Mitigating Phishing with ID-based Online/Offline Authentication

Qiong Ren, Yi Mu and Willy Susilo

Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, NSW 2522, Australia
Email: {qr02,ymu,wsusilo}@uow.edu.au

Abstract

Enforcing strong authentication is an option to mitigate phishing. However, existing authentication methods, like traditional digital signatures, require unrealistic full deployment of public key infrastructure(PKI) and destroy email users' privacy in that the identity of an email sender is automatically revealed to the public. There have been some works in the literature, where the technology of deniable authentication is adopted and sender's privacy can be protected. However, the additional computation introduced into the system is obviously a drawback. In this paper, we introduce the notion of online/offline authentication into anti-phishing, in order to construct an efficient and secure anti-phishing scheme. It is commonly known that a generic online/offline signature can be constructed with a traditional chameleon function. Nevertheless, a standard chameleon function suffers from so-called key-exposure attacks. To tackle this issue, we propose an efficient chameleon function without key-exposure, which is especially suitable for constructing efficient online/offline signatures that are applicable to mitigating phishing. We also demonstrate how to apply our novel scheme to a traditional email system.

Keywords: Anti-Phishing, Online/Offline Signature, Identity-Based System, Chameleon Hash Function.

1 Introduction

Email systems are essential components of the Internet infrastructure. As an asynchronous communication medium, email systems provide a free and extremely fast delivery service, which makes it become the most popular application on the Internet. With the growth of the Internet, numerous email servers have been established on the Internet. However, security attacks in email systems have also increased. The worst one is email-based phishing, which is designed to lead recipients to official-looking spoof websites in order to trick them into divulging their personal and financial information. The number of the victims and the cost of these attacks increased each year (Anti-Phishing. 1989). Despite the use of various technologies against phishing, it is still one of the most serious attacks against Internet users. Moreover, phishing attacks become more sophisticated recently. The reason of phishing attacks flooding on the Internet is SMTP (Simple Mail Transfer Protocol (Postel 1982)), the core technology of e-mail system, was designed without consideration of the sender address's authenticity.

To provide a solution to mitigating email phishing attacks, we are concerned with the following issues:

1. *Practicalness*: Deployment is easy and realistic.
2. *Authenticity*: An email is really from the origin address as it claimed.
3. *Non-repudiation*: An email sender can not deny that he has sent the email, while a judge deals with the case of a dispute.
4. *Non-Transferability*: the recipient can not convince others that the sender actually sent the message. This provides the sender's privacy as in the conventional mail system.

Strong authentication based on digital signatures has been considered for mitigating phishing in the literature (Anti-Phishing.). The idea of using digital signatures obviously works since they allow email recipients to check whether or not a sender is genuine. However, using a traditional signature scheme does not provide a desirable solution to email systems, since it destroys user privacy that is enjoyed by Internet users. By user privacy, we mean that in a traditional email system, an email sender can deny that he or she has sent a message to the public since email address does not provide any authentication to the sender and the message at all. This privacy is enjoyed by email users in current email systems. Therefore, we have to find a suitable way. Contrary to a traditional signature based authentication scheme, deniable authentication (Dwork et al. 1998) provides a useful property we can make use of. Using a scheme of deniable authentication, an email sender can convince a specific recipient that he is indeed the sender, while his identity remains ambiguous to others. The idea of deniable authentication for mitigating phishing is not new. There are several published works in the literature (Susilo et al. 2004, 2003, Naor 2002). Very recently, identity based deniable authentication has also been developed for applications in anti-phishing (Ren et al. 2007). All above mentioned schemes have a drawback of additional computational overhead. We understand that it is inevitable while cryptography is utilized. Our aim in this paper is to propose a more efficient approach for using identity-based deniable authentication, where we allow most of computational overhead to be handled on a *offline* phase and a signing process becomes very efficient.

In 1989, Even, Goldreich and Micali introduced the notion of online/offline signature (Even et al. 1989). It was designed to provide a fast signature generation, which can be applied to portable device applications like smart card applications. The main idea is to split the signature generation into two phases: offline phase and online phase. Before the actual message is ready, the signing algorithm runs in the offline phase and the most signature computation is done in this phase. Once the signing message is available, the signing algorithm runs in the online phase and retrieves the stored results of the offline phase to sign the real message.

In 2000, chameleon hash function was introduced by Krawczyk and Rabin (Krawczyk et al. 2000). Based on a chameleon hash function, Shamir and Tauman (Shamir et al. 2001) proposed a new "hash-sign-switch" paradigm to achieve more efficient online/offline signature scheme compared with the scheme due to Even, Goldreich and Micali. However, Shamir and Tauman's online/offline signature scheme suffers from the limitation of key exposure due to using a traditional chameleon hash function. This means that a recipient can possibly obtain a hash collision and use it to recover the sender's trapdoor information, i.e., the secret key. In the case of key exposure, the recipient can forge any signature. To avoid this problem, Chen et al. (Chen et al. 2007) proposed an improved online/offline signature scheme without key exposure, which solved the key exposure problem. Unfortunately, this signature scheme is not identity-based.

Our contributions are outlined as follows.

- *Construction of novel ID-based on-line/off-line signature scheme without key exposure.* Utilizing an existing ID-based deniable signature scheme and our ID-based chameleon hash without key exposure based on Shamir-Tauman's "hash-sign-switch" paradigm, we construct an ID-based online/offline signature scheme without key exposure.
- *Applications for Mitigating Phishing.* Our ID-based online/offline signature scheme is applicable to the authentication of email system. ID-based property simplifies the key generation and deployment. Our ID-based online/offline signature scheme without key exposure can be integrated into the SEFAP system (Ren et al. 2007).

The rest of this paper is organized as follows. In Section 2, we review the cryptographic tools and notions required throughout this paper and present our generic ID-based online/offline signature scheme without key exposure. In Section 3, based on our scheme, we describe an implementation which has potential applications in anti-phishing. In Section 4, we described a new protocol to authenticate email for anti-phishing. The final concluding remarks are given in Section 5.

2 ID-based Online/Offline Signature Scheme without Key Exposure

In this section, we present a novel ID-based online/offline signature scheme without key exposure. Shamir-Tauman's "hash-sign-switch" paradigm was based on the chameleon hash function. According to chameleon hash's property, no one except the holder of the information can easily generate a hash collision $hash(m', r') = hash(m, r)$. As the trapdoor holder, the signer can finish the most computation, like the calculation of hash value based on a random pair message-integer(m, r) and signature generation on the hash value, in the offline phase. Therefore, the computation in online phase is less and the signing process is efficient. Using the ID-based chameleon hash function extended in (Ren et al. 2007), our ID-based online/offline signature scheme eliminates the key exposure limitation from Shamir-Tauman's "hash-sign-switch" paradigm. Our ID-based online/offline signature scheme is a construction through combining any efficient secure deniable signature scheme and our ID-based chameleon hash function without key exposure and with properties of fast signing, non-repudiation and non-transferability for the authentication of email systems with preserved privacy.

2.1 Bilinear Pairings

Let $\mathbb{G}_1, \mathbb{G}_2$ be cyclic additive groups with a prime q generated by P_1, P_2, respectively. Let $\mathbb{G}_M$ be a cyclic multiplicative group with the same order q. We assume there is an isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ such that $\psi(P_2) = P_1$. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_M$ 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 $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_M$ 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 ID-based Chameleon Hash

Definition 1 *(ID-based Chameleon Hash) An ID-based chameleon hashing scheme consists of four efficiently computable algorithms (Setup, Extract, Hash, Forge):*

- Setup*: Private Key Generator (PKG) takes as input a security parameter λ, generates a pair of master secret key MSK and master public key MPK, and publishes master public key MPK and public parameters.*
- Extract*: A deterministic algorithm that, on input MSK and an identity string ID, outputs the trapdoor information S_{ID} corresponding to the identity.*
- Hash*: A probabilistic algorithm accepts MPK, an identity string ID, a message m and a random number $r \in Z_q^*$, and outputs a hashed value $h =$ $\mathsf{Hash}(MPK, ID, m, r)$.*
- Forge*: For any valid hash value h, an algorithm $\mathcal{F}$ accepts inputs MPK, the corresponding trapdoor information S_{ID} associated with ID, a message m', and a hash value h of a message m, and r, outputs r' that satisfies*

$$\mathsf{Hash}(MPK, ID, m', r') = \mathsf{Hash}(MPK, ID, m, r)$$

Note that a public key associated with an Identity String ID can be efficiently calculated by anyone using the public parameters. An ID-based chameleon hashing scheme has the following properties:

- *Collosion resistance*: There exists no probabilistic polynomial time algorithm $\mathcal{A}$ that accepts as inputs a message m, a random integer r, another message $m' \neq m$, and a public key Q_{ID} associated with ID, and outputs a random integer $r' \neq r$ that satisfies $\mathsf{Hash}(MPK, ID, m', r') = \mathsf{Hash}(MPK, ID, m, r)$, with non-negligible probability.

- *Trapdoor collisions*: There exists an efficient algorithm $\mathcal{F}$ that accepts as inputs a trapdoor information S_{ID} associated with an Identify String ID, a pair of a message m and a random number r, and another message $m' \neq m$, outputs r' that satisfies

  $$\mathsf{Hash}(MPK, ID, m', r') = \mathsf{Hash}(MPK, ID, m, r)$$

- *Semantic security*: From the hash value it is infeasible to determine which message is likely to have resulted in such value by an application of the hash algorithm.

2.3 An Efficient ID-based Chameleon Hash without key exposure

An efficient ID-based chameleon hash scheme without key exposure is introduced in this sub-section. This ID-based chameleon hash scheme not only has all the properties of a general ID-based chameleon hash scheme, but also prevents any third party from recovering the trapdoor holder's private key (S_{ID}) even after a hash collision was forged by the receiver. Moreover, in that deniable signature scheme, it prevents a signer from generating a hash collision using any information based on the existing hash value. To incorporate our Online/Offline scheme, in our hash scheme, the trapdoor information holder is the signer.

The scheme is described as follows.

◇ Setup: PKG chooses a random number $s \in Z_q^*$ as a master secret key and sets $P_{pub} = sP \in \mathbb{G}$ as the corresponding master public key. Define cryptographic hash functions H_0:$\{0,1\}^* \rightarrow \mathbb{G}$, h_1:$\{0,1\}^* \rightarrow Z_q^*$. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_M$ be a pairing. PKG publishes $\{\mathbb{G}$, $\mathbb{G}_M$, $\hat{e}$, q, κ, p, P_{pub}, H_0, $h_1\}$ as system parameters and keeps s which is known only to itself.

◇ Extract: User submits his identity information ID to PKG. PKG computes the user's public key as $Q_{ID} = H_0(ID)$, and returns $S_{ID} = s^{-1}Q_{ID}$ to the user as his private key.

◇ Hash: Given a message m, chooses a random integer $r \in Z_q^*$, an identity string ID, lets $B = \hat{e}(P, H_0(ID))$ and C=$\hat{e}(P_{pub}, H_0(ID))$, computes (B^r, C^r), outputs a hash value

$$h = Hash(MPK, ID, m, B^r, C^r) = B^{h_1(m)}C^r$$

$$= \hat{e}(P, H_0(ID))^{h_1(m)}\hat{e}(P_{pub}, H_0(ID))^r.$$

◇ Forge: For any valid hash value h, an algorithm $\mathcal{F}$ accepts inputs MPK, a trapdoor information S_{ID} associated with identity string ID, another message $m' \neq m$, and a hash value h of a message m, B^r and C^r, outputs $B^{r'}$ and $C^{r'}$

$$Hash(MPK, ID, m', B^{r'}, C^{r'})$$

$$= Hash(MPK, ID, m, B^r, C^r),$$

where

$$B^{r'} = B^r\hat{e}(P, S_{ID})^{(h_1(m)-h_1(m'))},$$

$$C^{r'} = C^rB^{(h_1(m)-h_1(m'))}$$

Note that

$$Hash(MPK, ID, m', B^{r'}, C^{r'}) = B^{h_1(m')}C^{r'}$$

$$= B^{h_1(m')}C^rB^{(h_1(m)-h_1(m'))}$$

$$= Hash(MPK, m, B^r, C^r)$$

and $< B, C, B^{r'}, C^{r'} >$ is a valid Diffie-Hellman tuple. Therefore, the forgery is successful.

The scheme has the property against revealing the trapdoor information to any third party if a collision forgery happened. This can be easily proved: collision forgery results in any third party to recover the information $\hat{e}(P, S_{ID})$:

$$B^{r'}/B^r = \hat{e}(P, S_{ID})^{(h_1(m)-h_1(m'))}$$

$$= \hat{e}(P, Q_{ID})^{s^{-1}(h_1(m)-h_1(m'))}$$

If s^{-1} can be recovered, then discrete log problem is not intractable. Recalling the definition of $S_{ID} = s^{-1}Q_{ID}$, we can easily reduce $\hat{e}(P, S_{ID}) = \hat{e}(P, Q_{ID})^{s^{-1}}$. However, any third party could not recover the trapdoor information S_{ID} from $\hat{e}(P, S_{ID})$. Hence, the holder of the trapdoor information can still surely use the secret key at anytime and anywhere after hash collision happened.

2.4 The Online/Offline Signature Scheme

In this section, we follow Shamir-Tauman's "Hash-Sign-Switch" paradigm (Shamir et al. 2001) to propose a much more efficient online/offline signature scheme. Compared with Chen et al. (Naor 2004), the advantage of Ren-Mu-Susilo's ID-based Chameleon Hash (Ren et al. 2007) is private key exposure-freeness in identity based system which is more practical. However, if we directly apply Ren-Mu-Susilo's ID-based Chameleon Hash (Ren et al. 2007) to our online/offline signature scheme, it requires to compute the chameleon hash value in the offline phase every time. To reduce the computation in the offline phase, we proposed a new ID-based chameleon hash scheme extended in (Ren et al. 2007) to make the chameleon hash value reusable. Compared with traditional chameleon signature scheme, in the online/offline signature scheme, the sender is the holder of the trapdoor information instead of the receiver. Ren-Mu-Susilo's ID-based Chameleon Hash (Ren et al. 2007) is not designed for the online/offline signature scheme. In that scheme, the receiver is the holder of the trapdoor information and the unique event δ requirement is used to guarantee the sender can not generate collision forgery after the receiver forges the collision. Hence, to achieve more efficient general online/offline scheme, we extended Ren-Mu-Susilo's ID-based Chameleon Hash (Ren et al. 2007) by removing the unique event δ requirement since signer is the holder of the trapdoor information and it is unnecessary to prevent a signer from re-using the hash value for a receiver. The main idea is that the chameleon hash scheme is private key exposure-freeness and the hash value can be securely reused. The most computation can be done in system setup and the offline phase. Hence, the signature can be efficiently generated in the online phase. Since the proposed ID-based chameleon hash scheme has no key exposure problem and also allows the reuse of the hash value, the hash value can be set in a system parameter generation phase.

The proposed ID-based online/offline signature scheme consists of four efficient algorithms as following:

- ◇ Parameters Setup: It consists of signature system setup and chameleon hash system setup.

  1. *ID-based Signature Scheme Setup*
     Let SG be a signature system parameters generation algorithm, SS be a signature signing algorithm and SV be a signature verification algorithm. Then, (SG, SS, SV) presents any secure ID-based signature scheme. Let SP_s be the public system parameters of the signature scheme.
  2. *Chameleon Scheme Setup*
     Let $\mathbb{G}$ be the cyclic additive groups generated P with a prime order q. Let $\mathbb{G}_M$ be a cyclic multiplicative group with the same order q. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_M$ be a bilinear mapping. PKG chooses a random number $s \in Z_q^*$ as master secret key MSK, which is only kept by itself, and sets $P_{pub} = sP$ $\in \mathbb{G}$ as master public key MPK. Define cryptographic hash functions H_0:$\{0,1\}^* \to$ $\mathbb{G}$, h_1:$\{0,1\}^* \to Z_q^*$. Let r be a random number from Z_q^*, m be a message, ID_s be a specified signer's identity string, and the ID-based chameleon hash function H_c is defined as follows:

     $$H_c(MPK, ID_s, m, B^r, C^r) = B^{h_1(m)}C^r$$

     Where

     $$B = \hat{e}(P, H_0(ID_s)), \quad C = \hat{e}(P_{pub}, H_0(ID_s))$$

     and $< B, C, B^r, C^r >$ is a valid Diffie-Hellman tuple.
     Let SP_{hc} be the chameleon hash scheme's public system parameters: $\{\mathbb{G}$, $\mathbb{G}_M$, $\hat{e}$, q, ℓ, P, P_{pub}, H_0, h_1, H_c $\}$. The public system parameters of the Online/Offline scheme are $\mathcal{SP} = \{SP_s, SP_{hc}\}$.

- ◇ Key Extraction:

  1. *Signing Private Key Extraction.* On input a security parameter k and an identity string ID_s, runs the key generation algorithm of the signature scheme, outputs the signing private key (SK_S).
  2. *Chameleon Scheme Private Key Extraction.* On input an identity string ID_s, computes public key $Q_{ID_s} = H_0(ID_s)$, runs the Key Extract algorithm of the chameleon hash scheme to obtain trapdoor key $S_{ID_s} = s^{-1}Q_{ID_s}$, computes $D = \hat{e}(P, S_{ID_s})$.
  3. Compute $B = \hat{e}(P, H_0(ID_s))$ and $C = \hat{e}(P_{pub}, H_0(ID_s))$ stored as constants, choose at random number $r_0 \in Z_q^*$ and a message m_0, compute (B^{r_0}, C^{r_0}), and output a hash value

     $$h = Hash(MPK, ID, m_0, B^{r_0}, C^{r_0})$$

     $$= B^{h_1(m_0)}C^{r_0}$$
  4. Store tuple $(m_0, B, C, D, B^{r_0}, C^{r_0}, h)$.

- ◇ Signature Generation:

  1. *Offline Phase*
     Retrieve the hash value h, run the signing algorithm SS with the key SK_s associated with the identify string ID_s to sign h, output partial signature $\sigma_i \leftarrow SS(h)$ on the message m_0, and store σ_i, where i denotes each signing event.
  2. *Online Phase*
     Given a new message m_i, retrieve the associated pair $(m_0, B^{r_0}, C^{r_0}, B, D, h, \sigma_i)$. Compute

     $$B^{r_i} = B^{r_0}D^{(h_1(m_0)-h_1(m_i))},$$

     $$C^{r_i} = C^{r_0}B^{(h_1(m_0)-h_1(m_i))}.$$

     Send the message-signature pair $(m_i, B^{r_i}, C^{r_i}, \sigma_i)$ to a verifier.

- ◇ Signature Verification:

  1. Compute the chameleon hash value as following:

     $$h = H_c(MPK, ID_s, m_i, B^{r_i}, C^{r_i})$$

     $$= B^{h_1(m_0)}C^{r_i},$$

     where

     $$B = \hat{e}(P, H_0(ID_s)), \quad C = \hat{e}(P_{pub}, H_0(ID_s)).$$

     Check if $< B, C, B^{r_i}, C^{r_i} >$ is a valid Diffie-Hellman tuple. If yes, then it is correct. Otherwise, it is an incorrect hash value.
  2. Verify the partial signature σ_i with associated signature verification algorithm.
     Note that

     $$\begin{aligned} h &= H_c(MPK, ID_s, m_i, B^{r_i}, C^{r_i}) \\ &= B^{h_1(m_i)}C^{r_i} \\ &= B^{h_1(m_i)}C^{r_0}B^{(h_1(m_0)-h_1(m_i))} \\ &= B^{h_1(m_0)}C^{r_0} \\ &= H_c(MPK, ID_s, m_0, B^{r_0}, C^{r_0}). \end{aligned}$$

3 Online/Offline Two-Party Ring Signature

In this section, we describe how to integrate the online/offline notion into an efficient ID-based two-party ring signature scheme. This scheme has potential applications in anti-phishing.

- ◇ Parameters Setup: It consists of signature system setup and chameleon hash system setup as following:

  1. *ID-based Signature System Setup*
     SG runs PKG to generate ring signature scheme's public global system params $\{\mathbb{G}_s$, $\mathbb{G}_{M_s}$, $\hat{e_s}$, q_s, κ, p_s, P_{pub_s}, h_{0^1}, $h_1^1\}$, where $h_0^1 : \{0,1\}^* \to G_s$, $h_1^1 : \{0,1\}^* \to Z_{q_s}$. Let $\hat{e}_s : \mathbb{G}_s \times \mathbb{G}_s \to \mathbb{G}_{M_s}$ be a bilinear mapping. Choose a random number $s_s \in Z_{q_s}^*$ as the master secret key and set $P_{pub_s} = s_sP_s$ as the master public key. Let SP_s be the public system parameters of the signature scheme: $\{\mathbb{G}_s$,$\mathbb{G}_{M_s}$, $\hat{e_s}$, q_s, κ, p_s, P_{pub_s}, h_0^1, $h_1^1\}$.

2. *ID-based Chameleon System Setup*
   Recall the definition in Section 2.3. Let SP_{hc} be the chameleon hash scheme's public system parameters: $\{\ \mathbb{G},\ \mathbb{G}_M,\ \hat{e},\ q,\ \ell,\ P,\ P_{pub},\ H_0,\ h_1,\ H_c\ \}$, where $MSK{=}s \in_R Z_q^*$, which is randomly chosen only kept by itself, $P_{pub} = sP \in \mathbb{G}$, $H_0{:}\{0,1\}^* \to \mathbb{G}$, $h_1{:}\{0,1\}^* \to Z_q^*$, $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_M$ be a pairing, and H_c is the ID-based chameleon hash function. Let r be a random number from Z_q^*, m be a message, ID_s be a specified signer's identity string, and the ID-based chameleon hash function H_c is defined as follows:

   $$H_c(MPK, ID_s, m, B^r, C^r) = B^{h_1(m)}C^r,$$

   where

   $$B = \hat{e}(P, H_0(ID_s)), C = \hat{e}(P_{pub}, H_0(ID_s))$$

   and $< B, C, B^r, C^r >$ is a valid Diffie-Hellman tuple.
   For simplicity, hereafter, we set $P_s = P$ and $P_{pub_s} = P_{pub}$. The public system parameters of an online/Offline scheme are $\mathcal{SP} = \{SP_S, SP_{hc}\}$.

◇ Key Extraction:

1. *Signing Private Key Extraction*
   Given an identity information ID_s, run *Extractor* to compute $Q_{ID_s} = h_0^1(ID_s) \in Z_{q_s}^*$ and return a secret key $SK_S{=}S_{ID_s} = (s_s{+}Q_{ID_s})^{-1}P \in \mathbb{G}$. Compute $E = \hat{e}(P, P)$ and store it.
2. *Chameleon Scheme Private Key Extraction*
   On input an identity string ID_s, compute public key $Q_{ID_s} = H_0(ID_s)$, run the Key Extract algorithm of the chameleon hash scheme to obtain trapdoor key $S_{ID_s} = s^{-1}Q_{ID_s}$, compute $D = \hat{e}(P, S_{ID_s})$.
3. Compute $B = \hat{e}(P, H_0(ID_s))$ and $C{=}\hat{e}(P_{pub}, H_0(ID_s))$ stored as a constant, choose at random number $r_0 \in Z_q^*$ and a message m_0, compute (B^{r_0}, C^{r_0}), and output a hash value

   $$h = Hash(MPK, ID_s, m_0, B^{r_0}, C^{r_0})$$
   $$= B^{h_1(m_0)}C^{r_0}.$$

4. Store tuple $(m_0, B, C, D, B^{r_0}, C^{r_0}, E, h)$.

◇ Signature Generation:

1. *Offline Phase*
   Retrieve the hash value h and a constant E, run the signing algorithm SS with the key SK_s associated with the identify string ID_s to sign h, output partial signature $\sigma_i \leftarrow SS(h)$ on the message m_0, and store σ_i, where i means each signing event.
   SS: Given a hash value h and a constant E, a set of identity string IDs (Sender's ID and Receiver's ID). Choose a random integer $r_s \in Z_{q_s}^*$ and do the following:
   – Compute $Y_{k+1} = h_1^1((h)||\hat{e}(P_s, r_sP_s)) \in Z_{q_s}^*$, where $k \in (0, 1)$, k denotes the sender, and assume that $k = 1$, and then get $Y_{k+1} = Y_0$ and $P_s = P_1$.
   – Generate the forward ring sequence: pick $T_0 \in \mathbb{G}_s$, compute

   $$Y_{i+1} = h_1^1((h)||\hat{e}(h_0^1(ID_i)P_i + P_{pub_i}, T_i)E^{Y_i}),$$

   where $i = 0$.
   – Compute $T_k = (r_sP_s - Y_kP_s)(s + Q_{ID_s})^{-1} = (r_s - Y_k)S_{ID_k}$, where $k = 1$.
   For simplicity, we assume $P_0 = P_1$. The message-signature tuple is $\sigma_i \leftarrow \{h, Y_0, T_0, T_1\}$.
2. *Online Phase*
   Given a new message m_i, retrieve the associated pair $(m_0, B^{r_0}, C^{r_0}, B, D, h, \sigma_i)$. Compute

   $$B^{r_i} = B^{r_0}D^{(h_1(m_0)-h_1(m_i))}$$
   $$C^{r_i} = C^{r_0}B^{(h_1(m_0)-h_1(m_i))}$$

   Send the message-signature pair $(m_i, B^{r_i}, C^{r_i}, \sigma_i)$ to a verifier.

◇ Signature Verification:

1. Compute the chameleon hash value as following:

   $$h = H_c(MPK, ID_s, m_i, B^{r_i}, C^{r_i})$$
   $$= B^{h_1(m_0)}C^{r_i},$$

   where

   $$B = \hat{e}(P, H_0(ID_s)),\ C = \hat{e}(P_{pub}, H_0(ID_s)).$$

   Check if $< B, C, B^{r_i}, C^{r_i} >$ is a valid Diffie-Hellman tuple. If yes, it is then correct. Otherwise, it is an incorrect hash value.
2. Verify the partial signature σ_i with associated signature verification algorithm.
   To verify the signature, retrieve system parameters $E = \hat{e}(P, P)$, and compute $Y_{i+1} = h_1^1((h)||\hat{e}(h_0^1(ID_i)P_i + P_{pub_i}, T_i)E^{Y_i})$ for $i \in \{0, 1\}$, Accept if $Y_2 = Y_0$.
   Thus

   $$Y_1 = h_1^1((h)||\hat{e}(h_0^1(ID_0)P_0 + P_{pub_0}, T_0)E^{Y_0}),$$
   $$Y_2 = h_1^1((h)||\hat{e}(h_0^1(ID_1)P_1 + P_{pub_s}, T_1)E^{Y_1}).$$

3. Correctness.

   $$T_k = T_1 = (r_sP_s - Y_kP_s)(s_s + Q_{ID_s})^{-1}$$
   $$= (r_s - Y_1)S_{ID_s}.$$
   $$Y_2 = h_1^1((h)||\hat{e}(h_0^1(ID_1)P_1 + P_{pub_1}, T_1)E^{Y_1})$$
   $$= h_1^1((h)||\hat{e}((h_0^1(ID_1) + s)P_s, (r_sP_s - Y_1P_s)$$
   $$(s + Q_{ID_1})^{-1})E^{Y_1})$$
   $$= h_1^1((h)||\hat{e}(P_s, r_sP_s - Y_1P_s + Y_1P_s)) = Y_0.$$

4 Application to Email Systems

Using our ID-based online/offline signature scheme, we can construct an efficient protocol for authentication in email systems. Assume that each email user has his own certificate generated by his own email server. The certificate is encrypted and can be stored in kinds of forms like encrypted file, or USB device, or smart card, and so on. The certificate contains the setup information like online/offline signature system parameters, user's identity string and his private key, which are generated though running the first two phases of online/offline signature scheme. To authenticate email's origin, the sender has to sign the outgoing message with his private key. According to our ID-based online/offline scheme, the signature generation is performed in two phases: online phase and offline phase. Before the sender clicks the send button, our scheme runs in the offline phase: signing the stored chameleon hash value. As soon as the real message is ready, our scheme performs the online phase: computing the hash collision based on the given message m. Since the online phase is efficient, using our ID-based online/offline signature scheme is much more applicable for email system authentication. The detail of this protocol is as follows.

Parameter Generation and Key Extraction

1. Alice chooses email address and registers as an legal email user.
2. Based on Alice identity string, the email server runs the parameter generation and key extraction algorithms of our ID-based online/offline signature scheme to generate Alice's certificate and deliveries it back to Alice with a secure channel. The certificate includes the system parameters of the current ID-based online/offline signature scheme and Alice's secret key.

The signing process is as described in the previous section.

5 Conclusion

In this paper, we consider the authentication of email system for mitigating email-based phishing attacks. To mitigate phishing by cryptography, the signature scheme should be ID-based and should have properties non-repudiation and non-transferability to enhance sender's privacy. The efficiency of the signature scheme is one of the most important considerations for email system. Online/offline signature scheme is a good option to provide fast signing process since the most computation is shifted to an offline phase and signing real message in an online phase is efficient. However, based on the "hash-sign-switch" paradigm (Shamir et al. 2001), online/offline signature schemes suffer from the key exposure issue. Our ID-based chameleon hash function eliminates the key exposure problem. With any secure ID-based deniable signature scheme and our ID-based chameleon hash function without key exposure, we proposed an ID-based online/offline signature scheme without key exposure which also can be applied to many areas. Compared with previous signature schemes for anti-phishing, our ID-based online/offline two-signer ring signature scheme is applicable to authentication in email systems

References

A. Menezes, T. Okamoto, S. Vanstone, (1993), 'Reducing elliptic curve logarithms to logarithms in a finite field', Vol. 39, IEEE Tran. on Info.Th., pp. 1639-1646.

Qiong Ren, Yi Mu, and Willy Susilo, (2007), 'Mitigating Phishing by a New ID-based Chameleon Hash without Key Exposure', *Proceedings of AusCERT Asia Pacific Information Technology Security Conference (AusCERT2007), Gold Coast, Australia*, University of Queensland, pp. 1–13.

Qiong Ren, Yi Mu, and Willy Susilo, (2007), 'SEFAP: An Email System for Anti-Phishing', in *The 6th IEEE International Conference on Computer and Information Science (ICIS 2007) , Melbourne, Australia*, IEEE Press, pp. 782–787.

A. Shamir and Y. Tauman, (2001), 'Improved Online/Offline signature schemes', Advances in Cryptology-Crypto 2001, LNCS 2139, Springer-Verlag, pp. 355–367.

Xiaofeng Chen, Fangguo Zhang, Willy Susilo, and Yi Mu, (2007), 'Efficient Generic Online/Offline Signatures Without Key Exposure', in *5th International Conference on Applied Cryptography and Network Security (ACNS'07), 5-8 June, 2007, Zhuhai, China*, Vol. 4521, LNCS, Springer Verlag, pp. 18–30.

Jonathan B. Postel. (1982), 'Simple Mail Transfer Protocol', RFC 821

Willy Susilo and Yi Mu, (2004), 'Deniable ring authentication revisited', in *Applied Cryptography and Network Security (ANCS) 04*, Vol. 3089, LNCS, pp. 149–163.

S. Even, O. Goldreich, and S. Micali, (1989), 'Online/Offline digital signatures', in *Advances in Cryptology-Crypto 1989*, Vol. 2442, LNCS, Springer-Verlag, pp. 263–277.

Anti-Phishing Working Group. Phishing activity trends report, (2006), Available at http://www.antiphishing.org.

H. Krawczyk and T. Rabin, (2000), 'Chameleon hashing and signatures,', in *Proceeding of Network and Distributed System Security 2000*, pp. 143–154.

Anti-Phishing Working Group. Digital Signatures to Fight Phishing Attacks. http://www.antiphishing.org/smim-dig-sig.htm.

Willy Susilo and Yi Mu, (2003), 'Non-interactive deniable ring signatures.', in *the 6th International Conference on Information Security and Cryptology (ICISC) 03*, pp. 397–412.

C. Dwork, M. Naor, and A. Sahai, (1998), ' Concurrent Zero-Knowledge', in *Proc. 30th ACM Symposium on the Theory of Computing*, pp. 409–418.

M. Naor, (2002), 'Deniable Ring Authentication', Vol. 2442, Advances in Cryptology - Crypto 2002, Lecture Notes in Computer Science, pp. 481–498.

X. Chen, F. Zhang, and K. Kim, (2004), 'Chameleon hashing without key exposure', 7th Information Security Conference (ISC 2004), Lecture Notes in Computer Science, Vol. 3225, Springer-Verlag, pp. 87–98.