Identity-Based Parallel Key-Insulated Signature: Framework and Construction

Jian Weng[1] **Shengli Liu**[1,2] **Kefei Chen**[1]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, P. R. China

[2] State Key Laboratory of Information Security, Graduate School of Chinese Academy of Sciences, Beijing, 100049, P. R. China

Xiangxue Li

School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai 200240, P. R. China
{jianweng, ssliu, kfchen, xxli}@sjtu.edu.cn

To minimize the damage caused by key-exposure in ID-based signatures, in ISPEC'06 Zhou et al *(2006) proposed an ID-based key-insulated signature (IBKIS) scheme. However, their scheme is not strong key-insulated, i.e, if a user's helper is compromised, the adversary can derive all of this user's secret keys for any time period. Moreover, in practice, to alleviate the damage in case of key-exposure, secret keys in IBKIS schemes have to be updated at very short intervals, which will increase the risk of helper key-exposure. It is important to note that even for an IBKIS scheme with strong key-insulated security, once a user's helper and one of his secret keys are both compromised, the adversary can also derive all of this user's secret keys for any time period. Is it possible to allow frequent key-updates without increasing the risk of helper key-exposure in IBKIS systems? In this paper, we extend Hanaoka* et al *(2006) parallel key-insulated mechanism to ID-based signature scenarios, and present an ID-based parallel key-insulated signature (IBPKIS) scheme. Compared with Zhou* et al *(2006) IBKIS scheme, our scheme enjoys three attractive features: (i) it is strong key-insulated; (ii) it can allow frequent key-updates without increasing the risk of helper key-exposure, and over all, enhances the security of the system; (iii) even if one of a user's helpers and some of his secret keys are both exposed, it is impossible for an adversary to derive all of this user's secret keys.*

Key words: Parallel Key-Insulated, Identity-Based Signature, Key-Exposure, Bilinear Pairings
ACM Classification: E.3 (Data Encryption)

1. INTRODUCTION

In 1984, Shamir (1984) introduced an innovative concept called identity-based (ID-based for short) cryptography, where users' identity information such as email or IP addresses instead of digital certificates can be used as public key for encryption or signature verification. As a result, ID-based cryptography significantly reduces the system complexity and the cost for establishing and managing the public key authentication framework known as Public Key Infrastructure (PKI). So far, a large number of papers have been published in this area, including many ID-based signature

Manuscript received: 17 November 2006
Communicating Editor: Rei Safavi-Naini

schemes (Cha and Cheon, 2003; Gentry and Silverberg, 2002; Hess, 2002; Zhang and Kim, 2002; Paterson, 2002; Yi, 2003).

Standard ID-based signatures rely on the assumption that secret keys are kept "perfectly secure". In practice, with more and more cryptographic primitives applied to insecure environments (e.g. mobile devices), it is easier for an adversary to obtain the secret key from a naive user than to break the computational assumption on which the system's security is based. The key-exposure problem is perhaps the most dangerous attack on a cryptosystem, since it typically means that security is entirely lost. In conventional public key infrastructures, certificate revocation list (CRL) can be used to revoke the compromised keys. However, straightforward implementation of CRL will not be the best solution to ID-based signatures. Remember that utilizing the CRL, public keys need to be renewed, while the public key in ID-based signatures represents an identity and is not desirable to be changed. For example, in an ID-based signature scheme where a user' identity card number acts as his public key, it is impractical to renew the identity card number.

Boneh and Franklin (2001) showed the first generalized method for key revocation in ID-based systems. In their mechanism, the Private Key Generator (PKG) generates each user's secret key whose corresponding public key is set to be the concatenation of user identity and time information, e.g. "recipient@xxx.xxx||2006.06.01-2006.06.02". In such a setting, the public key is renewed regularly by the PKG no matter whether it is revoked or not. However, as pointed out by Hanaoka *et al* (2005), there exist some disadvantages in this method. On the one hand, to alleviate the damage caused by key-exposure, the renewal interval has to be short (e.g. per day). This will require frequent interacting with the PKG, and increase the overhead of communication and computation cost. In those settings with a large number of users, this overhead will make the PKG insufferable. On the other hand, whenever the secret key is renewed, there is a need to frequently establish a secure channel between the PKG and the user.

To deal with the key-exposure problem, a natural try is to distribute the secret key across multiple servers to make key-exposure more difficult. This mechanism includes secret sharing (Shamir, 1979; Santis *et al*, 1994), threshold cryptosystems (Desmedt and Frankel, 1989) and proactive cryptosystems (Ostrovsky and Yung, 1991). However, such solutions tend to be quite costly, since they require many devices to participate in the cryptographic operations. While this may be acceptable in some scenarios, it does not seem appropriate for those settings where the risk of key-exposure is high but users need the ability to perform cryptographic computations on their own.

While secret sharing and threshold cryptography can be viewed as a separation of secret information in location, there is another approach, i.e., a separation of time. This mechanism includes forward security (Anderson, 1997; Bellare and Miner, 1999), intrusion-resilience (Itiks and Reyzin, 2002) and key-insulation (Dodis *et al*, 2002). The latter was introduced by Dodis *et al* (2002) in Eurocrypt'02. In this model, the lifetime of the secret key is divided into discrete time periods. The secret key is shared between the user and a physically secure device named helper. At the beginning of each time period, the user obtains from the helper an update key for the current time period. By combining this update key with the secret key for the previous time period, the user can derive the secret key for the current time period. A secret key for a given time period is used to sign a message during this time period without further access to the helper. Exposure of the secret key at a given time period will not enable an adversary to derive the secret keys for the remaining time periods. Thus the public key need not be revoked. This is a desirable property for dealing with the key-exposure problem in ID-based cryptosystems.

Following the pioneering work due to Dodis *et al* (2002), several elegant key-insulated encryption schemes including some ID-based key-insulated encryptions have been proposed

(Bellare and Palacio, 2002; Hanaoka *et al*, 2002; Hanaoka *et al*, 2005; Cheon *et al*, 2006; Hanaoka *et al*, 2006). Following Dodis *et al* (2003) first key-insulated signature schemes, efforts have also been devoted to the key-insulated signatures (Yum and Lee, 2003; González-Deleito *et al*, 2004; Le *et al*, 2004; Liu and Wong, 2005).

To minimize the damage caused by key-exposure in ID-based signatures, Zhou *et al* (2006) proposed an ID-based key-insulated signature (IBKIS) scheme. However, the full-fledged secret key in their scheme is just wholly stored in the helper. This means that their scheme cannot satisfy the strong key-insulated security i.e., if an adversary compromises a user's helper, he can obtain all of this user's secret keys, and then he can forge a signature on behalf of this user for any time period. It is worth pointing out that strong key-insulated security is an extremely important property for key-insulated cryptosystems, especially when the helper serves several users or the helper is untrustworthy.

Moreover, there exist some situations which the standard IBKIS scheme is hard to deal with. Consider the following example: Suppose a person works in the company's head office on the odd days, while on the even days he works in the branch. To alleviate the damage in case of key-exposure, he decides to update the secret key at very short intervals, e.g., once per day. Now, some problems happen: firstly, it is inconvenient but necessary for this person to remind himself to bring the helper between the head office and the branch back and forth; secondly, the short renewal interval means the high frequency of the helper's connection to insecure environments, and thus the risk of helper key-exposure is increased. It is important to note that the helper key-exposure is very dangerous for IBKIS systems, since even for an IBKIS scheme with strong key-insulated security, once a user's helper key and one of his secret keys are both exposed, the adversary can derive all of this user's secret keys. Is it possible to allow frequent key-updates without increasing the risk of helper key-exposure? Hanaoka *et al* (2006) introduced a very clever method named parallel key-insulation to deal with this problem for key-insulated public-key encryptions: based on Boneh-Franklin's ID-based encryption scheme (Boneh and Franklin, 2001), they proposed a parallel key-insulated public-key encryption scheme. Being different from the original key-insulated encryptions, their scheme introduced two distinct helpers which are alternately used to update the secret keys. The two helper keys are independent of each other, and they can successfully enhance the security of the system by allowing frequent key-updates without increasing the risk of helper key-exposure.

Weng *et al* (2006) extended the parallel key-insulated mechanism to ID-based encryption scenarios and proposed an ID-based parallel key-insulated encryption scheme. Based on Weng *et al*'s (2006) idea, in this paper, we will consider the parallel key-insulated mechanism in ID-based signature scenarios. We first formalize the definition and security model for ID-based parallel key-insulated signatures, and then propose an IBPKIS scheme. Compared with Zhou *et al* (2006) IBKIS scheme, our scheme enjoys the following features:

- Our scheme can allow frequent key-updates without increasing the risk of helper key-exposure. Therefore, the security of our scheme is enhanced.
- Our scheme is strong key-insulated, namely, even if an adversary compromises both of a user's two helper keys, he cannot derive any of this user's secret keys, and he cannot forge signatures on behalf of this user.
- Even if an adversary compromises one of a user's helper keys and some of this user's secret keys, it is still impossible for him to derive all of this user's secret keys. On the contrary, even for an IBKIS scheme with strong key-insulated security, once a user's helper key and one of this user's secret keys are both exposed, all of this user's secret keys are also exposed.

The rest of this paper is organized as follows. Section 2 gives an introduction to bilinear pairings and the computational Diffie-Hellman assumption. We formalize the definition and security notions for IBPKIS systems in Section 3. In Section 4, a concrete IBPKIS scheme is proposed. Section 5 gives the security proof for our proposed scheme. Section 6 concludes this paper.

2. PRELIMINARIES

Throughout this paper, let Z_q denote the set $\{0, 1, 2, \ldots, q-1\}$ and Z_q^* denote $Z_q \backslash \{0\}$. By $\in_R S$, it means choosing a random element from the set S with a uniform distribution. Now we proceed to give an introduction to the bilinear pairings, and we also briefly review the computational Diffie-Hellman assumption which will be used for our security analysis.

2.1 Bilinear Pairings

We briefly review the necessary about bilinear pairings. Let G_1 be a cyclic multiplicative group of prime order q, and G_2 be a cyclic multiplicative group of the same order q. A bilinear pairing is a map e: $\mathrm{G}_1 \times \mathrm{G}_1 \rightarrow G_2$ with the following properties:

1. Bilinearity: $\forall g_1, g_2 \in G_1$, $\forall a, b \in Z_q^*$, we have $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$;
2. Non-degeneracy: There exist $g_1, g_2 \in G_1$ such that $e(g_1, g_2) \neq 1$;
3. Computability: There exists an efficient algorithm to compute $e(g_1, g_2)$ for $\forall g_1, g_2 \in G_1$.

2.2 Computational Diffie-Hellman Assumption

Under such a group G_1, we can define the computational Diffie-Hellman (CDH) assumption which will be used for our security analysis.

Definition 1. The CDH problem in group G_1 is, given $(P, aP, bP) \in G_1^3$ for some unknown $a, b \in Z_q^*$, to compute $abP \in G_1$. For a polynomial-time adversary $\mathcal{A}$, we define his advantage against the CDH problem in group G_1 as

$$\mathrm{Adv}_{\mathcal{A}}^{CDH} = \Pr\left[P \in_R G_1, a, b \in_R Z_q^* : \mathcal{A}(P, aP, bP) = abP \right],$$

where the probability is taken over the random coins consumed by $\mathcal{A}$.

Definition 2. We say that the (t, ε)-CDH assumption holds in group G_1 if no t-time adversary $\mathcal{A}$ has advantage at least ε in solving the CDH problem in G_1.

3. FRAMEWORK OF ID-BASED PARALLEL KEY-INSULATED SIGNATURE

We now present the definition for IBPIKS, and thereafter specify what it means for IBPKIS scheme to be secure.

3.1 Definition of IBPKIS

Before formalizing the definition for IBPKIS systems, we first give an overview for IBPKIS systems. As original key-insulated signatures, the lifetime of IBPKIS systems is divided into discrete time periods. A user's identity acts as his public key and is fixed for all the lifetime, while his secret key is updated in every time period. Every user may have an arbitrary number of helpers (for an easy explanation, in the subsequent depiction, we assume that every user ID has two helpers which store $HK_{ID,1}$ and $HK_{ID,0}$ respectively). The two helper keys are alternately used to update this user's secret keys, namely, $HK_{ID,1}$ is used in odd time periods while $HK_{ID,0}$ is for even time periods. At time period t, user ID obtains an update key $UK_{ID,t}$ from the i-th helper (here $i = t \bmod 2$). Combining $UK_{ID,t}$ with the secret key $SK_{ID,t\text{-}1}$ for the previous time period, he can derive the secret key $SK_{ID,t}$ for the current time period. $SK_{ID,t}$ is used to sign a message during the corresponding time period without further access to the helpers. Note that unlike Boneh-Franklin's key revocation method (Boneh and Franklin, 2001), the key-update phase in IBPKIS systems adds no overhead on PKG since it needs not to interact with PKG. As to the aforementioned person who works in the head office and the branch, now he can put $HK_{ID,1}$ in the head office and $HK_{ID,0}$ in the branch, then he no longer needs to remind himself to bring the helper between the head office and branch back and forth. Moreover, due to the fact that $HK_{ID,1}$ and $HK_{ID,0}$ are alternately used, the risk of key-exposure for $HK_{ID,1}$ or $HK_{ID,0}$ will not be increased, even if this user's key-updates frequency is doubled.

Concretely, an IBPKIS scheme consists of the following six polynomial-time algorithms:

- **Setup**(k,N): the setup algorithm which, on input a security parameters k and (possibly) a total number of time periods N, outputs a public parameter *param* and a master key *msk*.
- **Extract**$(msk, param, ID)$: the key extraction algorithm which, on input *msk*, *param* and a user's identity $ID \in \{0, 1\}^*$, outputs an initial secret key $SK_{ID,0}$ and two helper keys $(HK_{ID,1},HK_{ID,0})$.
- **UpdH**$(t, ID,HK_{ID,i})$: the helper key-update algorithm performed by a user's helpers, taking as input a time period index t, a user's identity ID and the i-th helper key $HK_{ID,i}$ with $i = t \bmod 2$, returns an update key $UK_{ID,t}$.
- **UpdS**$(t, ID,UK_{ID,t}, SK_{ID,t\text{-}1})$: the secret key update algorithm performed by the user, taking as input a time period index t, a user's identity ID, a secret key $SK_{ID,t\text{-}1}$ and an update key $UK_{ID,t}$, returns a secret key $SK_{ID,t}$.
- **Sign**$(t, m, SK_{ID,t})$: the signing algorithm which, on input a time period index t, a message m and a secret key $SK_{ID,t}$, outputs a pair (t, σ) composed of the time period t and a signature σ.
- **Verify**$((t, \sigma), m, ID)$: the verification algorithm, on input a candidate signature (t, σ) on m and the user's identity ID, outputs 1 if (t, σ) is a valid signature, and 0 otherwise.

Consistency requires that $\forall t \in \{1,\text{L} ,\text{N}\}$, $\forall m \in \mathcal{M}$, $\forall ID \in \{0,1\}^*$, Verify$((t, \sigma), m, ID)$=1 holds, where (t, σ) = Sign$(t, m, SK_{ID,t})$ and $\mathcal{M}$ denotes the message space.

3.2 Security Model for IBPKIS

Based on Dodis *et al* (2003) security notions for KIS systems, we formalize the security notions for IBPKIS systems in this subsection. Since we consider them in the ID-based scenarios and two helpers are available for every user, our notions are somewhat different from those of Dodis *et al*.

Key-insulated security. Generally, the key-insulated security for KIS systems says that, if the helper key is not compromised, exposure of any of the secret keys does not enable an adversary to forge a valid signature for the non-exposed time periods. Here, the adversary we consider in IBPKIS systems is much more powerful: the adversary is allowed to compromise any of the non-challenged identities' secret keys and helper keys; for the challenged identity, the adversary is even allowed to compromise one of the helper keys and any of the secret keys; as usual, we also allow the adversary to issue signing queries. For such a powerful adversary, the key-insulated security for IBPKIS systems ensures that:

(i) If none of the challenged identity's helpers is compromised, exposure of any of the challenged identity's secret keys does not enable the adversary to forge a valid signature on behalf of this user for those non-exposed time periods.

(ii) Even if one of the challenged identity's helpers and some of this user's secret keys are both exposed, the adversary is still unable to forge signatures on behalf of this user for those time periods where the secret keys cannot be trivially derived from the exposed keys. Note that even for KIS schemes with strong key-insulated security, once the helper keys and one of the secret keys are both exposed, the adversary can forge a signature for any time periods.

Concretely, we define the key-insulated security for an IBPKIS scheme $\prod$ by the following game played by a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

Setup. Challenger $\mathcal{C}$ runs algorithm Setup and obtains the public parameters *param* and the master key *msk*. Adversary $\mathcal{A}$ is given *param* while *msk* is kept by challenger $\mathcal{C}$.

Queries. Adversary $\mathcal{A}$ issues a series of queries in an adaptive fashion. The following queries are allowed.

- *Extraction queries*. Upon receiving an extraction query $<ID>$, $\mathcal{C}$ runs algorithm Extract and obtains an initial secret key $SK_{ID,0}$ and two helper keys $(HK_{ID,1},HK_{ID,0})$. $\mathcal{C}$ then sends $SK_{ID,0}$ and $(HK_{ID,1},HK_{ID,0})$ to $\mathcal{A}$.
- *Helper key queries*. Upon receiving a helper key query $<ID, i>$ with $i \in \{0, 1\}$, $\mathcal{C}$ runs algorithm Extract to generate $HK_{ID,i}$ and returns it to $\mathcal{A}$.
- *Secret key queries*. Upon receiving a secret key query $<ID, t>$, $\mathcal{C}$ runs algorithm UpdS to obtain $SK_{ID,t}$, which is forwarded to $\mathcal{A}$.
- *Signing queries*. Upon receiving a signing query $<ID, t, m>$, $\mathcal{C}$ first runs algorithm UpdS to obtain $SK_{ID,t}$, and then runs algorithm Sign$(t, m, SK_{ID,t})$ to obtain a signature (t, σ), which is returned to $\mathcal{A}$.

Forge. Eventually, $\mathcal{A}$ outputs a message m^*, an identity ID^* and a signature (t^*, σ^*). We say that adversary $\mathcal{A}$ wins in this game if the following holds true: (1) Verify$((t^*, \sigma^*), m^*, ID^*) = 1$; (2) $\mathcal{A}$ has never issued a signing query on $<ID^*, t^*,m^*>$; (3) Adversary $\mathcal{A}$ never issue an extraction query $<ID^*>$; (4) Adversary $\mathcal{A}$ never make a secret key query $<ID^*, t^*>$; (5) Adversary $\mathcal{A}$ cannot issue both secret key query $<ID^*, t^*$-$1>$ and helper key query $<ID^*, t^* \bmod 2>$; (6) Adversary $\mathcal{A}$ cannot issue both secret key query $<ID^*,t^*+1>$ and helper key query $<ID^*, (t^*+1) \bmod 2>$; (7) Adversary $\mathcal{A}$ cannot issue helper key queries on both $<ID^*, 1>$ and $<ID^*, 0>$.

Remark 1. Conditions (3)-(6) prevent the adversary from deriving SK_{ID^*,t^*} trivially. For example, if $\mathcal{A}$ issues both secret key query $<ID^*, \mathrm{t}^*$-$1>$ and helper key query $<ID^*, t^* \bmod 2>$, he gets $SK_{ID^*,t^*\text{-}1}$ and $HK_{ID^*,t^* \bmod 2}$, then he can run algorithm UpdH and UpdS to derive SK_{ID^*,t^*}. Similarly to the explanation in Hanaoka *et al* (2006), we know that if an adversary issues both secret key query $<ID^*, t^* + 1>$ and helper key query $<ID^*, (t^* + 1) \bmod 2>$, he can derive SK_{ID^*,t^*} trivially.

Remark 2. To ensure the strong existential unforgeability, we can modify condition (2) to be (2)′ (t^*,σ^*) was never returned by $\mathcal{C}$ on input $\mathcal{A}$'s signing query $<ID^*, t^*,m^*>$. This means that the adversary is allowed to issue a signing query on $<ID^*, t^*,m^*>$, if only (t^*, σ^*) is not the corresponding output. Note that our proposed IBPKIS scheme can ensure this strong existential unforgeability.

We refer to the above game as a game of *existential unforgeable against chosen identity and adaptive chosen message attack under key-exposure* (UF-ID&KE-CMA), and we call such an adversary $\mathcal{A}$ as an UF-ID&KE-CMA adversary. We define $\mathcal{A}$'s advantage as

$$\mathrm{Adv}_{\mathcal{A},\Pi}^{\text{UF-ID\&KE-CMA}} = \Pr[\mathcal{A} \text{ wins the UF-ID\&KE-CMA game}],$$

where the probability is taken over the random bits consumed by $\mathcal{A}$.

Definition 3. We say that an IBPKIS scheme $\prod$ is (t, ε)-UF-ID&KE-CMA secure, if for any t-time UF-ID&KE-CMA adversary $\mathcal{A}$, we have $\mathrm{Adv}_{\mathcal{A},\Pi}^{\text{UF-ID\&KE-CMA}} < \varepsilon$.

Strong key-insulated security. The strong key-insulated security for KIS systems says that, if none of the secret keys is compromised, exposure of the helper key does not enable an adversary to forge a valid signature for any time period. This is an extremely important property for key-insulated systems if the helper serves several users or the helper is untrustworthy. Note that Zhou *et al*'s IBKIS scheme is not strong key-insulated. To model this security notion for IBPKIS systems, we allow the adversary to compromise all the helper keys for any identity, even including the challenged identity. As the strong key-insulated security for KIS systems, the adversary is disallowed to compromise any of the challenged identity's secret key. Note that we allow him to issue *secret key queries* for any non-challenged identity. Since these queries are implied by the *extraction queries*, we do not explicitly provide *secret key queries* for the adversary. Concretely, we define the strong key-insulated security for an IBPKIS scheme $\prod$ by the following game between a challenger C and an adversary $\mathcal{A}$:

Setup. The same as UF-ID&KE-CMA game.

Queries. Adversary $\mathcal{A}$ issues a series of queries in the same way as UF-ID&KE-CMA game except that the secret key queries are not provided for him.

Forge. Eventually, $\mathcal{A}$ outputs a message m^*, an identity ID^* and a signature (t^*,σ^*). We say that adversary $\mathcal{A}$ wins in this game if the following conditions are satisfied: (1) $\mathsf{Verify}((t^*,\sigma^*), m^*, ID^*) =1$; (2) $\mathcal{A}$ has never issued a signing query on $<ID^*, t^*,m^*>$; (3) $\mathcal{A}$ never issue an extraction query $<ID^*>$.

We refer to the above game as a *strongly*-UF-ID&KE-CMA game, and we call such an adversary $\mathcal{A}$ as a strongly-UF-ID&KE-CMA adversary. We define $\mathcal{A}$'s advantage as

$$\mathrm{Adv}_{\mathcal{A},\Pi}^{\text{strongly-UF-ID\&KE-CMA}} = \Pr[\mathcal{A} \text{ wins the strongly-UF-ID\&KE-CMA game}],$$

where the probability is taken over the random bits consumed by $\mathcal{A}$.

Definition 4. We say that an IBPKIS scheme $\prod$ is (t, ε)-strongly-UF-ID&KE-CMA secure, if for any t-time strongly-UF-ID&KE-CMA adversary $\mathcal{A}$, we have $\mathrm{Adv}_{\mathcal{A},\Pi}^{\text{strongly-UF-ID\&KE-CMA}} < \varepsilon$.

Secure key-updates. Finally, as in Dodis *et al* 2003, we address an adversary who compromises the user's storage when secret keys are being updated. Note that when a secret key $SK_{ID,t-1}$ is being updated to $SK_{ID,t}$, such an adversary can obtain the secret keys $SK_{ID,t-1}$ and $SK_{ID,t}$ as well as the update key $UK_{ID,t}$. To model this security notion, we define another game which is identical to the UF-ID&KE-CMA with the exception that the update key queries as below are also provided for the adversary:

- *Update key queries*. Upon receiving an update key query $<ID, t>$, C first runs algorithm $\mathsf{UpdH}(t, ID,HK_{ID,i})$ with $i = t \bmod 2$ to obtain the resulting update key $UK_{ID,t}$, which is passed to $\mathcal{A}$.

We refer to the above game as a game of *secure key-updates* (SKU), and we call such an adversary $\mathcal{A}$ as a SKU adversary. We define $\mathcal{A}$'s advantage as

$$\mathrm{Adv}_{\mathcal{A},\Pi}^{\text{SKU}} = \Pr[\text{A wins the SKU game}],$$

where the probability is taken over the random bits consumed by $\mathcal{A}$.

Definition 5. We say that an IBPKIS scheme $\prod$ is (t, ε) secure key updates, if for any t-time SKU adversary $\mathcal{A}$, we have $\mathrm{Adv}_{\mathcal{A},\Pi}^{\text{SKU}} < \varepsilon$.

Note that the security notions described in this subsection can be easily adapted to the random oracle model, where the adversary has access to a random hash function.

4. CONSTRUCTION OF AN IBPKIS SCHEME

4.1 On the straightforward Integration of two IBKIS Schemes

Intuitively, straightforward integration of two independent IBKIS schemes seems to be a solution. However, such integration is not a correct answer, since both of a user's helper keys are *simultaneously* used in such a combined system; therefore, with the key-updates frequency increasing, the risk of helper key-exposure is also increased accordingly.

4.2 Our Proposed Scheme

In this subsection, we present our IBPKIS scheme for the case of two helpers. Note that it can be extended to allow arbitrary number of helpers for any user trivially.

To describe our scheme, some global parameters are required to be defined in advance. Let G_1 and G_2 be two groups with prime order q of size k, P be a random generator of G_1, and e be a bilinear map such that e: $G_1 \times G_1 \rightarrow G_2$. Let H_1,H_2 and H_3 be cryptography hash functions such that H_1: $\{0, 1\}^* \rightarrow G_1$, H_2: $\{0, 1\}^* \rightarrow G_1$ and H_3: $\{0, 1\}^* \rightarrow G_1$. The proposed IBPKIS scheme consists of the following six algorithms:

- **Setup:** given a security parameter k, the PKG picks $s \in_R Z_q^*$, and sets $P_{pub} = sP$. The master key is $msk = s$ and the public parameter is $param = (G_1,G_2, e, q, P, P_{pub}, H_1, H_2, H_3)$.
- **Extract:** for an identity $ID \in \{0, 1\}^*$, PKG first picks two helper keys $HK_{ID,1}$, $HK_{ID,1} \in_R Z_q^*$. Next, it computes $T_{ID,1} = HK_{ID,1} \cdot$P , $T_{ID,0} = HK_{ID,0} \cdot P$ and
$$S_{ID,0} = sH_1(ID) + HK_{ID,1} \cdot H_2(ID,T_{ID,1},-1) + HK_{ID,0} \cdot H_2(ID,T_{ID,0},0). \tag{1}$$
The initial secret key for ID is $SK_{ID,0} = (S_{ID,0}, T_{ID,1}, T_{ID,0})$.
- **UpdH:** given an identity $ID \in \{0, 1\}^*$ and a time period index t, the i-th (here $i = t \bmod 2$) helper computes $T_{ID,i} = HK_{ID,i} \cdot P$ and returns the update key as
$$UK_{ID,t} = HK_{ID,i}\left(H_2(ID,T_{ID,i},\mathrm{t}) - H_2(ID,T_{ID,i},t-2)\right). \tag{2}$$
- **UpdS:** given an identity $ID \in \{0, 1\}^*$, a time period index t, an update key $UK_{ID,t}$ and the secret key $SK_{ID,t\text{-}1}$=($S_{ID,t\text{-}1}$, $T_{ID,1}$, $T_{ID,0}$) for the previous time period, user ID computes $S_{ID,t} = S_{ID,t\text{-}1} + UK_{ID,t}$. The secret key for time period t is $SK_{ID,t} = (S_{ID,t}, T_{ID,1}, T_{ID,0})$.

  Note that let $i = t \bmod 2$ and $i' = (t - 1) \bmod 2$, then the following equality always holds
$$S_{ID,t} = sH_1(ID) + HK_{ID,i'} \cdot H_2(ID,T_{ID,i'},t-1) + HK_{ID,i} \cdot H_2(ID,T_{ID,i},t). \tag{3}$$
- **Sign:** in time period t, given a message m and the secret key $SK_{ID,t}$= ($S_{ID,t}$, $T_{ID,1}$, $T_{ID,0}$), user ID chooses $u \in_R Z_q^*$, and compute $U = uP$, $W_m = H_3(t, ID, m,U)$, $V = S_{ID,t} + uW_m$. The signature is $\sigma = (t, (U, V, T_{ID,1}, T_{ID,0}))$.
- **Verify:** given a purported signature $(t, (U, V, T_{ID,1}, T_{ID,0}))$ on message m and identity ID, one can verify this signature as follows: Let $i = t \bmod 2$ and $i' = (t - 1) \bmod 2$. Compute $W_m = H_3(t, ID, m,U)$, return 1 if the following equality holds and 0 otherwise:
$$e(P,V) = e(P_{pub}, H_1(ID))e(T_{ID,i'},H_2(ID,T_{ID,i'},t-1))e(T_{ID,i},H_2(ID,T_{ID,i},t))e(U,W_m). \tag{4}$$

4.3 Correctness

Let $i = t \bmod 2$ and $i' = (t - 1) \bmod 2$, then we have

$$
\begin{aligned}
e(P,V) &= e(P,S_{ID,t} + uW_m) \\
&= e(P,sH_1(ID) + HK_{ID,i'} \cdot H_2(ID,T_{ID,i'},t-1) + HK_{ID,i} \cdot H_2(ID,T_{ID,i},t) + uW_m) \\
&= e(P_{pub},H_1(ID))e(T_{ID,i'},H_2(ID,T_{ID,i'},t-1))e(T_{ID,i},H_2(ID,T_{ID,i},t))e(U,W_m)
\end{aligned}
$$

5. SECURITY PROOF

To support our scheme, we proceed to give the security proof for our proposed scheme in the random oracle model.

Theorem 1. The proposed scheme is UF-ID&KE-CMA secure in the random oracle model. Concretely, given a (T, ε)-UF-ID&KE-CMA adversary $\mathcal{A}$ against our proposed scheme by asking at most q_{h_i} hash function queries to H_i (i = 1, 2, 3), q_e extraction queries, q_h helper key queries, q_k secret key queries and q_s signing queries, there exists a (T', ε')-adversary $\mathcal{B}$ that breaks the CDH assumption in group G_1 with

$$T' \leq T + \left(q_{h_1} + q_{h_2} + q_{h_3} + 5q_e + 2q_h + 5q_k + 7q_s + 4\right)t_{sm}, \qquad \varepsilon' \geq \frac{\varepsilon}{e\left(q_e + q_k + q_s + 1\right)},$$

where e denotes the base of the natural logarithm, and t_{sm} denotes the running time of computing a scalar multiplication in G_1.

Proof. We will show how to construct a (T', ε')-adversary $\mathcal{B}$ against the CDH assumption in group G_1. Suppose $\mathcal{B}$ is given a CDH instance $(P, X = aP, Y = bP) \in G_1^3$. $\mathcal{B}$'s goal is to derive abP with the help of adversary $\mathcal{A}$. $\mathcal{B}$ plays the role of $\mathcal{A}$'s challenger and works by interacting with $\mathcal{A}$ in an UF-ID&KE-CMA game as follows:

Setup. $\mathcal{B}$ sets $P_{pub} = X$ and gives the public parameter $param = (G_1, G_2, e, q, P, P_{pub}, H_1, H_2, H_3)$ to $\mathcal{A}$. Note that the master key is implicitly assigned to be $msk = a$, which is unknown to $\mathcal{B}$.

Queries. $\mathcal{B}$ answers a series of queries for $\mathcal{A}$ as follows:

- H_1 *queries*: $\mathcal{B}$ maintains a hash list H_1^{list} which is initially empty. When $\mathcal{A}$ issues a H_1 query on ID, as in Coron's proof technique (Coron, 2000), $\mathcal{B}$ responds in the following way: If H_1^{list} contains a tuple for this input, then the previously defined value is returned. Otherwise, $\mathcal{B}$ chooses $d \in Z_q^*$ and flips a random biased coin $c \in \{0, 1\}$ that yields 0 with probability δ and 1 with probability $1 - \delta$. If $c = 0$ then the hash value $H_1(ID)$ is defined as $Q = dP$, else $Q = dY$. Finally, Q is returned to $\mathcal{A}$ and (ID, c, d, Q) is added on H_1^{list} .
- H_2 *queries*: $\mathcal{B}$ maintains a hash list H_2^{list} which is initially empty. When a tuple $(ID, T_{ID,i}, t)$ with $i \in \{0,1\}$ is queried, $\mathcal{B}$ returns the previously defined value to $\mathcal{A}$ if H_2^{list} has contained a tuple for this input. Otherwise, $\mathcal{B}$ chooses $r \in_R Z_q^*$, computes $R = rP$, stores tuple $(ID, T_{ID,i}, t, r, R)$ in H_2^{list} and returns R to $\mathcal{A}$.

- H_3 *queries*: $\mathcal{B}$ maintains a hash list H_3^{list} which is initially empty. When a tuple (t, ID, m, U) is queried, $\mathcal{B}$ responds with the previously defined value if H_3^{list} has contained a tuple for this input. Otherwise, $\mathcal{B}$ chooses $w \in_R Z_q^*$, computes $Wm = wP$, adds tuple (t, ID, m, U, w, W_m) on H_3^{list} and returns W_m to $\mathcal{A}$.
- *Extraction queries*: $\mathcal{B}$ maintains a list D^{list} which is initially empty. When $\mathcal{A}$ issues an extraction query $<ID>$, $\mathcal{B}$ acts as below:
  1. Recover the tuple (ID, c, d, Q) from H_1^{list} (Wlog, we assume that ID was previously submitted to oracle H_1). If $c = 1$ then $\mathcal{B}$ outputs "failure" and aborts (event **E1**). Otherwise, it means that $H_1(ID)$ was previously assigned to be dP.
  2. If D^{list} has not contained a tuple for the input ID, $\mathcal{B}$ chooses $HK_{ID,1}, HK_{ID,2} \in_R Z_q^*$, computes $T_{ID,1} = HK_{ID,1} \cdot P$, $T_{ID,0} = HK_{ID,0} \cdot P$, adds tuple $(ID, HK_{ID,1}, HK_{ID,0}, T_{ID,1}, T_{ID,0})$ on D^{list}.
  3. Compute $S_{ID,0} = dX + HK_{ID,1} \cdot H_2(ID, T_{ID,1}, -1) + HK_{ID,0} \cdot H_2(ID, T_{ID,0}, 0)$. Return $SK_{ID,0} = (S_{ID,0}, T_{ID,1}, T_{ID,0})$ and $(HK_{ID,1}, HK_{ID,0})$ to $\mathcal{A}$.

     Note that $S_{ID,0}$ has the correct form as Eq. (1) and $SK_{ID,0}$ is a valid initial secret key for $\mathcal{A}$.
- *Helper key queries*: When a helper key query $<ID, i>$ with $i \in \{0, 1\}$ is coming, $\mathcal{B}$ returns the predefined value to $\mathcal{A}$ if D^{list} has contained a tuple for identity ID. Otherwise, $\mathcal{B}$ chooses $HK_{ID,1}, HK_{ID,0} \in_R Z_q^*$, computes $T_{ID,1} = HK_{ID,1} \cdot P, T_{ID,0} = HK_{ID,0} \cdot P$, adds tuple $(ID, HK_{ID,1}, HK_{ID,0}, T_{ID,1}, T_{ID,0})$ on D^{list}, and returns $HK_{ID,i}$ to $\mathcal{A}$.
- *Secret key queries*: Upon receiving a secret key query $<ID, t>$, $\mathcal{B}$ works as follows:
  1. Recover tuple (ID, c, d, Q) from H_1^{list} (Wlog, we assume that ID was previously submitted to oracle H_1). If $c = 1$ then $\mathcal{B}$ outputs "failure" and aborts (event **E2**). Otherwise, it means that $H_1(ID)$ was previously defined to be dP.
  2. If D^{list} has not contained a tuple for identity ID, $\mathcal{B}$ chooses $HK_{ID,1}, HK_{ID,0} \in_R Z_q^*$, computes $T_{ID,1} = HK_{ID,1} \cdot P$, $T_{ID,0} = HK_{ID,0} \cdot P$, adds tuple $(ID, HK_{ID,1}, HK_{ID,0}, T_{ID,1}, T_{ID,0})$ on D^{list}.
  3. Let $i = t \bmod 2$ and $i' = (t\text{-}1) \bmod 2$. Set

     $S_{ID,t} = dX + HK_{ID,i'} \cdot H_2(ID, T_{ID,i'}, t-1) + HK_{ID,i} \cdot H_2(ID, T_{ID,i}, t-1)$
  4. Return $SK_{ID,t} = (S_{ID,t}, T_{ID,1}, T_{ID,0})$ to $\mathcal{A}$.

     Note that $SK_{ID,t}$ has the correct form as Eq. (3) and $SK_{ID,t}$ is indeed a valid secret key for $\mathcal{A}$.
- *Signing queries*: When a signing query $<t, ID, m>$ is coming, $\mathcal{B}$ responds as below:
  1. Recover tuple (ID, c, d, Q) from H_1^{list} (Wlog, we assume that ID was previously submitted to oracle H_1). If $c = 1$ then $\mathcal{B}$ outputs "failure" and aborts (event **E3**). Otherwise, it means that $H_1(ID)$ was previously defined to be dP.

2. If D^{list} has not contained a tuple for identity ID, $\mathcal{B}$ chooses $HK_{ID,1}, HK_{ID,0} \in_R Z_q^*$, computes $T_{ID,1} = HK_{ID,1} \cdot P$, $T_{ID,0} = HK_{ID,0} \cdot P$, adds tuple $(ID, HK_{ID,1}, HK_{ID,0}, T_{ID,1}, T_{ID,0})$ on D^{list}.
3. Let $i = t \bmod 2$ and $i' = (t\text{-}1) \bmod 2$. Recover tuples $(ID, T_{ID,i}, t, r, R)$ and $(ID, T_{ID,i'}, t\text{–}1, r', R')$ from H_2^{list} (Wlog, we assume that $(ID, T_{ID,i}, t)$ and $(ID, T_{ID,i'}, t\text{–}1)$ were previously submitted to oracle H_2).
4. Choose $V \in_R G_1$, $u \in_R Z_q^*$, compute $U = uP$. If H_3^{list} has not contained a tuple for the input (t, ID, m, U), issue a H_3 query and assign the hash value $H_3(t, ID, m, U)$ to be $u^{-1}(V - dP_{pub} - rT_{ID,i} - r'T_{ID,i'})$. Return $(t, (U, V, T_{ID,1}, T_{ID,0}))$ as the signature on m. Note that it can be verified that this is indeed a valid signature.

Forge. Eventually, $\mathcal{A}$ outputs a signature $\sigma^* = (t^*, (U^*, V^*, T^*_{ID^*,1}, T^*_{ID^*,0}))$ on message m^* and identity ID^* with the constraints described in the UF-ID&KE-CMA game. Let $i^* = t^* \bmod 2$ and $i'^* = (t^*\text{–}1) \bmod 2$. $\mathcal{B}$ recovers tuples $(ID^*, T_{ID^*,i^*}, t^*, r^*, R^*)$ and $(ID^*, T_{ID^*,i'^*}, t^*, r'^*, R'^*)$ from H_2^{list}, and tuple (ID^*, c^*, d^*, Q^*) from H_1^{list}. If $c^* = 0$ then $\mathcal{B}$ outputs "**failure**" and aborts (event **E4**). Otherwise, $\mathcal{B}$ searches H_3^{list} for tuple $(t^*, ID^*, m^*, U^*, w^*, W^*_m)$. If $\mathcal{A}$ succeeds in this game, then we have

$$e(P, V^*) = e(P_{pub}, d^*Y)e(T^*_{ID^*,i'^*}, r'^*P)e(T_{ID^*,i^*}, r^*P)e(U^*, w^*P),$$

which implies that

$$e(P_{pub}, d^*Y)e(P, abP)^{d^*} = e(P, V^* - r'^*T^*_{ID^*,i'^*} - r^*T^*_{ID^*,i^*} - w^*U^*).$$

Thus $\mathcal{B}$ can derive abP as $abP = (d^*)^{-1}\left(V^* - r'^*T^*_{ID^*,i'^*} - r^*T^*_{ID^*,i^*} - w^*U^*\right)$ and solve the CDH instance successfully.

From the above description of $\mathcal{B}$, we know that $\mathcal{B}$'s running time T' is bounded by

$$T' \leq T + (q_{h_1} + q_{h_2} + q_{h_3} + 5q_e + 2q_h + 5q_k + 7q_s + 4)t_{sm}.$$

We now proceed to analyze the advantage of $\mathcal{B}$. Note that the responses to $\mathcal{A}$'s H_1,H_2 and H_3 queries are indistinguishable from the real environment, since each response is uniformly random and independently distributed in G_1. The responses of helper key queries provided for $\mathcal{A}$ are also valid. The responses for $\mathcal{A}$'s extraction queries (secret key queries, signing queries, resp.) are valid unless event **E1**(**E2**, **E3**, resp.) happens. So if none of events **E1**, **E2** and **E3** happens, the simulation provided for $\mathcal{A}$ is indistinguishable from the real environment. Furthermore, if $\mathcal{A}$ succeeds in forging a valid signature and events **E4** does not happen, then $\mathcal{B}$ can solve the CDH instance successfully. Now we try to bound the probability for events **E1**, **E2**, **E3** and **E4**.

From the description of the simulation, we have $\Pr[\neg \mathrm{E1} \wedge \neg \mathrm{E2} \wedge \neg \mathrm{E3} \wedge \neg \mathrm{E4}] = \delta^{q_e + q_k + q_s}(1 - \delta)$, which is maximized at $\delta_{opt} = \dfrac{q_e + q_k + q_s}{q_e + q_k + q_s + 1}$. Using δ_{opt}, the probability $\Pr[\neg \mathrm{E1} \wedge \neg \mathrm{E2} \wedge \neg \mathrm{E3} \wedge \neg \mathrm{E4}]$ is at least $\dfrac{1}{e(1 + q_e + q_k + q_s)}$. Therefore, $\mathcal{B}$'s advantage ε' satisfies

$$\varepsilon' \geq \frac{\varepsilon}{e(1 + q_e + q_k + q_s)}.$$

This concludes the proof.

Theorem 2. The proposed scheme is strongly UF-ID&KE-CMA secure in the random oracle model. Concretely, given a (T, ε)-strongly-UF-ID&KE-CMA adversary against our proposed scheme by asking at most q_{h_i} hash function queries to H_i (i = 1, 2, 3), q_e extraction queries, q_h helper key queries and q_s signing queries, there exists a (T', ε')-adversary $\mathcal{B}$ that breaks the CDH assumption in group G_1 with

$$T' \leq T + (q_{h_1} + q_{h_2} + q_{h_3} + 5q_e + 2q_h + 7q_s + 4)t_{sm}, \quad \varepsilon' \geq \frac{\varepsilon}{e(q_e + q_s + 1)},$$

where e and t_{sm} have the same meaning as Theorem 1.

The proof is similar to that of Theorem 3 except that we need not provide the secret key queries for adversary $\mathcal{A}$. Here we omit the proof.

Theorem 3. The proposed scheme has secure key-updates in the random oracle model. Concretely, given a (T, ε)-SKU adversary against our proposed scheme by asking at most q_{h_i} hash function queries to H_i (i = 1, 2, 3), q_e extraction queries, q_h helper key queries, q_u key-update queries, q_k secret key queries and q_s signing queries, there exists a (T', ε')-adversary $\mathcal{B}$ that breaks the CDH assumption in group G_1 with

$$T' \leq T + (q_{h_1} + q_{h_2} + q_{h_3} + 5q_e + 2q_h + 3q_u + 5q_k + 7q_s + 4)t_{sm}, \quad \varepsilon' \geq \frac{\varepsilon}{e(q_e + q_k + q_s + 1)},$$

where e and t_{sm} have the same meaning as Theorem 1.

Proof. The proof is the same as that of Theorem 1, except that $\mathcal{B}$ needs to answer the update key queries for $\mathcal{A}$ as below:

- *Update key queries*: When $\mathcal{A}$ issues an update key queries <ID, t>, $\mathcal{B}$ works as below:
  1. If D^{list} has not contained a tuple for the input ID, $\mathcal{B}$ chooses $HK_{ID,1}$,$HK_{ID,0} \in_R Z_q^*$, computes $T_{ID,1} = HK_{ID,1} \cdot P$, $T_{ID,0} = HK_{ID,0} \cdot P$, adds tuple ($ID$,$HK_{ID,1}$,$HK_{ID,0}$, $T_{ID,1}$, $T_{ID,0}$) on D^{list}.
  2. Let $i = t$ mod 2. Recover tuples (ID, $T_{ID,i}$, t, r,R) and (ID, $T_{ID,i}$, $t - 2$, r', R') from H_2^{list} (Wlog, we assume that (ID, $T_{ID,i}$, t) and (ID, $T_{ID,i}$, t-2) were previously submitted to oracle H_2).
  3. Return $HK_{ID,i} \cdot (R - R')$ to $\mathcal{A}$.

  Similar to the analysis in Theorem 1, the time complexity T' of $\mathcal{B}$ is bounded by

$$T' \leq T + (q_{h_1} + q_{h_2} + q_{h_3} + 5q_e + 2q_h + 3q_u + 5q_k + 7q_s + 4)t_{sm},$$

  and the advantage ε' of $\mathcal{B}$ satisfies

$$\varepsilon' \geq \frac{\varepsilon}{e(q_e + q_k + q_s + 1)}.$$

6. CONCLUSION

Classical ID-based signatures rely on the assumption that secret keys are kept perfectly secure. In practice, however, key-exposure seems inevitable. No matter how strong these ID-based signatures are, once the secret keys are exposed, their security is entirely lost. Thus it is worthwhile to deal with the key-exposure problem in ID-based signatures.

In this paper, we have extended Hanaoka *et al* (2006) parallel key-insulated mechanism to ID-based signatures and minimized the damage caused by key-exposure in ID-based signatures. We formalized the definition and security notions for IBPKIS systems, and at the same time proposed an IBPKIS scheme. The proposed scheme can allow frequent key-updates without increasing the risk of helper key-exposure, and eventually enhance the security of the system. This is an attractive advantage which the standard IBKIS schemes do not possess.

ACKNOWLEDGEMENTS

The author would like to thank the anonymous reviewers for their valuable comments and suggestions. This paper was partially supported by NSFC under Grants 90704003, 60573030 and 60673077.

REFERENCES

ANDERSON, R. (1997): Two remarks on public-key cryptology. Invited lecture, CCCS'97. Http://www.cl.cam.ac.uk/users/rja14/. Accessed 10-Sep-2006.

BONEH, D. and FRANKLIN, M. (2001): Identity based encryption from the Weil pairing. In *Advances in Cryptology-Crypto'01*, Kilian J. (Ed) LNCS 2139: 213-229, Springer-Verlag.

BELLARE, M. and MINEER, S. (1999): A forward-secure digital signature scheme. In *Advances in Cryptology-Crypto'99*, LNCS 1666: 431-448, Springer-Verlag.

BELLARE, M. and PALACIO, A. (2002): Protecting against key exposure: strongly key-insulated encryption with optimal threshold. Http://eprint.iacr.org/2002/064. Accessed 10-Sep-2006.

CHA, J. C. and CHEON, J. H. (2003): An identity-based signature from Gap Diffie-Hellman groups. In *Proceedings of PCK'03*, LNCS 2567: 18-30, Springer-Verlag.

CHEON, J. H., HOPPER, N., KIM, Y. and OSIPKOV, I. (2006): Timed-release and key-insulated public key encryption. In *Proceedings of FC'06*, LNCS 4107: 191-205, Springer-Verlag.

CORON, J. S.(2000): On the exact security of full domain hash. In *Advances in Cryptology-Crypto'00*, LNCS 1880: 229-235, Springer-Verlag.

DESMEDT, Y. and FRANKEL, Y. (1989): Threshold cryptosystems. In *Advances in Cryptology-Crypto'89*, LNCS 435: 307-315, Springer-Verlag.

DODIS, Y., KATZ, J., XU, S. and YUNG, M. (2003): Strong key-insulated signature schemes. In *Proceedings of PKC'03*, LNCS 2567: 130-144. Springer-Verlag.

DODIS, Y., KATZ, J., XU, S. and YUNG, M. (2002): Key-insulated public-key cryptosystems. In *Advances in Cryptology-Eurocrypt'02*, LNCS 2332: 65-82, Springer-Verlag.

GONZALEZ-DELEITO, N., MARKOVITCH, O. and DALL'OLIO, E. (2004): A new key-insulated signature scheme. In *Proceedings of ICICS'04*, LNCS 3269: 465-479, Springer-Verlag.

GENTRY, C. and SILVERBERG, A. (2002): Hierarchical ID-based cryptography. In *Advances in Cryptology-Asiacrypt' 02*, LNCS 2501: 548-566, Springer-Verlag.

HESS, F. (2002): Efficient identity based signature schemes based on pairings. In *Proceedings of Selected Areas in Cryptography'02*, LNCS 2595: 310-324, Springer-Verlag.

HANAOKA, G., HANAOKA, Y. and IMAI, H. (2006): Parallel key-insulated public key encryption. In *Proceedings of PKC'06*, LNCS 3958: 105-122, Springer-Verlag.

HANAOKA, Y., HANAOKA, G., SHIKATA, J. and IMAI, H. (2002): Unconditionally secure key insulated cryptosystems: models, bounds and constructions. In *Proceedings of ICICS'02*, LNCS 2513: 85-96, Springer-Verlag.

HANAOKA, Y., HANAOKA, G., SHIKATA, J. and IMAI, H. (2005): Identity-based hierarchical strongly key-insulated encryption and its application. In *Advances in Cryptology-Asiacrypt'05*, LNCS 3788: 495-514, Springer-Verlag.

ITIKS, G. and REYZIN, L. (2002): SiBIR: signer-base intrusion-resilient signatures. In *Advances in Cryptology-Crypto'02*, LNCS 2442: 499-514, Springer-Verlag.

LE, Z., OUYANG, Y., FORD, J. and MAKEDON, F. (2004): A hierarchical key-insulated signature scheme in the CA trust model. In *Proceedings of ISC'04*, LNCS 3225: 280-291. Springer-Verlag.

LIU, J. and WONG, D. (2005): Solutions to key exposure problem in ring signature. Http://eprint.iacr.org/2005/427. Accessed 10-Sep-2006.

OSTROVSKY, R. and YUNG, M. (1991): How to withstand mobile virus attacks. In *Proceedings of PODC'91*, 51-59, ACM Press.

PATERSON, K. G. (2006): ID-based signatures from pairings on elliptic curves. *IEEE Communications Letters*, 38(18):1025-1026.

SANTIS, A. D., DESMEDT, Y., FRANKEL, Y. and YUNG, M. (1994): How to share a function securely. In *Proceedings of STOC'94*, pp. 522-533, ACM Press.

SHAMIR, A. (1979): How to share a secret. *Communications of the ACM* 22(11):612-613.

SHAMIR, A. (1984): Identity-based cryptosystems and signature schemes. In *Advances in Cryptology-Crypto'84*, LNCS 196: 47-53, Springer-Verlag.

WENG, J., LIU, S., CHEN, K. and MA, C. (2006): Identity-based parallel key-insulated encryption without random oracles: security notions and construction. In *Proceedings of IndoCrypt'06*, LNCS 4329: 409-423, Springer-Verlag.

YI, X. (2003): An identity-based signature scheme from the Weil pairing. *IEEE Communications Letters*, 7(2): 76-78.

YUM, D.H. and LEE, P.J. (2003): Efficient key updating signature schemes based on IBS. In *Proceedings of Cryptography and Coding'03*, LNCS 2898: 16-18, Springer-Verlag.

ZHOU, Y., CAO, Z. and CHAI, Z. (2006): Identity based key insulated signature. In *Proceedings of ISPEC' 06*, LNCS 3903: 226-234, Springer-Verlag.

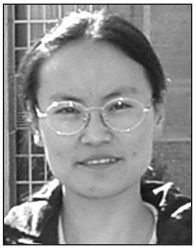
ZHANG, F. and KIM, K. (2002): ID-based blind signature and ring signature from pairings. In *Advances in Cryptology-Asiacrypt'02*, LNCS 2501: 533-547, Springer-Verlag.

BIOGRAPHICAL NOTES

Jian Weng obtained his MS degree in Computer Science from South China University of Technology in 2004. Since 2004, he has been a PhD candidate at Shanghai Jiao Tong University, and is also a member of the Laboratory of Cryptography and Information Security at the University where he has done research for his thesis in the fields of key-exposure protection mechanism. His other research interests include pairing-based cryptography, quantum cryptography, number theory, etc.

Jian Weng

Shengli Liu, obtained her first PhD degree from Xidian University in 2000, and obtained her second PhD degree from Eindhoven University of Technology, Holland in 2002. Her research areas include information theory, computer security, and classical cryptography, etc. She came to Shanghai Jiao Tong University in 2002 and became the adjunct professor at the Department of Computer Science and Engineering. She has published more than 20 high quality papers on cryptography and information security in journals and conferences.



Shengli Liu

Kefei Chen, obtained his PhD degree from Justus Liebig University Giessen, Germany in 1994. His main research areas include classical and modern cryptography, theory and technology of network security, etc. He came to Shanghai Jiao Tong University in 1996 and was appointed professor at the Department of Computer Science and Engineering. He is also the director of the Laboratory of Cryptography and Information Security in Shanghai Jiao Tong University. He has published more than 90 academic papers on cryptology and information security in journals and conferences.

Kefei Chen

Xiangxue Li obtained his MS degree in Mathematics from Nanjing University in 2000. From 2000 to 2003 he was a lecturer at Nanjing University of Posts and Telecommunications. He became a PhD candidate at the Shanghai Jiao Tong University in 2003, and he obtained his PhD degree in 2006. Subsequently, he became a lecturer at the School of Information Security Engineering, Shanghai Jiao Tong University. He has published a number of high quality papers in journals and conferences. His research interests include paring-based cryptography, code-based cryptography, information security, etc.

Xiangxue Li