcal{SN}$, outputs a signature σ.
- S-Verifier: A deterministic signature verification algorithm, on input the message M, the system public parameters W, a signature σ, the user's ID, the signature scheme SN, outputs true or false.

Figure 1 shows a usage scenario of view for the SEFAP system. It organized information collected during requirement elicitation into use-cases.

3.2 Information Flows

We show how a concrete implementation of the SEFAP mechanism is integrated into a standard SMTP Server. Figure 2 emphasizes the control/information flow of the SEFAP system. It describes the signing process flow. The functionality of the SEFAP Server is to setup system parameters, extract private key for each user, control verification payload balance using connection pool and balance equalizer, verify each incoming email, and decide whether the response action is to block incoming email or give a warning.

The interpretation of the above scenario has been given

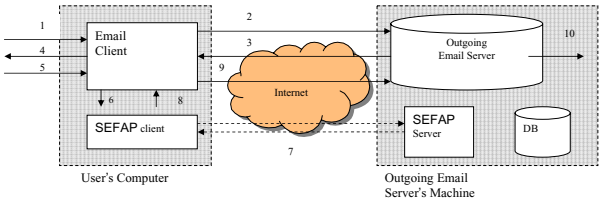

Figure 2. SEFAP Outgoing Email Process Methodology

An interpretation of the above outgoing email processes scenario follows:

1. Alice requires to access her email account though email client (e.g., web page email logging interface) using her username/ password.
2. The email server verifies if Alice's account and her password are correct.
3. The email server takes an appropriate action and replies to the email client.
4. The email client shows Alice's personal email management interface if the account is correct, otherwise, an error will be delivered to the email client.

0-7695-2841-4/07 $25.00

5. After composing message, Alice requests to send the email.
6. The email client requests the SEFAP client to sign the email and delivers the system parameters license to SEFAP client.
7. The SEFAP client verifies if its system parameters license is expired. If so, the system parameter synchronization is required and the SEFAP client downloads system parameters from the SEFAP server and also updates Alice's private key certificate.
8. The SEFAP client generates the signature on the email and delivers the signature back to the email Client.
9. The email client forwards the signature and message pair to the email server.
10. The email server sends the signed outgoing email to the destination incoming email server.

How to implement a verification process for the incoming email is given in Figure 3. Also the verbal interpretation of the scenario has been given in it.

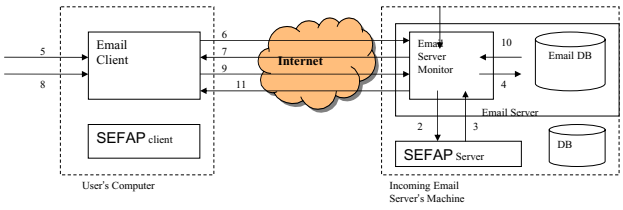

Figure 3. SEFAP Incoming Email Process Methodology

In Figure 3, the incoming email process, the email server consists of two components: email server monitor and email database. We interpret the process as follows:

1. The email server receives incoming email.
2. The email server requests the SEFAP server to verify the signature if the incoming email consists of a message signature pair.
3. The SEFAP server chooses the corresponding verification algorithm to verify message-signature pair under the specified signature scheme. Then, it responds with accept, reject, or unknown.
4. The email server takes an appropriate action to deal with the incoming email and updates the record in the database for the user identity of this incoming email.
5. Alice requests to access her email account though the email client (e.g., the web page email logging interface) using her username/ password.
6. The email server verifies if Alice's account and her password are correct.
7. The email client shows Alice's personal email management interface if the account is correct, otherwise, error page will be delivered to email client via browse.
8. Alice asks to read her email.
9. The email client requests the email server to extract the email from the server.
10. The email server retrieves the email and its identity verification status from the database.
11. The email server sends the result to the email client

The encrypted certificate including the private key is generated by the SEFAP Server and securely delivered to its email user, after his registration has been validated. Only its legal email user knows and keeps his private key (that could be stored a secure portable device). The SEFAP system provides the following security properties:

1. *Password Exposure Free*

   Since the SEFAP server will authenticate the user through the user's private key, even if the user's password has been revealed, attackers still can not access user's account. Thus, the malicious software such as standard keyloggers, which can only capture user input, can not compromise user's account. This will protect user's account from revealing the password.

2. *Enhanced Privacy*

   The SEFAP system utilizes privacy-enhanced signature schemes. We require that the signature schemes have ambiguity property. For example, we can make use of the ring structure scheme consisting of sender and recipient, where the recipient knows exactly the signature is indeed signed by the sender; However, he can not convince any third party that the email comes from the sender since he might forge the email by himself. This therefore protects sender's privacy.

3. *Authentication*

   Most of email-based phishing attacks succeed in large part because they fabricate the origin of email. Since a repudiable signature can only be generated by sender or receiver. When the recipient receives the signed email, he can ensure the email's origin since he has not signed it. Thus, SEFAP can ensure the authenticity of email's origin.

4. *Unforgeability*

   In our ID-based ring signature scheme, given a valid signature, the adversary can not generate a new valid signature in polynomial time.

5. *Non-interaction*: Because the use of Identity-based property, unrealistic and full deployment of Public Key Infrastructure (PKI) is no longer required.

4 Implementation: An ID-based Ring Signature Scheme for SEFAP

In this section, we show how to apply a ring signature scheme to SEFAP. We describe our construction for an ID-based Ring signature scheme, which could be applied to the SEFAP system against malicious impostor mails such as phishing attacks. Since the Identity-Based systems allow a user's Identity string such as email address as a public key, the Identity-Based property eliminates PKI. Moreover, The separable property in our ring signatures allows a sender and a receiver possess independent domain system parameters with the same signature scheme. Those two properties make the deployment realistic. The advantage of two-party ring signatures, where the signers of the ring only consists of the sender and the recipient, is to make the signed email have repudiability. We present our ID-Based ring signature scheme using pairings.

4.1 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 $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_M$ be a bilinear mapping with the following properties:

1. *Bilinearity*: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ 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 $\hat{e}(P, Q) \neq 1$.
3. *Computability*: There exists an efficient algorithm to compute $\hat{e}(P, Q)$ 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 $param = (p, \mathbb{G}_1, \mathbb{G}_M, \hat{e}, P)$ 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 $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_M$ 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.

There exist a sender, Alice, a recipient, Bob, and TA (Trustee, like SEFAP) in the system. TA executes the system setup and user's secret key generation. We assume that all of participants have setup their public key setting, and we donate by S_{ID_s} the secret key for the sender and by S_{ID_r} the secret key for the receiver R. An ID-based Ring signature scheme consists of four efficiently computable algorithms:

- $\mathcal{SETUP}$: Private Key Generator (PKG) takes as input a security parameter ℓ and generates a pair of master secret key MSK and master public key MPK.
- $\mathcal{EXTRACT}$: A deterministic algorithm that, on input MSK and an identity string ID, outputs the trapdoor information S_{ID} corresponding to the identity.
- $\mathcal{SG}$: A probabilistic polynomial algorithm accepts MPK, a set of identity string ID, a message m and a random number $r \in Z_q^*$, and outputs a signature $\sigma(m)$.
- $\mathcal{SV}$: A probabilistic polynomial algorithm takes input a signature $\sigma(m)$, a list of ID, and system parameters and outputs either accept or reject.

In addition to the above main algorithms, we also require the following,

- Unconditional Ambiguity. The adversary can not tell the identity of the signer with a probability larger than $1/r$, where r is the cardinality of the ring, even assuming that he has unlimited computing resources.
- Unforgeability. The adversary can not generate additional signatures based on original signature.

4.2 Our Scheme

In this section, we present our concrete construction of an ID-based Ring Signature scheme from the bilinear pairings. SEFAP initiates $\mathcal{SETUP}$ and $\mathcal{EXTRACT}$ operations for system setup. After that, SEFAP executes $\mathcal{SG}$ for outgoing emails and executes $\mathcal{SV}$ for incoming emails. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_2$ be a bilinear mapping, where $\mathbb{G}$ be a GDH group of prime order q. We assume $i \in \{0, 1\}$ representing either a sender or a receiver.

- $\diamond$ $\mathcal{SETUP}$: SEFAP runs PKG to generate and publish global system params $\{ \mathbb{G}_i, \hat{e}, q_i, \kappa_i, p_i, P_{pub_i}, h_0, h_1 \}$, where $h_0 : \{0,1\}^* \rightarrow \mathbb{G}$, $h_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$, $i \in \{A, B\}$. A represents a sender domain system and B represents a recipient domain system. Choose

a random number $s_i \in Z_{q_i}^*$ as the master secret key , keep s_i known by SEFAP itself, and sets $P_{pub_i} = s_i P_i$ as the master public key, which is published and could be gotten using the synchronization operation.

◇ $\mathcal{EXTRACT}$: Sender submits his identity information ID_s to SEFAP, then SEFAP runs $Extractor$ to compute $Q_{ID_s} = h_0(ID_s) \in Z_q^*$ and returns his secret key $S_{ID_s} = (s+Q_{ID_s})^{-1}P \in \mathbb{G}$ to the sender via a secure channel. The receiver does the same.

◇ $\mathcal{SG}$: Given a message m containing (email header, subject, content and attachment), a set of identity string IDs (Sender's ID and Receiver's ID), and $D_i = \hat{e}(P_i, P_i)$ stored in SEFAP, SEFAP chooses a random integer $r \in Z_q^*$ and does the following:

- Compute $C_{k+1} = h_1((M)||\hat{e}(P_i, rP_i)) \in Z_{q_i}^*$, where $i \in (0,1)$, $i = k$, k denotes the sender, and assumes that $k = 1$ and then gets $C_{k+1} = C_0$.
- Generate the forward ring sequence: pick $T_0 \in \mathbb{G}$, compute $C_{i+1} = h_1((M)||\hat{e}(h_0(ID_i)P_i + P_{pub_i}, T_i)D_i^{C_i}) \in Z_{q_i}^*$, where $i = 0$
- Compute $T_k = (rP_k - C_k P_k)(s + Q_{ID_k})^{-1} = (r - C_k)S_{ID_k}$

The message-signature tuple is $\sigma \leftarrow \{m, C_0, T_0, T_1\}$.

◇ $\mathcal{SV}$: To verify the signature, retrieve system parameters $D_i = \hat{e}(P_i, P_i)$ from SEFAP for sender and receiver, and compute

$C_{i+1} = h_1((M)||\hat{e}(h_0(ID_i)P_i + P_{pub_i}, T_i)D_i^{C_i})$ for $i \in \{0,1\}$, Accept if $C_2 = C_0$.

Thus

$$C_1 = h_1((M)||\hat{e}(h_0(ID_0)P_0 + P_{pub_0}, T_0)D_0^{C_0}),$$

$$C_2 = h_1((M)||\hat{e}(h_0(ID_1)P_1 + P_{pub_1}, T_1)D_1^{C_1}).$$

◇ Correctness.

$$T_k = (rP_k - C_k P_k)(s + Q_{ID_s})^{-1}$$

$$= (r - C_k)S_{ID_k}.$$

$$C_2 = h_1((M)||\hat{e}(h_0(ID_1)P_1 + P_{pub_1}, T_1)D_1^{C_1})$$

$$= h_1((M)||$$

$$\hat{e}((h_0(ID_1)+s)P_1, (rP_1 - C_1P_1)(s+Q_{ID_1})^{-1})D_1^{C_1})$$

$$= C_0.$$

The proposed ID-Based signature scheme satisfies the properties of enhanced privacy, non-repudiation, unforgeability, and non-interaction.

In our scheme, we reduce pairing computation in each ring node calculation compared with the one presented by Fangguo Zhang and Kwangjo Kim[9]. Also our scheme could be extended to multiple potential signers to construct the ring for other systems. Because the ring sequence generation except the real singer ring node is the same during $\mathcal{SG}$ and SV. Our method provides an alternative to other signature-based methods (e.g., those using methods given in [5], [7], [8]).

References

[1] MessageLabs. Annual Email Security Report, December 2004. http://www.messagelabs.com/intelligence/2004report.

[2] Jonathan B. Postel. Simple Mail Transfer Protocol. RFC 821, August 1982.

[3] Anti-Phishing Working Group. Phishing activity trends report, December, 2006. Available at http://www.antiphishing.org.

[4] E. Kartaltepe and S. Xu. On Automatically Detecting Malicious Imposter Emails. Frontiers in Artificial Intelligence and Applications, Volume 128, pages 33 - 47, IOS press, 2005.

[5] F. Zhang, R. Safavi-Naini, and W. Susilo. ID-Based Chameleon Hashes from Bilinear Pairings. Cryptology ePrint Archive, Report 2003/208, 2003. Available at http://eprint.iacr.org/2003/208

[6] E. Kartaltepe and S. Xu. Towards Blocking Outgoing Malicious Impostor Emails. In the Proceedings of the 2nd International Workshop on Trust, Security and Privacy for Ubiquitous Computing, pages 657-661, IEEE Press, 2006.

[7] G. Ateniese and B. de Medeiros, Identity-based chameleon hash and applications, Financial Cryptography 2004 (FC'04), Volume 3110/2004 of LNCS, pages 164-180, Springer-Verlag, 2004.

[8] D. Boneh, B. Lynn, and H. Shacham, Short signatures from the Weil pairing, Advances in Cryptology - ASIACRYPT 2001, Volume 2248/2001 of LNCS, pages 514-532, Springer-Verlag, 2001.

[9] F. Zhang, K. Kim, ID-based blind signature and ring signature from pairings. Advances in Cryptology - ASIACRYPT 2002, Volume 2501/2002 of LNCS, pages 533-547, Springer-Verlag, 2002.

0-7695-2841-4/07 $25.00