JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 24, 1143-1157 (2008)

DOI:10.1688/JISE.2008.24.4.9

Identity-Based Parallel Key-Insulated Signature Without Random Oracles*

JIAN WENG[1,2], XIANG-XUE LI[3], KE-FEI CHEN[2] AND SHENG-LI LIU[2]
[1]*Department of Computer Science*
Jinan University
Guangzhou 510632, P.R. China
[2]*Department of Computer Science and Engineering*
[3]*School of Information Security Engineering*
Shanghai Jiao Tong University
Shanghai 200240, P.R. China

We extend Hanaoka *et al.*'s parallel key-insulated mechanism to identity-based signature scenarios, and propose an identity-based parallel key-insulated signature scheme. The proposed scheme enjoys several attractive features: (i) it is provably secure without random oracles; (ii) it is strong key-insulated, and even if one of a user's helper key and some of his temporary secret keys are exposed, it is still impossible for an adversary to derive all of this user's temporary secret keys; (iii) it allows frequent key-updates without increasing the risk of helper key-exposure, and thus enhances the security of the system.

Keywords: parallel key-insulated, identity-based signature, key-exposure, random oracle model, bilinear paring

1. INTRODUCTION

In 1984, Shamir [1] introduced an innovative concept called identity-based cryptography, where an entity's public key is determined as his identity such as email address, and the corresponding private key is generated by a private key generator (PKG). The identity is a natural link to a user, hence it can eliminate the need for certificates as used in the traditional public key infrastructure. So far, a large number of papers have been published in this area (see [2] for some of these), including many ID-based signatures. Standard ID-based signatures rely on the assumption that secret keys are kept perfectly secure. However, with more and more cryptographic primitives being applied to insecure environments, key-exposure seems inevitable. This problem is perhaps the most devastating attack on a cryptosystem, since it typically means that security is entirely lost.

Key-evolving protocol is a practical method to deal with the key-exposure problem. This mechanism includes forward security [3, 4], intrusion-resilience [5] and key-insulation [6]. The latter was introduced by Dodis, Katz, Xu and Yung [6]. In this model, a physically-secure but computationally-limited device named helper is involved. The full-fledged secret key is divided into two parts: a helper key and a temporary secret key. The former is stored in the helper and the latter is kept by the user. The lifetime of the system is divided into discrete time periods. The public key remains unchanged throughout the lifetime, while temporary secret keys are updated periodically: at the beginning of each

Received October 20, 2006; revised March 6, 2007; accepted April 3, 2007.
Communicated by Tzong-Chen Wu.
* This paper was partially supported by NSFC under grants No. 90104005, 60573030 and 60673077.

time period, the user obtains from the helper an update key for the current time period; combining this update key with the temporary secret key for the previous time period, he can derive the temporary secret key for the current time period. A temporary secret key is used to sign a message during the corresponding time period without further access to the helper. Exposure of the temporary secret key at a given time period will not enable an adversary to derive temporary secret keys for the remaining time periods. More precisely, in a (l, N)-*key-insulated* system, the compromise of temporary secret keys for up to l time periods does not expose temporary secret keys for any of the remaining $N - l$ time periods. Therefore, there is no need to revoke the public key unless l time periods have been exposed. If $l = N - 1$ then the scheme is called *perfectly key-insulated*. This is a desirable property for dealing with the key-exposure problem in ID-based signatures. Additionally, *strong key-insulated* security guarantees that the helper is unable to derive the temporary secret key for any time period.

Following the pioneering work due to Dodis *et al.* [6], several elegant key-insulated encryption schemes including some ID-based key-insulated encryptions have been proposed [7-12]. Following Dodis *et al.*'s key-insulated signature schemes [13], efforts have also been devoted to the key-insulated signatures, *e.g.* [14-17].

To minimize the damage caused by key-exposure in ID-based signatures, Zhou *et al.* [18] proposed an ID-based key-insulated signature (IBKIS) scheme (ZCC06 for short). However, the full-fledged secret key in ZCC06 scheme is just wholly stored in the helper. Consequently, it can not satisfy the strong key-insulated security. That is, if an adversary compromises a user's helper, he can derive all of this user's temporary secret keys. Subsequently, Weng *et al.* [19] proposed a new IBKIS scheme with strong key-insulated security (WLCL06 for short). However, both ZCC06 and WLCL06 schemes are provably secure in the random oracle model. As pointed out in [20], a proof in the random oracle model can only serve as a heuristic argument, since it does not imply the security in the implementation. Moreover, there exist some situations that are hard for standard IBKIS schemes to deal with. For example, when key-exposure occurs in IBKIS cryptosystems, the temporary secret key has to be updated at very short intervals to alleviate the damage. Unfortunately, this in turn increases the frequency of helper's connection to insecure environments, and thus increases the risk of helper key-exposure. Keep in mind that even for an IBKIS scheme with strong key-insulated security, once a user's helper key and one of his temporary secret keys are exposed, the adversary can derive all of this user's temporary secret keys.

For deeper understanding, let's consider another example. Suppose a person works in a company's head office in the odd days, while in the even days he works in a branch. To alleviate the damage in case of key-exposure he decides to update his secret key at very short intervals, *e.g.*, once per day. Now, some problems exist: firstly, it is inconvenient but necessary for him to remind himself to bring the helper to the head office in odd days and to the branch in even days; secondly, bringing the helper back and forth means a frequent connection to insecure environments, and thus the risk of helper key-exposure is increased; thirdly, the short renewal interval also increases the risk of helper key-exposure. We notice that Hanaoka *et al.* [11] proposed an ID-based hierarchical key-insulated encryption, where the helper is formed into a hierarchical structure to improve its security. However, it might not be a desirable solution to this example. Because this user still has to bring the first level helper back and forth, and the risk of helper key-exposure will

be increased accordingly. In PKC'06, Hanaoka *et al.* [12] 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 [21], they proposed a parallel key-insulated public-key encryption scheme secure in the random oracle model. Being different from the original key-insulated encryptions, their scheme introduces two distinct helpers that are alternately used to update the secret keys. The 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.* [22] extended this mechanism to ID-based encryptions and proposed an ID-based parallel key-insulated encryption scheme. Since it's worthwhile to deal with the key-exposure problem in ID-based signatures, in this paper, we extend the parallel key-insulated mechanism to ID-based signature scenarios, and proposed an identity-based parallel key-insulated signature (IBPKIS) scheme.

The rest of this paper is organized as follows. Section 2 gives an introduction to bilinear pairings and computational Diffie-Hellman (CDH) assumption. We formalize the definition and security model for IBPKIS 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

Notations. Throughout this paper, let Z_q denote the set $\{0, 1, 2, \ldots, q-1\}$, and Z_q^* denote $Z_q - \{0, 1\}$. Let $d[i]$ denote the ith bit of an integer d in a binary representation. For a set S, we let $|S|$ denote its cardinality. By $\in_R S$, it means choosing a random element from the set S with a uniform distribution. We use $x \leftarrow A$ to denote that algorithm A is executed on some specified input and its output is assigned to the variable x.

2.1 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: $G_1 \times G_1 \rightarrow G_2$ with the following properties:

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

2.1 CDH Assumption

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

$$\mathrm{Adv}_A^{CDH} = \Pr[g \in_R G_1, a, b \in_R Z_q^* : A(g, g^a, b^b) = g^{ab}],$$

where the probability is taken over the random coins consumed by A.

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

3. FRAMEWORK OF IBPKIS

We first give an overview for the IBPKIS model. 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 temporary secret key is updated in every time period. Every user may have 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. We also remark that the framework of IBPKIS in this section and our proposed scheme in the next section can be easily extended to support arbitrary number of helpers for any user trivially). The two helper keys are alternately used to update this user's temporary secret keys, namely, $HK_{ID,1}$ is used in odd time periods, while $HK_{ID,0}$ is involved in even time periods. At time period t, user *ID* obtains an update key $UK_{ID,t}$ from the ith helper (here $i = t \bmod 2$). Combining $UK_{ID,t}$ with the temporary secret key $TSK_{ID,t\text{-}1}$ for the previous time period, he can derive the temporary secret key $TSK_{ID,t}$ for the current time period.

Definition 3 An *IBPKIS scheme* consists of a tuple of six polynomial-time algorithms:

- **Setup:** Takes as input the security parameter k and (possibly) the total number of time periods N. It returns a public parameter *param* and a master key *msk*. We write (*param*, *msk*) ← Setup(k, N);
- **Extract:** Takes as input *msk*, *param* and a user's identity *ID*. It returns the initial secret key $TSK_{ID,0}$ and two helper keys ($HK_{ID,1}$, $HK_{ID,0}$). We write ($TSK_{ID,0}$, ($HK_{ID,1}$, $HK_{ID,0}$)) ← Extract(*msk*, *param*, *ID*);
- **UpdH:** Takes as input a time period index t, a user's identity *ID* and the ith helper key $HK_{ID,i}$ with $i = t \bmod 2$. It returns an update key $UK_{ID,t}$. We write $UK_{ID,t}$ ← UpdH(t, *ID*, $HK_{ID,i}$);
- **UpdT:** Takes as input t, a user's identity *ID*, the temporary secret key $TSK_{ID,t\text{-}1}$ and the updated key $UK_{ID,t}$. It returns a temporary secret key $TSK_{ID,t}$. We write $TSK_{ID,t}$ ← UpdT(t, *ID*, $UK_{ID,t}$, $TSK_{ID,t\text{-}1}$);
- **Sign:** Takes as input a time period index t, a message m and the temporary secret key $TSK_{ID,t}$. It returns a pair (t, σ) composed of the time period t and a signature σ. We write (t, σ) ← Sign(t, m, $TSK_{ID,t}$);
- **Verify:** Takes as input a message m, a candidate signature (t, σ) on m and an identity *ID*. It returns 1 if (t, σ) is a valid signature, and 0 otherwise. We write (1 or 0) ← Verify((t, σ), m, *ID*).

Before giving the security notions for IBPKIS, we consider the following oracles which together model the abilities of an adversary:

- $KEO(\cdot)$: *a key-extraction oracle*, upon receiving a user's identity *ID*, returns this user's initial secret key $TSK_{ID,0}$ and two helper keys ($HK_{ID,1}$, $HK_{ID,0}$);

– *HKO*(·,·): *a helper key oracle*, upon receiving a tuple <*ID*, *i*> consists of a user's identity *ID* and an index *i* of the helper, returns the helper key $HK_{ID,i}$;
– *TKO*(·,·): *a temporary secret key oracle*, upon receiving a tuple <*ID*, *t*> consists of a user's identity *ID* and a time period index *t*, returns the temporary secret key $TSK_{ID,t}$.
– *SO*(·,·,·): *a signing oracle*, upon receiving a tuple <*ID*, *t*, *m*> consists of a user's identity *ID*, a time period index *t* and a message *m*, returns a signature Sign(*t*, *m*, $TSK_{ID,t}$).

To model the *key-insulated security* for IBPKIS, besides oracles *TKO* and *SO*, we further provide oracle *KEO* for him. Moreover, we even provide oracle *HKO* for the adversary and allow him to compromise one of the helper keys for the challenged identity.

Definition 4 Let $\prod$ be an IBPKIS scheme. For a polynomial-time adversary *A*, his advantage is defined as

$$\mathrm{Adv}^{\prod}_{A,KI}(k) = \Pr\left[\begin{array}{l}(param, msk) \leftarrow \mathrm{Setup}(k, N); ((t^*, \sigma^*), ID^*, m^*) \leftarrow \\ A^{KEO(\cdot);HKO(\cdot,\cdot);TKO(\cdot,\cdot);SO(\cdot,\cdot,\cdot)}(param) : \mathrm{Verify}((t^*, \sigma^*), ID^*, m^*) = 1\end{array}\right],$$

where it is mandated that: (1) *A* can not submit <ID^*, t^*, m^*> to oracle *SO*; (2) <ID^*, t^*> was never submitted to oracle *TKO*; (3) ID^* was never submitted to *KEO*; (4) *A* can not submit <ID^*, $t^* - 1$> to oracle *TKO* and <ID^*, $t^* \bmod 2$> to oracle *HKO* simultaneously; (5) A can not submit <ID^*, $t^* + 1$> to oracle *TKO* and <ID^*, $(t^* + 1) \bmod 2$> to oracle *HKO* simultaneously; (6) *A* can not compromise both of ID^*'s helper keys. We say that $\prod$ is *perfectly key-insulated* if for any polynomial-time adversary *A*, $Adv^{\prod}_{A,KI}(k)$ is negligible.

Remark 1: For those non-challenged identities, oracle *TKO* is of no help for adversary *A*, since he can derive any temporary secret key for these identities by querying oracle *KEO*. Therefore, without loss of generality, we require that adversary *A* only query oracle *TKO* on the challenged identity.

To model the *strong key-insulated* security for IBPKIS, We provide oracle *HKO* for the attacker and allow him to query all the helper keys for any identity, even including the challenged identity. However, the adversary is disallowed to query oracle *TKO* on the challenged identity for any time period. Note that we allow the adversary to query oracle *TKO* on the non-challenged identities for any time period. Since these queries are implied by the *KEO* queries, we do not explicitly provide oracle *TKO* for the adversary in the following definition.

Definition 5 Let $\prod$ be an IBPKIS scheme. For a polynomial-time adversary *A*, his *advantage* is defined as

$$\mathrm{Adv}^{\prod}_{A,SKI}(k) = \Pr\left[\begin{array}{l}(param, msk) \leftarrow \mathrm{Setup}(k, N); ((t^*, \sigma^*), ID^*, m^*) \leftarrow \\ A^{KEO(\cdot);HKO(\cdot,\cdot);SO(\cdot,\cdot,\cdot)}(param) : \mathrm{Verify}((t^*, \sigma^*), ID^*, m^*) = 1\end{array}\right],$$

where it is mandated that: (1) *A* can not query oracle *SO* on <ID^*, t^*, m^*>; (2) ID^* was never submitted to oracle *KEO*. We say that $\prod$ is *strong key-insulated* if for any polynomial-time adversary *A*, $Adv^{\prod}_{A,SKI}(k)$ is negligible.

Finally, as in [13], we address an adversary who compromises the user's storage while a key is being updated from $TSK_{ID,t-1}$ to $TSK_{ID,t}$, and we call it a key-update exposure at (*ID*, *t*). When this occurs, the adversary gets $TSK_{ID,t-1}$, $UK_{ID,t}$ and $TSK_{ID,t}$. We say an IBPKIS scheme *has secure key-updates* if a key-update exposure at (*ID*, *t*) is of no more help to the adversary than issuing *TKO* queries on <*ID*, *t* − 1> and <*ID*, *t*>.

Definition 6 An IBPKIS scheme *has secure key-updates* if the view of any adversary *A* making a key-update exposure at <*ID*, *t*> can be perfectly simulated by an adversary *A*′ making *TKO* queries on <*ID*, *t* − 1> and <*ID*, *t*>.

4. OUR PROPOSED IBPKIS SCHEME

4.1 Construction

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, g be a random generator of G_1, and e be a bilinear map such that e: $G_1 \times G_1 \to G_2$. Let H_1: $\{0, 1\}^* \to \{0, 1\}^{n_u}$ and H_1: $\{0, 1\}^* \to \{0, 1\}^{n_m}$ be two collision-resistant hash functions for some n_u, $n_m \in Z$. Let F be a pseudo-random function (PRF) such that given a k-bit seed s and a k-bit input x, it outputs a k-bit string $F_s(x)$ (refer to [23] for details about PRF). Based on Paterson-Schuldt's ID-based signature scheme [24], which is based on Water's ID-based encryption scheme [25], our proposed IBPKIS scheme consists of the following six algorithms:

Setup: The PKG picks $\alpha \in Z_q^*$, $g_2 \in_R G_1$ and defines $g_1 = g^\alpha$. Next, it chooses u', $m' \in_R G_1$. It also chooses a n_u-dimensional vector $\vec{U} = (\hat{u}_i)$ with $\hat{u}_i \in_R G_1$ for $i = 1, \ldots, n_u$. Another n_m-dimensional vector, $\vec{M} = (\hat{m}_j)$ with $\hat{m}_j \in_R G_1$ for $i = 1, \ldots, n_u$, is also chosen. The public parameter is $param = (g, g_1, g_2, u', m', \vec{U}, \vec{M}, H_1, H_2)$, and the master key is $msk = g_2^\alpha$.

For convenience, we define two functions L_1 and L_2 such that $L_1(S) = u' \prod_{i \in S} \hat{u}_i$, $L_2(S) = m' \prod_{j \in S'} \hat{m}_j$, where $S \subseteq \{1, \ldots, n_u\}$ and $S' \subseteq \{1, \ldots, n_m\}$. Furthermore, to make the notation easy to follow, for a given identity *ID*, time period *t* and message *m*, we hereafter use $U_{ID,t}$ to denote the set $\{i \mid S_1[i] = 1, S_1 = H_1(ID \| t)\}$, U'_{ID} to denote $\{j \mid S_2[i] = 1, S_2 = H_1(ID)\}$, and M_m to denote $\{k \mid S_3[k] = 1, S_3 = H_2(m)\}$.

Extract: Given an identity *ID*, the PKG randomly chooses two helper keys $HK_{ID,1}$, $HK_{ID,0} \in_R \{0,1\}^k$, and computes $k_{ID,-1} = F_{HK_{ID,1}}(-1 \| ID)$, $k_{ID,0} = F_{HK_{ID,0}}(0 \| ID)$. Next, it chooses $r \in_R Z_q^*$ and defines the initial secret key to be

$$TSK_{ID,0} = (g_2^\alpha L_1(U'_{ID})^r L_1(U_{ID,-1})^{k_{ID,-1}} L_1(U_{ID,0})^{k_{ID,0}}, g^{k_{ID,-1}}, g^{k_{ID,0}}, g^r). \quad (1)$$

Finally, it outputs the initial secret key $TSK_{ID,0}$ and the two helper keys $HK_{ID,1}$ and $HK_{ID,0}$.

UpdH: given an identity *ID* and a time period index *t*, the *i*th (here $i = t \bmod 2$) helper for

ID defines and returns the update key as $UK_{ID,t} = (L_1(U_{ID,t})^{k_{ID,t}} \big/ L_1(U_{ID,t-2})^{k_{ID,t-2}}, g^{k_{ID,t}})$, where $k_{ID,t-2} = F_{HK_{ID,i}}(t-2 \| ID)$, $k_{ID,t} = F_{HK_{ID,i}}(t \| ID)$.

UpdT: Given a time period index t, an update key $UK_{ID,t} = (\hat{Q}_{ID,t}, \hat{R}_{ID,t})$ and a temporary secret key $TSK_{ID,t-1} = (Q_{ID,t-1}, R_{ID,t-1}, R_{ID,t-2}, R)$, this algorithm returns the temporary secret key for user *ID* in time period t as $TSK_{ID,t} = (Q_{ID,t-1} \cdot \hat{Q}_{ID,t}, R_{ID,t-1}, \hat{R}_{ID,t}, R)$.

Note that if let $i = t \bmod 2$ and $j = (t-1) \bmod 2$, then the temporary secret key is always set to be

$$TSK_{ID,t} = (g_2^{\alpha} L_1(U'_{ID})^r L_1(U_{ID,t-1})^{k_{ID,t-1}} L_1(U_{ID,t})^{k_{ID,t}}, g^{k_{ID,t-1}}, g^{k_{ID,t}}, g^r), \quad (2)$$

where $k_{ID,t-1} = F_{HK_{ID,j}}(t-1 \| ID)$, $k_{ID,t} = F_{HK_{ID,i}}(t \| ID)$.

Sign: To produce a signature on m during time period t, the user *ID* with temporary secret key $TSK_{ID,t} = (Q_{ID,t}, R_{ID,t-1}, R_{ID,t}, R)$ first chooses $l'_{t-1}, l'_t, r_m \in_R Z_q^*$, and then computes $S_{ID,t-1} = R_{ID,t-1} \cdot g^{l'_{t-1}}$, $S_{ID,t} = R_{ID,t} \cdot g^{l'_t}$, $S_m = g^{r_m}$ and $V = Q_{ID,t} \cdot L_1(U_{ID,t-1})^{l'_{t-1}} L_1(U_{ID,t})^{l'_t} L_2(M_m)^{r_m}$. The signature is $\sigma = (t, V, S_{ID,t-1}, S_{ID,t}, R, S_m)$.

Note that let $i = t \bmod 2$ and $j = (t-1) \bmod 2$, the signature is always set to be

$$\sigma = (g_2^{\alpha} L_1(U'_{ID})^r L_1(U_{ID,t-1})^{k_{ID,t-1}} L_1(U_{ID,t})^{k_{ID,t}} L_2(M_m)^{r_m}, g^{k_{ID,t-1}}, g^{k_{ID,t}}, g^r, g^{r_m}), \quad (3)$$

where $l_{ID,t-1} = l'_{t-1} + F_{HK_{ID,j}}(t-1 \| ID)$, $l_{ID,t} = l'_t + F_{HK_{ID,j}}(t \| ID)$.

Verify: given a purported signature $\sigma = (t, V, S_{ID,t-1}, S_{ID,t}, R, S_m)$ of an identity *ID* on a message m, a verifier accepts σ iff. the following equality holds:

$$e(g, V) = e(g_1, g_2) e(R, L_1(U'_{ID})) e(S_{ID,t-1}, L_1(U_{ID,t-1})) e(S_{ID,t}, L_1(U_{ID,t})) e(S_m, L_2(M_m)). \quad (4)$$

Remark 2: There exist some implicit relations in the above scheme: (i) according to Eqs. (1)-(3), a given user's initial secret key, all of his temporary secret keys and all the signature signed by him share the same exponent r; (ii) all the temporary secret keys for a given user *ID* are mutually dependent on one another, namely, $TSK_{ID,t-1}$ and $TSK_{ID,t}$ share the same exponent $k_{ID,t-1}$, while $TSK_{ID,t}$ and $TSK_{ID,t+1}$ share the same exponent $k_{ID,t}$.

4.2 Comparison with IBKIS Scheme

In this subsection, we give a comparison between our IBPKIS scheme and IBKIS schemes. In our IBPKIS scheme, every user has two helper keys to *alternately* update his temporary secret key. As to the aforementioned person who works in both the head office and a branch, now he can put the first helper key in the head office and the second one in the branch. Then he no longer needs to remind himself to bring the helper back and forth. Moreover, since the two helper keys are alternately used, even if the frequency of temporary secret key-updates in our scheme is twice as those in IBKIS schemes, the risk of each helper's key-exposure is still the same as those of IBKIS schemes. This means that our proposed IBPKIS scheme allows frequent key-updates without increasing the risk of

helper key-exposure. Besides, even if one of a user's helper key and some of his temporary secret keys are exposed, it is still impossible for an adversary to derive all of this user's temporary secret keys. On the contrary, even for the strong key-insulated IBKIS scheme, once a user's helper key and one of his temporary secret keys are exposed, all of his temporary secret keys will be exposed. In the next section, we will prove that the proposed scheme enjoys desirable features such as secure key-updates, perfectly key-insulation and strong key-insulation in the standard model. We give a comparison between the proposed scheme and existing IBKIS schemes in Table 1.

Table 1. Comparison of the proposed IBPKIS scheme and existing IBKIS schemes.

	Proposed Scheme	ZCC06 [18]	WLCL06 [19]
Perfectly Key-Insulated	√	√	√
Strong Key-Insulated	√	×	√
Secure Key-Updates	√	√	√
Without Random Oracles	√	×	×
Allow Frequent Key-Updates	√	×	×

"Allow Frequent Key-Updates" means allowing frequent key-updates without increasing the risk of helper key-exposure.

We can use Naccache and Sarkar-Chatterjee's technique [26, 27] to reduce the size of public parameters (see section 6 in [24] for more details). However, we are quite aware that since our proposed scheme is based on Paterson-Schuldt's IBS scheme [24], our IBPKIS scheme is a little more expensive than ZCC06 and WLCL06 schemes. It's an open question to construct an IBPKIS scheme satisfying both high efficiency and provable security without random oracles.

5. SECURITY ANALYSIS

Theorem 1 The proposed scheme is perfectly key-insulated in the standard model under the CDH assumption in G_1. Concretely, given an adversary A that has advantage ε against the perfectly key-insulated security of our scheme by running within time T, asking at most q_k (q_h, q_t, q_s resp.) queries to oracle *KEO* (*HKO*, *TKO*, *SO*, resp.), there exists a (T', ε') adversary B that breaks the CDH assumption in G_1 with $T' \le T + O((q_k + q_t + q_s) t_e + (n_u(q_k + q_t) + (n_u + n_m)q_s)t_m)$ and $\varepsilon' \ge 9\varepsilon/1024(q_k + q_t + 3q_s)^3(n_u + 1)^3 q_s(n_u + 1)$, where t_e and t_m denote the running time of an exponentiation and a multiplication in G_1 respectively.

Proof: We will show how to construct a (T', ε')-adversary B against the CDH assumption in group G_1. Suppose B is given a tuple $(g, g^a, g^b) \in G_1^3$ for some unknown $a, b \in Z_q^*$. B's goal is to derive g^{ab} by interacting with A. B flips a fair coin $COIN \in \{0, 1\}$. If $COIN = 1$, B plays **Game 1** with A, else he plays **Game 2**.

Game 1: In this game, B acts as a challenger expecting that A will never corrupt the helper key of the challenged identity. Note that in this game, B can randomly choose the exponent $k_{ID,t}$ on his own for the challenged identity ID and a time period index t, since

$k_{ID,t}$ is the output of a PRF and A does not know the corresponding seeds $HK_{ID,1}$ and $HK_{ID,0}$. B interacts with A in the following way:

Setup: B randomly chooses two integers k_u and k_m such that $0 \le k_u \le n_u$, $0 \le k_m \le n_m$. Let $l_u = 4(q_k + q_t + 3q_s)/3$ and $l_m = 2q_s$. We here assume that $l_u(n_u + 1) < q$ and $l_m(n_m + 1) < q$. Next, it randomly chooses $x' \in_R Z_{l_u}$, $z' \in_R Z_{l_m}$, y', $w' \in_R Z_q$ and assigns $g_1 = g^a$, $g_2 = g^b$, $u' = g_2^{x'-l_u k_u} g^{y'}$ and $m' = g_2^{x'-l_m k_m} g^{w'}$. It also chooses $\hat{x}_i, \hat{y}_i \in_R Z_{l_u}$ and assigns $\vec{U} = (\hat{u}_i)$ with $\hat{u}_i = g_2^{\hat{x}_i} g^{\hat{y}_i}$ for $i = 1, \ldots, n_u$. Besides, it chooses $\hat{z}_j, \hat{w}_j \in_R Z_{l_m}$ and assigns $\vec{M} = (\hat{m}_i)$ with $\hat{m}_j = g_2^{\hat{z}_j} g^{\hat{w}_j}$ for $j = 1, \ldots, n_u$. Finally, B returns the public parameters to A.

Observe that from the perspective of the adversary, the distributions of these public parameters are identical to the real construction. Note that the master key is implicitly set to be $g_2^{\alpha} = g_2^a = g^{ab}$.

To make the notation easy to follow, we define four functions J_1, J_2, K_1 and K_2 such that $K_1(S) = x' - l_u k_u + \sum_{i \in S} \hat{x}_i$, $J_1(S) = y' + \sum_{i \in S} \hat{y}_i$, $K_2(S) = z' - l_m k_m + \sum_{j \in S'} \hat{z}_j$, $J_2(S') = w' + \sum_{j \in S'} \hat{w}_j$, where $S \subseteq \{1, \ldots, n_u\}$, $S' \subseteq \{1, \ldots, n_m\}$. Note that the following equalities always hold: $g_2^{K_1(S)} g^{J_1(S)} = L_1(S)$, $g_2^{K_2(S')} g^{J_2(S')} = L_2(S')$.

To embody the implicit relations mentioned in Remark 2, B forms a list named R^{list} as explained below. For easy explanation, an algorithm named *RQuery* is also defined.

Algorithm *RQuery*(ID, t):

If there exists a tuple in R^{list} for this input then output the predefined value.

Else if t = '–' then choose $\hat{r} \in_R Z_q^*$, add (ID, '–', $\hat{r}$) on R^{list}, return $\hat{r}$.

Else choose $\hat{k}_{ID,t} \in_R Z_q^*$ add $(ID, t, \hat{k}_{ID,t})$ on R^{list}, return $\hat{k}_{ID,t}$.

End if

End if

Oracles Simulations: B answers a series of oracle queries for A in the following way:

Oracle HKO simulation. B maintains a list HK^{list} which is initially empty. Upon receiving a helper key query <ID, i> with $i \in \{0, 1\}$. B first checks whether HK^{list} contains a tuple (ID, i, $HK_{ID,i}$). If it does, $HK_{ID,i}$ is returned to A. Otherwise, B chooses $HK_{ID,i} \in_R \{0,1\}^k$, adds ($ID$, i, $HK_{ID,i}$) into HK^{list} and returns $HK_{ID,i}$ to A.

Oracle KEO simulation. Upon receiving a *KEO* query on identity ID, B outputs "failure" and aborts if $K_1(U'_{ID}) \equiv 0 \bmod q$ (denote this event by **E1**). Otherwise, B issues *HKO* queries on <ID, 1> and <ID, 0> to obtain $HK_{ID,1}$ and $HK_{ID,0}$. Next, it computes $\hat{r}$ = *RQuery* (ID, '–') and assigns the initial secret key $TSK_{ID,0}$ to be

$$(g_1^{J_1(U'_{ID})/K_1(U'_{ID})} L_1(U'_{ID})^{\hat{r}} L_1(U_{ID,-1})^{k_{ID,-1}} L_1(U_{ID,0})^{k_{ID,0}}, g^{k_{ID,-1}}, g^{k_{ID,0}}, g^{-1/K_1(U'_{ID})} g^{\hat{r}}),$$

where $k_{ID,-1} = F_{HK_{ID,1}}(-1 \| ID)$ and $k_{ID,0} = F_{HK_{ID,0}}(0 \| ID)$. Finally, B returns ($TSK_{ID,0}$, $HK_{ID,1}$, $HK_{ID,0}$) to A.

Observe that, if let $r = \hat{r} - a/K_1(U'_{ID})$, then in a similar analysis as in [25], it can be verified that $TSK_{ID,0}$ has the correct form as Eq. (1).

Oracle TKO queries. As argued in Remark 1, we require that A just queries oracle *TKO* on the challenged identity. Upon receiving a *TKO* query <*ID*, *t*> (Wlog, we assume t is even, note that an odd t can be handled in a similar manner), B outputs "failure" and aborts if $K_1(U'_{ID}) \equiv K_1(U_{ID,t\text{-}1}) \equiv 0 \bmod q$ holds (denote this event by **E2**. Note that to embody the mutually dependent relation mentioned in Remark 2, we do not make use of the case $K_1(U_{ID,t\text{-}1}) \neq 0 \bmod q$ for an odd $t-1$). Otherwise, it first computes $\hat{r} = RQuery(ID, \text{'}-\text{'})$, $\hat{k}_{ID,t} = RQuery(ID, t)$ and $\hat{k}_{ID,t-1} = RQuery(ID, t-1)$. Next, if $K_1(U'_{ID}) \neq 0 \bmod q$, it defines and returns $TSK_{ID,t}$ as

$$(g_1^{J_1(U'_{ID})/K_1(U'_{ID})} L_1(U'_{ID})^{\hat{r}} L_1(U_{ID,t-1})^{\hat{k}_{ID,t-1}} L_1(U_{ID,t})^{\hat{k}_{ID,t}}, g^{\hat{k}_{ID,t-1}}, g^{\hat{k}_{ID,t}}, g^{-1/K_1(U'_{ID})} g^{\hat{r}}), \quad (5)$$

else if $K_1(U_{ID,t}) \neq 0 \bmod q$, it defines and returns $TSK_{ID,t}$ as

$$(g_1^{J_1(U_{ID,t})/K_1(U_{ID,t})} L_1(U'_{ID})^{\hat{r}} L_1(U_{ID,t-1})^{\hat{k}_{ID,t-1}} L_1(U_{ID,t})^{\hat{k}_{ID,t}}, g^{\hat{k}_{ID,t-1}}, g^{-1/K_1(U_{ID,t})} g^{\hat{k}_{ID,t}}, g^{\hat{r}}). \quad (6)$$

Observe that in both cases, $TSK_{ID,t}$ has the correct form as Eq. (2) and is indeed a valid temporary secret key.

Oracle SO simulation. Upon receiving a *SO* query <*ID*, *t*, *m*>, B outputs "failure" and aborts if $K_1(U'_{ID}) \equiv K_1(U_{ID,t-1}) \equiv K_1(U_{ID,t}) \equiv K_1(M_m) \equiv 0 \bmod q$ holds (denote this event by **E3**). Otherwise, B first computes $\hat{r} = RQuery(ID, \text{'}-\text{'})$, and then constructs the signature for A according to four cases:

– $K_1(U'_{ID}) \neq 0 \bmod q$: Choose $l_{ID,t\text{-}1}, l_{ID,t}, r_m \in_R Z_q^*$ and assigns the signature σ as

$$(t, g_1^{J_1(U'_{ID})/K_1(U'_{ID})} L_1(U'_{ID})^{\hat{r}} L_1(U_{ID,t-1})^{l_{ID,t-1}} L_1(U_{ID,t})^{l_{ID,t}} L_2(M_m)^{r_m}, g^{l_{ID,t-1}}, g^{l_{ID,t}}, g^{-1/K_1(U'_{ID})} g^{\hat{r}}, g^{r_m}).$$

– $(K_1(U'_{ID}) \equiv 0 \bmod q) \wedge (K_1(U_{ID,t\text{-}1}) \neq 0 \bmod q)$: Choose $\hat{l}_{ID,t-1}, l_{ID,t}, r_m \in_R Z_q^*$ and assigns the signature σ as

$$\left(t, g_1^{\frac{J_1(U_{ID,t-1})}{K_1(U_{ID,t-1})}} L_1(U'_{ID})^{\hat{r}} L_1(U_{ID,t-1})^{\hat{l}_{ID,t-1}} L_1(U_{ID,t})^{l_{ID,t}} L_2(M_m)^{r_m}, g^{\frac{-1}{K_1(U_{ID,t-1})}} g^{\hat{l}_{ID,t-1}}, g^{l_{ID,t}}, g^{\hat{r}}, g^{r_m}\right).$$

– $(K_1(U'_{ID}) \equiv K_1(U_{ID,t\text{-}1}) \equiv 0 \bmod q) \wedge (K_1(U_{ID,t}) \neq 0 \bmod q)$: Choose $l_{ID,t-1}, \hat{l}_{ID,t}, r_m \in_R Z_q^*$ and assigns the signature σ as

$$\left(t, g_1^{\frac{J_1(U_{ID,t})}{K_1(U_{ID,t})}} L_1(U'_{ID})^{\hat{r}} L_1(U_{ID,t-1})^{l_{ID,t-1}} L_1(U_{ID,t})^{\hat{l}_{ID,t}} L_2(M_m)^{r_m}, g^{l_{ID,t-1}}, g^{\frac{-1}{K_1(U_{ID,t})}} g^{\hat{l}_{ID,t}}, g^{\hat{r}}, g^{r_m}\right).$$

– $(K_1(U'_{ID}) \equiv K_1(U_{ID,t\text{-}1}) \equiv K_1(U_{ID,t}) \equiv 0 \bmod q) \wedge (K_2(M_m) \neq 0 \bmod q)$: Choose $l_{ID,t-1}, l_{ID,t}$, $\hat{r}_m \in_R Z_q^*$ and assigns the signature σ as

$$\left(t,\ g_1^{\frac{J_2(M_m)}{K_2(M_m)}} L_1(U'_{ID})^{\hat{r}}\, L_1(U_{ID,t-1})^{l_{ID,t-1}}\, L_1(U_{ID,t})^{l_{ID,t}}\, L_2(M_m)^{\hat{r}_m},\ g^{l_{ID,t-1}},\ g^{l_{ID,t}},\ g^{\hat{r}},\ g^{\frac{-1}{K_2(M_m)}} g^{\hat{r}_m}\right).$$

In all cases, it can be verified that σ has the correct form as Eq. (3).

Forge: Eventually, B returns a forged signature $\sigma^* = (t^*, V^*, S_{ID^*,t^*-1}, S_{ID^*,t^*}, R^*, S_{m^*})$ on an identity ID^*, a time period index t^* and a message m^* with the constraints described in Definition 4. B first checks whether A has corrupted one of ID^*'s helper keys during this game. If it does, B outputs "failure" and aborts (denote this event by **E4**). Otherwise, B further checks whether $K_1(U'_{ID^*}) \equiv K_1(U_{ID^*,t^*-1}) \equiv K_1(U_{ID^*,t^*}) \equiv K_2(M_{m^*}) \equiv \bmod q$ holds. holds. If not, B outputs "failure" and aborts (denote this event by **E5**); otherwise, B can successfully compute g^{ab} as $V^* \Big/ R^{*J_1(U'_{ID^*})} S_{ID^*,t^*-1}^{J_1(U_{ID^*,t^*-1})} S_{ID^*,t^*}^{J_1(U_{ID^*,t^*})} S_{m^*}^{J_2(M_{m^*})}$.

Game 2: In this game, B acts as a challenger expecting that A will corrupt exactly one of the helper keys on the challenged identity. B picks $\gamma \in_R \{0, 1\}$ and bets on that A queries on the γth helper. Wlog, we assume $\gamma = 1$ (the case of $\gamma = 0$ can be handled in a similar manner). Then for the challenged identity ID and an even time period index t, B can randomly choose the exponent $k_{ID,t}$ since F is a PRF and A does not know the corresponding seed $HK_{ID,0}$. B provides the simulation of **Setup**, oracles *HKO*, *KEO* and *SO* for A in the same way as Game 1. Here we let **F1** and **F3** denote the abort events in oracle *KEO* simulation and oracle *SO* simulation respectively. B provides oracle *TKO* simulation for A as follows:

Oracle TKO simulation. As argued in Remark 1, we also require that A just query oracle *TKO* on the challenged identity. For a *TKO* query $<ID, t>$, we explain how to deal with the case of an even t (the case of an odd t can be handled in a similar manner): B outputs "failure" and aborts if $K_1(U'_{ID}) \equiv K_1(U_{ID,t}) \equiv 0 \bmod q$ holds (denote this event by **F2**). Otherwise, B first computes $\hat{r} = RQuery(ID, '-')$, $\hat{k}_{ID,t} = RQuery(ID, t)$ and $l_{ID,t-1} = F_{HK_{ID,t}}(t - 1 \,||\, ID)$. If $K_1(U'_{ID}) \equiv 0 \bmod q$, it defines $TSK_{ID,t}$ similarly to Eq. (5), else if $K_1(U_{ID,t}) \neq 0 \bmod q$, it defines $TSK_{ID,t}$ similarly to Eq. (6). It can be verified that in both cases, $TSK_{ID,t}$ has the correct form as Eq. (2).

Forge: Eventually, A returns a forged signature $\sigma^* = (t^*, V^*, S_{ID^*,t^*-1}, S_{ID^*,t^*}, R^*, S_{m^*})$ on identity ID^*, messages m^* and time period index t^* with the constraints described in Definition 4. B first checks whether A has corrupted $HK_{ID^*,1}$ during this game. If not, B outputs "failure" and aborts (denote this event by **F4**). Otherwise, B further checks whether $K_1(U'_{ID^*}) \equiv K_1(U_{ID^*,t^*-1}) \equiv K_1(U_{ID^*,t^*}) \equiv (M_{m^*-1}) \equiv 0 \bmod q$ holds. If not, B outputs "failure" and aborts (denote this event by **F5**); else B can successfully compute g^{ab} in the same way as Game 1.

This completes the simulation. From the description of B, we know that the time complexity of B is dominated by the exponentiations and the multiplications in the simulation of oracles *TKO*, *KEO* and *SO*. Since there are $O(1)$ exponentiations in each stage, and $O(n_u)$, $O(n_u)$ and $O(n_u + n_m)$ multiplications in the above three oracle simulations respectively, we know that the time complexity of B is bounded by $T' \leq T + O((q_k + q_t + q_s)t_e + (n_u(q_k + q_t) + (n_u + n_m)q_s)t_m)$.

Next, we bound B's advantage against the CDH assumption in G_1. Let Pr[¬abort1] and Pr[¬abort2] denote the probabilities of B's not aborting in Games 1 and 2 respectively. Similarly to the analysis in [24], we have the following claim.

Claim 1: $\Pr[\neg\text{abort1}] \geq 9/1024(q_k + q_t + 3q_s)^3(n_u + 1)^3 q_s(n_m + 1)$,
$\Pr[\neg\text{abort2}] \geq 9/1024(q_k + q_t + 3q_s)^3(n_u + 1)^3 q_s(n_m + 1)$.

From the description of the simulation, we know that if B does not abort, the Setup phase and the responds to A's *HKO*, *KEO*, *TKO* and *SO* queries are valid, and the implicit relations mentioned in Remark 2 are also satisfied. Therefore, if B does not abort, A can successfully return a valid forged signature with advantage ε, then B can solve the CDH problem instance. Since *COIN* is a fair coin, we know that B can solve the CDH problem instance with advantage $\varepsilon' = \frac{1}{2}\Pr[\neg\text{abort1}] \cdot \varepsilon + \frac{1}{2}\Pr[\neg\text{abort2}] \cdot \varepsilon \geq 9\varepsilon/1024(q_k + q_t + 3q_s)^3(n_u + 1)^3 q_s(n_m + 1)$. ❑

Theorem 2 The proposed scheme is strong key-insulated in the standard model under the CDH assumption in group G_1. Concretely, given an adversary A that has advantage ε against the strong key-insulated security of our proposed scheme by running within time T, asking at most q_k (q_h, q_s, resp.) queries to oracle *KEO* (*HKO*, *SO*, resp.), there exists a (T', ε') adversary B that breaks the CDH assumption in group G_1 with $T' \leq T + O((q_k + q_s)t_e + (n_u q_k + (n_u + n_m)q_s)t_m)$ and $\varepsilon' \geq 9\varepsilon/1024(q_k + 3q_s)^3(n_u + 1)^3 q_s(n_m + 1)$, where t_e and t_m has the same meaning as Theorem 1.

***Proof*:** (Sketch) On input $(g, g^a, g^b) \in G_1^3$ for some unknown $a, b \in Z_q^*$, B's goal is to compute g^{ab}. B interacts with A as follows:

Setup: The same as Theorem 1 except that l_u is set to be $l_u = 4(q_k + 3q_s)/3$.

Oracle Simulation: To embody the implicit relations mentioned in Remark 2, we also define the algorithm *RQuery* as in Theorem 1. B provides the simulation of oracles *HKO*, *KEO* and *SO* for A in the same way as Game 1 in Theorem 1. Note that we need not provide oracle *TKO* for A.

Forge: Eventually, A returns a forged signature σ^* with the constraints described in Definition 4. B can derive g^{ab} in the same way as Game 1 in Theorem 1.

As the proof of Theorem 1, we can see that B's running time is bounded by $T' \leq T + O((q_k + q_s)t_e + (n_u q_k + (n_u + n_m)q_s)t_m)$. Similarly, the advantage of B can be bounded by $\varepsilon' \geq 9\varepsilon/1024(q_k + 3q_s)^3(n_u + 1)^3 q_s(n_m + 1)$. ❑

Theorem 3 The proposed scheme has secure key-updates.

This theorem follows from the fact that for any time period t and any identity *ID*, the update key $UK_{ID,t}$ can be derived from $TSK_{ID,t}$ and $TSK_{ID,t\text{-}1}$.

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 the 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 model for IBPKIS, 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. Our scheme is provably secure without resorting to the random oracle methodology, which is also a desirable feature since a proof in the random oracle model can only serve as heuristic argument and does not imply the security in the implementation.

REFERENCES

1. A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of CRYPTO on Advances in Cryptology*, LNCS 196, Springer-Verlag, 1984, pp. 47-53.
2. P. Barreto, "The pairing-based crypto lounge," http://paginas.terra.com.br/informatica/paulobarreto/pblounge.html.
3. R. Anderson, "Two remarks on public-key cryptology," in *Proceedings of the 4th ACM Conference on Computer and Communications Security*, 1997, http:// www.cl.cam.ac.uk/users/rja14/.
4. M. Bellare and S. Miner, "A forward-secure digital signature scheme," in *Proceedings of CRYPTO on Advances in Cryptology*, LNCS 1666, Springer-Verlag, 1999, pp. 431-448.
5. G. Itkis and L. Reyzin, "SiBIR: signer-base intrusion-resilient signatures," in *Proceedings of CRYPTO on Advances in Cryptology*, LNCS 2442, Springer-Verlag, 2002, pp. 499-514.
6. Y. Dodis, J. Katz, S. Xu, and M. Yung, "Key-insulated public-key cryptosystems," in *Proceedings of EUROCRYPT on Advances in Cryptology*, LNCS 2332, Springer-Verlag, 2002, pp. 65-82.
7. M. Bellare and A. Palacio, "Protecting against key exposure: strongly key-insulated encryption with optimal threshold," http://eprint.iacr.org/2002/064.
8. Y. Hanaoka, G. Hanaoka, J. Shikata, and H. Imai, "Unconditionally secure key insulated cryptosystems: models, bounds and constructions," in *Proceedings of the 4th International Conference on Information and Communications Security*, LNCS 2513, Springer-Verlag, 2002, pp. 85-96.
9. Y. Dodis and M. Yung, "Exposure-resilience for free: the hierarchical ID-based encryption case," in *Proceedings of the 1st International IEEE Security in Storage Workshop*, 2002, pp. 45-52.
10. J. H. Cheon, N. Hopper, Y. Kim, and I. Osipkov, "Authenticated key-insulated public key encryption and timed-release cryptography," http://eprint.iacr.org/2004/231.

11. Y. Hanaoka, G. Hanaoka, J. Shikata, and H. Imai, "Identity-based hierarchical strongly key-insulated encryption and its application," in *Proceedings of ASIACRYPT on Advances in Cryptology*, LNCS 3788, Springer-Verlag, 2005, pp. 495-514.
12. G. Hanaoka, Y. Hanaoka, and H. Imai, "Parallel key-insulated public key encryption," in *Proceedings of Public Key Cryptography*, LNCS 3958, Springer-Verlag, 2006, pp. 105-122.
13. Y. Dodis, J. Katz, S. Xu, and M. Yung, "Strong key-insulated signature schemes," in *Proceedings of Public Key Cryptography*, LNCS 2567, Springer-Verlag, 2003, pp. 130-144.
14. Z. Le, Y. Ouyang, J. Ford, and F. Makedon, "A hierarchical key-insulated signature scheme in the CA trust model," in *Proceedings of the 7th International Conference on Information Security*, LNCS 3225, Springer-Verlag, 2004, pp. 280-291.
15. N. González-Deleito, O. Markowitch, and E. Dall'Olio, "A new key-insulated signature scheme," in *Proceeding of the 6th International Conference on Information and Communications Security*, LNCS 3269, Springer-Verlag, pp. 465-479.
16. D. H. Yum and P. J. Lee, "Efficient key updating signature schemes based on IBS," in *Proceedings of Cryptography and Coding*, LNCS 2898, Springer-Verlag, 2003, pp. 16-18.
17. J. Liu and D. Wong, "Solutions to key exposure problem in ring signature," http://eprint.iacr.org/2005/427.
18. Y. Zhou, Z. Cao, and Z. Chai, "Identity based key insulated signature," in *Proceedings of Information Security Practice and Experience Conference*, LNCS 3903, Springer-Verlag, 2006, pp. 226-234.
19. J. Weng, S. Liu, K. Chen, and X. Li, "Identity-based key-insulated signature with secure key-updates," in *Proceedings of Information Security and Cryptology*, LNCS 4318, Springer-Verlag, 2006, pp. 13-26.
20. R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited," *Journal of the ACM*, Vol. 51, 2004, pp. 557-594.
21. D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing," in *Proceedings of CRYPTO on Advances in Cryptology*, LNCS 2139, Springer-Verlag, 2001, pp. 213-229.
22. J. Weng, S. Liu, K. Chen, and C. Ma, "Identity-based parallel key-insulated encryption without random oracles: security notions and construction," in *Proceedings of INDOCRYPT*, LNCS 4329, Springer-Verlag, 2006, pp. 409-423.
23. O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *Journal of the ACM*, Vol. 33, 1984, pp. 792-807.
24. K. Paterson and J. Schuldt, "Efficient identity-based signatures secure in the standard model," in *Proceedings of Australasian Conference on Information Security and Privacy*, LNCS 4058, Springer-Verlag, 2006, pp. 207-222.
25. B. Waters, "Efficient identity-based encryption without random oracles," in *Proceedings of EUROCRYPT*, LNCS 3494, Springer-Verlag, 2005, pp. 114-127.
26. D. Naccache, "Secure and practical identity-based encryption," http://eprint.iacr.org/2005/369.
27. P. Sarkar and S. Chatterjee, "Trading time for space: towards an efficient IBE scheme with short(er) public parameters in the standard model," in *Proceedings of*

International Conference on Information Security and Cryptology, LNCS 3935, Springer-Verlag, 2005, pp. 424-440.

Jian Weng (翁健) received his M.S. and B.S. degrees in Computer Science and engineering from South China University of Technology, Guangzhou, P.R. China, in 2004 and 2000, respectively. He is currently a Ph.D. candidate at Shanghai Jiaotong University. His research interests include key-exposure protection mechanism, provable security and pairing-based cryptography.

Xiang-Xue Li (李祥學) received the B.S. degree in Fundamental Mathematics from Nanjing Normal University, Nanjing, in 1997. He received the M.S. degree in Mathematics from Nanjing University, Nanjing, P.R. China, in 2000. He received the Ph.D. degree in Computer Science and Engineering from Shanghai Jiaotong University, Shanghai, in 2006. Since then, he has been with the Department of Computer Science and Engineering of Shanghai Jiaotong University. His areas of research include provable security and pairing-based cryptography.

Ke-Fei Chen (陳克非) received his Ph.D. degree from Justus Liebig University Giessen, Germany, in 1994. Since 1996, he came to Shanghai Jiao Tong University and became the Professor at the Department of Computer Science and Engineering. His areas of research include classical and modern cryptography, theory of network security, *etc*.

Sheng-Li Liu (劉勝利) received her first Ph.D. degree in Xidian University, Xian, P.R. China, in 2000, and received her second Ph.D. degree in Eindhoven University of Technology, Holland, in 2002. Since 2002, she came to Shanghai Jiao Tong University and became the Associate Professor at the Department of Computer Science and Engineering. Her areas of research include cryptography and information security.