Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Certificateless key-insulated signature without random oracles*

Zhong-mei WAN[†1,2], Xue-jia LAI[1], Jian WENG[3,4], Sheng-li LIU[1], Yu LONG[1], Xuan HONG[1]

([1]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*)

([2]*College of Science, Hohai University, Nanjing 210098, China*)

([3]*Department of Computer Science, Jinan University, Guangzhou 510630, China*)

([4]*School of Information Systems, Singapore Management University, Singapore 178902, Singapore*)

[†]E-mail: wanmei@sjtu.edu.cn

Received Oct. 15, 2008; Revision accepted Oct. 5, 2009; Crosschecked Aug. 14, 2009

Abstract: Leakage of the private key has become a serious problem of menacing the cryptosystem security. To reduce the underlying danger induced by private key leakage, Dodis *et al*.(2003) proposed the first key-insulated signature scheme. To handle issues concerning the private key leakage in certificateless signature schemes, we devise the first certificateless key-insulated signature scheme. Our scheme applies the key-insulated mechanism to certificateless cryptography, one with neither certificate nor key escrow. We incorporate Waters (2005)'s signature scheme, Paterson and Schuldt (2006)'s identity-based signature scheme, and Liu *et al*.(2007)'s certificateless signature scheme to obtain a certificateless key-insulated signature scheme. Our scheme has two desirable properties. First, its security can be proved under the non-pairing-based generalized bilinear Diffie-Hellman (NGBDH) conjecture, without utilizing the random oracle model; second, it solves the key escrow problem in identity-based key-insulated signatures.

Key words: Key-insulated, Key leakage, Certificateless, Bilinear map
doi:10.1631/jzus.A0820714 **Document code:** A **CLC number:** TP311

INTRODUCTION

Shamir (1984) introduced the concept of 'identity based cryptography' (IBC), in which any strings such as email addresses, server names or phone numbers, can be used as public keys. This greatly reduces the certificate cost of public key infrastructure (PKI) as needed in traditional public key cryptography. However, the private key generation of a user is fully controlled by a private key generator (PKG), which inevitably results in the key escrow of IBC. To handle this issue, Al-Riyami and Paterson (2003) described a new framework named 'certificateless public key cryptography' (CLPKC). Its main motivation is to get rid of the key-escrow issues in IBC without taking advantage of the certificates in certified public key cryptography (PKC). In CLPKC, any user's private key is composed of two parts: the partial key produced by the key generation center (KGC) and the secret key selected by the user. Hence, the KGC does not fully control any user's private key generation.

The first specific certificateless signature scheme was described by Al-Riyami and Paterson (2003). Since then, several others regarding certificateless signatures have been developed (Yum and Lee, 2004; Gorantla *et al*., 2005; Huang *et al*., 2005; Hu *et al*., 2006; Zhang *et al*., 2006; Liu *et al*., 2007).

Leakage of the private key has become a serious problem of menacing the cryptosystem security. To reduce the underlying danger induced by private key leakage, Dodis *et al*.(2003) introduced a new

* Project (Nos. 60573032, 60773092, 60842002, 60873229, and 90604036) supported by the National Natural Science Foundation of China

protection framework named 'key-insulated signature'. In this framework, a physically secure device called a 'helper' is used to keep a helper key. The lifetime of the signing key is separated into distinct period numbers. When a user needs to refresh his/her signing key, the helper produces an update key from its helper key and forwards the update key to the user. The user produces the new signing key of period number i from the update key and his/her signing key of period number j. The public key is not changed during the whole key updating. Since the helper is always supposed to be physically secure, revealing the signing key at some period numbers will not help a forger counterfeit signatures at other period numbers.

Since Dodis *et al*. proposed the first key-insulated encryption scheme (Dodis *et al*., 2002) and the first key-insulated signature (Dodis *et al*., 2003), there have been many other primitives described for encryption (Dodis and Yung, 2002; Hanaoka Y *et al*., 2002; Bellare and Palacio, 2006; Cheon *et al*., 2006; Hanaoka G *et al*., 2006) and signatures (González-Deleito *et al*., 2004; Le *et al*., 2004; Weng *et al*., 2006; 2007; Zhou *et al*., 2006). Zhou *et al*.(2006) addressed the issue of identity based key-insulated signature (IBKIS) and proposed a secure IBKIS scheme in the random oracle model. However, Canetti *et al*.(2004) showed that some secure cryptographic schemes in random oracle model may not be secure in actual applications. Later, Weng *et al*.(2007) presented the constructions of an IBKIS scheme without random oracles. However, their schemes cannot resolve the key escrow problem. Therefore, it is desirable to construct a certificateless key-insulated signature scheme without random oracles.

In this paper, we describe the first certificateless key-insulated signature scheme whose security is proven without making use of the random oracle model. We incorporate Waters (2005)'s signature scheme, Paterson and Schuldt (2006)'s identity-based signature scheme, and Liu *et al*.(2007)'s certificateless signature (CLS) scheme to obtain a certificateless key-insulated signature scheme. Our scheme has two desirable properties: first, its security can be proven under the non-pairing- based generalized bilinear Diffie-Hellman (NGBDH) conjecture, without utilizing the random oracle model; second, it solves the key escrow problem in identity-based key-insulated signatures.

PRELIMINARIES

Bilinear map

Here we briefly recall the bilinear map as presented in Waters (2005).

Suppose G_1, G_2 are two multiplicative cyclic groups of prime order q and G_1 is generated by g. A bilinear map is an efficiently computable map e: $G_1 \times G_1 \rightarrow G_2$, satisfying the following properties:

1. Bilinearity: $\forall g_1, g_2 \in G_1$ and $\forall a, b \in Z_q^*$,

$$e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}.$$

2. Non-degeneracy: $e(g, g) \neq 1$.

Complexity conjecture

Definition 1 (Non-pairing-based generalized bilinear Diffie-Hellman problem, NGBDH) (Liu *et al*., 2007) Provided $(g, g^a, g^b) \in G^3$ for some $a, b \in Z_q^*$, the NGBDH problem in group G is to compute (g^{abc}, g^c).

We define the advantage of an algorithm $\mathcal{B}$ in resolving the NGBDH problem as

$$\mathrm{Adv}(\mathcal{B})=\Pr[\mathcal{B}(g, g^a, g^b)=(g^{abc}, g^c)].$$

The (t, ε)-NGBDH conjecture remains valid in G if $\mathrm{Adv}(\mathcal{B})$ is less than ε for a t-algorithm $\mathcal{B}$.

Definition 2 (Many-DH problem) (Lysyanskaya, 2002) Provided $(g, g^a, g^b, g^c, g^{ab}, g^{ac}, g^{bc}) \in G^7$ for some $a, b, c \in Z_q^*$, the many-DH problem in group G is to compute g^{abc}.

We define the advantage of an algorithm $\mathcal{B}'$ in resolving the many-DH problem as

$$\mathrm{Adv}(\mathcal{B}')=\Pr[\mathcal{B}'(g, g^a, g^b, g^c, g^{ab}, g^{ac}, g^{bc})=g^{abc}].$$

The (t, ε)-many-DH conjecture remains valid in G if $\mathrm{Adv}(\mathcal{B}')$ is less than ε for a t-algorithm $\mathcal{B}'$.

FRAMEWORK OF CERTIFICATELESS KEY-INSULATED SIGNATURE

Syntax

A certificateless key-insulated signature (CLKIS) scheme consists of eight tuples of polynomial time algorithms as follows:

1. Setup. A KGC selects a security parameter 1^k, and returns a master key msk and system parameters mpk.

2. UserKeyGen. A user supplies mpk and an identity ID and returns a secret key usk_{ID} and a public key upk_{ID}.

3. ExtractPartialKey. A KGC inputs mpk, msk and ID and returns a partial key psk_{ID}.

4. Gen. A user supplies mpk, usk_{ID} and psk_{ID} and computes his/her preliminary signing key $x_{\text{ID},0}$ and a helper key hk_{ID}.

5. CL-Update*. A helper inputs mpk, ID, hk_{ID} and a period number t and returns an update key $\text{uk}_{\text{ID},t}$ of period number t.

6. CL-Update. A user takes mpk, ID, x_{ID,t_2} and $\text{uk}_{\text{ID},t_1}$ as input, and outputs the signing key x_{ID,t_1} of period number t_1.

7. CL-Sign. A signer provides mpk, t, $x_{\text{ID},t}$ and a message m and returns a signature (t, s).

8. CL-Ver. A user supplies m, ID and (t, s) and outputs 1 if (t, s) is a valid signature, and 0 otherwise.

Consistency requires that CL-Ver((t, s), ID, m, upk_{ID})=1, whenever $\forall t\in\{1, 2, \ldots, N\}$, $m\in\{0, 1\}^*$, $\forall \text{ID}\in\{0, 1\}^*$, (t, s)=CL-Sign(t, m, $x_{\text{ID},t}$).

Security notions

Similar to Al-Riyami and Paterson (2003), we discuss the security notions for a certificateless key-insulated signature scheme according to two kinds of forgers: a type I forger $\mathcal{A}_{\text{I}}$ and a type II forger $\mathcal{A}_{\text{II}}$. $\mathcal{A}_{\text{I}}$ does not possess the master key but can substitute the public key of any user with a new one he/she picks. $\mathcal{A}_{\text{II}}$ can corrupt the master key, which means that it can compute any user's partial key by itself, but cannot perform any public key replacement.

Like Hu *et al*.(2006), the KGC is supposed to produce the master key in the same way as the scheme specifies.

Definition 3 A CLKIS scheme Π is key-insulated-secure unforgeable against chosen message attacks, if the advantage of any forger $\mathcal{A}$ in the following experiment is negligible:

1. The challenger $\mathcal{B}$ executes Setup to produce msk and mpk. $\mathcal{B}$ sends mpk to $\mathcal{A}$ while keeping msk secret if $\mathcal{A}$ is a type I forger; otherwise, $\mathcal{A}$ is equipped with msk.

2. $\mathcal{A}$ performs the following queries:

(1) RequestPublicKey queries. $\mathcal{A}$ provides an identity ID, and $\mathcal{B}$ replies with the public key upk_{ID}.

(2) RequestSecretKey queries. $\mathcal{A}$ provides an identity ID, and $\mathcal{B}$ replies with the secret key usk_{ID} if the ID's public key is not substituted.

(3) RequestPartialkey queries. $\mathcal{A}$ provides an identity ID, and $\mathcal{B}$ replies with the partial key psk_{ID}.

(4) ReplacePublicKey queries. $\mathcal{A}$ provides a public key upk_{ID}^* and an identity ID, and the public key for ID is substituted with upk_{ID}^* by $\mathcal{B}$.

(5) RequestSigningKey queries. $\mathcal{A}$ provides a period number t and an identity ID, and $\mathcal{B}$ responds with the signing key $x_{\text{ID},t}$.

(6) Sign queries. $\mathcal{A}$ provides a period number t, an identity ID, and a message m, and $\mathcal{B}$ responds with a signature s.

$\mathcal{A}$ returns $(\text{ID}^*, \text{upk}_{\text{ID}*}, t^*, s^*, m^*)$ such that the following conditions hold:

(1) If $\mathcal{A}$ is a type I forger, it is not permitted to ask for both a RequestPartialKey query and a ReplacePublicKey query or a RequestSecretKey query on ID^*.

(2) If $\mathcal{A}$ is a type II forger, it is not permitted to ask for a RequestSecretKey query on ID^*.

(3) $\mathcal{A}$ is not permitted to ask for a RequestSigningKey query on (ID^*, t^*).

(4) $\mathcal{A}$ is not permitted to ask for a Sign query on (ID^*, t^*).

(5) CL-Ver(mpk, ID^*, $\text{upk}_{\text{ID}*}$, t^*, s^*, m^*)=1. We define $\mathcal{A}$'s advantage as its probability of success.

Our proposed scheme

Choose groups G_1, G_2 defined as above such that a bilinear map e: $G_1\times G_1\rightarrow G_2$ can be constructed. Define a cryptographic hash function H: $\{0, 1\}^*\rightarrow \{0, 1\}^n$.

1. Setup.

(1) Randomly choose $\alpha \in Z_q^*$ and $g_2 \in G_1$. Assign $g_1=g^{\alpha}$. Furthermore, choose a vector $\boldsymbol{V}=(v_i)\in_R G_1^{n+1}$.

(2) Define a function f by $f(W)=v_0\prod_{i\in W}v_i$, where $W\subseteq\{1, 2, \ldots, n\}$.

(3) Output $\text{msk}=g_2^{\alpha}$ and mpk=$\{g, g_1, g_2, q, \boldsymbol{V}, f, H\}$.

2. UserKeyGen. Pick $x\in_R Z_q^*$, set $\text{usk}_{\text{ID}}=x$ and compute $\text{upk}_{\text{ID}}=(g^x, g_1^x)=(\text{upk}_{\text{ID}}^{(1)}, \text{upk}_{\text{ID}}^{(2)})$.

3. ExtractPartialKey. Compute $V_{\text{ID}}=H(\text{ID})$ and define $\mathcal{V}_{\text{ID}}=\{i\,|\,V_{\text{ID}}[i]=1,\ i\in\{1,2,\cdots,n\}\}$. The KGC selects $d_u\in_R Z_q^*$ and computes the user ID's partial key as $\text{psk}_{\text{ID}}=(\text{psk}_{\text{ID}}^{(1)},\text{psk}_{\text{ID}}^{(2)})=(g_2^{\alpha}f(\mathcal{V}_{\text{ID}})^{d_u}, g^{d_u})$.

4. Gen. The user does the following:

(1) Pick $d_v\in_R Z_q^*$, and compute $a_{\text{ID}}=(\text{psk}_{\text{ID}}^{(2)})^x g^{d_v}$ and $\text{hk}_{\text{ID}}=(\text{psk}_{\text{ID}}^{(1)})^x f(\mathcal{V}_{\text{ID}})^{d_v}$.

(2) Pick $b_{\text{ID},0}$, $c_{\text{ID},0}\in_R G_1$, and define the preliminary signing key as $x_{\text{ID},0}=(a_{\text{ID}}, b_{\text{ID},0}, c_{\text{ID},0})$.

(3) Output $x_{\text{ID},0}$ and a helper key hk_{ID}.

5. CL-Update*. Supplying a period number t and an identity ID, the helper executes the following:

(1) Pick $d_t\in_R Z_q^*$, and set $\hat{c}_{\text{ID},t}=g^{d_t}$.

(2) Compute $V_{\text{ID},t}=H(\text{ID},t)$.

(3) Let $\mathcal{V}_{\text{ID},t}=\{i\,|\,V_{\text{ID},t}[i]=1,\ i\in\{1,2,\cdots,n\}\}$.

(4) Compute $\hat{b}_{\text{ID},t}=\text{hk}_{\text{ID}}f(\mathcal{V}_{\text{ID},t})^{d_t}$.

(5) Output the update key $\text{uk}_{\text{ID},t}=(\hat{b}_{\text{ID},t},\hat{c}_{\text{ID},t})$.

6. CL-Update. On receiving a period number t_1, an update key $\text{uk}_{\text{ID},t_1}$ and a signing key x_{ID,t_2}, user ID generates the following signing key for the period number t_1:

(1) Parse $x_{\text{ID},t_2}=(a_{\text{ID}},b_{\text{ID},t_2},c_{\text{ID},t_2})$ and $\text{uk}_{\text{ID},t_1}=(\hat{b}_{\text{ID},t_1}, \hat{c}_{\text{ID},t_1})$.

(2) Set $b_{\text{ID},t_1}=\hat{b}_{\text{ID},t_1}$, $c_{\text{ID},t_1}=\hat{c}_{\text{ID},t_1}$.

(3) Output the signing key $x_{\text{ID},t_1}=(a_{\text{ID}},b_{\text{ID},t_1}, c_{\text{ID},t_1})$.

7. CL-Sign. Providing a message m, a period number t and a signing key $x_{\text{ID},t}$, user ID computes the signature as follows:

(1) Parse $x_{\text{ID},t}=(a_{\text{ID}}, b_{\text{ID},t}, c_{\text{ID},t})$.

(2) Define $M=H(m)$ and put $\mathcal{M}=\{i\,|\,M[i]=1,\ i\in\{1,2,...,n\}\}$.

(3) Pick $d_m\in_R Z_q^*$, and compute $D=g^{d_m}$ and $B=b_{\text{ID},t}f(\mathcal{M})^{d_m}$.

(4) Return the final signature $(t, s)=(t, (D, B, a_{\text{ID}}, c_{\text{ID},t}))$.

8. CL-Ver. When receiving a message m, a signature (t, s) and user ID's public key $\text{upk}_{\text{ID}}=(\text{upk}_{\text{ID}}^{(1)}, \text{upk}_{\text{ID}}^{(2)})$, this algorithm performs the following:

(1) Parse $s=(D, B, a_{\text{ID}}, c_{\text{ID},t})$.

(2) Let $M=H(m)$, $V_{\text{ID},t}=H(\text{ID}, t)$, and $V_{\text{ID}}=H(\text{ID})$.

(3) Let $\mathcal{M}$, $\mathcal{V}_{\text{ID},t}$, $\mathcal{V}_{\text{ID}}$ be the sets as ever described.

(4) Test if $e(\text{upk}_{\text{ID}}^{(1)}, g_1)=e(\text{upk}_{\text{ID}}^{(2)}, g)$ and

$$e(B,g)=e(g_2,\text{upk}_{\text{ID}}^{(2)})e(f(\mathcal{V}_{\text{ID}}),a_{\text{ID}})e(f(\mathcal{V}_{\text{ID},t}),c_{\text{ID},t})\cdot e(f(\mathcal{M}),D)$$

hold. If both hold, return 1; else, return 0.

Security analysis

Theorem 1 The proposed scheme is (t, ε)-key-insulated-secure unforgeable against type I forgers, supposing that the (t', ε')-NGBDH conjecture remains valid in G_1, where

$$\varepsilon'\geq\frac{27\varepsilon}{256(t_{\text{pk}}+t_{\text{sk}}+t_{\text{s}})^3(n+1)^3},$$

$$t'=t+O[(t_{\text{pk}}+t_{\text{sk}}+t_{\text{s}})n\rho+(t_{\text{ps}}+t_{\text{pk}}+t_{\text{sk}}+t_{\text{s}})\tau].$$

Here t_{pk}, t_{ps}, t_{sk}, and t_{s} represent the numbers of asking for RequestPublicKey queries and RequestSecretKey queries altogether, RequestPartialKey queries, RequestSigningKey queries, and Sign queries, respectively, while ρ and τ represent the computational cost of an exponentiation and a multiplication in G_1, respectively.

Proof Suppose $\mathcal{A}_{\text{I}}$ is a type I forger against our proposed scheme. We will show how to produce another t'-time algorithm $\mathcal{B}$ which can solve the NGBDH problem with an advantage of at least ε' with the help of $\mathcal{A}_{\text{I}}$. We here perform the method analogous to that of Liu *et al.*(2007).

Presume $\mathcal{B}$ is given (g^a, g^b) as an instance of the NGBDH problem. $\mathcal{B}$ acts as a challenger and replies to the queries for $\mathcal{A}_\mathrm{I}$ by interplaying with $\mathcal{A}_\mathrm{I}$ as the following.

1. Setup. $\mathcal{B}$ generates the system parameters mpk for $\mathcal{A}_\mathrm{I}$ as follows:

(1) Put $k=4(t_{pk}+t_{sk}+t_s)/3$ and $k(n+1)<q$ and randomly pick $v\in\{0, 1, ..., n\}$. Without loss of generality, we here suppose that $3|(t_{pk}+t_{sk}+t_s)$, since otherwise more queries are performed to satisfy this condition.

(2) Pick two vectors $\boldsymbol{X}=(x_i)\in Z_l^{n+1}$ and $\boldsymbol{Y}=(y_i)\in Z_q^{n+1}$.

(3) Define $g_1=g^a$, $g_2=g^b$, and a vector $\boldsymbol{V}=(v_i)\in G_1^{n+1}$ with $v_0=g_2^{-kv+x_0}g^{y_0}$ and $v_i=g_2^{x_i}g^{y_i}$ for $i=1, 2, \ldots, n$.

(4) For $\mathcal{W}\subseteq\{1,2,\ldots,n\}$, we assign

$$S_1(\mathcal{W})=-kv+x_0+\sum_{i\in\mathcal{W}}x_i,$$
$$S_2(\mathcal{W})=y_0+\sum_{i\in\mathcal{W}}y_i,$$
$$f(\mathcal{W})=g_2^{S_1(\mathcal{W})}g^{S_2(\mathcal{W})}.$$

It is easy to show that $g_2^{\alpha}=g_2^a=g^{ab}$, and $f(\mathcal{W})=v_0\prod_{i\in\mathcal{W}}v_i$.

(5) Send mpk=$(q, g, g_1, g_2, \boldsymbol{V}, f, H)$ to $\mathcal{A}_\mathrm{I}$.

Challenger $\mathcal{B}$ replies to $\mathcal{A}_\mathrm{I}$'s queries in the attack phase of the game as follows:

2. RequestPublicKey queries. $\mathcal{B}$ has a list L_{list} of records (ID, $\mathrm{upk_{ID}}$, $\mathrm{usk_{ID}}$). When $\mathcal{A}_\mathrm{I}$ provides an identity ID, $\mathcal{B}$ tests if a record (ID, $\mathrm{upk_{ID}}$, $\mathrm{usk_{ID}}$) exists in L_{list}. If no, $\mathcal{B}$ runs algorithm UserKeyGen to obtain $\mathrm{upk_{ID}}/\mathrm{usk_{ID}}$. It adds the record (ID, $\mathrm{upk_{ID}}$, $\mathrm{usk_{ID}}$) to L_{list} and sends $\mathrm{upk_{ID}}$ to $\mathcal{A}_\mathrm{I}$.

3. RequestSecretKey queries. When $\mathcal{A}_\mathrm{I}$ provides an identity ID, $\mathcal{B}$ tests if a record (ID, $\mathrm{upk_{ID}}$, $\mathrm{usk_{ID}}$) exists in L_{list}. If no, $\mathcal{B}$ runs algorithm UserKeyGen to obtain $\mathrm{upk_{ID}}/\mathrm{usk_{ID}}$. It adds the record (ID, $\mathrm{upk_{ID}}$, $\mathrm{usk_{ID}}$) to L_{list} and sends $\mathrm{usk_{ID}}$ to $\mathcal{A}_\mathrm{I}$.

4. RequestPartialKey queries. When $\mathcal{A}_\mathrm{I}$ furnishes an identity ID, $\mathcal{B}$ first computes $V_{ID}=H(ID)$. Let $\mathcal{V}_{ID}$ represent the set as ever described. If $S_1(\mathcal{V}_{ID})\neq 0 \bmod q$, $\mathcal{B}$ picks $d_u\in_R Z_q^*$, and sets $\mathrm{psk_{ID}}=(\mathrm{psk_{ID}^{(1)}},\mathrm{psk_{ID}^{(2)}})=(g_1^{-S_2(\mathcal{V}_{ID})/S_1(\mathcal{V}_{ID})}f(\mathcal{U}_{ID})^{d_u}, g_1^{-1/S_1(\mathcal{V}_{ID})}g^{d_u})$. Note that by setting $\tilde{d}_u=d_u-a/S_1(\mathcal{V}_{ID})$, we can verify the validity of $\mathrm{psk_{ID}}$ as follows:

$$\begin{aligned}\mathrm{psk_{ID}^{(1)}}&=g_1^{-S_2(\mathcal{V}_{ID})/S_1(\mathcal{V}_{ID})}f(\mathcal{V}_{ID})^{d_u}\\&=g_2^{a}(g_2^{S_1(\mathcal{V}_{ID})}g^{S_2(\mathcal{V}_{ID})})^{-a/S_1(\mathcal{V}_{ID})}(g_2^{S_1(\mathcal{V}_{ID})}g^{S_2(\mathcal{V}_{ID})})^{d_u}\\&=g_2^{a}(g_2^{S_1(\mathcal{V}_{ID})}g^{S_2(\mathcal{V}_{ID})})^{d_u-a/S_1(\mathcal{V}_{ID})}\\&=g_2^{a}(f(\mathcal{V}_{ID}))^{\tilde{d}_u},\\\mathrm{psk_{ID}^{(2)}}&=g_1^{-1/S_1(\mathcal{V}_{ID})}g^{d_u}=g^{d_u-a/S_1(\mathcal{V}_{ID})}=g^{\tilde{d}_u}.\end{aligned}$$

If $S_1(\mathcal{V}_{ID})=0 \bmod q$, $\mathcal{B}$ halts. For simplicity, we assume that $\mathcal{B}$ will halt if $S_1(\mathcal{V}_{ID})=0 \bmod k$. Since we can deduce kv, $x_0+\sum_{i\in\mathcal{V}_{ID}}\hat{x}_i\in[0,q)$ from $k(n+1)<q$, we then derive $S_1(\mathcal{V}_{ID})\in(-q,q)$, which infers if $S_1(\mathcal{V}_{ID})=0 \bmod q$, then $S_1(\mathcal{V}_{ID})=0 \bmod k$. Hence, $S_1(\mathcal{V}_{ID})\neq 0 \bmod k$ infers $S_1(\mathcal{V}_{ID})\neq 0 \bmod q$. Therefore, if $S_1(\mathcal{V}_{ID})\neq 0 \bmod k$, we can compute a partial key without halting.

5. ReplacePublicKey queries. $\mathcal{A}_\mathrm{I}$ may substitute public keys with new ones that it selects and $\mathcal{B}$ records all these changes.

6. RequestSigningKey queries. When $\mathcal{A}_\mathrm{I}$ supplies an identity ID and a period number t, $\mathcal{B}$ first computes $V_{ID}=H(ID)$, $V_{ID,t}=H(ID, t)$ and then searches L_{list} to test if the user ID's public key is the original one. If no, assume the new public key is $(\mathrm{upk_{ID}^{(1)}}, \mathrm{upk_{ID}^{(2)}})$ and assume $S_1(\mathcal{V}_{ID,t})\neq 0 \bmod k$. Using the aforementioned argument, it infers $S_1(\mathcal{V}_{ID,t})\neq 0 \bmod q$, given the conjecture $k(n+1)<q$. $\mathcal{B}$ computes the signing key as follows: picking $d_u, d_t\in_R Z_q^*$ and defining

$$a_{ID}=g^{d_u},\ c_{ID,t}=(\mathrm{upk_{ID}^{(2)}})^{-1/S_1(\mathcal{V}_{ID,t})}g^{d_t},$$

$$b_{\mathrm{ID},t}=(\mathrm{upk}_{\mathrm{ID}}^{(2)})^{-S_2(\mathcal{V}_{\mathrm{ID},t})/S_1(\mathcal{V}_{\mathrm{ID},t})}f(\mathcal{V}_{\mathrm{ID}})^{d_u}f(\mathcal{V}_{\mathrm{ID},t})^{d_t},$$

$\mathcal{B}$ returns the signing key $x_{\mathrm{ID},t}=(a_{\mathrm{ID}}, b_{\mathrm{ID},t}, c_{\mathrm{ID},t})$ to $\mathcal{A}_{\mathrm{I}}$. Note that if $S_1(\mathcal{V}_{\mathrm{ID},t})=0 \bmod k$, $\mathcal{B}$ halts.

The validity can be verified as follows:

$$\begin{aligned}b_{\mathrm{ID},t}&=(\mathrm{upk}_{\mathrm{ID}}^{(2)})^{-S_2(\mathcal{V}_{\mathrm{ID},t})/S_1(\mathcal{V}_{\mathrm{ID},t})}f(\mathcal{V}_{\mathrm{ID}})^{d_u}f(\mathcal{V}_{\mathrm{ID},t})^{d_t}\\&=g_2^{ax}(g_2^{S_1(\mathcal{V}_{\mathrm{ID},t})}g^{S_2(\mathcal{V}_{\mathrm{ID},t})})^{\frac{-ax}{S_1(\mathcal{V}_{\mathrm{ID},t})}}(g_2^{S_1(\mathcal{V}_{\mathrm{ID},t})}g^{S_2(\mathcal{V}_{\mathrm{ID},t})})^{d_t}f(\mathcal{V}_{\mathrm{ID}})^{d_u}\\&=g_2^{ax}(g_2^{S_1(\mathcal{V}_{\mathrm{ID},t})}g^{S_2(\mathcal{V}_{\mathrm{ID},t})})^{\tilde{d}_t}f(\mathcal{V}_{\mathrm{ID}})^{d_u},\\c_{\mathrm{ID},t}&=(\mathrm{upk}_{\mathrm{ID}}^{(2)})^{-1/S_1(\mathcal{V}_{\mathrm{ID},t})}g^{d_t}=g^{\tilde{d}_t},\end{aligned}$$

where $\tilde{d}_t=d_t-ax/S_1(\mathcal{V}_{\mathrm{ID},t})$. The signing key produced above is identical to the actual construction.

If the user ID's public key is the original one, $\mathcal{B}$ produces the signing key in terms of the following two situations:

(1) If $S_1(\mathcal{V}_{\mathrm{ID}})\neq 0 \bmod k$, $\mathcal{B}$ first performs the above simulation for RequestPartialKey queries, and then tests if a record (ID, $\mathrm{upk}_{\mathrm{ID}}$, $\mathrm{usk}_{\mathrm{ID}}$) exists in L_{list}. If no, $\mathcal{B}$ runs algorithm UserKeyGen to obtain $\mathrm{upk}_{\mathrm{ID}}/\mathrm{usk}_{\mathrm{ID}}$. It adds the record (ID, $\mathrm{upk}_{\mathrm{ID}}$, $\mathrm{usk}_{\mathrm{ID}}$) to L_{list}. Otherwise, it executes algorithm CL-Update to generate a signing key on (ID, t).

(2) If $S_1(\mathcal{V}_{\mathrm{ID}})=0 \bmod k$ and $S_1(\mathcal{V}_{\mathrm{ID},t})\neq 0 \bmod k$ (this infers $S_1(\mathcal{V}_{\mathrm{ID},t})\neq 0 \bmod q$, given the conjecture $k(n+1)<q$), $\mathcal{B}$ picks $d_u, d_t \in_R Z_q^*$, obtains $\mathrm{usk}_{\mathrm{ID}}$ from L_{list} (if it does not exist in L_{list}, $\mathcal{B}$ first executes the algorithm UserKeyGen to generate it), sets $x=\mathrm{usk}_{\mathrm{ID}}$, and produces the signing key as

$$\begin{aligned}x_{\mathrm{ID},t}&=(g^{d_u}, g_1^{-xS_2(\mathcal{V}_{\mathrm{ID},t})/S_1(\mathcal{V}_{\mathrm{ID},t})}f(\mathcal{V}_{\mathrm{ID}})^{d_u}f(\mathcal{V}_{\mathrm{ID},t})^{d_t x},\\&\quad g_1^{-x/S_1(\mathcal{V}_{\mathrm{ID},t})}g^{d_t x})\\&=(g^{d_u}, g_2^{ax}(g_2^{S_1(\mathcal{V}_{\mathrm{ID},t})}g^{S_2(\mathcal{V}_{\mathrm{ID},t})})^{\tilde{d}_t}f(\mathcal{V}_{\mathrm{ID}})^{d_u}, g^{\tilde{d}_t})\\&=(a_{\mathrm{ID}}, b_{\mathrm{ID},t}, c_{\mathrm{ID},t}),\end{aligned}$$

where $\tilde{d}_t=d_t x-ax/S_1(\mathcal{V}_{\mathrm{ID},t})$.

If $S_1(\mathcal{V}_{\mathrm{ID},t})=0 \bmod k$, $\mathcal{B}$ will halt.

7. Sign queries. When $\mathcal{A}_{\mathrm{I}}$ supplies an identity ID and a period number t, $\mathcal{B}$ computes $V_{\mathrm{ID}}=H(\mathrm{ID})$, $V_{\mathrm{ID},t}=H(\mathrm{ID}, t)$ and $M=H(m)$. Let $\mathcal{M}$, $\mathcal{V}_{\mathrm{ID},t}$, $\mathcal{V}_{\mathrm{ID}}$ represent the sets as ever described. $\mathcal{B}$ searches L_{list} to test if the user ID's public key is the original one. If no, suppose a new public key is $(\mathrm{upk}_{\mathrm{ID}}^{(1)}, \mathrm{upk}_{\mathrm{ID}}^{(2)})$, and $\mathcal{B}$ generates the signature as follows. If both $S_1(\mathcal{V}_{\mathrm{ID},t})=0 \bmod k$ and $S_1(\mathcal{M})=0 \bmod k$ hold, $\mathcal{B}$ simply halts. Otherwise, $\mathcal{B}$ randomly picks integers $d_t, d_v, d_m \in_R Z_q^*$, and generates the signature in terms of the following two situations: if $S_1(\mathcal{V}_{\mathrm{ID},t})\neq 0 \bmod k$, $\mathcal{B}$ assigns

$$B=(\mathrm{upk}_{\mathrm{ID}}^{(2)})^{-S_2(\mathcal{V}_{\mathrm{ID},t})/S_1(\mathcal{V}_{\mathrm{ID},t})}f(\mathcal{V}_{\mathrm{ID}})^{d_v}f(\mathcal{V}_{\mathrm{ID},t})^{d_t}f(\mathcal{M})^{d_m},$$
$$D=g^{d_m},\ a_{\mathrm{ID}}=g^{d_v},\ c_{\mathrm{ID},t}=(\mathrm{upk}_{\mathrm{ID}}^{(2)})^{-1/S_1(\mathcal{V}_{\mathrm{ID},t})}g^{d_t};$$

otherwise, $\mathcal{B}$ assigns

$$B=(\mathrm{upk}_{\mathrm{ID}}^{(2)})^{-S_2(\mathcal{M})/S_1(\mathcal{M})}f(\mathcal{V}_{\mathrm{ID}})^{d_v}f(\mathcal{V}_{\mathrm{ID},t})^{d_t}f(\mathcal{M})^{d_m},$$
$$D=(\mathrm{upk}_{\mathrm{ID}}^{(2)})^{-1/S_1(\mathcal{M})}g^{d_m},\ a_{\mathrm{ID}}=g^{d_v},\ c_{\mathrm{ID},t}=g^{d_t}.$$

Finally, $\mathcal{B}$ sends $(t, (D, B, a_{\mathrm{ID}}, c_{\mathrm{ID},t}))$ to $\mathcal{A}_{\mathrm{I}}$. It is easy to see that this is identical to the actual construction for $\mathcal{A}_{\mathrm{I}}$.

If yes, $\mathcal{B}$ generates the signature as follows.

If $S_1(\mathcal{V}_{\mathrm{ID}})\neq 0 \bmod k$, $\mathcal{B}$ first performs the above simulation for RequestPartialKey queries, and then tests if a record (ID, $\mathrm{upk}_{\mathrm{ID}}$, $\mathrm{usk}_{\mathrm{ID}}$) exists in L_{list}. If no, $\mathcal{B}$ runs algorithm UserKeyGen to obtain $\mathrm{upk}_{\mathrm{ID}}/\mathrm{usk}_{\mathrm{ID}}$. It adds the record (ID, $\mathrm{upk}_{\mathrm{ID}}$, $\mathrm{usk}_{\mathrm{ID}}$) to L_{list}. Otherwise, it executes algorithm CL-Update to generate a signing key on (ID, t), and then runs algorithm CL-Sign to generate a signature for (ID, t).

If $S_1(\mathcal{V}_{\mathrm{ID}})=0 \bmod k$, $\mathcal{B}$ produces the signature as follows. If both $S_1(\mathcal{V}_{\mathrm{ID},t})=0 \bmod k$ and $S_1(\mathcal{M})=0 \bmod k$ hold, $\mathcal{B}$ simply halts. Otherwise, $\mathcal{B}$ picks $d_t, d_v, d_m \in_R Z_q^*$, obtains $\mathrm{usk}_{\mathrm{ID}}$ from L_{list} (if it does not exist in L_{list}, $\mathcal{B}$ first executes the algorithm UserKeyGen to generate it), sets $x=\mathrm{usk}_{\mathrm{ID}}$, and generates the signature in terms of the following two situations: if $S_1(\mathcal{V}_{\mathrm{ID},t})\neq 0 \bmod k$, $\mathcal{B}$ puts

$$B = g_1^{-xS_2(\mathcal{V}_{\mathrm{ID},t})/S_1(\mathcal{V}_{\mathrm{ID},t})} f(\mathcal{V}_{\mathrm{ID}})^{d_v} f(\mathcal{V}_{\mathrm{ID},t})^{d_t x} f(\mathcal{M})^{d_m},$$
$$D = g^{d_m},\ a_{\mathrm{ID}} = g^{d_v},\ c_{\mathrm{ID},t} = g_1^{-x/S_1(\mathcal{V}_{\mathrm{ID},t})} g^{d_t x};$$

otherwise, $\mathcal{B}$ puts

$$B = g_1^{-xS_2(\mathcal{M})/S_1(\mathcal{M})} f(\mathcal{V}_{\mathrm{ID}})^{d_v} f(\mathcal{V}_{\mathrm{ID},t})^{d_t} f(\mathcal{M})^{d_m x},$$
$$D = g_1^{-x/S_1(\mathcal{M})} g^{d_m x},\ a_{\mathrm{ID}} = g^{d_v},\ c_{\mathrm{ID},t} = g^{d_t}.$$

Finally, $\mathcal{B}$ returns $(t, (D, B, a_{\mathrm{ID}}, c_{\mathrm{ID},t}))$ to $\mathcal{A}_{\mathrm{I}}$. It is easy to verify the validity of this signature.

8. Forgery. If $\mathcal{B}$ finishes the simulation without halting, $\mathcal{A}_{\mathrm{I}}$ will produce a tuple $(\mathrm{ID}^*, m^*, s^*, \mathrm{upk}_{\mathrm{ID}*})$ with advantage at least ε, where $s^*=(t^*, (D^*, B^*, a_{\mathrm{ID}*}, c_{\mathrm{ID}*,t*}))$. $\mathcal{B}$ computes $V_{\mathrm{ID}*}=H(\mathrm{ID}^*)$, $V_{\mathrm{ID}*,t*}=H(\mathrm{ID}^*,t^*)$, $M^*=H(m^*)$.

Let

$$\mathcal{V}_{\mathrm{ID}*} = \{i : V_{\mathrm{ID}*}[i] = 1, i \in \{1,2,...,n\}\},$$
$$\mathcal{V}_{\mathrm{ID}*,t*} = \{i : V_{\mathrm{ID}*,t*}[i] = 1, i \in \{1,2,...,n\}\},$$
$$\mathcal{M}^* = \{i : M^*[i] = 1, i \in \{1,2,...,n\}\}.$$

If $S_1(\mathcal{V}_{\mathrm{ID}*}) = 0 \bmod q$ or $S_1(\mathcal{V}_{\mathrm{ID}*,t*}) = 0 \bmod q$ or $S_1(\mathcal{M}^*) = 0 \bmod q$, $\mathcal{B}$ halts. Otherwise, $\mathcal{B}$ computes

$$\frac{B^*}{D^{*S_2(\mathcal{M}^*)} a_{\mathrm{ID}*}^{S_2(\mathcal{V}_{\mathrm{ID}*})} c_{\mathrm{ID}*}^{S_2(\mathcal{V}_{\mathrm{ID}*,t*})}}$$
$$= g_2^{ax} (g_2^{S_1(\mathcal{V}_{\mathrm{ID}*})} g^{S_2(\mathcal{V}_{\mathrm{ID}*})})^{d_v} (g_2^{S_1(\mathcal{V}_{\mathrm{ID}*,t*})} g^{S_2(\mathcal{V}_{\mathrm{ID}*,t*})})^{d_t}$$
$$\cdot (g_2^{S_1(\mathcal{M}^*)} g^{S_2(\mathcal{M}^*)})^{d_m} g^{-S_2(\mathcal{M}^*)d_m} g^{-S_2(\mathcal{V}_{\mathrm{ID}*})d_v} g^{-S_2(\mathcal{V}_{\mathrm{ID}*,t*})d_t}$$
$$= g_2^{ax} = g^{abx}$$

and returns $(g^{abx}, \mathrm{upk}_{\mathrm{ID}*}^{(1)}) = (g^{abx}, g^x)$, which solves the NGBDH problem.

Eventually, the lower limit regarding the probability of $\mathcal{B}$ not halting will be deduced. To finish the simulation without halting, the following conditions need to be satisfied simultaneously:

(1) RequestPartialKey queries on an identity ID have $S_1(\mathcal{V}_{\mathrm{ID}}) \neq 0 \bmod k$.

(2) RequestSigningKey queries on (ID, t) will have either $S_1(\mathcal{V}_{\mathrm{ID}}) \neq 0 \bmod k$ or $S_1(\mathcal{V}_{\mathrm{ID},t}) \neq 0 \bmod k$.

(3) Sign queries (ID, t) will have either $S_1(\mathcal{V}_{\mathrm{ID}}) \neq 0 \bmod k$ or $S_1(\mathcal{V}_{\mathrm{ID},t}) \neq 0 \bmod k$ or $S_1(\mathcal{M}) \neq 0 \bmod k$, if $\mathcal{A}_{\mathrm{I}}$ does not substitute the user ID's public key. Otherwise, it requires $S_1(\mathcal{V}_{\mathrm{ID},t}) \neq 0 \bmod k$ or $S_1(\mathcal{M}) \neq 0 \bmod k$.

(4) $S_1(\mathcal{V}_{\mathrm{ID}*}) = 0 \bmod k$, $S_1(\mathcal{V}_{\mathrm{ID}*,t*}) = 0 \bmod k$, and $S_1(\mathcal{M}^*) = 0 \bmod k$.

Assuming $\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_{t_{\mathrm{I}}}$ are all the distinct $\mathcal{V}_{\mathrm{ID}}$ and $\mathcal{V}_{\mathrm{ID},t}$ arising in RequestPartialKey queries, RequestSigningKey queries, and Sign queries. Obviously, we have $t_{\mathrm{I}} \leq t_{\mathrm{sk}}+t_{\mathrm{pk}}+t_{\mathrm{s}}$. We consider the following cases:

A_i: $S_1(\mathcal{V}_i) = 0 \bmod k$, where $i=1, 2, \ldots, t_{\mathrm{I}}$,
A^*: $S_1(\mathcal{V}_{\mathrm{ID}*}) = 0 \bmod q$,
C'^*: $S_1(\mathcal{V}_{\mathrm{ID}*,t*}) = 0 \bmod q$,
C^*: $S_1(\mathcal{M}^*) = 0 \bmod q$.

The probability of $\mathcal{B}$ not halting is

$$\Pr[\text{not halt}] \geq \Pr[\wedge_{i=1}^{t_{\mathrm{I}}} A_i \wedge A^* \wedge C'^* \wedge C^*].$$

It is easy to see that the cases A^*, C'^* and C^* are independent.

Note that the conjecture $k(n+1)<q$ leads to the fact $S_1(\mathcal{V}_{\mathrm{ID}*}) = 0 \bmod q \Rightarrow S_1(\mathcal{V}_{\mathrm{ID}*}) = 0 \bmod k$. Also, it is easy to see that, if $S_1(\mathcal{V}_{\mathrm{ID}*}) = 0 \bmod k$, there exists a unique $v \in [0, n]$ such that $S_1(\mathcal{V}_{\mathrm{ID}*}) = 0 \bmod q$. Since v, x_0 and $\boldsymbol{X}$ are selected at random, we have

$$\Pr[A^*] = \Pr[S_1(\mathcal{V}_{\mathrm{ID}*}) = 0 \bmod q \wedge S_1(\mathcal{V}_{\mathrm{ID}*}) = 0 \bmod k]$$
$$= \Pr[S_1(\mathcal{V}_{\mathrm{ID}*}) = 0 \bmod k]$$
$$\cdot \Pr[S_1(\mathcal{V}_{\mathrm{ID}*}) = 0 \bmod q \mid S_1(\mathcal{V}_{\mathrm{ID}*}) = 0 \bmod k]$$
$$= 1/[k(n+1)].$$

Similarly, we can obtain

$$\Pr[C'^*] = \frac{1}{k(n+1)},\quad \Pr[C^*] = \frac{1}{k(n+1)}.$$

Therefore, we have $\Pr[A^* \wedge C'^* \wedge C^*] \geq \frac{1}{k^3(n+1)^3}$. Likewise, we can know that A_i and A^*, C'^*, C^* are independent for any i, hence

$$\Pr[\neg A_i \wedge A^* \wedge C'^* \wedge C^*] \geq 1/k,$$

and

$$\begin{aligned}
&\Pr[\text{not halt}] \\
&\geq \Pr[\wedge_{i=1}^{t_1} A_i \wedge A^* \wedge C'^* \wedge C^*] \\
&= \Pr[A^* \wedge C'^* \wedge C^*]\Pr[\wedge_{i=1}^{t_1} A_i \mid A^* \wedge C'^* \wedge C^*] \\
&\geq \frac{1}{k^3(n+1)^3}(1-\Pr[\vee_{i=1}^{t_1} \neg A_i \mid A^* \wedge C'^* \wedge C^*]) \\
&\geq \frac{1}{k^3(n+1)^3}\left(1-\sum_{i=1}^{t_1}\Pr[\neg A_i \mid A^* \wedge C'^* \wedge C^*]\right) \\
&\geq \frac{1}{k^3(n+1)^3}\left(1-\frac{t_1}{k}\right) \\
&= \frac{27}{256(t_{sk}+t_{pk}+t_s)^3(n+1)^3}.
\end{aligned}$$

If $\mathcal{B}$ proceeds, the simulation provided for $\mathcal{A}_I$ is perfect, and hence $\mathcal{A}_I$ will provide a valid signature with an advantage of at least ε. Thus, $\mathcal{B}$ can solve the NGBDH problem with an advantage

$$\varepsilon' \geq \frac{27\varepsilon}{256(t_{sk}+t_{pk}+t_s)^3(n+1)^3}.$$

Note that $\mathcal{B}$'s overhead is determined by multiplication and exponentiation computations executed in the RequestPublicKey, RequestSecretKey, RequestPartialKey, RequestSigningKey, and Sign queries. Since it takes $O(1)$ exponentiations in the RequestPartialKey, RequestSigningKey, and Sign queries and $O(n)$ multiplications in every phase, it is easy to obtain $t'=t+O[(t_{pk}+t_{sk}+t_s)n\rho+(t_{ps}+t_{pk}+t_{sk}+t_s)\tau]$.

Theorem 2 The proposed scheme is (t, ε)-key-insulated-secure unforgeable against type II forgers, supposing the (t', ε')-many-DH conjecture remains valid in group G_1, where

$$\varepsilon' \geq \frac{27\varepsilon}{256t_{ps}(t_{sk}+t_s)^3(n+1)^3},$$

$$t' = t + O[(t_{sk}+t_s)n\rho + (t_{ps}+t_s+t_{sk})\tau].$$

Here t_{ps}, t_{sk}, and t_s represent the numbers of RequestPublicKey and RequestSecretKey queries altogether, RequestSigningKey queries, and Sign queries, respectively; ρ and τ represent the cost of performing an exponentiation and a multiplication in G_1, respectively.

Proof Suppose $\mathcal{A}_{II}$ is a type II forger against the proposed scheme. We will show how to generate another t'-time $\mathcal{B}$ that can solve the many-DH problem with advantage at least ε'.

Suppose $\mathcal{B}$ is provided a many-DH instance (g^a, g^b, g^x, g^{ab}, g^{ax}, g^{bx}). To deal with this instance with the help of $\mathcal{A}_{II}$, $\mathcal{B}$ acts as a challenger and answers the queries for $\mathcal{A}_{II}$ as follows.

1. Setup. $\mathcal{B}$ generates the system parameters mpk for $\mathcal{A}_{II}$ as follows:

(1) Put $k=4(t_{sk}+t_s)/3$ and $k(n+1)<q$ and randomly select an integer $v\in[0, n]$. Without loss of generality, we here suppose $3|(t_{sk}+t_s)$, since otherwise more queries are performed to satisfy this condition.

(2) Pick two vectors $\boldsymbol{X}=(x_i)\in Z_k^{n+1}$ and $\boldsymbol{Y}=(y_i)\in Z_q^{n+1}$.

(3) Define $g_1=g^a$, $g_2=g^b$, $g_2^\alpha = g^{ab}$, $\text{upk}_{\text{ID}^*}^{(1)}=g^x$, $\text{upk}_{\text{ID}^*}^{(2)} = g^{ax}$, an n-element vector $\boldsymbol{V}=(v_i)\in G_1^{n+1}$ with $v_i = g_2^{x_i}g^{y_i}$ $(1\leq i\leq n)$ and $v_0 = g_2^{-kv+x_0}g^{y_0}$. For convenience, consider the following three functions:

$$\begin{aligned}
f(\mathcal{W}) &= g_2^{S_1(\mathcal{W})}g^{S_2(\mathcal{W})}, \\
S_2(\mathcal{W}) &= y_0 + \sum_{i\in\mathcal{W}} y_i, \\
S_1(\mathcal{W}) &= -kv + x_0 + \sum_{i\in\mathcal{W}} x_i,
\end{aligned}$$

where $\mathcal{W}\subseteq\{1,2,\ldots,n\}$. Note that we implicitly assign the master key $g_2^\alpha = g_2^a = g^{ab}$, and $f(\mathcal{W}) = v_0\prod_{i\in\mathcal{W}} v_i$.

(4) Finally, send all the system parameters mpk and the master key $g_2^\alpha = g_2^a = g^{ab}$ to $\mathcal{A}_{II}$.

$\mathcal{B}$ replies to forger $\mathcal{A}_{II}$'s queries in the attack phase of the game as follows:

2. RequestPublicKey queries. $\mathcal{B}$ has a list L_{list} of records (ID, upk_{ID}, usk_{ID}). $\mathcal{B}$ first inserts the record (ID^*, $\text{upk}_{\text{ID}*}$, $\perp$) into the list L_{list}. When $\mathcal{A}_{II}$ provides

an identity ID, $\mathcal{B}$ tests if a record (ID, upk_{ID}, usk_{ID}) exists in L_{list}. If no, $\mathcal{B}$ runs algorithm UserKeyGen to obtain $\text{upk}_{\text{ID}}/\text{usk}_{\text{ID}}$. It adds the record (ID, upk_{ID}, usk_{ID}) to L_{list} and sends upk_{ID} to $\mathcal{A}_{\text{II}}$.

3. RequestSecretKey queries. Suppose $\mathcal{A}_{\text{II}}$ provides an identity ID. If ID=ID*, $\mathcal{B}$ halts; otherwise, $\mathcal{B}$ tests if a record (ID, upk_{ID}, usk_{ID}) exists in L_{list}. If no, $\mathcal{B}$ runs algorithm UserKeyGen to obtain $\text{upk}_{\text{ID}}/\text{usk}_{\text{ID}}$. It adds the record (ID, upk_{ID}, usk_{ID}) to L_{list} and outputs usk_{ID} to $\mathcal{A}_{\text{II}}$.

4. RequestSigningKey queries. When $\mathcal{A}_{\text{II}}$ asks for a signing key query for (ID, t), $\mathcal{B}$ first computes $V_{\text{ID}}=H(\text{ID})$, $V_{\text{ID},t}=H(\text{ID}, t)$ and then tests if ID=ID*. If yes, it generates the signing key as follows. Assume $S_1(\mathcal{V}_{\text{ID},t})\neq 0 \bmod k$. Using the aforementioned argument, it infers that $S_1(\mathcal{V}_{\text{ID},t})\neq 0 \bmod q$ assuming $k(n+1)<q$. Then $\mathcal{B}$ picks $d_u, d_t \in_R Z_q^*$ and generates the signing key as follows:

$$a_{\text{ID}} = g^{d_u},\ c_{\text{ID},t} = (\text{upk}_{\text{ID}^*}^{(2)})^{-1/S_1(\mathcal{V}_{\text{ID},t})} g^{d_t},$$
$$b_{\text{ID},t} = (\text{upk}_{\text{ID}^*}^{(2)})^{-S_2(\mathcal{V}_{\text{ID},t})/S_1(\mathcal{V}_{\text{ID},t})} f(\mathcal{V}_{\text{ID}})^{d_u} f(\mathcal{V}_{\text{ID},t})^{d_t}.$$

Then $\mathcal{B}$ returns the signing key $x_{\text{ID},t}=(a_{\text{ID}}, b_{\text{ID},t}, c_{\text{ID},t})$ to $\mathcal{A}_{\text{II}}$. If $S_1(\mathcal{V}_{\text{ID},t})=0 \bmod k$, $\mathcal{B}$ simply halts.

If ID≠ID*, $\mathcal{B}$ generates the signing key as follows.

If $S_1(\mathcal{V}_{\text{ID}})\neq 0 \bmod k$, $\mathcal{B}$ first generates a partial key as in the RequestPartialKey queries of Theorem 1, obtains usk_{ID} from L_{list} (if it does not exist in L_{list}, $\mathcal{B}$ executes algorithm UserKeyGen and adds the record (ID, upk_{ID}, usk_{ID}) to L_{list}), and performs algorithm CL-Update to generate a signing key on ID and t.

If $S_1(\mathcal{V}_{\text{ID}})=0 \bmod k$ and $S_1(\mathcal{V}_{\text{ID},t})\neq 0 \bmod k$ (this infers that $S_1(\mathcal{V}_{\text{ID},t})\neq 0 \bmod q$ from the conjecture $k(n+1)<q$), $\mathcal{B}$ randomly picks $d_u, d_t \in_R Z_q^*$, gets usk_{ID} from L_{list} (if it does not exist in L_{list}, $\mathcal{B}$ executes algorithm UserKeyGen to generate it), sets $x=\text{usk}_{\text{ID}}$, and computes the signing key as

$$\begin{aligned} x_{\text{ID},t} &= (g^{d_u}, g_1^{-xS_2(\mathcal{V}_{\text{ID},t})/S_1(\mathcal{V}_{\text{ID},t})} f(\mathcal{V}_{\text{ID}})^{d_u} f(\mathcal{V}_{\text{ID},t})^{d_t x}, \\ &\quad g_1^{-x/S_1(\mathcal{V}_{\text{ID},t})} g^{d_t x}) \\ &= (g^{d_u}, g_2^{ax} (g_2^{S_1(\mathcal{V}_{\text{ID},t})} g^{S_2(\mathcal{V}_{\text{ID},t})})^{\tilde{d}_t} f(\mathcal{V}_{\text{ID}})^{d_u}, g^{\tilde{d}_t}) \\ &= (a_{\text{ID}}, b_{\text{ID},t}, c_{\text{ID},t}), \end{aligned}$$

where $\tilde{d}_t = d_t x - ax / S_1(\mathcal{V}_{\text{ID},t})$.

If $S_1(\mathcal{V}_{\text{ID},t})=0 \bmod k$, the simulator halts.

5. Sign queries. Assuming $\mathcal{A}_{\text{II}}$ provides (ID, t), $\mathcal{B}$ first computes $V_{\text{ID}}=H(\text{ID})$, $V_{\text{ID},t}=H(\text{ID}, t)$, and $M=H(m)$. Let $\mathcal{M}, \mathcal{V}_{\text{ID},t}, \mathcal{V}_{\text{ID}}$ denote the sets as ever described. $\mathcal{B}$ tests if ID=ID*. If yes, it generates the signature as follows. If both $S_1(\mathcal{V}_{\text{ID},t})=0 \bmod k$ and $S_1(\mathcal{M})=0 \bmod k$ hold, $\mathcal{B}$ simply terminates. Otherwise, $\mathcal{B}$ chooses $d_t, d_v, d_m \in_R Z_q^*$, and generates the signature in terms of the following two situations: if $S_1(\mathcal{V}_{\text{ID},t})\neq 0 \bmod k$, $\mathcal{B}$ assigns

$$B = (\text{upk}_{\text{ID}^*}^{(2)})^{-S_2(\mathcal{V}_{\text{ID},t})/S_1(\mathcal{V}_{\text{ID},t})} f(\mathcal{V}_{\text{ID}})^{d_v} f(\mathcal{V}_{\text{ID},t})^{d_t} f(\mathcal{M})^{d_m},$$
$$D = g^{d_m},\ a_{\text{ID}} = g^{d_v},\ c_{\text{ID},t} = (\text{upk}_{\text{ID}^*}^{(2)})^{-1/S_1(\mathcal{V}_{\text{ID},t})} g^{d_t};$$

otherwise, $\mathcal{B}$ assigns

$$B = (\text{upk}_{\text{ID}^*}^{(2)})^{-S_2(\mathcal{M})/S_1(\mathcal{M})} f(\mathcal{V}_{\text{ID}})^{d_v} f(\mathcal{V}_{\text{ID},t})^{d_t} f(\mathcal{M})^{d_m},$$
$$D = (\text{upk}_{\text{ID}^*}^{(2)})^{-1/S_1(\mathcal{M})} g^{d_m},\ a_{\text{ID}} = g^{d_v},\ c_{\text{ID},t} = g^{d_t}.$$

Finally, $\mathcal{B}$ returns $(t, (D, B, a_{\text{ID}}, c_{\text{ID},t}))$ to $\mathcal{A}_{\text{II}}$. It is easy to verify the validity of this signature.

If ID≠ID*, $\mathcal{B}$ generates the signature as follows.

If $S_1(\mathcal{V}_{\text{ID}})\neq 0 \bmod k$, $\mathcal{B}$ produces a partial key as in the RequestPartialKey queries of Theorem 1, obtains the secret key usk_{ID} from L_{list} (if it does not exist in L_{list}, $\mathcal{B}$ executes algorithm UserKeyGen and adds the record (ID, upk_{ID}, usk_{ID}) to L_{list}), performs algorithm CL-Update to generate a signing key, and then performs algorithm CL-Sign to generate a signature for t and ID.

If $S_1(\mathcal{V}_{\text{ID}})=0 \bmod k$, $\mathcal{B}$ generates the signature as follows. If both $S_1(\mathcal{V}_{\text{ID},t})=0 \bmod k$ and

$S_1(\mathcal{M})=0 \bmod k$ hold, $\mathcal{B}$ simply halts. Otherwise, $\mathcal{B}$ picks $d_t, d_v, d_m \in_R Z_q^*$, obtains usk_{ID} from L_{list} (if it does not exist in L_{list}, $\mathcal{B}$ first executes algorithm UserKeyGen to obtain it), and sets $x=\text{usk}_{\text{ID}}$. The signature is generated in terms of the following two situations: if $S_1(\mathcal{V}_{\text{ID},t})\neq 0 \bmod k$, $\mathcal{B}$ assigns

$$B=g_1^{-xS_2(\mathcal{V}_{\text{ID},t})/S_1(\mathcal{V}_{\text{ID},t})} f(\mathcal{V}_{\text{ID}})^{d_v} f(\mathcal{V}_{\text{ID},t})^{d_t x} f(\mathcal{M})^{d_m},$$
$$D=g^{d_m},\ a_{\text{ID}}=g^{d_v},\ c_{\text{ID},t}=g_1^{-x/S_1(\mathcal{V}_{\text{ID},t})} g^{d_t x}.$$

Otherwise, $\mathcal{B}$ assigns

$$B=g_1^{-xS_2(\mathcal{M})/S_1(\mathcal{M})} f(\mathcal{V}_{\text{ID}})^{d_v} f(\mathcal{V}_{\text{ID},t})^{d_t} f(\mathcal{M})^{d_m x},$$
$$D=g_1^{-x/S_1(\mathcal{M})} g^{d_m x},\ a_{\text{ID}}=g^{d_v},\ c_{\text{ID},t}=g^{d_t}.$$

Finally, $\mathcal{B}$ returns $(t, (D, B, a_{\text{ID}}, c_{\text{ID},t}))$ to $\mathcal{A}_{\text{II}}$. From $\mathcal{A}_{\text{II}}$'s viewpoint, the resulting signature is identical to the actual construction.

6. Forgery. If $\mathcal{B}$ proceeds during the simulation, $\mathcal{A}_{\text{II}}$ will, with an advantage of at least ε, provide a tuple $(m^*, \text{ID}^*, \text{upk}_{\text{ID}*}, s^*)$, where $s^*=(t^*, (D^*, B^*, a_{\text{ID}*}, c_{\text{ID}*,t*}))$. $\mathcal{B}$ computes $V_{\text{ID}*}=H(\text{ID}^*)$, $V_{\text{ID}*,t*}=H(\text{ID}^*, t^*)$, and $M^*=H(m^*)$.

Let

$$\mathcal{V}_{\text{ID}*}=\{i: V_{\text{ID}*}[i]=1, i\in\{1,2,\cdots,n\}\},$$
$$\mathcal{V}_{\text{ID}*,t*}=\{i: V_{\text{ID}*,t*}[i]=1, i\in\{1,2,\cdots,n\}\},$$
$$\mathcal{M}^*=\{i: M^*[i]=1, i\in\{1,2,\cdots,n\}\}.$$

If $S_1(\mathcal{V}_{\text{ID}*})=0 \bmod q$ or $S_1(\mathcal{V}_{\text{ID}*,t*})=0 \bmod q$ or $S_1(\mathcal{M}^*)=0 \bmod q$, $\mathcal{B}$ simply halts. Otherwise, $\mathcal{B}$ computes g^{abx} as follows:

$$\frac{B^*}{D^{*S_2(\mathcal{M}^*)} a_{\text{ID}*}^{S_2(\mathcal{V}_{\text{ID}*})} c_{\text{ID}*}^{S_2(\mathcal{V}_{\text{ID}*,t*})}}$$
$$=g_2^{ax}(g_2^{S_1(\mathcal{V}_{\text{ID}*})} g^{S_2(\mathcal{V}_{\text{ID}*})})^{d_v}(g_2^{S_1(\mathcal{V}_{\text{ID}*,t*})} g^{S_2(\mathcal{V}_{\text{ID}*,t*})})^{d_t}$$
$$\cdot(g_2^{S_1(\mathcal{M}^*)} g^{S_2(\mathcal{M}^*)})^{d_m} g^{-S_2(\mathcal{M}^*)d_m} g^{-S_2(\mathcal{V}_{\text{ID}*})d_v} g^{-S_2(\mathcal{V}_{\text{ID}*,t*})d_t}$$
$$=g_2^{ax}=g^{abx}.$$

The analysis for the probability and time complexity is analogous to that for Theorem 1, except for the elimination of the RequestPartialKey queries for the type II forgers.

CONCLUSION

In this paper, we proposed the first certificateless key-insulated signature scheme without random oracles. Our proposed scheme has two desirable properties. First, its security can be proved under the non-pairing-based generalized bilinear Diffie-Hellman (NGBDH) conjecture, without making use of the random oracle model. Second, it gets rid of the key escrow in identity-based key-insulated signatures.

References

Al-Riyami, S.S., Paterson, K.G., 2003. Certificateless public key cryptography. *LNCS*, **2894**:452-473. [doi:10.1007/b94617]

Bellare, M., Palacio, A., 2006. Protecting against key exposure: strongly key insulated encryption with optimal threshold. *LNCS*, **16**(6):379-396. [doi:10.1007/s00200-005-0183-y]

Canetti, R., Goldreich, O., Halevi, S., 2004. The random oracle methodology, revisited. *J. ACM*, **51**(4):557-594. [doi:10.1145/1008731.1008734]

Cheon, J.H., Hopper, N., Kim, Y., Osipkov, I., 2006. Timed-release and key-insulated public key encryption. *LNCS*, **4107**:191-205. [doi:10.1007/11889663]

Dodis, Y., Yung, M., 2002. Exposure-resilience for Free: The Hierarchical ID-based Encryption Case. Proc. IEEE Security in Storage Workshop, p.45-52.

Dodis, Y., Katz, J., Xu, S., Yung, M., 2002. Key-insulated public-key cryptosystems. *LNCS*, **2332**:65-82. [doi:10.1007/3-540-46035-7]

Dodis, Y., Katz, J., Xu, S., Yung, M., 2003. Strong key-insulated signature schemes. *LNCS*, **2567**:130-144. [doi:10.1007/3-540-36288-6]

González-Deleito, N., Markowitch, O., Dall'Olio, E., 2004. A new key-insulated signature scheme. *LNCS*, **3269**:465-479.

Gorantla, M., Gangishetti, R., Das, M., Saxena, A., 2005. An Effective Certificateless Signature Scheme Based on Bilinear Pairings. Proc. 3rd Int. Workshop on Security in Information Systems, p.31-39.

Hanaoka, G., Hanaoka, Y., Imai, H., 2006. Parallel key-insulated public key encryption. *LNCS*, **3958**:105-122. [doi:10.1007/b101042]

Hanaoka, Y., Hanaoka, G., Shikata, J., Imai, H., 2002. Unconditionally secure key insulated cryptosystems: models, bounds and constructions. *LNCS*, **2513**:85-96. [doi:10.1007/3-540-36159-6]

Hu, B., Wong, D., Zhang, Z., Deng, X., 2006. Key replacement attack against a generic construction of certificateless signature. *LNCS*, **4058**:235-246. [doi:10.1007/11780656]

Huang, X., Susilo, W., Mu, Y., Zhang, F., 2005. On the security of certificateless signature schemes from Asiacrypt 2003. *LNCS*, **3810**:13-25. [doi:10.1007/11599371]

Le, Z., Ouyang, Y., Ford, J., Makedon, F., 2004. A hierarchical key-insulated signature scheme in the CA trust model. *LNCS*, **3225**:280-291. [doi:10.1007/b100936]

Liu, J.K., Au, M.H., Susilo, W., 2007. Self-generated-certificate Public Key Cryptography and Certificateless Signature/Encryption Scheme in the Standard Model. ACM Symp. on Information, Computer and Communications Security, p.273-283.

Lysyanskaya, A., 2002. Unique signatures and verifiable random functions from the DH-DDH separation. *LNCS*, **2442**:597-612. [doi:10.1007/3-540-45708-9]

Paterson, K., Schuldt, J., 2006. Efficient identity-based signatures secure in the standard model. *LNCS*, **4058**:207-222. [doi:10.1007/11780656]

Shamir, A., 1984. Identity-based cryptosystems and signature schemes. *LNCS*, **196**:47-53. [doi:10.1007/3-540-39568-7]

Waters, B., 2005. Efficient identity-based encryption without random oracles. *LNCS*, **3494**:114-127. [doi:10.1007/b136415]

Weng, J., Chen, K.F., Liu, S.L., Li, X.X., 2006. Identity-based key-insulated signature with secure key-updates. *LNCS*, **4318**:13-26. [doi:10.1007/11937807]

Weng, J., Liu, S.L., Chen, K.F., Ma, C.S., 2007. Identity-based key-insulated signature without random oracles. *LNAI*, **4456**:470-480. [doi:10.1007/978-3-540-74377-4]

Yum, D.H., Lee, P.J., 2004. Generic construction of certificateless signature. *LNCS*, **3108**:200-211. [doi:10.1007/b98755]

Zhang, Z.F., Wong, D.S., Xu, J., Feng, D.G., 2006. Certificateless public-key signature: security model and efficient construction. *LNCS*, **3989**:293-308. [doi:10.1007/11767480]

Zhou, Y., Cao, Z., Chai, Z., 2006. Identity-based key insulated signature. *LNCS*, **3903**:226-234. [doi:10.1007/11689522]