International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **7(A)**, July **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 4797–4810

ID-BASED KEY-INSULATED SIGNATURE SCHEME WITH BATCH VERIFICATIONS AND ITS NOVEL APPLICATION

Tsu-Yang Wu, Yuh-Min Tseng* and Ching-Wen Yu

Department of Mathematics
National Changhua University of Education
Jin-De Campus, No. 1, Jin-De Road, Chang-Hua 500, Taiwan
*Corresponding author: ymtseng@cc.ncue.edu.tw

Received March 2011; revised September 2011

Abstract. *For a digital signature scheme, loss of private keys will cause a devastating effect on e-commerce and Internet-based transaction applications in the present network environment. Key-insulated public-key system is introduced to reduce damage caused by private key exposure. Over the last few years, identity (ID)-based key-insulated cryptography using bilinear pairings has received much attention from cryptographic researchers. In this paper, we propose a new and efficient ID-based key-insulated signature scheme with batch verifications. As compared with the recently proposed ID-based key-insulated signature schemes, our scheme has the best performance for batch verifications. For security analysis, we demonstrate that the proposed scheme is a provably secure ID-based key-insulated signature in the random oracle model and under the computational Diffie-Hellman assumption. Meanwhile, to demonstrate the additional advantage of our ID-based key-insulated signature scheme, we present a novel application based on the proposed scheme, called ID-based proxy signature scheme with full delegation and time restriction. This new type of proxy signature scheme provides flexible management for the delegated proxy signers.*
Keywords: Key-insulated, Signature, Batch verification, ID-based, Proxy signature

1. **Introduction.** Exposure of private keys in cryptographic mechanisms (e.g., digital signature and cryptosystem) is the greatest harm to users and means that all security goals are entirely lost. In 2002, Dodis et al. introduced the first key-insulated public-key system [8,9] to solve the problem of regarding cryptographic system injury caused by user's private key exposure. In their key-insulated public-key system, the private keys at discrete various time periods are different and could be stored on an insecure device. Each user must periodically refresh her/his private key through a physically-secure device, named a helper, and the user's public key remains unchanged and fixed throughout the lifetime of the key-insulated public-key system. If an adversary revealed a user's present private key, the adversary is still unable to compute the user's former or later private keys. Hence, the key-insulated public-key system can be used to reduce damage caused by private key exposure.

In 1984, Shamir [17] proposed the first identity (ID)-based public-key cryptosystem. In an ID-based public key system, a user's public key is determined by his/her identity (e.g., name, e-mail address, or social security number). As compared with the traditional certificate-based public-key systems, ID-based public-key system may simplify certificate management. However, Shamir's system is not easy to be realized in practice. In 2001, Boneh and Franklin [3] proposed a practical ID-based cryptosystem from the Weil pairing defined on elliptic curves. Later on, many ID-based cryptographic schemes and protocols from bilinear pairings were proposed in [4,6,7,24,25].

Recently, ID-based key-insulated cryptography using bilinear pairings has received much attention from cryptographic researchers. In 2005, Hanaoka et al. [12] proposed the first ID-based key-insulated cryptosystem from bilinear pairings. Then Zhou et al. [27] presented the first ID-based key-insulated signature scheme. However, Zhou et al.'s signature scheme does not satisfy the strong key-insulated property. This strong key-insulated property means that even if the helper is corrupted by an adversary, the adversary is still unable to compute user's private keys. In 2006, Weng et al. [22] proposed a strong ID-based key-insulated signature scheme. Afterwards, many ID-based key-insulated cryptographic schemes and protocols were proposed such as parallel signature schemes [19,20], encryption schemes [1,21], and parallel encryption schemes [11,23]. In particular, the parallel signature/encryption schemes use two helpers to update the user's private key, alternately.

In the past, many group-oriented signatures or authentications often use the batch verification technique to decrease the computational cost of verification. Various signature schemes with batch verifications [2,4,10,13,18,26] have been presented and these signature schemes allow a verifier to validate several signatures at one time. Until now, to our best knowledge, the related research of ID-based key-insulated signature scheme with batch verifications is not addressed. Two types of batch verifications are defined as follows.

Type 1: A signer signs multiple messages in identical time period.

Type 2: A signer signs multiple messages in the different time periods.

In this paper, we first define the framework and the security model of an ID-based key-insulated signature scheme with batch verifications (IDKISBV). Then, a new and concrete IDKISBV scheme using bilinear pairings is proposed. In the random oracle model and under the computational Diffie-Hellman assumption, we demonstrate that the proposed ID-based key-insulated signature scheme is provably secure and satisfies the strong key-insulated property. As compared with the previously proposed ID-based key-insulated signature schemes, our scheme has the best performance for two types of batch verifications.

Finally, to demonstrate the additional advantage of our ID-based key-insulated signature, we also present a novel application based on our proposed scheme. This new application is an ID-based proxy signature scheme with full delegation and time restriction. As compared with the traditional proxy signature scheme with full-delegation, this new type of proxy signature scheme provides flexible management for the delegated proxy signers. This means that a delegated proxy signer cannot forge valid proxy signatures in other non-delegation time periods. The ID-based proxy signature with full delegation and time restriction is a novel cryptographic application and first presented in this article.

The remainder of this paper is organized as follows. In Section 2, we present the preliminaries of bilinear pairings and the related mathematical assumptions. The framework and the security model of the ID-based key-insulated signature scheme with batch verifications (IDKISBV) are presented in Section 3. In Section 4, we propose a concrete IDKISBV scheme. Security analysis is given in Section 5. In Section 6, we demonstrate the performance analysis and comparisons. A novel application based on our proposed scheme is presented in Section 7. Finally, conclusions are made in Section 8.

2. **Preliminaries.**

In this section, we briefly review the concept of bilinear pairings. Then, we introduce several important security problems and assumptions for bilinear pairings defined on elliptic curves.

2.1. **Bilinear pairings.**

Let G_1 be an additive cyclic group of a large prime order q and G_2 be a multiplicative cyclic group of the same order q. Let P be a generator of the group

G_1. An admissible bilinear pairing is a map e: $G_1 \times G_1 \to G_2$ and satisfies the following properties:

(1) Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$ for all P, $Q \in G_1$ and a, $b \in Z_q^*$.

(2) Non-degenerate: there exist P, $Q \in G_1$ such that $e(P, Q) \neq 1$.

(3) Computability: for all P, $Q \in G_1$, there is an efficient algorithm to compute $e(P, Q)$.

A bilinear map which satisfies the above three properties is called an admissible bilinear map. Such non-degenerate admissible bilinear maps can be obtained from the Weil, Tate or Ate pairings over supersingular elliptic curves or abelian varieties. For the details of bilinear pairings, readers can refer to [3,6] for full descriptions.

2.2. **Mathematical assumption.** To prove the security of our proposed scheme, we summarize some important security problems and assumptions for bilinear pairings defined on elliptic curves as follows:

- Decision Diffie-Hellman (DDH) problem: given P, aP, bP, c$P \in G_1$ for some a, b, $c \in Z_q^*$, it is easy to distinguish $e(aP, bP)$ from $e(P, cP)$, i.e., the DDH problem in G_1 is easy.
- Computational Diffie-Hellman (CDH) problem: given P, aP, $bP \in G_1$ for some a, $b \in Z_q^*$, the CDH problem is to compute $abP \in G_1$.
- CDH assumption: no probabilistic polynomial time (PPT) algorithm with a non-negligible advantage can solve the CDH problem.

We call the group G_1 is a gap Diffie-Hellman group if the DDH problem can be solved in polynomial time, but no probabilistic polynomial time algorithm can solve the CDH problem with a non-negligible advantage in the group G_1.

2.3. **Notations.** Here, we define the following notations that are used through the whole paper:

- e: an admissible bilinear map, e: $G_1 \times G_1 \to G_2$.
- s: the system secret key.
- P_{pub}: the system public key $P_{pub} = s \cdot P$.
- hsk: the helper secret key.
- P_{hlp}: the helper public key $P_{hlp} = hsk \cdot P$.
- ID_u: the identity of a user u.
- $DID_{u,0}$: the user u's initial private key.
- i: a time period i, where $1 \leq i \leq z$ and the whole lifetime of the system is divided into distinct time periods $1, 2, \ldots, z$.
- $HSK_{u,i}$: a user$u$'s helper key in the time period i.
- $DID_{u,i}$: a signer u's private key in the time period i.
- $H_1()$: a hash function H_1: $\{0, 1\}^* \times G_1 \to Z_q^*$.
- $H_G()$: a hash function H_G: $\{0, 1\}^* \to G_1$.
- $H_{GID}()$: a hash function H_{GID}: $\{0, 1\}^* \to G_1$.

3. **Framework and Security Model.** In this section, we define the formal framework and the security model of an ID-based key-insulated signature scheme with batch verifications by extending ID-based key-insulated signature schemes in [9,18,22].

3.1. **Framework.** An ID-based key-insulated signature scheme with batch verifications (IDKISBV) consists of six polynomial-time algorithms: *System Setup*, *Initial Private Key Generating*, *Key Update*, *Signing*, *Verifying*, and *Batch Verifying algorithms*.

***System Setup algorithm*:** A probabilistic algorithm takes a security parameter l as input. It returns public parameters, a helper secret key hsk, and a system secret key s.

***Initial Private Key Generating algorithm*:** Taking input public parameters, the master secret key s, and a user's identity ID_u, it returns the user's initial private key $DID_{u,0}$, where 0 denotes the initial time period.

***Key Update algorithm*:** This algorithm consists of two deterministic algorithms:

- **Helper Key Update algorithm.** Inputting a time period i, the helper secret key *hsk*, and a user's identity ID_u, it returns the user's helper key $HSK_{u,i}$ for the time period i.
- **Private Key Update algorithm.** This algorithm is a deterministic algorithm which takes input a time period i, a helper key $HSK_{u,i}$, and a private key $DID_{u,i-1}$ of the time period $i-1$. It returns a private key $DID_{u,i}$ for the time period i.

***Signing algorithm*:** A probabilistic algorithm takes input a message m, a time period i, public parameters, and a signer's private key $DID_{u,i}$. It returns a signature σ.

***Verifying algorithm*:** A deterministic verification algorithm takes input a message m, public parameters, the signer's identity, and a signature σ. It returns "1", if σ is valid. Otherwise, it returns "0".

***Batch Verifying algorithm*:** The verifier may use this algorithm to verify a k-batch signature including n signatures $\{(\mathrm{ID}_u, m_j, t_j, \sigma_j) \mid j = 1, 2, \ldots, n \text{ and } n \leq k\}$, where t_j are valid time periods. It returns "1" if n signatures are valid. Otherwise, it returns "0".

3.2. **Security model of IDKISBV scheme.** Here, we define the security model of an ID-based key-insulated signature scheme with batch verifications (IDKISBV). In particular, the strong key-insulated property means that an adversary can get the entire helper key information in some time period from oracle.

Definition 3.1. *An ID-based key-insulated signature scheme provides existential unforgeability and strongly key-insulated property against adaptive chosen-message attacks and ID attacks, if no probabilistic polynomial-time adversary A has a non-negligible advantage in the following game played between a challenger C and the adversary A.*

(1) ***Initialization.*** The challenger C takes a security parameter l and runs the *System Setup algorithm* to generate public parameters, a system secret key s, and a helper secret key *hsk*. Then, C sends the public parameters to the adversary A.

(2) ***Queries.*** The adversary A can issue a series of queries as follows.

- **Initial key query (ID_u).** In this query, the adversary A sends an identity ID_u to the challenger C. Then, C runs the *Initial Private Key Generating algorithm* to generate an initial private key $DID_{u,0}$ corresponding to ID_u and sends it to the adversary A.
- **Helper key query (ID_u, i).** In this query, the adversary A sends a pair (ID_u, i) to the challenger C, where i is a time period. Then, C runs the *Helper Key Generating algorithm* to generate a helper key $HSK_{u,i}$ for the time period i corresponding to ID_u and sends it to the adversary A.
- **Signing query (ID_u, m, i).** The adversary A sends a tuple (ID_u, m, i) to the challenger C. Then, C runs the *Signing algorithm* to generate a signature σ and sends it to the adversary A.

(3) ***Forgery.*** Finally, the adversary A generates a tuple $(\mathrm{ID}_u^*, m^*, i^*, \sigma^*)$. We say that the adversary A wins the game, if the verification algorithm outputs "1" and the following conditions hold:

(i) The identity ID_u^* did not appear in the initial key query.

(ii) The tuple $(\mathrm{ID}_u^*, m^*, i^*)$ did not appear in the signing query.
The advantage of the adversary A is defined as the probability that A wins the game.

Definition 3.2. *An ID-based key-insulated signature scheme with batch verifications provides k-batch existential unforgeability and strongly key-insulated property against adaptive chosen-message attacks and ID attacks, if no probabilistic polynomial-time adversary A has a non-negligible advantage in the following game played between a challenger C and the adversary A.*

(1) ***Initialization.*** The phase is the same as one defined in Definition 3.1.
(2) ***Queries.*** Queries issued by the adversary A are also same as ones in Definition 3.1.
(3) ***k-batch forgery.*** For some integer k, the adversary A outputs n signatures $\{(\mathrm{ID}_u^*, m_j^*, t_j^*, \sigma_j^*) | j = 1, 2, \ldots, n \text{ and } n \leq k\}$, where t_j^* are valid time periods. We say that the adversary A wins the game, if the verification algorithm outputs "1" and the following conditions hold:
(i) The identity ID_u^* did not appear in the Initial key query.
(ii) There exists at least one tuple $(\mathrm{ID}_u^*, m_j^*, t_j^*)$ did not appear in the Signing query.
The advantage of the adversary A is defined as the probability that A wins the game.

4. **Proposed Scheme.** As mentioned in Subsection 3.1, our proposed scheme consists of six phases that include the system setup phase, the key generation phase, the key update phase, the signing phase, the verifying phase, and the batch verifying phase. We describe them in details as follows:

[System Setup phase]

A Key Generation Center (KGC) takes input a security parameter l to generate all required parameters and functions. Firstly, the KGC selects a bilinear map e: $G_1 \times G_1 \to G_2$, where G_1 is a subgroup of additive cyclic group of a prime order q and G_2 is a subgroup of multiplicative cyclic group with the same order q. Then, the KGC chooses a random number $s \in Z_q^*$ keeping as the system secret key and computes $P_{pub} = s \cdot P$ as the system public key, where P is a generator of the group G_1. Meanwhile, the KGC randomly chooses a helper secret key $hsk \in Z_q^*$ and computes the helper public key $P_{hlp} = hsk \cdot P$. Finally, the KGC picks two hash functions $H_1 : \{0,1\}^* \times G_1 \to Z_q^*$, $H_G : \{0,1\}^* \to G_1$ and $H_{GID} : \{0,1\}^* \to G_1$. The public parameters and functions are defined as Params $= \{e, G_1, G_2, q, P_{pub}, P, P_{hlp}, H_1, H_G, H_{GID}\}$.

[Key Generation phase]

When a user submits her/his identity ID_u to the KGC, the KGC first computes $DID_{u,0} = s \cdot H_{GID}(\mathrm{ID}_u) + hsk \cdot H_G(\mathrm{ID}_u, 0)$. Then, the KGC sends $DID_{u,0}$ to the user as her/his initial private key and hsk to the user's helper as the helper secret key via a secure channel.

[Key Update phase]

At start of the time period i, the user's helper computes a helper key $HSK_{u,i} = hsk \cdot [H_G(\mathrm{ID}_u, i) - H_G(\mathrm{ID}_u, i-1)]$ and sends it to the user. Then, the user updates her/his private key $DID_{u,i}$ for time period i by $DID_{u,i} = HSK_{u,i} + DID_{u,i-1}$. Finally, the user erases two values $HSK_{u,i}$ and $DID_{u,i-1}$.

[Signing phase]

Given a message m, the signer ID_u first chooses a random number $r \in Z_q^*$, and then computes $U_1 = r \cdot H_{GID}(\mathrm{ID}_u)$, $U_2 = r \cdot H_G(\mathrm{ID}_u, i)$, $h = H_1(m, U_1, U_2, i)$, and $V = (r + h) \cdot DID_{u,i}$. The tuple (U_1, U_2, V) is a signature σ on the message m for the time period i.

[Verifying phase]

Any verifier can verify the signature tuple $(\mathrm{ID}_u, m, i, \sigma = (U_1, U_2, V))$ as follows. The verifier computes $h = H_1(m, U_1, U_2, i)$ and then checks $e(P, V) = e(P_{pub}, U_1 + h \cdot H_{GID}(\mathrm{ID}_u)) \cdot e(P_{hlp}, U_2 + h \cdot H_G(\mathrm{ID}_u, i))$. It returns "1", if σ is a valid signature. Otherwise, it returns "0".

[Batch Verifying phase of Type 1]

For batch verifications of Type 1, the verifier can use the Batch Verifying algorithm to verify a k-batch signature $\{(\mathrm{ID}_u, m_j, t_j, \sigma_j) | j = 1, 2, \ldots, n \text{ and } n \leq k\}$, where $t_1 = t_2 = \cdots = t_n$ is a valid time period. The verifier computes $h_j = H_1(m_j, U_{1,j}, U_{2,j}, t_1)$, then checks

$$e\left(P, \sum_{j=1}^{n} V_j\right) = e\left(P_{pub}, \sum_{j=1}^{n} U_{1,j} + \sum_{j=1}^{n} h_j \cdot H_{GID}(ID_u)\right) \cdot e\left(P_{hlp}, \sum_{j=1}^{n} U_{2,j} + \left(\sum_{j=1}^{n} h_j\right) \cdot H_G(ID_u, t_1)\right)$$

It returns "1", if n signatures are valid. Otherwise, it returns "0".

[Batch Verifying phase of Type 2]

For batch verifications of Type 2, the verifier can use the Batch Verifying algorithm to verify a k-batch signature including n signatures $\{(\mathrm{ID}_u, m_j, t_j, \sigma_j) | j = 1, 2, \ldots, n \text{ and } n \leq k\}$, where t_j are valid time periods. The verifier computes $h_j = H_1(m_j, U_{1,j}, U_{2,j}, t_j)$ and checks

$$e\left(P, \sum_{j=1}^{n} V_j\right) = e\left(P_{pub}, \sum_{j=1}^{n} U_{1,j} + \sum_{j=1}^{n} h_j \cdot H_{GID}(ID_u)\right) \cdot e\left(P_{hlp}, \sum_{j=1}^{n} U_{2,j} + \sum_{j=1}^{n} [h_j \cdot H_G(ID_u, t_j)]\right).$$

It returns "1", if n signatures are valid. Otherwise, it returns "0".

5. **Security Analysis.** In this section, we show that the proposed ID-based key-insulated signature scheme with batch verifications provides existential unforgeability and k-batch existential unforgeability, and satisfies the strong key-insulated properties to adaptive chosen message attacks and ID attacks in the random oracle model and under the computational Diffie-Hellman (CDH) assumption.

[Single signature]

Firstly, we demonstrate the security of our proposed ID-based key-insulated signature scheme for single signature.

Theorem 5.1. *In the random oracle model, assume that a probabilistic polynomial-time adversary A can break the proposed ID-based key-insulated signature scheme under adaptive chosen message attacks and ID attacks with an advantage* $\varepsilon_0 \geq \frac{10(q_s+1)(q_s+q_{H1})q_{HGID}}{q-1}$ *and within running time* t_0*. Here, A may make the Initial key, Helper key,* H_1*,* H_G*,* H_{GID}*, and Sign queries at most* q_I*,* q_{Hlp}*,* q_{H1}*,* q_{HG}*,* q_{HGID}*, and* q_S *times, respectively. Then, there exists a probabilistic polynomial-time algorithm C with a non-negligible advantage* $\varepsilon_2 \geq \frac{1}{9}$ *that can solve the computational Diffie-Hellman problem within a running time* $t_2 \leq \frac{23 \cdot q_{H1} \cdot q \cdot t_0 \cdot q_{HGID}}{\varepsilon_0(1-1/q)}$.

Proof: In the random oracle model, assume that A is an algorithm within running time t_0 and with advantage ε_0 to perform an adaptive chosen message attack and an ID-attack to our scheme. Using Lemma 1 in [4], it implies that there is an algorithm $B$for an adaptive chosen message attack and given fixed ID-attack which has running time $t_1 \leq t_0$ and advantage $\varepsilon_1 \leq \varepsilon_0(1 - 1/q)/q_{HGID}$, where q_{HGID} is the maximum number of oracle queries to H_{GID} hash function asked by A. Without loss of generality, we refer the given fixed ID to the identity ID_U of the user U.

If there exists the above algorithm B with a non-negligible advantage ε_1, it implies that an adversary C without knowing the secret key $DID_{u,i}$ of U can use B to solve the CDH problem. We assume that the algorithm C receives a random instance (P, aP, bP) in G_1 for some unknown a, $b \in Z_q^*$ and C wants to compute the value abP. Here, C acts as a challenger in the game (defined in Definition 3.1). At beginning of the game, the challenger C generates the public parameters $\{e, G_1, G_2, P, P_{pub}, P_{hlp}\}$, where $P_{pub} = a \cdot P$ and $P_{hlp} = hsk \cdot P$. Here, a denotes the system secret key and is unknown to the challenger C. Then, C sends the public parameters to the adversary B. In addition, C needs to maintain six lists L_j, where $j \in \{1, 2, 3, 4, 5, 6\}$ that are initially empty and are used to keep track of answers to the following queries. C is responsible to answer the different queries made by B as below:

- H_{GID} query (ID_u). The adversary B sends a pair (ID_u, i) to the challenger C. Then, C randomly selects a value $x \in Z_q^*$ and computes the hash value $H_G(\mathrm{ID}_u) = x \cdot P \in G_1$. Finally, the challenger C sends xP to the adversary B. The challenger C adds (ID_u, x, xP) into the list L_1.
- H_G query (ID_u, i). Upon receiving this query with (ID_u, i), the challenger C chooses a random value $w \in Z_q^*$ and computes the hash value $w \cdot P \in G_1$ of H_G. Then, the challenger C sends wP to the adversary B and adds (ID_u, i, w, wP) into the list L_2.
- Initial key query (ID_u). When the adversary B issues this query on an identity ID_u to the challenger C, the challenger C accesses the corresponding tuple (ID_u, x, xP) from the list L_1. Then C returns the ID_u's initial private key $DID_{u,0} = x \cdot P_{pub}$ to B and adds ($\mathrm{ID}_u, DID_{u,0}$) into the list L_3.
- Helper key query (ID_u, i). Upon receiving this query with (ID_u, i), the challenger C accesses the corresponding tuple (ID_u, i, w, wP) from the list L_2. Then the challenger C returns a helper key $HSK_{u,i} = w \cdot P_{hlp}$ for the time period i to B and adds ($\mathrm{ID}_u, i, HSK_{u,i}$) into the list L_4.
- H_1 query (m, U_1, U_2, i). When the adversary B issues this query on a tuple (m, U_1, U_2, i) to the challenger C, then C computes the hash value R_H of H_1 on the requested input and sends it to B. Finally, the challenger C adds (m, U_1, U_2, i, R_H) into the list L_5.
- Signing query (ID_u, m, i). When the adversary B issues this query on a tuple (ID_u, m, i) to the challenger C, then C returns a signature σ to the adversary B and adds ($\mathrm{ID}_u, m, i, \sigma$) into the list L_6.

Assume that B can output a valid signature tuple (ID_U, m', i', $\sigma' = (U_1', U_2', V')$) with a non-negligible advantage. The challenger C first checks whether the pair (ID_U, i') appeared in both the list L_2 and in the list L_1. If they are not, the challenger C aborts it. Following the Forking Lemma in [16], this lemma adopts the "oracle replay attack" using a polynomial replay of the attack with the same random tape and a different oracle. If there is an algorithm B with a non-negligible probability ε_1 to generate a valid signature $\sigma' = (U_1', U_2', V')$ for the message (ID_U, m', i'), then the algorithm B can generate two valid message signatures (ID_U, m', i', $\sigma' = (U_1', U_2', V')$) and ($\mathrm{ID}_U, m', i', \sigma'' = (U_1', U_2', V'')$) with a non-negligible probability at least $\varepsilon_1/2$ such that

$$e(P, V') = e(P_{pub}, U_1 + h' \cdot H_{GID}(\mathrm{ID}_U)) \cdot e(P_{hlp}, U_2 + h' \cdot H_G(\mathrm{ID}_U, i'))$$

and

$$e(P, V'') = e(P_{pub}, U_1 + h'' \cdot H_{GID}(\mathrm{ID}_U)) \cdot e(P_{hlp}, U_2 + h'' \cdot H_G(\mathrm{ID}_U, i')),$$

where $h' \neq h''$ are two hash values from H_1 query.

Let $H_{GID}(\mathrm{ID}_U) = b \cdot P$. By the bilinear pairing properties, we obtain

$$e(P, V' - V'') = e(aP, (h' - h'') \cdot bP) \cdot e(P_{hlp}, (h' - h'') \cdot H_G(\mathrm{ID}_U, i')).$$

And it implies

$$e(P, (V' - V'') - (h' - h'') \cdot abP) = e(P_{hlp}, (h' - h'') \cdot H_G(\mathrm{ID}_U, i'))$$

with $H_G(\mathrm{ID}_U, i) = w \cdot P$ for some know $w \in Z_q^*$. Hence we have

$$e(P, (V' - V'') - (h' - h'') \cdot abP) = e(P_{hlp}, (h' - h'') \cdot wP).$$

The adversary B can easily obtain the value $abP = [(V' - V'') - (h' - h'') \cdot wP_{hlp}]/(h' - h'')$. Thus, the challenger C can run the adversary B as a subroutine to obtain the value abP with the probability $\varepsilon_2 \geq \frac{1}{9}$ and within the running time $t_2 \leq \frac{23 \cdot q_{H1} \cdot t_1}{\varepsilon_1}$ from a random instance (P, aP, bP). For these values ε_1, ε_2 and t_2, readers can refer to [17, Lemma 4]. This is a contradiction for the computational Diffie-Hellman (CDH) assumption.

[Multiple signatures]

Now, we focus on the security of the presented ID-based key-insulated signature scheme with batch verifications for multiple signatures. To prove the security of a k-batch ID-based key-insulated signature scheme, it must offer k-batch existential unforgeability to adaptive chosen message attacks and ID attacks. Since the batch verifications for Type 1 and Type 2 respectively are used to verify multiple messages for an identical time period and the multiple time periods, we can use [4, Lemma 1] to reduce this problem to the variant: the k-batch ID-based key-insulated signature scheme offers k-batch existential unforgeability to adaptive chosen message and given ID attacks.

Theorem 5.2. *In the random oracle model, assume that an adversary A can break the k-batch signature of Type 1 in the proposed scheme under the adaptive chosen message attacks and ID attacks with a non-negligible advantage $\varepsilon_0 \geq \frac{(12 \cdot Vq_{H1,k} + 6(q_{H1} + k \cdot q_S)^2) q_{HGID}}{q-1}$ and within running time t_0, where $Vq_{H1,k}$ denotes the k times the number of k-permutations of q_{H1} elements. And, A may make the Initial key, Helper key, H_1, H_G, H_{GID}, and Sign queries at most q_I, q_{Hlp}, q_{H1}, q_{HG}, q_{HGID} and q_S times, respectively. Then, there exists an adversary C to solve the computational Diffie-Hellman (CDH) problem with the advantage $\varepsilon_2 \geq \frac{1}{9}$ and within the running time $t_2 \leq \frac{144823 \cdot Vq_{H1,k} \cdot (1 + q_S) \cdot q_{HGID} \cdot t_0}{\varepsilon_0 (1 - 1/q)}$.*

Proof: Assume that there exists an algorithm A that can forge a k-batch signature of Type 1 for the adaptive chosen message attacks and ID attacks, with an advantage ε_0, within running time t_0. Using [4, Lemma 1], we may construct another algorithm B that can forge a k-batch signature of Type 1 for the adaptive chosen message and given ID attacks with the advantage $\varepsilon_1 \leq \varepsilon_0(1 - 1/q)/q_{HGID}$ and within the running time $t_1 \leq t_0$.

Now, we want to construct an algorithm C to solve the CDH problem using B. We assume that the algorithm C receives a random instance (P, aP, bP) in G_1 for unknown $a, b \in Z_q^*$ and C wants to compute the value abP. Here, C acts as a challenger in the game (defined in Definition 3.1). At beginning of the game, the challenger C generates the public parameters $\{e, G_1, G_2, P, P_{pub}, P_{hlp}\}$ and sends them to the adversary B, where $P_{pub} = a \cdot P$ and $P_{hlp} = hsk \cdot P$. Then, the challenger C is responsible to answer Initial key, Helper key, H_1, H_G, H_{GID}, and Signing queries issued by the adversary B in the same way as Theorem 5.1. If the algorithm C does not fail, B outputs a k-batch signature of Type 1 $\{(\mathrm{ID}'_u, m'_j, t'_j, \sigma'_j) | j = 1, 2, \ldots, n \text{ and } n \leq k\}$ with the non-negligible advantage ε_1.

Using [26, Lemma 1], assume B can generate a k-batch signature of Type 1 $\{(\mathrm{ID}'_u, m'_j, t'_j, \sigma'_j) \mid j = 1, 2, \ldots, n \text{ and } n \leq k\}$ with the advantage $\varepsilon_1 \geq \frac{12 \cdot Vq_{H1,k} + 6(q_{H1} + k \cdot q_S)^2}{q}$ and within the running t_1, where $Vq_{H1,k}$ denotes the k times the number of k-permutations of q_{H1} elements, i.e., $Vq_{H1,k} = k \cdot q_{H1}(q_{H1} - 1) \cdots (q_{H1} - k + 1)$. Then, there is another probabilistic polynomial-time adversary B' which has controlled over the machine obtained from B by simulation. In this case, it can generate the other set of multiple signatures

$\{(\mathrm{ID}'_u, m'_j, t'_j, \sigma'_j)|j = 1, 2, \ldots, n\}$ and $\{(\mathrm{ID}'_u, m''_j, t'_j, \sigma''_j)\,|j = 1, 2, \ldots, n\}$ such that hash values $h'_l \neq h''_l$ for some $l \in \{1, 2, \ldots, n\}$ and $h'_j = h''_j$ for all $j = 1, 2, \ldots, n$, excepting for $j = l$ within the running time $t' = \frac{144823 \cdot Vq_{H1,k} \cdot (1+q_S) \cdot t_0}{\varepsilon_0}$. That is, we have

$$e\left(P, \sum_{j=1}^{n} V'_j\right) = e\left(P_{pub}, \sum_{j=1}^{n} U_{1,j} + \sum_{j=1}^{n} h'_j \cdot H_{GID}(\mathrm{ID}'_u)\right) \cdot e\left(P_{hlp}, \sum_{j=1}^{n} U_{2,j} + \sum_{j=1}^{n} h'_j \cdot H_G(\mathrm{ID}'_u, t'_1)\right)$$

and

$$e\left(P, \sum_{j=1}^{n} V''_j\right) = e\left(P_{pub}, \sum_{j=1}^{n} U_{1,j} + \sum_{j=1}^{n} h''_j \cdot H_{GID}(\mathrm{ID}'_u)\right) \cdot e\left(P_{hlp}, \sum_{j=1}^{n} U_{2,j} + \sum_{j=1}^{n} h''_j \cdot H_G(\mathrm{ID}'_u, t'_1)\right).$$

Let $H_{GID}(\mathrm{ID}'_u) = b \cdot P$. By the bilinear pairing properties, we obtain

$$e\left(P, \sum_{j=1}^{n} V'_j - \sum_{j=1}^{n} V''_j\right) = e\left(aP, \left(\sum_{j=1}^{n} h'_j - \sum_{j=1}^{n} h''_j\right) \cdot bP\right) \cdot e\left(P_{hlp}, \left(\sum_{j=1}^{n} h'_j - \sum_{j=1}^{n} h''_j\right) \cdot H_G(\mathrm{ID}'_u, t'_1)\right)$$

And it implies

$$e\left(P, \sum_{j=1}^{n} V'_j - \sum_{j=1}^{n} V''_j\right) = e\left(P, (h'_l - h''_l) \cdot abP\right) \cdot e\left(P_{hlp}, (h'_l - h'') \cdot H_G(\mathrm{ID}'_u, t'_1)\right)$$

with $H_G(\mathrm{ID}'_u, t'_1) = w \cdot P$ for some know $w \in Z_q^*$. Hence we have

$$e\left(P, \sum_{j=1}^{n} V'_j - \sum_{j=1}^{n} V''_j\right) = e(P, (h'_l - h'') \cdot abP) \cdot e(P_{hlp}, (h'_l - h'') \cdot wP).$$

The adversary C can easily obtain the value $abP = \left[\sum_{j=1}^{n} (V'_j - V''_j) - (h'_l - h'') \cdot wP_{hlp}\right] / (h'_l - h'')$ with $\varepsilon_2 \geq \frac{1}{9}$ and within $t_2 \leq \frac{144823 \cdot Vq_{H1,k} \cdot (1+q_S) \cdot q_{HGID} \cdot t_0}{\varepsilon_0(1-1/q)}$.

Theorem 5.3. *In the random oracle model, assume that an adversary A can break the k-batch signature of Type 2 in the proposed scheme under the adaptive chosen message attacks and ID attacks with a non-negligible advantage $\varepsilon_0 \geq \frac{(12 \cdot Vq_{H1,k} + 6(q_{H1} + k \cdot q_S)^2) q_{HGID}}{q-1}$ and within running time t_0, where $Vq_{H1,k}$ denotes the k times the number of k-permutations of q_{H1} elements. Here, A may make the Initial key, Helper key, H_1, H_G, H_{GID}, and Sign queries at most q_I, q_{Hlp}, q_{H1}, q_{HG}, q_{HGID}, and q_S times, respectively. Then, there exists an adversary C to solve the computational Diffie-Hellman problem with the advantage $\varepsilon_2 \geq \frac{1}{9}$ and within the running time $t_2 \leq \frac{144823 \cdot Vq_{H1,k} \cdot (1+q_S) \cdot q_{HGID} \cdot t_0}{\varepsilon_0(1-1/q)}$.*

Proof: Assume that there exists an algorithm A that can forge a k-batch signature of Type 2 for the adaptive chosen message attacks and ID attacks, with an advantage ε_0, within running time t_0. Using [4, Lemma 1], we may construct another algorithm B that can forge a k-batch signature of Type 2 for the adaptive chosen message attacks and given ID attacks with the advantage $\varepsilon_1 \leq \varepsilon_0(1 - 1/q)/q_{HGID}$ and within the running time $t_1 \leq t_0$.

Now, we want to construct an algorithm C to solve the CDH problem using B. We assume that the algorithm C receives a random instance (P, aP, bP) in G_1 for unknown a, $b \in Z_q^*$ and C wants to compute the value abP. Here, C acts as a challenger in the game (defined in Definition 3.1). At beginning of the game, the challenger C generates the public parameters $\{e, G_1, G_2, P, P_{pub}, P_{hlp}\}$ and sends them to the adversary B, where $P_{pub} = a \cdot P$ and $P_{hlp} = hsk \cdot P$. Then, the challenger C is responsible to answer Initial key,

Helper key, H_1, H_G, H_{GID}, and Signing queries issued by the adversary B in the same way as Theorem 5.1. If the algorithm C does not fail, B outputs a k-batch signature of Type 2 $\{(\mathrm{ID}'_u, m'_j, t'_j, \sigma'_j)|j = 1, 2, \ldots, n \text{ and } n \leq k\}$ with the non-negligible advantage ε_1.

Using [26, Lemma 1], assume B can generate a k-batch signature of Type 2 $\{(\mathrm{ID}'_u, m'_j, t'_j, \sigma'_j)|j = 1, 2, \ldots, n \text{ and } n \leq k\}$ with the advantage $\varepsilon_1 \geq \frac{12 \cdot Vq_{H1,k} + 6(q_{H1} + k \cdot q_S)^2}{q}$ and within the running t_1, where $Vq_{H1,k}$ denotes the k times the number of k-permutations of q_{H1} elements, i.e., $Vq_{H1,k} = k \cdot q_{H1}(q_{H1} - 1) \cdots (q_{H1} - k + 1)$. Then, there is another probabilistic polynomial-time adversary B' which has controlled over the machine obtained from B by simulation. And it can generate the other set of multiple signatures $\{(\mathrm{ID}'_u, m'_j, t'_j, \sigma'_j)|j = 1, 2, \ldots, n\}$ and $\{(\mathrm{ID}'_u, m''_j, t'_j, \sigma''_j)|j = 1, 2, \ldots, n\}$ such that the hash values $h'_l \neq h''_l$ for some $l \in \{1, 2, \ldots, n\}$ and $h'_j = h''_j$ for all $j = 1, 2, \ldots, n$, excepting for $j = l$ within the running time $t' = \frac{144823 \cdot Vq_{H1,k} \cdot (1 + q_S) \cdot t_0}{\varepsilon_0}$. Without loss of generality, we assume that $l = 1$, that is, we have $e\left(P, \sum_{j=1}^{n} (V'_j - V''_j)\right) = e\left(P_{pub}, (h'_1 - h''_1) \cdot H_{GID}(ID'_u)\right) \cdot e\left(P_{hlp}, (h'_1 - h''_1) \cdot H_G(ID'_u, t'_1)\right)$.

Let $H_{GID}(\mathrm{ID}'_u) = b \cdot P$. By the bilinear pairing properties, we obtain

$$e\left(P, \sum_{j=1}^{n} (V'_j - V''_j)\right) = e(aP, (h'_1 - h''_1) \cdot bP) \cdot e(P_{hlp}, (h'_1 - h''_1) \cdot H_G(\mathrm{ID}'_u, t'_1)).$$

And it implies

$$e\left(P, \sum_{j=1}^{n} (V'_j - V''_j)\right) = e(P, (h'_1 - h''_1) \cdot abP) \cdot e(P_{hlp}, (h'_1 - h''_1) \cdot H_G(\mathrm{ID}'_u, t'_1)$$

with $H_G(\mathrm{ID}'_u, t'_1) = w \cdot P$ for some know $w \in Z_q^*$. Hence we have

$$e\left(P, \sum_{j=1}^{n} (V'_j - V''_j)\right) = e(P, (h'_1 - h''_1) \cdot abP) \cdot e(P_{hlp}, (h'_1 - h''_1) \cdot wP).$$

The adversary C can easily obtain the value $abP = \left[\sum_{j=1}^{n} (V'_j - V''_j) - (h_1 - h'_1) \cdot wP_{hlp}\right] \Big/ (h_1 - h'_1)$ with the advantage $\varepsilon_2 \geq \frac{1}{9}$ and within the running time $t_2 \leq \frac{144823 \cdot Vq_{H1,k} \cdot (1 + q_S) \cdot q_{HG} \cdot t_0}{\varepsilon_0 (1 - 1/q)}$.

6. **Performance Analysis and Comparisons.** For convenience to evaluate the computational cost, we consider the following time-consuming operations in pairing-based cryptography:

- TG_e: The time of executing the bilinear map operation e.
- TG_{mul}: The time of executing the scalar multiplication of points operation.
- TG_H: The time of executing the hash function $H_G()$ or $H_{GID}()$.

Considering the computational cost of our proposed scheme, in the *Key Update phase*, it requires $TG_{mul} + 2TG_H$. In the *Signing phase*, $3TG_{mul} + TG_H$ is required to compute (U_1, U_2, V). In the *Verifying phase*, it requires $3TG_e + 2TG_{mul} + 2TG_H$ for verifying $(\mathrm{ID}_u, m, \sigma = (i, U_1, U_2, V))$. As a result, it totally requires $3TG_e + 6TG_{mul} + 5TG_H$. More importantly, the batch verifications (BV) of Types 1 and 2 only require constant bilinear pairing operations, $3TG_e$. In Table 1, we demonstrate the comparisons between our scheme and the previously proposed ID-based key-insulated signature schemes [22,27] in terms of the computational costs and security property. It is easy to see that our

scheme has better performance for batch verifications and satisfies strong key-insulated (KI) property.

TABLE 1. Comparisons between our scheme and the previously proposed schemes

	Zhou et al.'s scheme [29]	Weng et al.'s scheme [23]	Our scheme
Key Update	$2TG_{mul} + 2TG_H$	$2TG_{mul} + 2TG_H$	$TG_{mul} + 2TG_H$
Signing	$2TG_{mul} + TG_H$	$2TG_{mul} + TG_H$	$3TG_{mul} + TG_H$
Verifying	$4TG_e + 3TG_H$	$4TG_e + 3TG_H$	$3TG_e + 2TG_{mul} + 2TG_H$
BV of Type 1	$(n+3)TG_e + (n+2)TG_H$	$(n+3)TG_e + (n+2)TG_H$	$3TG_e + 2TG_{mul} + 2TG_H$
BV of Type 2	$(2n+2)TG_e + (2n+1)TG_H$	$(n+3)TG_e + (2n+1)TG_H$	$3TG_e + (n+1)TG_{mul}$ $+(n+1)TG_H$
Strong KI	No	Yes	Yes

7. **A Novel Application.** Proxy signature [5,14,15] allow that an original signer delegates her/his signing capability to a designated proxy signer. Generally, there are four kinds of delegation called *full delegation*, *partial delegation*, *delegation by warrant*, and *partial delegation with warrant* were defined in [14,15]. In this section, we use our ID-based key-insulated signature scheme to present a novel ID-based proxy signature with full delegation and time restriction. Firstly, we briefly review the four types of delegation.

- **Full delegation:** The original signer gives her/his private key to the proxy signer as the signing key, completely. Hence, the original signature generated by the original signer and the proxy signature generated by the proxy signer are indistinguishable.
- **Partial delegation:** In this type, the original signer uses her/his private key to create a proxy signing key and sends it to the proxy signer. Hence, the original signature generated by the original signer and the proxy signature generated by the proxy signer are distinguishable.
- **Delegation by warrant:** The original signer first gives a warrant to the proxy signer. Then, the proxy signer generates a proxy signature σ using his private key and sends σ with the warrant to verifiers. The verifiers can verify whether the proxy signer is legal or not by the warrant.
- **Partial delegation with warrant:** This type possesses the properties of both the partial delegation and the delegation by warrant.

Remark 7.1. *For the above proxy signature scheme with full delegation, it is easy to see that the proxy signer may forge the original signer's signature willfully. Furthermore, any verifier is unable to distinguish the produced signature from the original signer or the proxy signer.*

By the remark, we have known that the traditional proxy signature scheme with full delegation has a security limitation. Thus, we define a new type of full delegation with time restriction as follows.

Definition 7.1. Full delegation with time restriction. *In a proxy signature scheme with full delegation and time restriction, an original signer directly gives his private key to a designated proxy signer as the signing key, where private key is a temporary key at the delegation time period. Then, the proxy signer can use the key to generate a proxy signature on behalf of the original signer for the delegation time period. Otherwise, the proxy signer cannot generate a proxy signature on behalf of the original signer.*

Here, we present an ID-based proxy signature scheme with full delegation and time restriction based on our scheme as follows. The proposed proxy signature scheme is a new type of proxy signature scheme. Until now, no such a proxy signature scheme is

proposed. Assume that an original signer wants to delegate her/his signing capability to a designated proxy signer in a time period i.

(1) The original signer first requests her/his helper to obtain a helper key $HSK_{u,i}$ and computes her/his private key $DID_{u,i}$ for the time period i.
(2) The original signer sends $DID_{u,i}$ to the proxy signer directly via a secure channel.
(3) In the delegation time period i, the proxy signer can produce a proxy signature $\sigma = (i, U_1, U_2, V)$ on behalf of the original signer.
(4) Any verifier can check the validity of σ by a proxy signature verification algorithm. Note that the verification algorithm is the same as the verifying algorithm described in Section 4.

It is easy to see that the proxy signer is unable to forge the original signer's signature for the time period $j \neq i$. Because the original signer's private key $DID_{u,j}$ is different to $DID_{u,i}$, the original signer uses the key $DID_{u,j}$ to generate a signature for the time period j. Therefore, the security of our proposed ID-based proxy signature scheme with full delegation and time restriction is better than one of the traditional ID-based proxy signature scheme with full delegation. In Theorem 7.1, we present the security of our ID-based proxy signature with full delegation and time restriction.

Theorem 7.1. *In the random oracle model and the CDH assumption, the proposed ID-based proxy signature scheme with full delegation and time restriction provides existential unforgeability under adaptive chosen-message attacks and ID attacks.*

Proof: Here, an adversary is able to get the target identity's private key for the time period $j-1$, $DID_{u,j-1}$, but he/she is unable to get the target identity's helper key for the time period j, $HSK_{u,j}$. Thus, he cannot directly compute the target identity's $DID_{u,j}$ by $DID_{u,j-1}$ and $HSK_{u,j}$. If the adversary can forge a valid proxy signature, then the computational Diffie-Hellman (CDH) problem can be solved by the similar method of Theorem 5.1.

8. **Conclusions.** In this paper, we have defined the framework and security model of an ID-based key-insulated signature scheme with batch verifications (IDKISBV). Meanwhile, we proposed a concrete ID-based key-insulated signature scheme with batch verifications (IDKISBV) from bilinear pairing. Performance analysis is given to demonstrate that our scheme requires only constant bilinear pairing operations for batch verifications. In summary, Our scheme provides the following merits. (1) Simplifying certificate management of users' public keys, with compared to certificated-based public key systems. (2) Providing existential unforgeability and k-batch existential unforgeability against adaptive chosen message and ID attacks. (3) Offering the best performance as compared to the previously proposed schemes. (4) Providing a novel application based on the proposed scheme, called ID-based proxy signature scheme with full delegation and time restriction, which provides flexible management for the delegated proxy signers. In the random oracle model and under the computational Diffie-Hellman assumption, we have shown that our scheme is provably secure and satisfies strong key-insulated property.

Acknowledgements. The authors would like to thank the anonymous referees for their valuable comments and constructive suggestions. This research was partially supported by National Science Council, Taiwan, under contract No. NSC100-2221-E-018-027.

REFERENCES

[1] M. Bellare and A. Palacio, Protecting against key-exposure: Strongly key-insulated encryption with optimal threshold, *Applicable Algebra in Engineering, Communication and Computing*, vol.16, pp.379-396, 2006.

[2] M. Bellare, J. Garay and T. Rabin, Fast batch verification for modular exponentiation and digital signatures, *Proc. of EUROCRYPT, LNCS*, vol.1403, pp.236-250, 1998.

[3] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, *Proc. of CRYPTO, LNCS*, vol.2139, pp.213-229, 2001.

[4] J. C. Cha and J. H. Cheon, An identity-based signature from gap Diffie-Hellman groups, *Proc. of PKC, LNCS*, vol.2567, pp.18-30, 2003.

[5] Y.-F. Chang and C.-C. Chang, Robust t-out-of-n proxy signature based on RSA cryptosystems, *International Journal of Innovative Computing, Information and Control*, vol.4, no.2, pp.425-431, 2008.

[6] L. Chen, Z. Cheng and N. P. Smart, Identity-based key agreement protocols from pairings, *International Journal of Information Security*, vol.6, no.4, pp.213-241, 2007.

[7] S. Cui, P. Duan and C. W. Chan, An efficient identity-based signature scheme with batch verifications, *Proc. of the 1st International Conference on Scalable Information Systems*, Hong Kong, 2006.

[8] Y. Dodis, J. Katz, S. Xu and M. Yung, Key-insulated public key cryptosystem, *Proc. of EUROCRYPTO, LNCS*, vol.2332, pp.65-82, 2002.

[9] Y. Dodis, J. Katz, S. Xu and M. Yung, Strong key-insulated signature scheme, *Proc. of PKC, LNCS*, vol.2567, pp.130-144, 2003.

[10] A. Fiat, Batch RSA, *Journal of Cryptology*, vol.10, no.2, pp.75-88, 1997.

[11] G. Hanaoka, Y. Hanaoka and H. Imai, Parallel key-insulated public key encryption, *Proc. of PKC, LNCS*, vol.3958, pp.105-122, 2006.

[12] Y. Hanaoka, G. Hanaoka, J. Shikata and H. Imai, Identity-based hierarchical strongly key-insulated encryption and its application, *Proc. of ASIACRYPT, LNCS*, vol.3788, pp.495-514, 2005.

[13] L. Harn and J. Ren, Efficient identity-based RSA multisignatures, *Computers & Security*, vol.27, pp.12-15, 2008.

[14] S. Kim, S. Park and D. Won, Proxy signatures, revisited, *Proc. of ICICS, LNCS*, vol.1334, pp.223-232, 1997.

[15] M. Mambo, K. Usuda and E. Okamoto, Proxy signatures: Delegation of the power to sign messages, *IEICE Trans. Fundamentals*, vol.E79-A, no.9, pp.1338-1354, 1996.

[16] D. Pointcheval and J. Stern, Security arguments for digital signatures and blind signatures, *Journal of Cryptology*, vol.13, no.3, pp.361-396, 2000.

[17] A. Shamir, Identity-based cryptosystems and signature schemes, *Proc. of CRYPTO, LNCS*, vol.196, pp.47-53, 1984.

[18] Y.-M. Tseng, T.-Y. Wu and J.-D. Wu, An efficient and provably secure ID-based signature scheme with batch verifications, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(A), pp.3911-3922, 2009.

[19] J. Weng, X. Li, K. Chen and S. Liu, Identity-based parallel key-insulated signature without random oracles, *Journal of Information Science and Engineering*, vol.24, no.4, pp.1143-1157, 2008.

[20] J. Weng, S. Liu and K. Chen, Identity-based parallel key-insulated signature: Framework and construction, *Journal of Research and Practice in Information Technology*, vol.40, no.1, pp.55-68, 2008.

[21] J. Weng, S. Liu, K. Chen, D. Zheng and W. Qiu, Identity-based threshold key-insulated encryption without random oracles, *Proc. of CT-RSA, LNCS*, vol.4964, pp.203-220, 2008.

[22] J. Weng, S. Liu, K. Chen and X. Li, Identity-based key-insulated signature with secure key-updates, *Proc. of INSCRYPT, LNCS*, vol.4318, pp.13-26, 2006.

[23] J. Weng, S. Liu, K. Chen and C. Ma, Identity-based parallel key-insulated encryption without random oracles: Security notions and construction, *Proc. of INDOCRYPT, LNCS*, vol.4329, pp.409-423, 2006.

[24] T.-Y. Wu and Y.-M. Tseng, An efficient user authentication and key exchange protocol for mobile client-server environment, *Computer Networks*, vol.52, no.9, pp.1520-1530, 2010.

[25] T.-Y. Wu and Y.-M. Tseng, An ID-based mutual authentication and key exchange protocol for low-power mobile devices, *The Computer Journal*, vol.53, no.7, pp.1062-1070, 2010.

[26] H.-J. Yoon, J.-H. Cheon and Y. Kim, Batch verifications with ID-based signatures, *Proc. of ICISC, LNCS*, vol.3506, pp.233-248, 2004.

[27] Y. Zhou, Z. Cao and Z. Chai, Identity based key insulated signature, *Proc. of ISPEC, LNCS*, vol.3903, pp.226-234, 2006.