INFORMATICA, 2012, Vol. 23, No. 1, 27–45

Identity-Based Key-Insulated Signcryption

Jianhong CHEN[1,2,3], Kefei CHEN[1,3], Yongtao WANG[1,3], Xiangxue LI[4], Yu LONG[1], Zhongmei WAN[5]

[1] *Department of Computer Science and Engineering, Shanghai Jiao Tong University 800 Dongchuan Road, Shanghai 200240, China*
[2] *School of Computer Engineering, Huaiyin Institute of Technology*
[3] *Shanghai Key Laboratory of Scalable Computing and System, Shanghai Jiao Tong University*
[4] *Department of Computer Science and Technology, East China Normal University 500 Dongchuan Road, Shanghai 200241, China*
[5] *College of Science, Hohai University 1 Xikang Road, Nanjing 210098, China*
e-mail: chen-kf@cs.sjtu.edu.cn, wyt_sjtu@yahoo.com.cn, xxli@cs.ecnu.edu.cn, longyu@sjtu.edu.cn, wan-zmei@163.com

Received: April 2009; accepted: September 2011

Abstract. Key-insulated cryptography is an important technique to protect private keys in identity-based (IB) cryptosytems. Despite the flurry of recent results on IB key-insulated encryption (IBKIE) and signature (IBKIS), a problem regarding the security and efficiency of practicing IBKIE and IBKIS as a joint IB key-insulated signature/encryption scheme with a common set of parameters and keys remains open. To deal with the above question, we propose an identity-based key-insulated signcryption (IBKISC) scheme. Compared with the Sign-then-Encrypt (StE) and Encrypt-then-Sign (EtS) using IBKIE and IBKIS in the standard model, our proposed IBKISC scheme is the fastest with the shortest ciphertext size.

Keywords: key-insulated, signcryption, standard model.

1. Introduction

In CRYPTO 1984, Shamir (1984) introduced a novel cryptography primitive named an identity-based cryptosystem in order to remove public key certificates. Since then, many identity-based encryption and signature schemes have been proposed. However, none of them are fully functioning until Boneh and Franklin put forward an identity-based encryption based on the Weil pairing over elliptic curves (Boneh, 2001).

Dodis *et al.* (2002) introduced a key insulation mechanism, which can protect private keys in public key cryptosystems. In a key-insulated cryptosystem, a physically-secure but computationally-limited device, named a helper, is involved. A private key is split into two parts: a temporary private key hold by the user and a helper key stored in the helper. The key-insulated cryptosystem refreshes the temporary private keys at discrete time periods via interaction between the user and the helper, and the public key remains unchanged throughout the lifetime of the system. A compromise of some periods of a key-insulated cryptosystem leaves the remaining time periods unharmed. Besides, a scheme

is called strongly key-insulated when adversaries corrupting the helper remain unable to perform private key operations on behalf of the user.

Privacy and authenticity are two of the most important aims offered by cryptography. Encryption and signature can be used to achieve these aims. Zheng (1997) proposed a primitive called signcryption in order to combine encryption and signature. A signcryption scheme can be more efficient than a composition of an encryption scheme and a signature scheme. In this paper, we propose an identity-based key-insulated signcryption (IBKISC) scheme.

1.1. *Our Contributions*

In this paper, we give a formal definition and security model for identity-based key-insulated signcryption (IBKISC) schemes, and then we propose an IBKISC scheme from bilinear pairings which is provably secure in the standard model. To the best of our knowledge, this is the first IBKISC scheme up to now.

1.2. *Related Work*

Several identity-based signcryption (IBSC) schemes (Barreto, 2005; Boyen, 2003; Chen, 2005; Chow, 2003; Malone-Lee, 2002; Nalla, 2003; Yuen, 2005) have been proposed so far. All the above schemes are provably secure in the random oracle model. Recently, Yu *et al.* (2009) put forward an identity-based signcryption scheme without random oracles. Then, Jin *et al.* (2010) improved it.

In identity-based key-insulated scenarios, identity-based hierarchical strongly key-insulated encryption was introduced by Hanaoka *et al.* (2005). Then, Zhou *et al.* (2006) put forward an identity-based key-insulated signature (IBKIS) scheme. However, their scheme is not strongly key-insulated. Weng et al. re-formalized the definition and security notions for IBKIS schemes and proposed a strongly key-insulated and perfectly key-insulated scheme (Weng, 2006). Then an IBKIS scheme without random oracles was given in Weng (2008).

1.3. *Organization*

In the upcoming sections, we first recall some preliminaries for an IBKISC scheme. Sections 3 and 4 give the syntax definition and security notions of the scheme respectively. Our scheme and its security are analyzed in Sections 5 and 6 repectively. We compare our IBKISC scheme with the related schemes in Section 7. We draw our conclusions in Section 8.

2. Preliminaries

Throughout this paper, we let Z_p denote the set $\{0, 1, 2, \ldots, p-1\}$ and Z_p^* denote $Z_p \backslash \{0\}$. In addition, we often equate a user with her/his identity $\mathbf{u}$.

2.1. *Bilinear Pairings*

Our IBKISC scheme uses a bilinear map, which is often called a "pairing". We describe bilinear maps and related mathematics in a more general format here.

Let G and G_T be two cyclic multiplicative groups with the same prime order p. Let e: $G \times G \rightarrow G_T$ be a pairing which satisfies the following conditions:

- *Bilinear*: For all $g_1, g_2 \in G$ and for all $a, b \in Z_p^*$, we have $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
- *Non-degenerate*: There exists $g_1, g_2 \in G$ such that $e(g_1, g_2) \neq 1$.
- *Computable*: There is an efficient algorithm to compute $e(g_1, g_2)$ for all $g_1, g_2 \in G$.

2.2. *Decisional Bilinear Diffie–Hellman (DBDH) Assumption*

DEFINITION 1. Let G and G_T be two cyclic multiplicative groups with the same prime order p, and e: $G \times G \rightarrow G_T$ be a bilinear pairing. Let $a, b, c \in Z_p$ be chosen at random and g be a generator of G. The DBDH assumption (Waters, 2005) is that no probabilistic polynomial-time algorithm $\mathcal{B}$ can distinguish the tuple $(g, A = g^a, B = g^b, C = g^c, e(g, g)^{abc}) \in G^4 \times G_T$ from the tuple $(g, A = g^a, B = g^b, C = g^c, e(g, g)^z)$ with more than a negligible advantage.

2.3. *Computational Diffie–Hellman (CDH) Assumption*

DEFINITION 2 (Computational Diffie–Hellman (CDH) problem). The CDH problem is, given $g, g^a, g^b \in G$ for unknown $a, b \in Z_p^*$, to compute g^{ab}.

DEFINITION 3. We say that the (t, ϵ)-CDH assumption holds in a group G if no algorithm running in time at most t can solve the CDH problem in G with probability at least ϵ.

3. Syntax of Identity-Based Key-Insulated Signcryption

An IBKISC scheme consists of six algorithms:

- Setup: Given a security parameter κ, the private key generator (PKG) uses this key generation algorithm to generate a set of public parameters, cp, and a master secret key msk.
- Extract: Given a user identity $\mathbf{u}$, the PKG uses this key extraction algorithm to compute an initial private key $d_{\mathbf{u},0}$ and a helper key $\text{HK}_{\mathbf{u}}$ corresponding to $\mathbf{u}$. The helper key is kept by the helper and the user $\mathbf{u}$ keeps the initial private key.
- HelperUpt: Given period indices t' and t, an identity $\mathbf{u}$ and its helper key $\text{HK}_{\mathbf{u}}$, the helper uses this helper key-update algorithm to compute the key-update information for $\mathbf{u}$ from period t' to period t, $\text{UI}_{\mathbf{u},t',t}$.
- UserUpt: Given an identity $\mathbf{u}$, the temporary private key $d_{\mathbf{u},t'}$ corresponding to $\mathbf{u}$ and t', and the key-update information for $\mathbf{u}$ from period t' to period t, $\text{UI}_{\mathbf{u},t',t}$, the user uses this user key-update algorithm to compute the temporary private key $d_{\mathbf{u},t}$ corresponding to $\mathbf{u}$ and t.

- Signcrypt: In period t, to send a message $\mathbf{m}$ to Bob whose identity is $\mathbf{b}$, Alice with identity $\mathbf{a}$ obtains a ciphertext (t, σ) by computing Signcrypt$(t, \mathbf{m}, d_{\mathbf{a},t}, \mathbf{b})$.
- Unsigncrypt: After Bob receives the ciphertext (t, σ), he computes Unsigncrypt $((t, \sigma), d_{\mathbf{b},t})$ and obtains the message $\mathbf{m}$ or the symbol $\perp$ indicating that the ciphertext is invalid.

4. Security Notions

In this subsection, we formalize the security notions for IBKISC schemes. This is based on the security definitions in key-insulated cryptography (Dodis, 2002) and IBSC systems (Boyen, 2003).

4.1. *Key-Insulated Security*

DEFINITION 4. For an IBKISC scheme, its semantic security against an adaptive chosen ciphertext attack under key-exposure (IND-IBKISC-KI-CCA) can be defined via the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$:

- **Setup**: $\mathcal{B}$ runs algorithm Setup to generate a set of public parameters, cp, and a master secret key msk. $\mathcal{B}$ gives cp to $\mathcal{A}$ and keeps msk to itself.
- **Phase 1**: $\mathcal{A}$ issues a series of queries in an adaptive fashion. The following queries are allowed.
  - Extract Queries. Upon receiving a user's identity $\mathbf{u}$, $\mathcal{B}$ runs algorithm Extract and obtains an initial private key $d_{\mathbf{u},0}$ and a helper key $\mathrm{HK}_{\mathbf{u}}$. $\mathcal{B}$ then sends $d_{\mathbf{u},0}$ and $\mathrm{HK}_{\mathbf{u}}$ to $\mathcal{A}$.
  - Temporary Private Key Queries. Upon receiving a tuple $\langle \mathbf{u}, t \rangle$ consisting of identity $\mathbf{u}$ and period t, $\mathcal{B}$ responds by running algorithms HelperUpt and UserUpt to generate $d_{\mathbf{u},t}$. $\mathcal{B}$ then returns it to $\mathcal{A}$.
  - Signcrypt Queries. Upon receiving a tuple $\langle \mathbf{m}, \mathbf{a}, \mathbf{b}, t \rangle$ consisting of a message $\mathbf{m}$, two identities, $\mathbf{a}$ and $\mathbf{b}$, and period t, $\mathcal{B}$ generates a ciphertext (t, σ).
  - Unsigncrypt Queries. Upon receiving a tuple $\langle (t, \sigma), \mathbf{a}, \mathbf{b} \rangle$ consisting of a ciphertext (t, σ) and two identities, $\mathbf{a}$ and $\mathbf{b}$, $\mathcal{B}$ outputs the decryption outcome and the verification result.
- **Challenge**: At the end of **Phase 1**, $\mathcal{A}$ outputs two identities, $\mathbf{a}^*$ and $\mathbf{b}^*$, a period index t^* and two equal-length messages, $\mathbf{m}_0$ and $\mathbf{m}_1$. $\mathcal{A}$ knows $d_{\mathbf{a}^*,t^*}$ and must not have made an extract query on $\mathbf{b}^*$. $\mathcal{B}$ picks a random bit $\gamma \in \{0, 1\}$ and computes Signcrypt$(t^*, m_\gamma, d_{\mathbf{a}^*,t^*}, \mathbf{b}^*)$ to obtain (t^*, σ^*). $\mathcal{B}$ sends (t^*, σ^*) to $\mathcal{A}$.
- **Phase 2**: $\mathcal{A}$ continues to issue additional queries as in Phase 1, and $\mathcal{B}$ responds these queries as in **Phase 1**.
- **Guess**: Eventually, $\mathcal{A}$ outputs γ' as the guess of γ. $\mathcal{B}$ ignores the answer. $\mathcal{A}$ wins the game if $\gamma' = \gamma$.

$\mathcal{A}$'s advantage is defined to be

$$Adv^{IND\text{-}IBKISC\text{-}KI\text{-}CCA}(\mathcal{A}) = \left|\Pr[\gamma' = \gamma] - \frac{1}{2}\right|.$$

We say that adversary $\mathcal{A}$ wins in this game if the following conditions are satisfied: (1) the target identity $\mathbf{b}^*$ does not appear in extraction queries; (2) $\langle \mathbf{b}^*, t^* \rangle$ does not appear in temporary private key queries; (3) the unsigncrypt query for ciphertext (t^*, σ^*) is disallowed; (4) $\mathcal{A}$ can not obtain the sender Alice's identity; (5) the sender is honest (i.e., the sender does not launch inner attacks).

DEFINITION 5. We say that an IBKISC scheme in the standard model is said to be existentially unforgeable against chosen-message attacks under key-exposure (EU-IBKISC-KI-CMA) if no PPT adversary $\mathcal{A}$ has a non-negligible advantage against a challenger $\mathcal{B}$ in the following game:

- **Setup**: The same as Definition 4.
- **Queries**: $\mathcal{A}$ issues queries to the same oracles as those in Definition 4.
- **Forgery**: Eventually, $\mathcal{A}$ outputs a tuple $((t^*, \sigma^*), \mathbf{a}^*, \mathbf{b}^*)$ (i.e., a tuple that was not produced by the signcryption oracle) and wins the game if the result of $\mathsf{Unsigncrypt}((t^*, \sigma^*), \mathbf{a}^*, d_{\mathbf{b}^*, t^*})$ is not the $\perp$ symbol.

$\mathcal{A}$'s advantage is defined to be

$$Adv^{EU\text{-}IBKISC\text{-}KI\text{-}CMA}(\mathcal{A}) = \Pr[\mathcal{A} \text{ wins the } EU\text{-}IBKISC\text{-}KI\text{-}CMA \text{ game }].$$

We say that adversary $\mathcal{A}$ wins in this game if the following conditions are satisfied: (1) the target identity $\mathbf{a}^*$ does not appear in extraction queries; (2) $\langle \mathbf{a}^*, t^* \rangle$ does not appear in temporary private key queries; (3) (t^*, σ^*) should not be computed by signcrypt queries oracle; (4) $\mathcal{A}$ can not obtain the sender Alice's identity; (5) the sender is honest (i.e., the sender does not launch inner attacks).

4.2. *Strongly Key-Insulated Security*

DEFINITION 6. For an IBKISC scheme, its semantic security against an adaptive chosen ciphertext attack under strong key exposure (IND-IBKISC-SKI-CCA) can be defined via the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$:

- **Setup**: The same as Definition 4.
- **Phase 1**: $\mathcal{A}$ issues a series of queries in an adaptive fashion. The following queries are allowed.
  - Extract Queries. The same as Definition 4.
  - Helper Key Queries. Upon receiving a user's identity $\mathbf{u}$, $\mathcal{B}$ runs algorithm Extract to generate $\mathrm{HK}_{\mathbf{u}}$ and sends it to $\mathcal{A}$.
  - Signcrypt Queries. The same as Definition 4.
  - Unsigncrypt Queries. The same as Definition 4.
- **Challenge**: The same as Definition 4.
- **Phase 2**: The same as Definition 4.
- **Guess**: The same as Definition 4.

$\mathcal{A}$'s advantage is defined to be

$$Adv^{IND\text{-}IBKISC\text{-}SKI\text{-}CCA}(\mathcal{A}) = \left|\Pr[u' = u] - \frac{1}{2}\right|.$$

We say that adversary $\mathcal{A}$ wins in this game if the following conditions are satisfied: (1) the target identity $\mathbf{b}^*$ does not appear in extraction queries; (2) An unsigncrypt query for ciphertext (t^*, σ^*) is disallowed; (3) $\mathcal{A}$ can not obtain the sender Alice's identity; (4) the sender is honest (i.e., the sender does not launch inner attacks).

DEFINITION 7. We say that an IBKISC scheme in the standard model is said to be existentially unforgeable against chosen-message attacks under strong key-exposure (EU-IBKISC-SKI-CMA) if no PPT adversary $\mathcal{A}$ has a non-negligible advantage against a challenger $\mathcal{B}$ in the following game:

- $\mathbf{Setup}$: The same as Definition 4.
- $\mathbf{Queries}$: $\mathcal{A}$ issues queries to the same oracles as those in Definition 6.
- $\mathbf{Forgery}$: The same as Definition 5.

$\mathcal{A}$'s advantage is defined to be

$$Adv^{EU\text{-}IBKISC\text{-}SKI\text{-}CMA}(\mathcal{A}) = \Pr[\mathcal{A} \text{ wins the } EU\text{-}IBKISC\text{-}SKI\text{-}CMA \text{ game}].$$

We say that adversary $\mathcal{A}$ wins in this game if the following conditions are satisfied: (1) the target identity $\mathbf{a}^*$ does not appear in extraction queries; (2) (t^*, σ^*) should not be computed by signcrypt queries oracle; (3) $\mathcal{A}$ can not obtain the sender Alice's identity; (4) the sender is honest (i.e., the sender does not launch inner attacks).

5. Our Construction

We will use the notation $x \overset{U}{\leftarrow} S$ as a short-hand for choosing a value x uniformly at random from the set S. Inspired by the cryptographic applications of pseudo-random

function (PRF) (Goldreich, 1985), we also use a PRF F such that given a κ-bit seed s and a κ-bit argument x, it outputs a κ-bit string $F_s(x)$. Let a bitstring of length n_u represent an identity for some $n_u \in Z$. Let a bitstring of length n_m represent a message for some $n_m \in Z$. Our construction follows.

- Setup: Choose groups G and G_T of prime order p of size κ such that an admissible pairing e: $G \times G \rightarrow G_T$ can be constructed and pick a generator g of G. Let H_u: $\{0,1\}^* \rightarrow \{0,1\}^{n_u}$ be a collision-resistant hash function. Let H_v: $\{0,1\}^{n_m} \rightarrow \{0,1\}^{n_v}$ be a collision-resistant hash function for some $n_v \in Z$. Then, pick a secret $\alpha \overset{U}{\leftarrow} Z_p$, compute $g_1 = g^\alpha$ and pick $g_2 \overset{U}{\leftarrow} G$. Let $Y = e(g_1, g_2)$. Next, define Γ to be a subset of $\{0,1\}^{n_u+n_m+n_v}$ with p elements, and define V: $\Gamma \rightarrow G_T$ to be a bijective function while V^{-1} is its inverse mapping. In addition, pick elements $u', m' \overset{U}{\leftarrow} G$ and vectors $\vec{U} = (u_i)$, $\vec{M} = (m_i)$ of length n_u and n_v, respectively, whose entries are random elements from G. The public parameters and the master secret key are

$$cp = \big(G, G_T, e, g, g_1, g_2, Y, u', \overrightarrow{U}, m', \overrightarrow{M}, H_u, H_v, V\big), \quad msk = g_2^\alpha.$$

- Extract: Let $\mathbf{u}$ be a bitstring of length n_u representing an identity and let $\mathbf{u}[i]$ be the ith bit of $\mathbf{u}$. Define $\mathcal{U}_\mathbf{u} \subseteq \{1, \ldots, n_u\}$ to be the set of indices i such that $\mathbf{u}[i] = 1$. Let $\mathbf{w}_{\mathbf{u},0}$ be the output of $H_u(\mathbf{u}\|0)$ and let $\mathbf{w}_{\mathbf{u},0}[i]$ be the ith bit of $\mathbf{w}_{\mathbf{u},0}$. Define $\mathcal{W}_{\mathbf{u},0} \subseteq \{1, \ldots, n_u\}$to be the set of indices i such that $\mathbf{w}_{\mathbf{u},0}[i] = 1$. Pick a helper key $\mathrm{HK}_\mathbf{u} \overset{U}{\leftarrow} \{0,1\}^\kappa$ and compute $k_{\mathbf{u},0} = F_{\mathrm{HK}_\mathbf{u}}(0)$. To construct the initial private key, $d_{\mathbf{u},0}$, of the identity $\mathbf{u}$, pick $r_\mathbf{u} \overset{U}{\leftarrow} Z_p$, and compute:

$$\begin{aligned} d_{\mathbf{u},0} &= \big(d_{\mathbf{u},0}^{\langle 1\rangle}, d_{\mathbf{u},0}^{\langle 2\rangle}, d_{\mathbf{u},0}^{\langle 3\rangle}\big) \\ &= \left(g_2^\alpha \left(u' \prod_{i \in \mathcal{U}_\mathbf{u}} u_i\right)^{r_\mathbf{u}} \left(u' \prod_{i \in \mathcal{W}_{\mathbf{u},0}} u_i\right)^{k_{\mathbf{u},0}}, g^{k_{\mathbf{u},0}}, g^{r_\mathbf{u}}\right). \end{aligned}$$

Therefore, the sender Alice's helper key is $\mathrm{HK}_\mathbf{a}$ and her initial private key is

$$\begin{aligned} d_{\mathbf{a},0} &= \big(d_{\mathbf{a},0}^{\langle 1\rangle}, d_{\mathbf{a},0}^{\langle 2\rangle}, d_{\mathbf{a},0}^{\langle 3\rangle}\big) \\ &= \left(g_2^\alpha \left(u' \prod_{i \in \mathcal{U}_\mathbf{a}} u_i\right)^{r_\mathbf{a}} \left(u' \prod_{i \in \mathcal{W}_{\mathbf{a},0}} u_i\right)^{k_{\mathbf{a},0}}, g^{k_{\mathbf{a},0}}, g^{r_\mathbf{a}}\right). \end{aligned}$$

The receiver Bob's helper key is $\mathrm{HK}_\mathbf{b}$ and his initial private key is

$$\begin{aligned} d_{\mathbf{b},0} &= \big(d_{\mathbf{b},0}^{\langle 1\rangle}, d_{\mathbf{b},0}^{\langle 2\rangle}, d_{\mathbf{b},0}^{\langle 3\rangle}\big) \\ &= \left(g_2^\alpha \left(u' \prod_{i \in \mathcal{U}_\mathbf{b}} u_i\right)^{r_\mathbf{b}} \left(u' \prod_{i \in \mathcal{W}_{\mathbf{b},0}} u_i\right)^{k_{\mathbf{b},0}}, g^{k_{\mathbf{b},0}}, g^{r_\mathbf{b}}\right). \end{aligned}$$

- HelperUpt: As in the Extract algorithm, let $\mathbf{w}_{\mathbf{u},t}$ be the output of $H_u(\mathbf{u}\|t)$, let $\mathbf{w}_{\mathbf{u},t}[i]$ be the ith bit of $\mathbf{w}_{\mathbf{u},t}$ and define $\mathcal{W}_{\mathbf{u},t} \subseteq \{1,\ldots,n_u\}$to be the set of indices i such that $\mathbf{w}_t[i] = 1$. Likewise, let $\mathbf{w}_{\mathbf{u},t'}$ be the output of $H_u(\mathbf{u}\|t')$, let $\mathbf{w}_{\mathbf{u},t'}[i]$ be the ith bit of $\mathbf{w}_{\mathbf{u},t'}$ and define $\mathcal{W}_{\mathbf{u},t'} \subseteq \{1,\ldots,n_u\}$to be the set of indices i such that $\mathbf{w}_{\mathbf{u},t'}[i] = 1$. Compute $k_{\mathbf{u},t} = F_{\mathrm{HK}_\mathbf{u}}(t)$ and $k_{\mathbf{u},t'} = F_{\mathrm{HK}_\mathbf{u}}(t')$. To construct the key-update information for identity $\mathbf{u}$ from period t' to period t, $\mathrm{UI}_{\mathbf{u},t',t}$, compute:

$$\begin{aligned}\mathrm{UI}_{\mathbf{u},t',t} &= \big(\mathrm{UI}^{\langle 1\rangle}_{\mathbf{u},t',t}, \mathrm{UI}^{\langle 2\rangle}_{\mathbf{u},t',t}\big)\\ &= \left(\left(u'\prod_{i\in\mathcal{W}_{\mathbf{u},t}} u_i\right)^{k_{\mathbf{u},t}} \Big/ \left(u'\prod_{i\in\mathcal{W}_{\mathbf{u},t'}} u_i\right)^{k_{\mathbf{u},t'}}, g^{k_{\mathbf{u},t}}\right).\end{aligned}$$

Likewise, Alice and Bob's key-update information from t' to t are

$$\begin{aligned}\mathrm{UI}_{\mathbf{a},t',t} &= \big(\mathrm{UI}^{\langle 1\rangle}_{\mathbf{a},t',t}, \mathrm{UI}^{\langle 2\rangle}_{\mathbf{a},t',t}\big)\\ &= \left(\left(u'\prod_{i\in\mathcal{W}_{\mathbf{a},t}} u_i\right)^{k_{\mathbf{a},t}} \Big/ \left(u'\prod_{i\in\mathcal{W}_{\mathbf{a},t'}} u_i\right)^{k_{\mathbf{a},t'}}, g^{k_{\mathbf{a},t}}\right),\\ \mathrm{UI}_{\mathbf{b},t',t} &= \big(\mathrm{UI}^{\langle 1\rangle}_{\mathbf{b},t',t}, \mathrm{UI}^{\langle 2\rangle}_{\mathbf{b},t',t}\big)\\ &= \left(\left(u'\prod_{i\in\mathcal{W}_{\mathbf{b},t}} u_i\right)^{k_{\mathbf{b},t}} \Big/ \left(u'\prod_{i\in\mathcal{W}_{\mathbf{b},t'}} u_i\right)^{k_{\mathbf{b},t'}}, g^{k_{\mathbf{b},t}}\right).\end{aligned}$$

- UserUpt: Parse the temporary private key for identity $\mathbf{u}$ and period t' as $(d_{\mathbf{u},t'} = d^{\langle 1\rangle}_{\mathbf{u},t'}, d^{\langle 2\rangle}_{\mathbf{u},t'}, d^{\langle 3\rangle}_{\mathbf{u},t'})$. Parse the key-update information for identity $\mathbf{u}$ from period t' to period t as $\mathrm{UI}_{\mathbf{u},t',t} = (\mathrm{UI}^{\langle 1\rangle}_{\mathbf{u},t',t}, \mathrm{UI}^{\langle 2\rangle}_{\mathbf{u},t',t})$. To construct the temporary private key for identity $\mathbf{u}$ and period t, $d_{\mathbf{u},t}$, the user $\mathbf{u}$ computes:

$$d_{\mathbf{u},t} = \big(d^{\langle 1\rangle}_{\mathbf{u},t'} \cdot \mathrm{UI}^{\langle 1\rangle}_{\mathbf{u},t',t}, \mathrm{UI}^{\langle 2\rangle}_{\mathbf{u},t',t}, d^{\langle 3\rangle}_{\mathbf{u},t'}\big).$$

Note that at time period t, $d_{\mathbf{u},t}$ is always set to be

$$d_{\mathbf{u},t} = \left(g_2^\alpha\left(u'\prod_{i\in\mathcal{U}_\mathbf{u}} u_i\right)^{r_\mathbf{u}}\left(u'\prod_{i\in\mathcal{W}_{\mathbf{u},t}} u_i\right)^{k_{\mathbf{u},t}}, g^{k_{\mathbf{u},t}}, g^{r_\mathbf{u}}\right).$$

Likewise, Alice and Bob's temporary private keys for period t are

$$\begin{aligned}d_{\mathbf{a},t} &= \left(g_2^\alpha\left(u'\prod_{i\in\mathcal{U}_\mathbf{a}} u_i\right)^{r_\mathbf{a}}\left(u'\prod_{i\in\mathcal{W}_{\mathbf{a},t}} u_i\right)^{k_{\mathbf{a},t}}, g^{k_{\mathbf{a},t}}, g^{r_\mathbf{a}}\right),\\ d_{\mathbf{b},t} &= \left(g_2^\alpha\left(u'\prod_{i\in\mathcal{U}_\mathbf{b}} u_i\right)^{r_\mathbf{b}}\left(u'\prod_{i\in\mathcal{W}_{\mathbf{b},t}} u_i\right)^{k_{\mathbf{b},t}}, g^{k_{\mathbf{b},t}}, g^{r_\mathbf{b}}\right).\end{aligned}$$

- Signcrypt: Let $\mathbf{m} \in \{0,1\}^{n_m}$ be a bitstring representing a message. In period t, to signcrypt a message $\mathbf{m}$ to Bob, Alice parses her temporary private key as $d_{\mathbf{a},t} = (d_{\mathbf{a},t}^{\langle 1\rangle}, d_{\mathbf{a},t}^{\langle 2\rangle}, d_{\mathbf{a},t}^{\langle 3\rangle})$, picks $r_{\mathbf{m}}, r'_t \overset{U}{\leftarrow} Z_p^*$, lets $r_t = r'_t + k_{\mathbf{a},t}$, picks $\mathbf{r} \overset{U}{\leftarrow} \{0,1\}^{n_v}$ such that $\mathbf{a}\|\mathbf{m}\|\mathbf{r} \in \Gamma$. Let $\mathcal{M}_{\mathbf{m}} \subseteq \{1,\ldots,n_v\}$ be the set of indices j for which the jth bit of $H_v(\mathbf{m})$ is different from that of $\mathbf{r}$, i.e., $\mathcal{M}_{\mathbf{m}} = \{j \in Z\colon H_v(\mathbf{m})[j] \oplus \mathbf{r}[j] = 1\}$. Then, Alice computes:

$$\begin{aligned}
\sigma^{\langle 1\rangle} &= Y^{r_{\mathbf{m}}} \cdot V(\mathbf{a}\|\mathbf{m}\|\mathbf{r}), \qquad \sigma^{\langle 2\rangle} = g^{r_{\mathbf{m}}},\\
\sigma^{\langle 3\rangle} &= \left(u' \prod_{i\in\mathcal{U}_{\mathbf{b}}} u_i\right)^{r_{\mathbf{m}}}, \qquad \sigma^{\langle 4\rangle} = \left(w' \prod_{i\in\mathcal{W}_{\mathbf{b},t}} w_i\right)^{r_{\mathbf{m}}},\\
\sigma^{\langle 5\rangle} &= d_{\mathbf{a},t}^{\langle 1\rangle} \cdot \left(u' \prod_{i\in\mathcal{W}_{\mathbf{a},t}} u_i\right)^{r'_t} \cdot \left(m' \prod_{i\in\mathcal{M}_{\mathbf{m}}} m_i\right)^{r_{\mathbf{m}}}\\
&= g_2^{\alpha}\left(u' \prod_{i\in\mathcal{U}_{\mathbf{a}}} u_i\right)^{r_{\mathbf{a}}}\left(u' \prod_{i\in\mathcal{W}_{\mathbf{a},t}} u_i\right)^{r'_t+k_{\mathbf{a},t}}\left(m' \prod_{i\in\mathcal{M}_{\mathbf{m}}} m_i\right)^{r_{\mathbf{m}}}\\
&= g_2^{\alpha}\left(u' \prod_{i\in\mathcal{U}_{\mathbf{a}}} u_i\right)^{r_{\mathbf{a}}}\left(u' \prod_{i\in\mathcal{W}_{\mathbf{a},t}} u_i\right)^{r_t}\left(m' \prod_{i\in\mathcal{M}_{\mathbf{m}}} m_i\right)^{r_{\mathbf{m}}},\\
\sigma^{\langle 6\rangle} &= d_{\mathbf{a},t}^{\langle 2\rangle} \cdot g^{r'_t} = g^{r'_t+k_{\mathbf{a},t}} = g^{r_t}, \qquad \sigma^{\langle 7\rangle} = d_{\mathbf{a},t}^{\langle 3\rangle} = g^{r_{\mathbf{a}}}.
\end{aligned}$$

  Alice outputs a ciphertext $(t,\sigma) = (t, (\sigma^{\langle 1\rangle}, \sigma^{\langle 2\rangle}, \sigma^{\langle 3\rangle}, \sigma^{\langle 4\rangle}, \sigma^{\langle 5\rangle}, \sigma^{\langle 6\rangle}, \sigma^{\langle 7\rangle}))$ and sends it to Bob.
- Unsigncrypt: Bob receives the ciphertext $(t,\sigma) = (t, (\sigma^{\langle 1\rangle}, \sigma^{\langle 2\rangle}, \sigma^{\langle 3\rangle}, \sigma^{\langle 4\rangle}, \sigma^{\langle 5\rangle}, \sigma^{\langle 6\rangle}, \sigma^{\langle 7\rangle}))$, parses his temporary private key as $d_{\mathbf{b},t} = (d_{\mathbf{b},t}^{\langle 1\rangle}, d_{\mathbf{b},t}^{\langle 2\rangle}, d_{\mathbf{b},t}^{\langle 3\rangle})$ and decrypts the ciphertext as follows.
  1. Compute $V^{-1}(\sigma^{\langle 1\rangle} \cdot e(d_{\mathbf{b},t}^{\langle 3\rangle}, \sigma^{\langle 4\rangle}) e(d_{\mathbf{b},t}^{\langle 2\rangle}, \sigma^{\langle 3\rangle}) / e(d_{\mathbf{b},t}^{\langle 1\rangle}, \sigma^{\langle 2\rangle})) \rightarrow \mathbf{a}\|\mathbf{m}\|\mathbf{r}$.
  2. Generate $\{j \in Z\colon H_v(\mathbf{m})[j] \oplus \mathbf{r}[j] = 1\} \rightarrow \mathcal{M}_{\mathbf{m}}$.
  3. Accept the message if the following equality holds:

$$\begin{aligned}
e\big(\sigma^{\langle 5\rangle}, g\big) = Y \cdot e\left(\sigma^{\langle 7\rangle}, u' \prod_{i\in\mathcal{U}_{\mathbf{a}}} u_i\right) e\left(\sigma^{\langle 6\rangle}, u' \prod_{i\in\mathcal{W}_{\mathbf{a},t}} u_i\right)\\
\times e\left(\sigma^{\langle 2\rangle}, m' \prod_{i\in\mathcal{M}_{\mathbf{m}}} m_i\right).
\end{aligned}$$

It is easy to see that the above Unsigncrypt algorithm is consistent.

6. Proof of Security

Theorem 1. *Our IBKISC scheme is IND-IBKISC-KI-CCA secure in the standard model, assuming that the DBDH assumption holds in groups (G, G_T), the hash function H is collision-resistant, F is a pseudo-random function and V is a bijective function.*

Concretely, if there exists an IND-IBKISC-KI-CCA adversary $\mathcal{A}$ that is able to distinguish two valid ciphertexts during the game defined in Definition 4 *with advantage at least ϵ when running in time at most t and asking at most q_e extract queries, q_t temporary private key queries, q_s signcryption queries and q_u unsigncryption queries, there exists a challenger that can solve an instance of the DBDH problem in time $t' < t + O((q_e + q_s + q_t + q_u)n_u t_m + (q_e + q_t + q_s)t_e + q_u t_p)$ with advantage*

$$\epsilon' > \frac{\epsilon}{54q_s(q_e + q_t + q_s)^2(n_u + 1)^2(n_v + 1)},$$

where t_m, t_e and t_p denote the time for a multiplication, an exponentiation in G and a pairing computation respectively.

Proof. We build a simulator $\mathcal{B}$ running in polynomial time that solves the DBDH problem with a non-negligible advantage ϵ'. $\mathcal{B}$ will take BDH challenge $(g,\ A = g^a,\ B = g^b,\ C = g^c,\ \mathcal{Z})$. $\mathcal{B}$ simulates a challenger for $\mathcal{A}$ in the following way:

- **Setup**: $\mathcal{B}$ sets $l_u = \frac{3(q_e+q_t+q_s)}{2}$ and $l_m = 2q_s$, and randomly chooses two integers k_u and k_m, with $0 < k_u < n_u$ and $0 < k_m < n_v$. We will assume that $l_u(n_u+1) < p$ and $l_m(n_v+1) < p$ for the given values of q_e, q_s, n_u and n_v. The simulator then chooses $x' \overset{U}{\leftarrow} Z_{l_u}$ and a vector $\vec{X} = (x_i)$ of length n_u with $x_i \overset{U}{\leftarrow} Z_{l_u}$ for all i. Likewise, it chooses $z' \overset{U}{\leftarrow} Z_{l_m}$ and a vector $\vec{Z} = (Z_j)$ of length n_v, with $z_j \overset{U}{\leftarrow} Z_{l_m}$ for all j. Lastly, $\mathcal{B}$ chooses $y', w' \overset{U}{\leftarrow} Z_p$ and two vectors, $\vec{Y} = (y_i)$ and $\vec{W} = (w_j)$, of length n_u and n_v, respectively, with $y_i, w_j \overset{U}{\leftarrow} Z_p$ for all i and j. To make the notation easier to follow, the following two pairs of functions are defined for an identity $\mathbf{u}$ and a message $\mathbf{m}\|\mathbf{r}$ respectively:

$$I(\mathbf{u}) = x' + \sum_{i \in \mathcal{U}_\mathbf{u}} x_i - l_u k_u \quad \text{and} \quad J(\mathbf{u}) = y' + \sum_{i \in \mathcal{U}_\mathbf{u}} y_i, \tag{1}$$

$$K(\mathbf{m}\|\mathbf{r}) = z' + \sum_{j \in \mathcal{M}_\mathbf{m}} z_j - l_m k_m \quad \text{and} \quad L(\mathbf{m}\|\mathbf{r}) = w' + \sum_{j \in \mathcal{M}_\mathbf{m}} w_j. \tag{2}$$

For $\mathbf{w}_{\mathbf{u},t}$, the output of $H_u(\mathbf{u}\|t)$, we have

$$I(\mathbf{w}_{\mathbf{u},t}) = x' + \sum_{i \in \mathcal{W}_{\mathbf{u},t}} x_i - l_u k_u \quad \text{and} \quad J(\mathbf{w}_{\mathbf{u},t}) = y' + \sum_{i \in \mathcal{W}_{\mathbf{u},t}} y_i.$$

Now, $\mathcal{B}$ constructs a set of public parameters for the IBKISC scheme by making the following assignments:

$$\begin{aligned}
&g_1 = g^a, \qquad g_2 = g^b, \\
&u' = g_2^{-l_u k_u + x'} g^{y'}, \qquad u_i = g_2^{x_i} g^{y_i}, \quad 1 < i < n_u, \\
&m' = g_2^{-l_m k_m + z'} g^{w'}, \qquad m_j = g_2^{z_j} g^{w_j}, \quad 1 < j < n_v.
\end{aligned}$$

Note that these public parameters will have the same distribution as in the game between the challenger and $\mathcal{A}$. Furthermore, this assignment means that the master secret will be $g_2^{\alpha} = g_2^{a} = g^{ab}$ and that for any identity $\mathbf{u}$ and message $\mathbf{m}\|\mathbf{r}$, the equations

$$u' \prod_{i \in \mathcal{U}_{\mathbf{u}}} u_i = g_2^{I(\mathbf{u})} g^{J(\mathbf{u})} \quad \text{and} \quad m' \prod_{j \in \mathcal{M}_{\mathbf{m}}} m_j = g_2^{K(\mathbf{m}\|\mathbf{r})} g^{L(\mathbf{m}\|\mathbf{r})}$$

hold. Hence, the equation

$$u' \prod_{i \in \mathcal{W}_{\mathbf{u},t}} u_i = g_2^{I(\mathbf{w}_{\mathbf{u},t})} g^{J(\mathbf{w}_{\mathbf{u},t})}$$

holds. All public parameters are passed to $\mathcal{A}$.

- **Phase 1**: When running the adversary, extract, temporary private key, signcryption and unsigncryption queries can occur. $\mathcal{B}$ answers these in the following way:
  - Extract Queries. $\mathcal{B}$ maintain two lists, HK^{list} and r^{list}, which are initially empty. Consider a query for the helper key and the initial private key of an identity $\mathbf{u}$. $\mathcal{B}$ searches HK^{list} for tuple $(\mathbf{u}, \text{HK}_{\mathbf{u}})$ (if HK^{list} does not contain this tuple, it chooses $\text{HK}_{\mathbf{u}} \overset{U}{\leftarrow} \{0,1\}^{\kappa}$ and adds $(\mathbf{u}, \text{HK}_{\mathbf{u}})$ into HK^{list}). $\mathcal{B}$ does not know the master secret, but assuming $I(\mathbf{u}) \neq 0 \bmod p$, it can construct an initial private key by searching r^{list} for tuple $(\mathbf{u}, r_{\mathbf{u}})$ (if r^{list} does not contain this tuple, it chooses $r_{\mathbf{u}} \overset{U}{\leftarrow} Z_p$ and adds $(\mathbf{u}, r_{\mathbf{u}})$ into r^{list}), computing $k_{\mathbf{u},0} = F_{\text{HK}_{\mathbf{u}}}(0)$ and setting the initial private key $d_{\mathbf{u},0} = (d_{\mathbf{u},0}^{\langle 1 \rangle}, d_{\mathbf{u},0}^{\langle 2 \rangle}, d_{\mathbf{u},0}^{\langle 3 \rangle})$ as follows.

$$d_{\mathbf{u},0}^{\langle 1 \rangle} = g_1^{-J(\mathbf{u})/I(\mathbf{u})} \left(u' \prod_{i \in \mathcal{U}_{\mathbf{u}}} u_i \right)^{r_{\mathbf{u}}} \left(u' \prod_{i \in \mathcal{W}_{\mathbf{u},0}} u_i \right)^{k_{\mathbf{u},0}},$$
$$d_{\mathbf{u},0}^{\langle 2 \rangle} = g^{k_{\mathbf{u},0}}, \qquad d_{\mathbf{u},0}^{\langle 3 \rangle} = g_1^{-1/I(\mathbf{u})} g^{r_{\mathbf{u}}}.$$

    Writing $\hat{r}_{\mathbf{u}} = r_{\mathbf{u}} - a/I(\mathbf{u})$, it can be verified that defining $d_{\mathbf{u},0}$ in this manner yields a valid initial private key of $\mathbf{u}$. Therefore, to $\mathcal{A}$, all initial private keys computed by $\mathcal{B}$ will be indistinguishable from the keys generated by a true challenger. If, on the other hand, $I(\mathbf{u}) = 0 \bmod p$, the above computation cannot be performed and the simulation will abort. To make the analysis of the simulation easier, we will force the simulation to abort whenever $I(\mathbf{u}) = 0 \bmod l_u$. Given the assumption $l_u(n_u + 1) < p$ which implies $0 < l_u k_u < p$ and $0 < x' + \sum_{i \in \mathcal{U}_{\mathbf{u}}} x_i < p$, it is easy to see that $I(\mathbf{u}) = 0 \bmod p$ implies that $I(\mathbf{u}) = 0 \bmod l_u$ according to (1). Hence, $I(\mathbf{u}) \neq 0 \bmod l_u$ implies $I(\mathbf{u}) \neq 0 \bmod p$, so the former condition will be a sufficient requirement to ensure that an initial private key for $\mathbf{u}$ can be constructed.
  - Temporary Private Key Queries. Consider a query for the temporary private key of an identity $\mathbf{u}$ and period t. $\mathcal{B}$ does not know the master secret, but

assuming that $I(\mathbf{u}) \equiv I(\mathbf{w}_{\mathbf{u},t}) \equiv 0 \bmod p$ does not hold, it can construct a temporary private key by searching r^{list} for tuple $(\mathbf{u}, r_{\mathbf{u}})$ (if r^{list} does not contain this tuple, it chooses $r_{\mathbf{u}} \overset{U}{\leftarrow} Z_p$ and adds $(\mathbf{u}, r_{\mathbf{u}})$ into r^{list}), picking $k_{\mathbf{u},t} \overset{U}{\leftarrow} Z_p^*$ ($\mathcal{B}$ can freely define $k_{\mathbf{u},t}$ since $k_{\mathbf{u},t}$ is the output of a PRF and $\mathcal{A}$ does not know $\text{HK}_{\mathbf{u}}$) and computing the temporary private key $d_{\mathbf{u},t} = (d_{\mathbf{u},t}^{\langle 1\rangle}, d_{\mathbf{u},t}^{\langle 2\rangle}, d_{\mathbf{u},t}^{\langle 3\rangle})$ as follows.

* If $I(\mathbf{u}) \neq 0 \bmod p$,

$$d_{\mathbf{u},t}^{\langle 1\rangle} = g_1^{-J(\mathbf{u})/I(\mathbf{u})} \left(u' \prod_{i \in \mathcal{U}_{\mathbf{u}}} u_i\right)^{r_{\mathbf{u}}} \left(u' \prod_{i \in \mathcal{W}_{\mathbf{u},t}} u_i\right)^{k_{\mathbf{u},t}},$$
$$d_{\mathbf{u},t}^{\langle 2\rangle} = g^{k_{\mathbf{u},t}}, \qquad d_{\mathbf{u},t}^{\langle 3\rangle} = g_1^{-1/I(\mathbf{u})} g^{r_{\mathbf{u}}};$$

* else if $I(\mathbf{w}_{\mathbf{u},t}) \neq 0 \bmod p$,

$$d_{\mathbf{u},t}^{\langle 1\rangle} = g_1^{-J(\mathbf{w}_{\mathbf{u},t})/I(\mathbf{w}_{\mathbf{u},t})} \left(u' \prod_{i \in \mathcal{U}_{\mathbf{u}}} u_i\right)^{r_{\mathbf{u}}} \left(u' \prod_{i \in \mathcal{W}_{\mathbf{u},t}} u_i\right)^{k_{\mathbf{u},t}},$$
$$d_{\mathbf{u},t}^{\langle 2\rangle} = g_1^{-1/I(\mathbf{w}_{\mathbf{u},t})} g^{k_{\mathbf{u},t}}, \qquad d_{\mathbf{u},t}^{\langle 3\rangle} = g^{r_{\mathbf{u}}}.$$

Therefore, to $\mathcal{A}$, all temporary private keys computed by $\mathcal{B}$ will be indistinguishable from the keys generated by a true challenger. If, on the other hand, $I(\mathbf{u}) \equiv I(\mathbf{w}_{\mathbf{u},t}) \equiv 0 \bmod p$, the above computation cannot be performed and the simulation will abort. To make the analysis of the simulation easier, we will force the simulation to abort whenever $I(\mathbf{u}) \equiv I(\mathbf{w}_{\mathbf{u},t}) \equiv 0 \bmod l_u$. As Extract Queries, it is easy to see that $I(\mathbf{u}) \equiv I(\mathbf{w}_{\mathbf{u},t}) \equiv 0 \bmod l_u$ implies that $I(\mathbf{u}) \equiv I(\mathbf{w}_{\mathbf{u},t}) \equiv 0 \bmod p$. Hence, $I(\mathbf{u}) \neq 0 \bmod p$ or $I(\mathbf{w}_{\mathbf{u},t}) \neq 0 \bmod p$ implies $I(\mathbf{u}) \neq 0 \bmod l_u$ or $I(\mathbf{w}_{\mathbf{u},t}) \neq 0 \bmod l_u$, so the former condition will be a sufficient requirement to ensure that a temporary private key for u can be constructed.

– Signcrypt Queries. Upon receiving a signcrypt query $\langle \mathbf{m}, \mathbf{a}, \mathbf{b}, t\rangle$, $\mathcal{B}$ aborts if $I(\mathbf{a}) \equiv I(\mathbf{w}_{\mathbf{a},t}) \equiv 0 \bmod p$. Otherwise, $\mathcal{B}$ searches r^{list} for tuple $(\mathbf{a}, r_{\mathbf{a}})$ (if r^{list} does not contain this tuple, it chooses $r_{\mathbf{a}} \overset{U}{\leftarrow} Z_p$ and adds $(\mathbf{a}, r_{\mathbf{a}})$ into r^{list}), picking $k_{\mathbf{a},t} \overset{U}{\leftarrow} Z_p^*$ and computes the temporary private key $d_{\mathbf{a},t} = (d_{\mathbf{a},t}^{\langle 1\rangle}, d_{\mathbf{a},t}^{\langle 2\rangle}, d_{\mathbf{a},t}^{\langle 3\rangle})$ as follows.

* If $I(\mathbf{a}) \neq 0 \bmod p$,

$$d_{\mathbf{a},t}^{\langle 1\rangle} = g_1^{-J(\mathbf{a})/I(\mathbf{a})} \left(u' \prod_{i \in \mathcal{U}_{\mathbf{a}}} u_i\right)^{r_{\mathbf{a}}} \left(u' \prod_{i \in \mathcal{W}_{\mathbf{a},t}} u_i\right)^{k_{\mathbf{a},t}},$$
$$d_{\mathbf{a},t}^{\langle 2\rangle} = g^{k_{\mathbf{a},t}}, \qquad d_{\mathbf{a},t}^{\langle 3\rangle} = g_1^{-1/I(\mathbf{a})} g^{r_{\mathbf{a}}};$$

* else if $I(\mathbf{w}_{\mathbf{a},t}) \neq 0 \bmod p$,

$$d_{\mathbf{a},t}^{\langle 1\rangle} = g_1^{-J(\mathbf{w}_{\mathbf{a},t})/I(\mathbf{w}_{\mathbf{a},t})}\left(u' \prod_{i\in\mathcal{U}_{\mathbf{a}}} u_i\right)^{r_{\mathbf{a}}}\left(u' \prod_{i\in\mathcal{W}_{\mathbf{a},t}} u_i\right)^{k_{\mathbf{a},t}},$$
$$d_{\mathbf{a},t}^{\langle 2\rangle} = g_1^{-1/I(\mathbf{w}_{\mathbf{a},t})} g^{k_{\mathbf{a},t}}, d_{\mathbf{a},t}^{\langle 3\rangle} = g^{r_{\mathbf{a}}}.$$

Then, $\mathcal{B}$ performs the Signcrypt algorithm to obtain the ciphertext (t, σ). As Temporary Private Key Queries, to make the analysis of the simulation easier, we will force the simulation to abort whenever $I(\mathbf{a}) \equiv I(\mathbf{w}_{\mathbf{a},t}) \equiv 0 \bmod l_u$.

– Unsigncrypt Queries. Upon receiving a unsigncrypt query $\langle (t, \sigma), \mathbf{a}, \mathbf{b}\rangle$, $\mathcal{B}$ aborts if $I(\mathbf{b}) \equiv I(\mathbf{w}_{\mathbf{b},t}) \equiv 0 \bmod p$. Otherwise, $\mathcal{B}$ searches r^{list} for tuple $(\mathbf{b}, r_{\mathbf{b}})$ (if r^{list} does not contain this tuple, it chooses $r_{\mathbf{b}} \overset{U}{\leftarrow} Z_p$ and adds $(\mathbf{b}, r_{\mathbf{b}})$ into r^{list}), picking $k_{\mathbf{b},t} \overset{U}{\leftarrow} Z_p^*$ and computes the temporary private key $d_{\mathbf{b},t} = (d_{\mathbf{b},t}^{\langle 1\rangle}, d_{\mathbf{b},t}^{\langle 2\rangle}, d_{\mathbf{b},t}^{\langle 3\rangle})$ as follows:

  * If $I(\mathbf{b}) \neq 0 \bmod p$,

$$d_{\mathbf{b},t}^{\langle 1\rangle} = g_1^{-J(\mathbf{b})/I(\mathbf{b})}\left(u' \prod_{i\in\mathcal{U}_{\mathbf{b}}} u_i\right)^{r_{\mathbf{b}}}\left(u' \prod_{i\in\mathcal{W}_{\mathbf{b},t}} u_i\right)^{k_{\mathbf{b},t}},$$
$$d_{\mathbf{b},t}^{\langle 2\rangle} = g^{k_{\mathbf{b},t}}, \qquad d_{\mathbf{b},t}^{\langle 3\rangle} = g_1^{-1/I(\mathbf{b})} g^{r_{\mathbf{b}}};$$

  * else if $I(\mathbf{w}_{\mathbf{b},t}) \neq 0 \bmod p$,

$$d_{\mathbf{b},t}^{\langle 1\rangle} = g_1^{-J(\mathbf{w}_{\mathbf{b},t})/I(\mathbf{w}_{\mathbf{b},t})}\left(u' \prod_{i\in\mathcal{U}_{\mathbf{b}}} u_i\right)^{r_{\mathbf{b}}}\left(u' \prod_{i\in\mathcal{W}_{\mathbf{b},t}} u_i\right)^{k_{\mathbf{b},t}},$$
$$d_{\mathbf{b},t}^{\langle 2\rangle} = g_1^{-1/I(\mathbf{w}_{\mathbf{b},t})} g^{k_{\mathbf{b},t}}, \qquad d_{\mathbf{b},t}^{\langle 3\rangle} = g^{r_{\mathbf{b}}}.$$

Then, $\mathcal{B}$ performs the Unsigncrypt algorithm to obtain the message $\mathbf{m}$ and the verification result. As Temporary Private Key Queries, to make the analysis of the simulation easier, we will force the simulation to abort whenever $I(\mathbf{b}) \equiv I(\mathbf{w}_{\mathbf{b},t}) \equiv 0 \bmod l_u$.

- **Challenge**: $\mathcal{A}$ next will submit a period index t^*, two messages $\mathbf{m}_0, \mathbf{m}_1 \in G_T$ and two identities $\mathbf{a}^*, \mathbf{b}^*$ on which he wishes to challenge. $\mathcal{B}$ will flip a fair coin, γ, and construct the signcryption ciphertext as follows. $\mathcal{B}$ picks $\mathbf{r}^* \overset{U}{\leftarrow} \{0,1\}^{n_v}$ such that $\mathbf{a}^* \| \mathbf{m}_\gamma \| \mathbf{r}^* \in \Gamma$ and generates $\{j \in Z\colon H_v(\mathbf{m}_\gamma)[j] \oplus \mathbf{r}^*[j] = 1\} \rightarrow \mathcal{M}_{\mathbf{m}}$. $\mathcal{B}$ will abort if $I(\mathbf{b}^*) = I(\mathbf{w}_{\mathbf{b}^*,t^*}) = 0 \bmod p$ does not hold or $I(\mathbf{a}^*) \neq 0 \bmod p \wedge I(\mathbf{w}_{\mathbf{a}^*,t^*}) \neq 0 \bmod p$ or $I(\mathbf{a}^*) = 0 \bmod p \wedge I(\mathbf{w}_{\mathbf{a}^*,t^*}) = 0 \bmod p$ or $K(\mathbf{m}_\gamma \| \mathbf{r}^*) \neq 0 \bmod p$. Otherwise, $\mathcal{B}$ searches r^{list} for tuple $(\mathbf{a}^*, r_{\mathbf{a}^*})$ (if r^{list} does not contain this tuple, it chooses $r_{\mathbf{a}^*} \overset{U}{\leftarrow} Z_p$ and adds $(\mathbf{a}^*, r_{\mathbf{a}^*})$ into r^{list}), picks $k_{\mathbf{a}^*,t^*} \overset{U}{\leftarrow} Z_p^*$ and sets the signcryption ciphertext as $(t^*, \sigma^*) =$

$(t^*, (\sigma^{*\langle 1\rangle}, \sigma^{*\langle 2\rangle}, \sigma^{*\langle 3\rangle}, \sigma^{*\langle 4\rangle}, \sigma^{*\langle 5\rangle}, \sigma^{*\langle 6\rangle}, \sigma^{*\langle 7\rangle}))$, where

$$\sigma^{*\langle 1\rangle} = \mathcal{Z}V(\mathbf{a}^*\|\mathbf{m}_\gamma\|\mathbf{r}^*), \sigma^{*\langle 2\rangle} = C, \sigma^{*\langle 3\rangle} = C^{J(\mathbf{b}^*)}, \sigma^{*\langle 4\rangle} = C^{J(\mathbf{w}_{\mathbf{b}^*,t^*})}.$$

If $I(\mathbf{a}^*) \neq 0 \bmod p \wedge I(\mathbf{w}_{\mathbf{a}^*,t^*}) = 0 \bmod p$, set

$$\begin{aligned}\sigma^{*\langle 5\rangle} &= g_1^{-J(\mathbf{a}^*)/I(\mathbf{a}^*)}\left(u'\prod_{i\in\mathcal{U}_{\mathbf{a}^*}} u_i\right)^{r^*_{\mathbf{a}}}\left(u'\prod_{i\in\mathcal{W}_{\mathbf{a}^*,t^*}} u_i\right)^{k_{\mathbf{a}^*,t^*}} \\ &\quad \times C^{J(\mathbf{w}_{\mathbf{a}^*,t^*})}C^{L(\mathbf{m}_\gamma\|\mathbf{r}^*)}, \\ \sigma^{*\langle 6\rangle} &= g^{k_{\mathbf{a}^*,t^*}}, \qquad \sigma^{*\langle 7\rangle} = g_1^{-1/I(\mathbf{a}^*)}g^{r^*_{\mathbf{a}}}.\end{aligned}$$

If $I(\mathbf{a}^*) = 0 \bmod p \bigwedge I(\mathbf{w}_{\mathbf{a}^*,t}) \neq 0 \bmod p$, set

$$\begin{aligned}\sigma^{*\langle 5\rangle} &= g_1^{-J(\mathbf{w}_{\mathbf{a}^*,t})/I(\mathbf{w}_{\mathbf{a}^*,t^*})}\left(u'\prod_{i\in\mathcal{U}_{\mathbf{a}^*}} u_i\right)^{r^*_{\mathbf{a}}}\left(u'\prod_{i\in\mathcal{W}_{\mathbf{a}^*,t^*}} u_i\right)^{k_{\mathbf{a}^*,t^*}} \\ &\quad \times C^{J(\mathbf{a}^*)}C^{L(\mathbf{m}_\gamma\|\mathbf{r}^*)}, \\ \sigma^{*\langle 6\rangle} &= g_1^{-1/I(\mathbf{w}_{\mathbf{a}^*,t})}g^{k_{\mathbf{a}^*,t^*}}, \qquad \sigma^{*\langle 7\rangle} = g^{r^*_{\mathbf{a}}}.\end{aligned}$$

Suppose that $\mathcal{B}$ was given a BDH tuple, that is $\mathcal{Z} = e(g,g)^{abc}$. Then we have

$$\begin{aligned}\sigma^{*\langle 1\rangle} &= e(g,g)^{abc}V(\mathbf{a}^*\|\mathbf{m}_\gamma\|\mathbf{r}^*) = e(g_1,g_2)^cV(\mathbf{a}^*\|\mathbf{m}_\gamma\|\mathbf{r}^*), \\ \sigma^{*\langle 2\rangle} &= g^c, \\ \sigma^{*\langle 3\rangle} &= g^{cJ(\mathbf{b}^*)} = \left(g_2^0g^{J(\mathbf{b}^*)}\right)^c = \left(g_2^{I(\mathbf{b}^*)}g^{J(\mathbf{b}^*)}\right)^c = \left(u'\prod_{i\in\mathcal{U}_{\mathbf{b}^*}} u_i\right)^c, \\ \sigma^{*\langle 4\rangle} &= g^{cJ(\mathbf{w}_{\mathbf{b}^*,t})} = \left(g_2^0g^{J(\mathbf{w}_{\mathbf{b}^*,t^*})}\right)^c = \left(g_2^{I(\mathbf{w}_{\mathbf{b}^*,t^*})}g^{J(\mathbf{w}_{\mathbf{b}^*,t^*})}\right)^c, \\ &= \left(u'\prod_{i\in\mathcal{W}_{\mathbf{b}^*,t^*}} u_i\right)^c.\end{aligned}$$

If $I(\mathbf{a}^*) \neq 0 \bmod p \wedge I(\mathbf{w}_{\mathbf{a}^*,t^*}) = 0 \bmod p$, let $\hat{r}_{\mathbf{a}^*} = r^*_{\mathbf{a}} - a/I(\mathbf{a}^*)$ and $c' = k_{\mathbf{a}^*,t^*} + c$. If $I(\mathbf{a}^*) = 0 \bmod p \wedge I(\mathbf{w}_{\mathbf{a}^*,t}) \neq 0 \bmod p$, let $\hat{k}_{\mathbf{a}^*,t} = k_{\mathbf{a}^*,t^*} - a/I(\mathbf{w}_{\mathbf{a}^*,t^*})$ and $c'' = r^*_{\mathbf{a}} + c$. We see that (t^*, σ^*) is a valid signcryption ciphertext of $\mathbf{m}_\gamma$. Otherwise, we have that $\mathcal{Z}$ is a random element of G_T. In that case the signcryption ciphertext will give no information about $\mathcal{B}$'s choice of γ.

- **Phase 2**: $\mathcal{A}$ issues the rest of queries as in Phase 1 with the restriction described in Definition 4. $\mathcal{B}$ responds to these queries for $\mathcal{A}$ in the same way as Phase 1.
- **Guess**: Finally, the adversary $\mathcal{A}$ outputs a guess γ' of γ. If $\gamma' = \gamma$ then $\mathcal{B}$ outputs 1 meaning $\mathcal{Z} = e(g,g)^{abc}$; otherwise, it outputs 0 meaning that $\mathcal{Z}$ is a random element in G_T.

This completes the description of the simulation. The probability analysis is very similar to Waters (2005).

Theorem 2. *Our IBKISC scheme is EU-IBKISC-KI-CMA secure in the standard model, assuming that the CDH assumption holds in groups G, the hash function H is collision-resistant, F is a pseudo-random function and V is a bijective function. Concretely, if there exists an EU-IBKISC-KI-CMA adversary $\mathcal{A}$ that is able to produce a forgery during the game defined in Definition* 5 *with advantage at least ϵ when running in time at most t and asking at most q_e extract queries, q_t temporary private key queries, q_s signcryption queries and q_u unsigncryption queries, there exists a challenger that can solve an instance of the CDH problem in time $t' < t + O((q_e + q_s + q_t + q_u)n_u t_m + (q_e + q_t + q_s)t_e + q_u t_p)$ with advantage*

$$\epsilon' > \frac{\epsilon}{27q_s(q_e + q_t + q_s)^2(n_u + 1)^2(n_v + 1)},$$

where t_m, t_e and t_p denote the same quantities as in Theorem 1.

Proof. Suppose there exists an adversary, $\mathcal{A}$, that can attack our scheme in the standard model. We build a simulator $\mathcal{B}$ that solves the CDH problem. $\mathcal{B}$ will be given a group G, a generator g and the elements g^a and g^b. To be able to use $\mathcal{A}$ to compute g^{ab}, $\mathcal{B}$ must be able to simulate a challenger for $\mathcal{A}$. Such a simulation can be created in the following way:

- **Setup**: This algorithm is the same as that of Theorem 1 except that $\mathcal{B}$ assigns $g_1 = g^a$ and $g_2 = g^b$.
- **Queries**: Extraction Queries, Temporary Private Key Queries, Signcrypt Queries and Unsigncrypt Queries are the same as those in the proof of Theorem 1.
- **Forgery**: Eventually, $\mathcal{A}$ returns two identities, $\mathbf{a}^*$ and $\mathbf{b}^*$, a forged ciphertext

  $$(t^*, \sigma^*) = \left(t^*, (\sigma^{*\langle 1\rangle}, \sigma^{*\langle 2\rangle}, \sigma^{*\langle 3\rangle}, \sigma^{*\langle 4\rangle}, \sigma^{*\langle 5\rangle}, \sigma^{*\langle 6\rangle}, \sigma^{*\langle 7\rangle})\right)$$

  on message $\mathbf{m}^*$, some $\mathbf{r}^* \in \{0,1\}^{n_v}$ such that $\mathbf{a}^* \| \mathbf{m}^* \| \mathbf{r}^* \in \Gamma$, with the constraint described in Definition 5. $\mathcal{B}$ generates $\{j \in Z\colon H_v(\mathbf{m}^*)[j] \oplus \mathbf{r}^*[j] = 1\} \rightarrow \mathcal{M}_{\mathbf{m}}$. $\mathcal{B}$ outputs $\perp$ and aborts if $I(\mathbf{a}^*) \equiv I(\mathbf{w}_{\mathbf{a}^*,t^*}) \equiv K(\mathbf{m}^* \| \mathbf{r}^*) \equiv 0 \bmod p$ does not hold. Otherwise, $\mathcal{B}$ can successfully derive g^{ab} as

  $$\frac{\sigma^{*\langle 5\rangle}}{\sigma^{*\langle 7\rangle\, J(\mathbf{a}^*)} \sigma^{*\langle 6\rangle\, J(\mathbf{w}_{\mathbf{a}^*,t^*})} \sigma^{*\langle 2\rangle\, L(\mathbf{m}^* \| \mathbf{r}^*)}}$$
  $$= \frac{g_2^{\alpha}(u' \prod_{i \in \mathcal{U}_{\mathbf{a}}} u_i)^{r_{\mathbf{a}}} (u' \prod_{i \in \mathcal{W}_{\mathbf{a},t}} u_i)^{r_t} (m' \prod_{i \in \mathcal{M}_{\mathbf{m}}} m_i)^{r_{\mathbf{m}}}}{g^{r_a J(\mathbf{a}^*)} g^{r_t J(\mathbf{w}_{\mathbf{a}^*,t^*})} g^{r_{\mathbf{m}} L(\mathbf{m}^* \| \mathbf{r}^*)}} = g^{ab},$$

  which is the solution to the given CDH problem.

Theorem 3. *Our IBKISC scheme is IND-IBKISC-SKI-CCA secure in the standard model, assuming that the DBDH assumption holds in groups (G, G_T), the hash function*

H is collision-resistant, F is a pseudo-random function and V is a bijective function. Concretely, if there exists an IND-IBKISC-SKI-CCA adversary $\mathcal{A}$ that is able to distinguish two valid ciphertexts during the game defined in Definition 6 *with advantage at least ϵ when running in time at most t and asking at most q_e extract queries, q_s signcryption queries and q_u unsigncryption queries, there exists a challenger that can solve an instance of the DBDH problem in time $t' < t+O((q_e+q_s+q_u)n_u t_m+(q_e+q_s)t_e+q_u t_p)$ and asking at most q_e extract queries, q_s extract queries and extract queries with advantage*

$$\epsilon' > \frac{\epsilon}{54q_s(q_e+q_s)^2(n_u+1)^2(n_v+1)},$$

where t_m, t_e and t_p denote the same quantities as in Theorem 1.

Proof. Suppose there exists a polynomial-time adversary, $\mathcal{A}$, that can attack our scheme in the standard model with advantage ϵ. We build a simulator $\mathcal{B}$ running in polynomial time that solves the DBDH problem with a non-negligible advantage ϵ'. $\mathcal{B}$ will take BDH challenge $(g, A = g^a, B = g^b, C = g^c, \mathcal{Z})$. To be able to use $\mathcal{A}$ to output a guess, β', as to whether the challenge is a BDH tuple, $\mathcal{B}$ must be able to simulate a challenger for $\mathcal{A}$. Such a simulation can be created in the following way:

- **Setup**: The same as Theorem 1.
- **Phase 1**: When running the adversary, extract, helper key, signcryption and unsigncryption queries can occur. $\mathcal{B}$ answers these in the following way:
  - Extraction Queries, Signcrypt Queries and Unsigncrypt Queries are the same as those in the proof of Theorem 1.
  - Helper Key Queries. Consider a query for the helper key of an identity $\mathbf{u}$. $\mathcal{B}$ searches HK^{list} for tuple $(\mathbf{u}, \text{HK}_{\mathbf{u}})$ (if HK^{list} does not contain this tuple, it chooses $\text{HK}_{\mathbf{u}} \overset{U}{\leftarrow} \{0,1\}^{\kappa}$ and adds $(\mathbf{u}, \text{HK}_{\mathbf{u}})$ into HK^{list}).
- **Challenge**: The same as Theorem 1.
- **Phase 2**: $\mathcal{A}$ issues the rest of queries as in Phase 1 with the restriction described in Definition 6. $\mathcal{B}$ responds to these queries for $\mathcal{A}$ in the same way as Phase 1.
- **Guess**: Finally, the adversary $\mathcal{A}$ outputs a guess γ' of γ. If $\gamma' = \gamma$ then $\mathcal{B}$ outputs 1 meaning $\mathcal{Z} = e(g,g)^{abc}$; otherwise, it outputs 0 meaning that $\mathcal{Z}$ is a random element in G_T.

Theorem 4. *Our IBKISC scheme is EU-IBKISC-SKI-CMA secure in the standard model, assuming that the CDH assumption holds in groups G, the hash function H is collision-resistant, F is a pseudo-random function and V is a bijective function. Concretely, if there exists an EU-IBKISC-SKI-CMA adversary $\mathcal{A}$ that is able to produce a forgery during the game defined in Definition* 7 *with with advantage at least ϵ when running in time at most t and asking at most q_e extract queries, q_s signcryption queries and q_u unsigncryption queries, there exists a challenger that can solve an instance of the CDH problem in time $t' < t + O((q_e + q_s + q_u)n_u t_m + (q_e + q_s)t_e + q_u t_p)$ and asking at most q_e extract*

queries, q_s extract queries and extract queries with advantage

$$\epsilon' > \frac{\epsilon}{27q_s(q_e+q_s)^2(n_u+1)^2(n_v+1)},$$

where t_m, t_e and t_p denote the same quantities as in Theorem 1.

Proof. Suppose there exists an adversary, $\mathcal{A}$, that can attack our scheme in the standard model. We build a simulator $\mathcal{B}$ that solves the CDH problem. $\mathcal{B}$ will be given a group G, a generator g and the elements g^a and g^b. To be able to use $\mathcal{A}$ to compute g^{ab}, $\mathcal{B}$ must be able to simulate a challenger for $\mathcal{A}$. Such. Such a simulation can be created in the following way:

- **Setup**: The same as Theorem 2.
- **Queries**: Extraction Queries, Signcrypt Queries and Unsigncrypt Queries are the same as those in the proof of Theorem 1. Help Key Queries is the same as that in the proof of Theorem 3.
- **Forgery**: Eventually, $\mathcal{A}$ returns a forged ciphertext (t^*, σ^*) with the constraint described in Definition 7. $\mathcal{B}$ can derive the g^{ab} in the same way as Theorem 2.

7. Comparisons

In this section, we compare our scheme with other IBKISC schemes. The computation time includes the number of pairings and exponential computation. We let t_p be computation time of pairings and t_e be the exponential computation time. As for the ciphertext size, we let $|t|$ be the number of bits required to represent a time period, $|ID|$ be the number of bits required to represent an identity, $|G|$ be the size of a G element and $|G_T|$ be the size of a G_T element.

In Table 1, we compare our scheme with the Sign-then-Encrypt(StE) and Encrypt-then-Sign(EtS) using a CPA (chosen plaintext attack)-secure IBKIE scheme in the standard model (Weng, 2006) and an IBKIS scheme in the standard model (Weng, 2008). Note that we can obtain a CPA-secure IBKIE scheme in the standard model from Weng (2006) by letting the number of helpers be 1. As we can see in Table 1, our proposed IBKISC scheme is the fastest with the shortest ciphertext size.

Table 1

A performance comparison with other IBKISC schemes

Scheme	Ciphertext size	Signcrypt time	Unsigncrypt time
E(CPA)tS	$1\|t\| + 7\|G\| + 1\|G_T\|$	$8t_e$	$7t_p$
StE(CPA)	$1\|t\| + 7\|G\| + 1\|G_T\|$	$8t_e$	$7t_p$
Ours(CCA)	$1\|t\| + 6\|G\| + 1\|G_T\| + 1\|ID\|$	$7t_e$	$7t_p$

8. Conclusion

Using Dodis *et al.* key-insulation mechanism, we propose an identity-based key-insulated signcryption (IBKISC) paradigm. Furthermore, we present a concrete IBKISC scheme. The proposed scheme is proved to be secure in the standard model.

Acknowlegments. Supported by the National Natural Science Foundation of China (Nos. 60970111, 60903189, 60903020 and 61103183), the National High Technology Research and Development Program (863) of China (No. 2009AA01Z418), the National Basic Research Program (973) of China (No. 2007CB311201) and the Foundation of NLMC (9140C1103020803).

References

Barreto, P.S.L.M., Libert, B., McCullagh, N., Quisquater, J.J. (2005). Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. *Asiacrypt 2005*, 515–532.

Boneh, D., Franklin, M. (2001). Identity based encryption from the weil pairing. In: *CRYPTO 2001*, pp. 213–229.

Boyen, X. (2003). Multipurpose identity-based signcryption: a swiss army knife for identity-based cryptography. In: *CRYPTO 2003*, pp. 383–399.

Chen, L., Malone-Lee, J. (2005) Improved identity-based signcryption. In: *PKC 2005*, pp. 362–379.

Chow, S.S.M., Yiu, S.M., Hui, L.C.K., Chow, K.P. (2003). Efficient forward and provably secure id-based signcryption scheme with public verifiability and public ciphertext authenticity. In: *ICISC 2003*, pp. 352–369.

Dodis, Y., Katz, J., Xu, S., Yung, M. (2002). Key-insulated public-key cryptosystem. In: *Eurocrypt 2002*, pp. 65–82.

Goldreich, E., Goldwasser, S., Micali, S. (1985). On the cryptographic applications of random functions. In: *Crypto 1985*, pp. 276–288.

Hanaoka, Y., Hanaoka, G., Shikata, J., Imai, H. (2005). Identity-based hierarchical strongly key-insulated encryption and its application. In: *Asiacrypt 2005*, pp. 495–514.

Jin, Z., Wen, Q., Du, H. (2010). An improved semantically-secure identity-based signcryption scheme in the standard model. *Computers and Electrical Engineering*, 36(3), 545–552.

Libert, B., Quisquater, J.J. (2003). New identity-based signcryption schemes from pairings. In: *ITW2003*, pp. 155–158.

Malone-Lee, J. (2002). Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098.

Nalla, D., Reddy, K.C. (2003). Signcryption scheme for identity-based cryptosystems. *Cryptology ePrint Archive*, Report 2003/066.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In: *CRYPTO 1984*, pp. 47–53.

Waters, B. (2005). Efficient identity-based encryption without random oracles. In: *Eurocrypt 2005*, pp. 114–127.

Weng, J., Liu, S., Chen, K., Ma, C. (2006). Identity-based parallel key-insulated encryption without random oracles: security notions and construction. In: *Indocrypt 2006*, pp. 409–423.

Weng, J., Chen, K., Liu, S., Li, X. (2008). Identity-based strong key-insulated signature without random oracles. *Journal of Software*, 19(6), 1555–1564.

Yu, Y., Yang, B., Sun, Y., Zhu, S. (2009). Identity based signcryption scheme without random oracles. *Computer Standards & Interfaces*, 31(1), 56–62 (2009)

Yuen, T.H., Wei, V.K. (2005). Fast and proven secure blind identity-based signcryption from pairings. In: *CT-RSA 2005*, pp. 305–322.

Zheng, Y. (1997). Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost(signature) + cost(encryption). In: *Crypto 1997*, pp. 165–179.

Zhou, Y., Cao, Z. (2006). Identity based key insulated signature. In: *ISPEC 2006*, pp. 226–234.

J.H. Chen received his MS and BS degrees in computer science and engineering from University of Science and Technology Liaoning, Anshan, China, in 2001 and 1994, respectively. He is currently a PhD candidate at Shanghai Jiao Tong University. His research interests include public key cryptosystem and pairing based cryptosystem.

K.F. Chen received his PhD 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.

Y.T. Wang received his MS degree in computer science and engineering from Xihua University, Chengdu, China, in 2007. He is currently a PhD candidate at Shanghai Jiao Tong University. His research interests include public key cryptosystem and key agreement protocol.

X.X. Li is currently an associate professor in the School of Information, East China Normal University, Shanghai. His current research interests include lightweight cryptography, applied cryptography, coding theory, disaster recovery, and information security.

Y. Long is a research associate of the Department of Computer Science and Engineering of the Shanghai Jiao Tong University. Her main research interests include threshold technique and public key encryption in the distributed network surrounding.

Z.M. Wan is a lecturer in the College of Science, Hohai University. Her research interests include the analysis, design, and application of digital signatures, and certificateless cryptography.

Tapatumu grįstas padalinto rakto šifravimas su pasirašymu

Jianhong CHEN, Kefei CHEN, Yongtao WANG, Xiangxue LI, Yu LONG, Zhongmei WAN

Padalinto į dvi dalis rakto kriptografija patikimai apsaugo privačiuosius raktus tapatumu grįstose šifravimo sistemose. Nepaisant sąmyšio kilusio dėl tapatumu grįsto padalinto rakto šifravimo (IBKIE) ir parašo (IBKIS) saugumo, šio metodo saugumo problema yra neišspręsta. Šios problemos sprendimui straipsnio autoriai siūlo tapatumu grįstą padalinto rakto šifravimo su pasirašymu schemą (IBKISC). Lyginant su „pasirašyk, o po to užšifruok“ (StE) ir su „užšifruok, o po to pasirašyk“ (EtS) schemomis, kurios naudojamos standartiniuose IBKIE ir IBKIS modeliuose, pasiūlytoji schema yra spartesnė, o užšifruotas tekstas yra trumpesnis.