Ann. Telecommun. (2015) 70:395–405
DOI 10.1007/s12243-015-0461-z

Strongly secure certificateless key-insulated signature secure in the standard model

Yanan Chen · Weixiang Xu · Hu Xiong

Received: 15 April 2014 / Accepted: 8 March 2015 / Published online: 1 April 2015

Abstract To protect signing rights against the compromise of secret key, the key-insulated signature (KIS) has attracted a lot of attention from the industry and academia. It would be interesting to investigate the notion of KIS in the certificateless public key cryptography (CL-PKC) environment to solve the problem of certificate management and key escrow simultaneously. To capture the seeming neglected attack mounted by the malicious key generation center (KGC), a stronger security model for the CL-PKC should be considered. In this paper, we first show that the only known CL-KIS scheme is vulnerable against malicious KGC attack, and then propose the first CL-KIS scheme secure against malicious KGC attack, with security proof in the standard model.

Keywords Certificateless cryptosystem · Key-insulated signature · Malicious-but-passive KGC attack · Standard model

Y. Chen (✉) · W. Xu (✉)
The MOE key Laboratory for Transportation Complex Systems Theory and Technology School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, People's Republic of China
e-mail: cynbjtu@gmail.com
e-mail: wxxu@bjtu.edu.cn

Y. Chen
School of Software, JiangXi University of Science and Technology, Nanchang 330013, People's Republic of China

H. Xiong
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
e-mail: xionghu.uestc@gmail.com

1 Introduction

The digital signature [9, 13, 23–25], as a counterpart to hand-written signature in the electronic world, has found wide applications in the fields such as electronic commerce, electronic government, and copyright protection by providing authentication, unforgeability, and non-repudiation. However, some issues should be addressed before the digital signature can be employed in practice. To protect the signing rights against the compromise of secret key, key evolving mechanisms including key-insulated [10, 11], intrusion-resilience [15, 31] and forward secure [8, 14] have been introduced independently. In a key-insulated signature (KIS) scheme [11], the whole lifetime of the signing key is divided into different time periods and this key is refreshed at each time period. On the other hand, the public key associated with this signing key remains unchanged during the whole lifetime. More specifically, the full signing key of the signer is updated by incorporating a temporary signing key held by the signer and a helper key from a physically secure device (which is usually named helper). In this case, the KIS scheme can achieve backward and forward security since the adversary who steals the signing key in present time period cannot break the signature scheme in the former or later time periods. To reduce the overhead caused by the certificate management, the key insulated mechanism has been investigated in identity-based public key cryptography (ID-PKC) [22] setting and several concrete identity-based key-insulated signature (ID-KIS) schemes [28, 29, 32] have already been proposed so far.

In ID-PKC, the public key of a user can be easily derived from its publicly known identity (e.g., email address or cell phone number), whereas the private key of this user is generated according to his/her identity by a fully-trusted private key generator (PKG). Unfortunately, the ID-PKC suffers

from the key escrow problem in the sense that PKG has any user's private key and can forge the signature or decrypt the ciphertext on behalf of any user without being detected. To solve the problems of certificate management in traditional public key cryptosystem (PKC) and key escrow in ID-PKC, respectively, a new paradigm called certificateless public key cryptography (CL-PKC) has been introduced, firstly by Al-Riyami and Paterson [1]. The basic idea of CL-PKC is to construct a full private key for a user by combining a partial private key generated by a semi-trusted key generation center (KGC) with a random secret value chosen by the user himself/herself. CL-PKC is not ID-based since the full public key of each user in the CL-PKC consists not only of the public identity, but also of the user public key calculated by the user himself/herself. Different from traditional PKC, the user public key in the CL-PKC does not need to be certified by any trusted third party (TTP) because the structure of CL-PKC guarantees the validity of the public key without a certificate issued by TTP. On the other hand, the inherent key escrow problem in ID-PKC has also been successfully solved in the CL-PKC environment since the KGC cannot access the full private key of the user. It would be interesting to investigate the notion of KIS in the CL-PKC environment. Recently, Wan et al. [26] present a formal definition of certificateless KIS (CL-KIS) by integrating the notions of CL-PKC and KIS altogether. Furthermore, a concrete CL-KIS scheme along with the formal security proof in the standard model [7] has also been given. In parallel to the work in [26], the certificate-based key insulated signature scheme [12, 16] has also been proposed to enjoy the merits of ID-PKC without suffering from the key escrow problem. Unfortunately, we observe that the seeming neglected attack mounted by the malicious-but-passive KGC has not been captured by Wan et al.'s security model. According to [3], the KGC can forge the signature on behalf of every signer by means of generating the system parameters maliciously.

In this paper, we demonstrate that Wan et al.'s CL-KIS scheme is subjected to the malicious-but-passive KGC attack. It is fair to say devising a CL-KIS scheme secure in the standard model remains an open question. We attempt to close this open issue by devising a provably secure CL-KIS scheme in the standard model. It is proven that our CL-KIS scheme satisfies unforgeability against outside adversary and malicious-but-passive KGC assuming the hardness of computational Diffie-Hellman (CDH) problem. The proofs do not rely on random oracles.

The rest of this paper is organized as follows. In Section 2, we describe the formal model of CL-KIS scheme and the mathematical backgrounds. In Section 3, we review and analyze Wan et al.'s CL-KIS scheme. After that, the improved scheme along with the security analysis in the standard model has been given in Sections 4 and 5, respectively. Finally, the conclusions are given in Section 7.

2 Preliminaries

In this section, we will review the mathematical notions and formal models for CL-KIS scheme.

2.1 Notations

The notations used throughout this paper are listed in Table 1.

2.2 Definitions of CL-KIS Schemes

In order to solve the key escrow problem in the ID-PKC, the signing key of the user in the CL-PKC is split into the user private key calculated by the user himself/herself and the partial private key issued by the KGC. In this way, the key escrow problem is avoided due to the fact that the signing key of the user (especially the user secret key) cannot be accessed by the KGC. In the traditional certificateless signature (CLS) scheme, both of the partial private key and the user private key are kept by the user himself/herself. However, to reduce the damage caused by the key leakage, in the certificateless key insulated signature (CL-KIS) scheme, the combination of the partial private key and the user private key consists of two independent parts, i.e., the helper key stored in the helper and temporary signing key kept by the user. Only the signing key in present time period is compromised, the signing rights are not affected in the former or later time periods. The illustration of the helper key and the temporary signing key is shown in Fig. 1.

According to [26], a CL-KIS scheme consists of a tuple (*Setup, UserKeyGen, ExtractPartialKey, User-key-generation, Gen*, *CL-Update* *, *CL-Update, CL-Sign, and CL-Ver*) described as follows.

1. *Setup.* Given a security parameter $k \in \mathbb{N}$ as input, this algorithm is executed by KGC to generate the public parameters mpk, a master secret key msk, and the total number of time periods N.
2. *UserKeyGen.* Given the public parameters mpk and user identity ID, this algorithm is executed by the user himself/herself to generate a user public/secret key pair (upk_{ID}, usk_{ID}).
3. *ExtractPartialKey.* Given the master secret key msk along with the user identity ID, this algorithm is executed by KGC to generate a user's partial private key psk, which will be sent to this user securely.
4. *Gen.* By integrating the user secret key and partial secret key, the helper key hk_{ID} and the initial signing key $x_{ID,0}$ are generated by the user himself/herself. We stress that the initial secret key which will be used in the *CL-Sign* algorithm cannot be accessed by the KGC, and thus the key escrow problem in ID-PKC can be avoided.

Table 1 Notations

Notations	Descriptions
CL-PKC:	certificateless public key cryptography
ID-PKC:	identity-based public key cryptography
KGC	Key Generation Center
PKG	Private Key Generator
Helper:	An absolutely secure but computationally limited device
N:	The total number of time periods.
t:	Time period such that $t \leq N$.
$\mathbb{G}_1, \mathbb{G}_2$:	Two cyclic groups of same order p
$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$:	A bilinear map
g:	The generator of $\mathbb{G}_1$
(upk_{ID}, usk_{ID}):	User public/secret key pair chosen by the user himself/herself
psk_{ID}:	Partial private key issued by the KGC
hk_{ID}:	The helper key kept by the helper
$x_{ID,0}$:	Initial signing key for the identity ID
uk_{ID,t_1}:	Update key for the identity ID in the time period t_1
x_{ID,t_1}:	Signing key for the identity ID in the time period t_1
$H(\cdot)$:	A hash function such as $H : \{0,1\}^* \rightarrow \{0,1\}^n$

5. *CL-Update**. Given the public parameters mpk, a time period t, an identity ID and the helper key hk_{ID}, the helper outputs the update key $uk_{ID,t}$.
6. *CL-Update.* Given a time period t_1, a update key uk_{ID,t_1}, and a signing key x_{ID,t_2}, user associated with identity ID generates the signing key x_{ID,t_1} for the time period t_1.
7. *CL-Sign.* Given a message $m \in \{0,1\}^*$, time period t, an identity ID, user public key upk_{ID}, and signing key $x_{ID,t}$, this algorithm is executed by the user himself/herself to generate a signature (s, t).
8. *CL-Ver.* Given the public parameters mpk, user identity ID, user public key upk_{ID}, message m, and signature (s, t), this algorithm is executed by the verifier to return 1 for accept or 0 for reject.

2.3 Security models

Taking into account of the malicious-but-passive KGC, the security model defined in [26] is reconsidered as follows. Firstly, the malicious outsider who can compromise the user private key or replace the user public key is defined as type I adversary $\mathcal{A}_1$. Secondly, the malicious-but-passive KGC who is responsible for the generation of the public system parameter and master public/secret key pair is specified as adversary $\mathcal{A}_2$. The restrictions regarding to these security models include that $\mathcal{A}_1$ cannot compromise the master secret key nor get access to the user partial key and $\mathcal{A}_2$ cannot mount the key replacement attack. The oracles which can be accessed by the adversaries are described as follows.

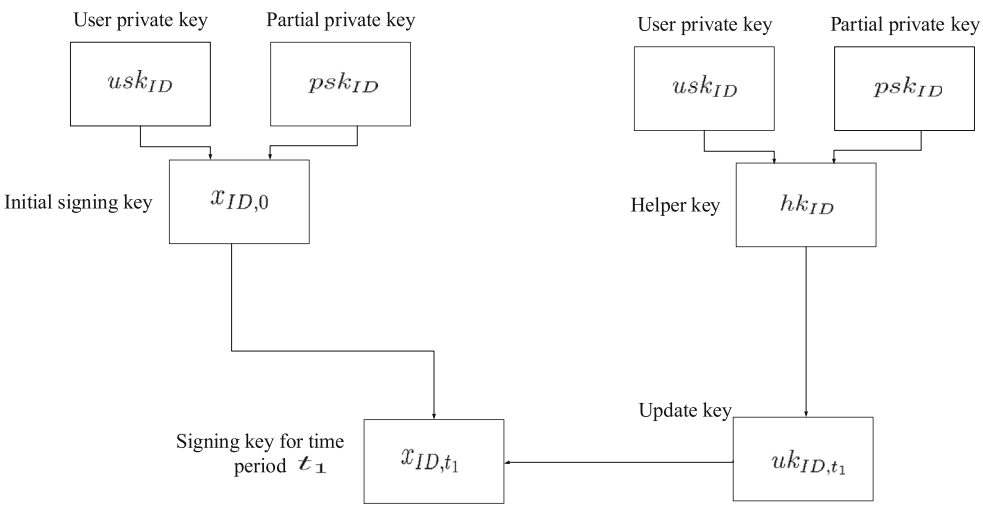

Fig. 1 The illustration of the helper key and the temporary signing key

1. Request-Public-Key Oracle: Given a query on identity ID, this oracle returns the matching user public key upk_{ID}.
2. Reveal-Partial-Private-Key Oracle: Given a query on identity ID, this oracle outputs the partial secret key psk_{ID} associated with this identity.
3. Reveal-Secret-Key Oracle: Given a query on identity ID, this oracle outputs a user secret key usk_{ID} associated with this identity.
4. Replace-Public-Key Oracle: Given a identity ID and a new user public key upk_{ID}, this oracle replaces the associated user's public key with the new public key upk'_{ID}.
5. Reveal-Signing-Key Oracle: Given a query on identity ID and time period t, this oracle outputs a signing key $x_{ID,t}$.
6. Sign Oracle: Upon receiving an identity ID, the corresponding user public key upk_{ID}, a time period t, and a message m, challenger $\mathcal{C}$ returns the resulting signature to the adversary.

In order to capture the attacks launched by $\mathcal{A}_1$ and $\mathcal{A}_2$, two games (*Game I* and *Game II*) are defined respectively.

Game I: Let $\mathcal{C}$ be the game simulator/challenger with the input of security parameter $k \in \mathbb{N}$.

1. *Initial.* $\mathcal{C}$ first executes *Setup* to generate the master public/secret key pair and public parameters, and then publishes the public parameters mpk and keeps the master secret key secret.
2. *Attack.* In this phase, $\mathcal{A}_1$ adaptively issues a polynomial bounded number of Request-Public-Key, Reveal-Partial-Private-Key, Reveal-Secret-Key, Replace-Public-Key, Reveal-Signing-Key, and Sign queries.
3. *Forgery.* $\mathcal{A}_1$ is to output $(ID^*, upk_{ID^*}, m^*, (s^*, t^*))$. $\mathcal{A}_1$ wins if $CL\text{-}Ver(mpk, (ID^*, upk_{ID^*}, m^*, (s^*, t^*))) = 1$ for some created ID^*, and the Sign has never been queried with $(ID^*, upk_{ID^*}, m^*, t^*)$. One additional restriction is that $\mathcal{A}_1$ has never queried Reveal-Partial-Private-Key oracle to get the partial private key of the target user.

Definition 1 A CL-KIS scheme is said to be Type-I secure if there is no probabilistic polynomial-time adversary $\mathcal{A}_1$ who wins *Game I* with non-negligible advantage.

Game II: Let $\mathcal{C}$ be the game challenger with the input of security parameter $k \in \mathbb{N}$.

1. *Initial.* $\mathcal{A}_2$ executes *Setup* to generate the master public/secret key pair and public system parameters, and sends the public system parameters $params$ and the master public/secret key pair to challenger $\mathcal{C}$. We should keep in mind that $\mathcal{A}_2$ generates $params$ and msk by itself.
2. *Attack.* In this phase, $\mathcal{A}_2$ adaptively issues a polynomial bounded number of Request-Public-Key, Reveal-Secret-Key, Reveal-Signing-Key, and Sign queries.
3. *Forgery.* Finally, $\mathcal{A}_2$ is to output $(ID^*, upk_{ID^*}, m^*, (s^*, t^*))$. $\mathcal{A}_2$ wins if $CL\text{-}Ver(mpk, (ID^*, upk_{ID^*}, m^*, (s^*, t^*))) = 1$ for some created ID^* and the Sign has never been queried with $(ID^*, upk_{ID^*}, m^*, t^*)$. One additional restriction is that $\mathcal{A}_2$ has never queried Replace-Public-Key to replace the public key of the target user.

Definition 2 A CL-KIS scheme is said to be Type-II secure if there is no probabilistic polynomial-time adversary $\mathcal{A}_2$ who wins *Game II* with non-negligible advantage.

A CL-KIS scheme is said to be existentially unforgeable under adaptive chosen message attacks, if there exists neither polynomial time Type I adversary nor polynomial time Type II adversary who has a non-negligible success probability in *Game I* and *Game II*, respectively.

2.4 Bilinear pairing

Let $\mathbb{G}_1, \mathbb{G}_2$ denote two multiplicative cyclic groups of prime order p. Let $\hat{e}$ be a bilinear map such that $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ with the following properties:

1. Bilinearity: For all $g_1, g_2 \in \mathbb{G}_1$, and $a, b \in \mathbb{Z}_p$, $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$.
2. Non-degeneracy: $\hat{e}(g_1, g_2) \neq 1_{\mathbb{G}_2}$.
3. Computability: It is efficient to compute $\hat{e}(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}_1$.

The modified Weil pairing and the Tate pairing are admissible maps of this kind. We refer for more details to [6].

Definition 3 Given the elements g, g^a, and g^b, for some random values $a, b \in \mathbb{Z}_p$ the Computational Diffie-Hellman (CDH) problem consists of computing the element g^{ab}. More details of CDH problem can be found in [5, 9].

3 Analysis of Wan et al.'s scheme

3.1 Overview of Wan et al.'s scheme

Now, we review Wan et al.'s [26] CL-KIS scheme which incorporates the idea of Waters's signature scheme

[27], Paterson and Schuldt [21]'s identity-based signature scheme, and Liu et al.'s certificateless signature scheme [17].

Let $H : \{0,1\}^* \rightarrow \{0,1\}^n$ be a collision-resistant cryptographic hash function. Select a pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ where $\mathbb{G}_1, \mathbb{G}_2$ denote two multiplicative cyclic groups of prime order p. Assume N be the total number of time periods.

1. *Setup.*
   (a) Randomly select $\alpha \leftarrow_R \mathbb{Z}_p$ and $g_2 \leftarrow_R \mathbb{G}_1$. Compute $g_1 = g^{\alpha}$, where g is a generator of $\mathbb{G}_1$. Furthermore, choose a vector $\mathbf{V} = (v_i) \leftarrow_R \mathbb{G}_1^{n+1}$.
   (b) Define a function f by $f(\mathcal{W}) = v_0 \prod_{i \in \mathcal{W}} v_i$, where $\mathcal{W} \subset \{1, \ldots, n\}$.
   (c) The public parameters are $mpk = \{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, g, g_1, g_2, f, \mathbf{V}, H, N\}$ and master secret is $msk = g_2^{\alpha}$.
2. *UserKeyGen.* User selects a secret value $x \leftarrow_R \mathbb{Z}_p$ as his secret key usk_{ID} and computes his public key as $upk_{ID} = (upk_{ID}^{(1)}, upk_{ID}^{(2)}) = (g^x, g_1^x)$.
3. *ExtractPartialKey.* Compute $V_{ID} = H(ID)$ and define $\mathcal{V}_{ID} \subset \{1, \ldots, n\}$ to be the set of indices i such that $V_{ID}[i] = 1$. To construct the partial secret key of identity ID, the KGC randomly picks $d_u \leftarrow_R \mathbb{Z}_p$ and computes:

   $$psk_{ID} = (psk_{ID}^{(1)}, psk_{ID}^{(2)}) = (g_2^{\alpha}(f(\mathcal{V}_{ID}))^{d_u}, g^{d_u}).$$

4. *Gen.* The user performs the following steps:
   (a) Pick $d_v \leftarrow_R \mathbb{Z}_p$ and compute $a_{ID} = (psk_{ID}^{(2)})^x g^{d_v}$ and $hk_{ID} = (psk_{ID}^{(1)})^x f(\mathcal{V}_{ID})^{d_v}$.
   (b) Pick $b_{ID,0}, c_{ID,0} \leftarrow_R \mathbb{G}_1$ and define the initial signing key as $x_{ID,0} = (a_{ID}, b_{ID,0}, c_{ID,0})$.
   (c) Output the helper key hk_{ID} and the initial signing key $x_{ID,0}$.
5. *CL-Update*.* Given a time period t and an identity ID, the helper performs the following steps:
   (a) Pick $d_t \leftarrow_R \mathbb{Z}_p$ and assign $\hat{c}_{ID,t} = g^{d_t}$.
   (b) Compute $V_{ID,t} = H(ID, t)$.
   (c) Define $\mathcal{V}_{ID,t} \subset \{1, \ldots, n\}$ to be the set of indices i such that $V_{ID,t}[i] = 1$.
   (d) Compute $\hat{b}_{ID,t} = hk_{ID} \cdot f(\mathcal{V}_{ID,t})^{d_t}$.
   (e) Output the update key $uk_{ID,t} = (\hat{b}_{ID,t}, \hat{c}_{ID,t})$.
6. *CL-Update.* Given a time period t_1, a update key uk_{ID,t_1}, and a signing key x_{ID,t_2}, user associated with identity ID generates the signing key in the time period t_1 as follows:
   (a) Parse $x_{ID,t_2} = (a_{ID}, b_{ID,t_2}, c_{ID,t_2})$ and $uk_{ID,t_1} = (\hat{b}_{ID,t_1}, \hat{c}_{ID,t_1})$.
   (b) Set $b_{ID,t_1}, = \hat{b}_{ID,t_1}, c_{ID,t_1} = \hat{c}_{ID,t_1}$.
   (c) Output the signing key $x_{ID,t_1} = (a_{ID}, b_{ID,t_1}, c_{ID,t_1})$
7. *CL-Sign.* To sign a message $m \in \{0,1\}^*$ in time period t, the signer with identity ID and signing key $x_{ID,t}$ generates the signature as follows:
   (a) Parse $x_{ID,t} = (a_{ID}, b_{ID,t}, c_{ID,t})$.
   (b) Compute $M = H(m)$. Let $M[i]$ be the i-th bit of M and let $\mathcal{M} \subset \{1, \ldots, n\}$ be the set of indices i such that $M[i] = 1$.
   (c) Pick $d_m \leftarrow_R \mathbb{Z}_p$ and compute $D = g^{d_m}$ and $B = b_{ID,t} \cdot f(\mathcal{M})^{d_m}$.
   (d) Output the signature $(s,t) = ((D, B, a_{ID}, c_{ID,t}), t)$ on the message m in time period t.
8. *CL-Ver.* Given a signature (s,t) for an identity ID and public key $(upk_{ID}^{(1)}, upk_{ID}^{(2)})$ on a message m, a verifier executes the following steps to check the validity of the signature.
   (a) Parse $(D, B, a_{ID}, c_{ID,t})$.
   (b) Compute $M = H(m)$. Let $M[i]$ be the i-th bit of M and let $\mathcal{M} \subset \{1, \ldots, n\}$ be the set of indices i such that $M[i] = 1$.
   (c) Let $\mathcal{V}_{ID}, \mathcal{V}_{ID,t}$ be the sets described in the *ExtractPartialKey* and *CL-Update* algorithms respectively.
   (d) Check whether $\hat{e}(upk_{ID}^{(1)}, g_1) = \hat{e}(upk_{ID}^{(2)}, g)$ and

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

   Output valid if both equalities hold. Otherwise output invalid. The signature is valid if all the steps pass. Otherwise it is invalid.

3.2 Analysis

According to [26], their scheme is semantically secure against Type I and Type II adversary in the standard model. However, the attack mounted by the malicious KGC has been neglected in [26]. In fact, we demonstrate that their scheme is not unforgeable against the malicious-but-passive KGC (Type II adversary $\mathcal{A}_2$) attack in this subsection. Intuitively, the insecurity of Wan et al.'s scheme lies in the fact that, given a signature, a malicious-but-passive KGC (Type II adversary $\mathcal{A}_2$) can derive the user's signing key, and hence can certainly forge signature on behalf of this user. The attack is described in detail as follows.

1. In the initial phase, adversary $\mathcal{A}_2$ generates the master public key mpk and master secret key for challenger $\mathcal{C}$.

In particular, adversary $\mathcal{A}_2$ computes $v_0 \leftarrow_R \mathbb{G}_1$ and $v_i \leftarrow_R \mathbb{G}_1$ for $i = 1, \ldots, n$ as follows:

- Choose random values $\beta_0, \beta_1, \cdots, \beta_n$ in $\mathbb{Z}_p$.
- Compute $v_0 = g^{\beta_0}$ and $v_i = g^{\beta_i}$ for $i = 1, \ldots, n$.

2. In the attack phase, $\mathcal{A}_2$ issues a signature query by submitting an identity ID^*, a time period t^*, public key $(upk^{(1)}_{ID^*}, upk^{(2)}_{ID^*})$, and a message m^*. Then adversary $\mathcal{A}_2$ is given a signature $(s^*, t^*) = ((D^*, B^*, a_{ID^*}, c_{ID^*,t^*}), t^*)$ under the identity ID^* and public key $(upk^{(1)}_{ID^*}, upk^{(2)}_{ID^*})$ on the message m^* in time period t^* such that

$$\hat{e}\left(upk^{(1)}_{ID^*}, g_1\right) = \hat{e}\left(upk^{(2)}_{ID^*}, g\right)$$
$$\hat{e}(B^*, g) = \hat{e}\left(g_2, upk^{(2)}_{ID^*}\right)\hat{e}(f(\mathcal{V}_{ID^*}), a_{ID^*}) \times \hat{e}(f(\mathcal{V}_{ID^*,t^*}), c_{ID^*,t^*})\hat{e}\left(f(n*), D^*\right)$$

where $V_{ID^*} = H(ID^*)$ and $\mathcal{V}_{ID^*} \subset \{1, \ldots, n\}$ denote the set of indices i such that $V_{ID^*}[i] = 1$, $V_{ID^*,t^*} = H(ID^*, t^*)$ and $\mathcal{V}_{ID^*,t^*} \subset \{1, \ldots, n\}$ denote the set of indices i such that $V_{ID^*,t^*}[i] = 1$, $M^* = H(m^*)$ and $\mathcal{M}^* \subset \{1, \ldots, n\}$ denote the set of indices i such that $M^*[i] = 1$.

From $D^* = g^{d^*_m}$ and $B^* = b_{ID^*,t^*} \cdot f(\mathcal{M}^*))^{d^*_m}$, adversary $\mathcal{A}_2$ can derive the user's signing key b_{ID^*,t^*} by computing $\dfrac{B^*}{(D^*)^{\beta' + \sum_{j \in \mathcal{M}} \beta_j}}$, since

$$\begin{aligned}
\frac{B^*}{(D^*)^{\beta_0 + \sum_{j \in \mathcal{M}^*} \beta_j}} &= \frac{b_{ID^*,t^*} \cdot f(\mathcal{M}^*)^{d^*_m}}{(g^{d^*_m})^{\beta' + \sum_{j \in \mathcal{M}^*} \beta_j}} \\
&= \frac{b_{ID^*,t^*} \cdot f(\mathcal{M}^*)^{d^*_m}}{(g^{\beta' + \sum_{j \in \mathcal{M}^*} \beta_j})^{d^*_m}} \\
&= \frac{b_{ID^*,t^*} \cdot f(\mathcal{M}^*)^{d^*_m}}{(g^{\beta'} \prod_{j \in \mathcal{M}^*} g^{\beta_j})^{d^*_m}} \\
&= \frac{b_{ID^*,t^*} \cdot f(\mathcal{M}^*)^{d^*_m}}{f(\mathcal{M}^*)^{d^*_m}} \\
&= b_{ID^*,t^*}
\end{aligned}$$

Recall that (a_{ID^*}, c_{ID^*,t^*}) can be extracted from the signature (s^*, t^*) directly. Thus, adversary $\mathcal{A}_2$ can obtain the user's signing key $x_{ID^*,t^*} = (a_{ID^*}, b_{ID^*,t^*}, c_{ID^*,t^*})$. Equipped with this signing key, $\mathcal{A}_2$ can definitely forge valid signature on behalf of this user within the time period t^*.

Our result shows that Wan et al.'s scheme cannot provide existential unforgeability against the malicious-but-passive KGC.

4 An improved scheme

In this section, we provide an improvement of the CL-KIS scheme proposed in [26] which can resist the attacks mounted by the outsider adversary (type I adversary) and malicious-but-passive KGC (type II adversaries) simultaneously. In fact, the user public key has not been incorporated in the part of signature associated with the message to be signed. In this section, we show how to fix this problem.

Without losing of generality, we only describe the *CL-Sign* and *CL-Ver* algorithms since the other algorithms are identical to the one defined in [26].

1. *CL-Sign.* To sign a message $m \in \{0, 1\}^*$ in time period t, the signer with identity ID and signing key $x_{ID,t}$ generates the signature as follows:
   (a) Parse $x_{ID,t} = (a_{ID}, b_{ID,t}, c_{ID,t})$.
   (b) Compute $M = H(m)$. Let $M[i]$ be the i-th bit of M and let $\mathcal{M} \subset \{1, \ldots, n\}$ be the set of indices i such that $M[i] = 1$.
   (c) Pick $d_m \leftarrow_R \mathbb{Z}_p$ and compute $D = g^{d_m}$ and $B = b_{ID,t} \cdot f(\mathcal{M})^{d_m} \cdot (upk^{(1)}_{ID})^{H_1(m)d_m}$, where $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_p$ is a collision-resistant cryptographic hash function defined in the public parameters mpk.
   (d) Output the signature $(s, t) = ((D, B, a_{ID}, c_{ID,t}), t)$ on the message m in time period t.

2. *CL-Ver.* Given a signature (s, t) for an identity ID and public key $(upk^{(1)}_{ID}, upk^{(2)}_{ID})$ on a message m, a verifier executes the following steps to check the validity of the signature.
   (a) Parse $(D, B, a_{ID}, c_{ID,t})$.
   (b) Compute $M = H(m)$. Let $M[i]$ be the i-th bit of M and let $\mathcal{M} \subset \{1, \ldots, n\}$ be the set of indices i such that $M[i] = 1$.
   (c) Let $\mathcal{V}_{ID}, \mathcal{V}_{ID,t}$ be the sets described in the *ExtractPartialKey* and *CL-Update* algorithms respectively.
   (d) Check whether $\hat{e}(upk^{(1)}_{ID}, g_1) = \hat{e}(upk^{(2)}_{ID}, g)$ and

$$\hat{e}(B, g) = \hat{e}\left(g_2, upk^{(2)}_{ID}\right)\hat{e}(f(\mathcal{V}_{ID}), a_{ID}) \times \hat{e}(f(\mathcal{V}_{ID,t}), c_{ID,t})\hat{e}\left(f(\mathcal{M})(upk^{(1)}_{ID}\right)^{H_1(m)}, D)$$

   Output valid if both equalities hold. Otherwise output invalid. The signature is valid if all the steps pass. Otherwise it is invalid.

Remark 1 To resist the above malicious-but-passive KGC attack, the signature part $B = b_{ID,t} \cdot f(\mathcal{M})^{d_m}$ in the original Wan et al.'s scheme has been replaced with $B = b_{ID,t} \cdot$

$f(\mathcal{M})^{d_m} \cdot (upk_{ID}^{(1)})^{H_1(m)d_m}$. In this way, the user public key $upk_{ID}^{(1)}$ has been embedded into the signature such that the malicious Type-II adversary $\mathcal{A}_2$ cannot extract $b_{ID,t}$ from B and D without the knowledge of the corresponding user secret key usk_{ID}.

5 Analysis of the improved scheme

We prove that our CL-KIS scheme is existentially unforgeable against Type I and II adversary, in the standard model, given that the CDH problem is hard.

Theorem 1 *(Type I Existential Unforgeability). Our CL-KIS scheme is ϵ-existential unforgeable against Type I adversary with advantage at most ϵ, assuming that the ϵ'-CDH assumption holds in $\mathbb{G}_1$, where $\epsilon' \geq \frac{\epsilon}{64(q_e+q_{se}+q_s)(q_{se}+q_s)q_s(n+1)^3}$ where q_e is the number of queries made to the oracle* Reveal-Partial-Private-Key*, q_{se} is the number of queries made to the* Reveal-Signing-Key *oracle, q_s is the number of queries made to the* Sign *oracle, q_k is the number of queries made to the oracles* Reveal-Secret-Key *and* Request-Public-Key *altogether.*

Proof We assume that an ϵ-Type I adversary $\mathcal{A}_1$ for our scheme exists. Resorting to this forger, an algorithm $\mathcal{C}$ can be constructed to solve CDH problem with probability at least ϵ'.

The aim of algorithm $\mathcal{C}$ is to output g^{ab} by given a group $\mathbb{G}_1$ of prime order p with generator g and elements $g^a, g^b \in \mathbb{G}_1$ where a, b are selected uniformly at random from $\mathbb{Z}_p^*$. $\mathcal{C}$ makes use of $\mathcal{A}_1$ by simulating a challenger for $\mathcal{A}_1$. Such a simulation can be depicted as follows:

Initial. $\mathcal{C}$ sets $l = 2(q_e + q_s + q_{se})$ and randomly chooses a integer k with $0 \leq k \leq n$. We assume that $l(n+1) < p$ for the given values of q_e, q_s, q_k, and n. The simulator then chooses an integer $x_0 \leftarrow_R \mathbb{Z}_l$ and a vector (x_i) of length n, with $x_i \leftarrow_R \mathbb{Z}_l$ for all i. Lastly, $\mathcal{C}$ chooses a integer $y_0 \leftarrow_R \mathbb{Z}_p$ and a vector (y_i) of length n with $y_i \leftarrow_R \mathbb{Z}_p$ for all i. A pair of functions are defined as follows:

$$F(\mathcal{W}) = x_0 + \sum_{i \in \mathcal{W}} x_i - l \cdot k \quad J(\mathcal{W}) = y_0 + \sum_{i \in \mathcal{W}} y_i$$

$\mathcal{C}$ assigns $g_1 = g^a$, $g_2 = g^b$, $u_0 = g_2^{-l \cdot k + x_0} g^{y_0}$, $u_i = g_2^{x_i} g^{y_i}$ $(1 \leq i \leq n)$, and the public parameters $mpk = (g_1, g_2, u_0, (u_i))$ are sent to $\mathcal{A}_1$. Moreover, this assignment of parameter means that the master secret is $g_2^{\alpha} = g_2^a = g^{ab}$ and we have the following equations:

$$f(\mathcal{W}) = u_0 \prod_{i \in \mathcal{W}} u_i = g_2^{F(\mathcal{W})} g^{J(\mathcal{W})}$$

Attack. $\mathcal{C}$ maintains a list $\mathcal{L}$ which is initially empty and consists of entry $(ID, psk_{ID}, upk_{ID}, usk_{ID})$. $\mathcal{C}$ simulates all oracles as follows:

Request-Public-Key Oracle: Given an identity ID, $\mathcal{C}$ looks up its list $\mathcal{L}$ to find out the corresponding entry. If it does not exist, $\mathcal{C}$ runs *UserKeyGen* and *Extract-PartialKey* algorithms to generate the secret/public key pair and partial private key respectively. It then stores $(ID, psk_{ID}, upk_{ID}, usk_{ID})$ into the list $\mathcal{L}$. In both cases, upk_{ID} is returned.

Reveal-Partial-Private-Key Oracle: Consider a query for the partial private key of an identity ID ($\mathcal{C}$ computes $V_{ID} = H(ID)$ and sets $\mathcal{V}_{ID} \subset \{1, \ldots, n\}$ to be the set of indices i such that $V_{ID}[i] = 1$). $\mathcal{C}$ does not know the master secret, but assuming $F(\mathcal{V}_{ID}) \neq 0 \bmod p$, it can construct a partial private key by choosing $r_u \leftarrow_R \mathbb{Z}_p$ and computing $\left(psk_{ID}^{(1)}, psk_{ID}^{(2)}\right) = \left(g_1^{-\frac{J(\mathcal{V}_{ID})}{F(\mathcal{V}_{ID})}} \left(u_0 \prod_{i \in \mathcal{V}_{ID}} u_i\right)^{r_u}, g_1^{-\frac{1}{F(\mathcal{V}_{ID})}} g^{r_u}\right)$. It can be verified that defining $(psk_{ID}^{(1)}, psk_{ID}^{(2)})$ in this manner yields a valid user partial key of ID assuming $\tilde{r}_u = r_u - a/F(\mathcal{V}_{ID})$, since that

$$\begin{aligned} psk_{ID}^{(1)} &= g_1^{-\frac{J(\mathcal{V}_{ID})}{F(\mathcal{V}_{ID})}} \left(u_0 \prod_{i \in \mathcal{V}_{ID}} u_i\right)^{r_u} \\ &= g_2^a \left(g_2^{F(\mathcal{V}_{ID})} g^{J(\mathcal{V}_{ID})}\right)^{-a/F(\mathcal{V}_{ID})} \left(g_2^{F(\mathcal{V}_{ID})} g^{J(\mathcal{V}_{ID})}\right)^{r_u} \\ &= g_2^a (g_2^{F(\mathcal{V}_{ID})} g^{J(\mathcal{V}_{ID})})^{r_u - a/F(\mathcal{V}_{ID})} \\ &= g_2^a \left(u_0 \prod_{i \in \mathcal{V}_{ID}} u_i\right)^{\tilde{r}_u} \end{aligned}$$

and $psk_{ID}^{(2)} = g_1^{-\frac{1}{F(\mathcal{V}_{ID})}} g^{r_u} = g^{r_u - a/F(\mathcal{V}_{ID})} = g^{\tilde{r}_u}$. In this way, all the partial private keys computed by $\mathcal{C}$ are indistinguishable from the real one. However, the above simulation aborts if $F(\mathcal{V}_{ID}) = 0 \bmod p$. Given the assumption $l(n+1) < p$ which implies $0 \leq l \cdot k < p$ and $0 \leq x_0 + \sum_{i \in \mathcal{V}_{ID}} x_i < p$, it is easy to find that $F(\mathcal{V}_{ID}) = 0 \bmod p$ implies that $F(\mathcal{V}_{ID}) = 0 \bmod l$. Therefore, $F(\mathcal{V}_{ID}) \neq 0 \bmod l$ implies $F(\mathcal{V}_{ID}) \neq 0 \bmod p$.

Reveal-Secret-Key Oracle: Upon receiving a query for a public key of an identity ID, $\mathcal{C}$ looks up $\mathcal{L}$ to find out the corresponding entry. If it does not exist, $\mathcal{C}$ runs *UserKeyGen* to generate a secret and public key pair. It stores the key pair in $\mathcal{L}$ and returns the secret key usk_{ID}.

Replace-Public-Key Oracle: Upon receiving a query for a public key replace oracle request of an identity

ID, $\mathcal{C}$ looks up $\mathcal{L}$ to replace the corresponding entry. If it does not exist, $\mathcal{C}$ creates a new entry for this identity.

Reveal-Signing-Key Oracle: Consider a query for a signing key of identity ID and period t, $\mathcal{C}$ first computes $V_{ID} = H(ID)$ and $V_{ID,t} = H(ID, t)$, and then sets $\mathcal{V}_{ID} \subset \{1, \ldots, n\}$ to be the set of indices i such that $V_{ID}[i] = 1$ and $\mathcal{V}_{ID,t} \subset \{1, \ldots, n\}$ to be the set of indices i such that $V_{ID,t}[i] = 1$ respectively. $\mathcal{C}$ looks up $\mathcal{L}$ to check whether the user public key of ID has been replaced or not. If this user public key has already been replaced with the new user public key $upk_{ID} = (upk_{ID}^{(1)}, upk_{ID}^{(2)})$, $\mathcal{C}$ computes the signing key in case $F(\mathcal{V}_{ID,t}) \neq 0 \bmod q$ and $F(\mathcal{V}_{ID,t}) \neq 0 \bmod l$ as follows:

$\mathcal{C}$ picks d_u, $d_t \leftarrow_R \mathbb{Z}_p$ and computes $a_{ID} = g^{d_u}$, $b_{ID,t} = (upk_{ID}^{(2)})^{-J(\mathcal{V}_{ID,t})/F(\mathcal{V}_{ID,t})} f(\mathcal{V}_{ID})^{d_u} f(\mathcal{V}_{ID,t})^{d_t}$, $c_{ID,t} = \left(upk_{ID}^{(2)}\right)^{-1/F(\mathcal{V}_{ID,t})} g^{d_t}$. After that, $\mathcal{C}$ returns $(a_{ID}, b_{ID,t}, c_{ID,t})$ as the signing key.

The correctness can be shown as below:

$$
\begin{aligned}
b_{ID,t} &= \left(upk_{ID}^{(2)}\right)^{-J(\mathcal{V}_{ID,t})/F(\mathcal{V}_{ID,t})} f(\mathcal{V}_{ID})^{d_u} f(\mathcal{V}_{ID,t})^{d_t} \\
&= g_1^{-\frac{J(\mathcal{V}_{ID,t})}{F(\mathcal{V}_{ID,t})}x} \left(g_2^{F(\mathcal{V}_{ID,t})} g^{J(\mathcal{V}_{ID,t})}\right)^{d_t} f(\mathcal{V}_{ID})^{d_u} \\
&= g^{-\frac{axJ(\mathcal{V}_{ID,t})}{F(\mathcal{V}_{ID,t})}} \left(g_2^{F(\mathcal{V}_{ID,t})} g^{J(\mathcal{V}_{ID,t})}\right)^{\frac{ax}{F(\mathcal{V}_{ID,t})}} \left(g_2^{F(\mathcal{V}_{ID,t})} g^{J(\mathcal{V}_{ID,t})}\right)^{-\frac{ax}{F(\mathcal{V}_{ID,t})}} \left(g_2^{F(\mathcal{V}_{ID,t})} g^{J(\mathcal{V}_{ID,t})}\right)^{d_t} f(\mathcal{V}_{ID})^{d_u} \\
&= g^{-\frac{axJ(\mathcal{V}_{ID,t})}{F(\mathcal{V}_{ID,t})}} g^{abx} g^{\frac{axJ(\mathcal{V}_{ID,t})}{F(\mathcal{V}_{ID,t})}} \left(g_2^{F(\mathcal{V}_{ID,t})} g^{J(\mathcal{V}_{ID,t})}\right)^{\tilde{d}_t} f(\mathcal{V}_{ID})^{d_u} \\
&= g_2^{ax} f(\mathcal{V}_{ID,t})^{\tilde{d}_t} f(\mathcal{V}_{ID})^{d_u} \\
c_{ID,t} &= \left(upk_{ID}^{(2)}\right)^{-1/F(\mathcal{V}_{ID,t})} g^{d_t} \\
&= g^{\tilde{d}_t}
\end{aligned}
$$

Here, $\tilde{d}_t = d_t - \frac{a}{F(\mathcal{V}_{ID,t})}x$.

If the user public key has not been replaced yet and assuming $F(\mathcal{V}_{ID}) = 0 \bmod l$ and $F(\mathcal{V}_{ID,t}) \neq 0 \bmod l$, $\mathcal{C}$ picks $d_u, d_t \leftarrow_R \mathbb{Z}_p$ and extracts the user secret key usk_{ID} from the list $\mathcal{L}$. After that, $\mathcal{C}$ computes $a_{ID} = g^{d_u}$, $b_{ID,t} = g^{-usk_{ID}J(\mathcal{V}_{ID,t})/F(\mathcal{V}_{ID,t})} f(\mathcal{V}_{ID})^{d_u} f(\mathcal{V}_{ID,t})^{d_t \cdot usk_{ID}}$, $c_{ID,t} = g^{-usk_{ID}/F(\mathcal{V}_{ID,t})} g^{usk_{ID} \cdot d_t}$. After that, $\mathcal{C}$ returns $(a_{ID}, b_{ID,t}, c_{ID,t})$ as the signing key.

Otherwise, if the user public key has not been replaced yet and assuming $F(\mathcal{V}_{ID}) \neq 0 \bmod l$, $\mathcal{C}$ generates the signing key by performing *CL-Update* algorithm.

Sign Oracle: Consider a query for a signature of ID on m in time period t, $\mathcal{C}$ executes Oracle to get the signing key and generates the signature by performing *CL-Sign* algorithm.

Forgery. If $\mathcal{C}$ does not abort as a consequence of one of the queries above, $\mathcal{A}_1$, with probability at least ϵ, returns an identity ID^*, a user public key upk_{ID^*}, a message m^*, and valid forgery $(s^*, t^*) = \left(\left(D^*, B^*, a_{ID^*}, c_{ID^*,t^*}\right), t^*\right)$ on m^*. $\mathcal{C}$ computes $V_{ID^*} = H(ID^*)$, $V_{ID^*,t^*} = H(ID^*, t^*)$ and $M^* = H(m^*)$, and then sets $\mathcal{V}_{ID^*} \subset \{1, \ldots, n\}$ to be the set of indices i such that $V_{ID^*}[i] = 1$, $\mathcal{V}_{ID^*,t^*} \subset \{1, \ldots, n\}$ to be the set of indices i such that $V_{ID^*,t^*}[i] = 1$ and $M^*[i]$ be the i-th bit of M^* and lets $\mathcal{M}^* \subset \{1, \ldots, n\}$ be the set of indices i such that $M^*[i] = 1$, respectively. If $F(\mathcal{V}_{ID^*}) = 0 \bmod p$ or $F(\mathcal{V}_{ID^*,t^*}) = 0 \bmod p$ or $F(\mathcal{M}^*) = 0 \bmod p$, then $\mathcal{C}$ aborts. Otherwise, $\mathcal{C}$ extracts the user secret key usk_{ID^*} of the target user ID^* and computes

$$
\begin{aligned}
&\frac{B^*}{a_{ID^*}^{J(\mathcal{V}_{ID^*})} c_{ID^*,t^*}^{J(\mathcal{V}_{ID^*,t^*})} (D^*)^{J(\mathcal{M}^*)+usk_{ID^*} \cdot H_1(m^*)}} \\
&= \frac{(g_2^{\alpha} f(\mathcal{V}_{ID^*})^{d_u})^{usk_{ID^*}} \cdot f(\mathcal{V}_{ID^*})^{d_v} \cdot f(\mathcal{V}_{ID^*,t^*})^{d_t} \cdot f(\mathcal{M}^*)^{d_m} \cdot (upk_{ID^*}^{(1)})^{H_1(m^*)d_m}}{(g^{d_u \cdot usk_{ID^*}} g^{d_v})^{J(\mathcal{V}_{ID^*})} (g^{d_t})^{J(\mathcal{V}_{ID^*,t^*})} (g^{d_m})^{J(\mathcal{M}^*)+usk_{ID^*} \cdot H_1(m^*)}} \\
&= g^{ab \cdot usk_{ID^*}}
\end{aligned}
$$

$\mathcal{C}$ outputs $g^{ab} = g^{ab \cdot usk_{ID^*}} / g^{usk_{ID^*}}$ as the solution to the CDH problem instance. □

Probability analysis For the simulation to complete without aborting, we require the following conditions fulfilled:

1. All Reveal-Partial-Private-Key queries on an identity ID have $F(\mathcal{V}_{ID}) \neq 0 \bmod l$.
2. All Reveal-Signing-Key queries ID in period t have either $F(\mathcal{V}_{ID}) \neq 0 \bmod l$ or $F(\mathcal{V}_{ID,t}) \neq 0 \bmod l$.
3. All Sign queries (ID, m) in period t have either $F(\mathcal{V}_{ID}) \neq 0 \bmod l$ or $F(\mathcal{V}_{ID,t}) \neq 0 \bmod l$ or $F(\mathcal{M}) \neq 0 \bmod l$.
4. $F(\mathcal{V}_{ID^*}) = 0 \bmod l$ or $F(\mathcal{V}_{ID^*,t^*}) = 0 \bmod l$ or $F(\mathcal{M}^*) = 0 \bmod l$.

Define the events A_i, A^*, B_j, B^*, C_k, C^* as

$A_i : F(\mathcal{V}_{ID_i}) \neq 0 \bmod l$ where $i = 1, \ldots, q_e + q_{se} + q_s$
$A^* : F(\mathcal{V}_{ID^*}) = 0 \bmod p$
$B_j : F(\mathcal{V}_{ID_j,t_j}) \neq 0 \bmod l$ where $j = 1, \ldots, q_{se} + q_s$
$B^* : F(\mathcal{V}_{ID^*,t^*}) = 0 \bmod p$
$C_k : F(\mathcal{M}_k) \neq 0 \bmod l$ where $k = 1, \ldots, q_s$ $C^* : F(\mathcal{M}^*) = 0 \bmod p$

The probability of $\mathcal{C}$ not aborting is : Pr[not abort]$\geq Pr[(\bigwedge_{i=1}^{q_e+q_{se}+q_s} A_i \wedge A^*) \wedge (\bigwedge_{j=1}^{q_{se}+q_s} B_j \wedge B^*) \wedge (\bigwedge_{k=1}^{q_s} C_k \wedge C^*)]$. It is easy to see that the events $(\bigwedge_{i=1}^{q_e+q_{se}+q_s} A_i \wedge A^*)$, $(\bigwedge_{j=1}^{q_{se}+q_s} B_j \wedge B^*)$ and $(\bigwedge_{k=1}^{q_s} C_k \wedge C^*)$ are independent.

The assumption $l \cdot (n+1) < p$ implies if $F(\mathcal{V}_{ID_i}) = 0 \bmod p$ then $F(\mathcal{V}_{ID_i}) = 0 \bmod l$. In addition, it also implies that if $F(\mathcal{V}_{ID_i}) = 0 \bmod l$, there is a unique choice of k with $0 \leq k \leq n$ such that $F(\mathcal{V}_{ID_i}) = 0 \bmod p$. Since k, x_0 and vector (x_i) of length n are randomly chosen, we have

$$\begin{aligned} Pr[A^*] &= Pr[F(\mathcal{V}_{ID^*}) = 0 \bmod p \wedge F(\mathcal{V}_{ID^*}) = 0 \bmod l] \\ &= Pr[F(\mathcal{V}_{ID^*}) = 0 \bmod l] \\ &\quad \times Pr[F(\mathcal{V}_{ID^*}) = 0 \bmod p | F(\mathcal{V}_{ID^*}) = 0 \bmod l] \\ &= \frac{1}{l}\frac{1}{n+1} \end{aligned}$$

On the other hand, we have $Pr[\bigwedge_{i=1}^{q_e+q_{se}+q_s} A_i | A^*] = 1 - Pr[\bigvee_{i=1}^{q_e+q_{se}+q_s} \overline{A}_i | A^*] \geq 1 - \sum_{i=1}^{q_e+q_{se}+q_s} Pr[\overline{A}_i | A^*]$ where $\overline{A}_i$ denotes the event $F(\mathcal{V}_{ID_i}) = 0 \bmod l$.

Hence , we have

$$\begin{aligned} Pr[\bigwedge_{i=1}^{q_e+q_{se}+q_s} A_i \wedge A^*] &= Pr[A^*] Pr[\bigwedge_{i=1}^{q_e+q_{se}+q_s} A_i | A^*] \\ &\geq \frac{1}{l \cdot (n+1)} (1 - \frac{q_e + q_{se} + q_s}{l}) \end{aligned}$$

and setting $l = 2(q_e + q_{se} + q_s)$ as in the simulation gives $Pr[\bigwedge_{i=1}^{q_e+q_{se}+q_s} A_i \wedge A^*] \geq \frac{1}{4(q_e+q_{se}+q_s)(n+1)}$

A similar analysis for the sign queries gives the result $Pr[\overline{B}_j | B^*] \geq \frac{1}{4(q_{se}+q_s)(n+1)}$ and $Pr[\overline{C}_k | C^*] \geq \frac{1}{4(q_s(n+1)}$, and finally we get that Pr[not abort]$\geq Pr[\overline{A}_i | A^*] Pr[\overline{B}_j | B^*] \geq \frac{1}{64(q_e+q_{se}+q_s)(q_{se}+q_s)q_s(n+1)^3}$

If the simulation does not abort, $\mathcal{A}_1$ will produce a forged signature with probability at least ϵ. Thus, $\mathcal{C}$ can solve the CDH problem instance with probability $\epsilon' \geq \frac{\epsilon}{64(q_e+q_{se}+q_s)(q_{se}+q_s)q_s(n+1)^3}$

Theorem 2 *(Type II Existential Unforgeability). Our CL-KIS scheme is ϵ-existential unforgeable against Type II adversary with advantage at most ϵ, assuming that the ϵ'-CDH assumption holds in $\mathbb{G}_1$, where $\epsilon' \geq \frac{\epsilon}{64(q_{se}+q_s)(q_{se}+q_s)q_s(n+1)^3}$ where q_{se} is the number of queries made to the Reveal-Signing-Key oracle, q_s is the number of queries made to the Sign oracle, q_k is the number of queries made to the oracle Reveal-Secret-Key and Request-Public-Key altogether.*

Proof We assume that an ϵ-Type II adversary $\mathcal{A}_2$ for our scheme exists. From this forger, we construct an algorithm $\mathcal{C}$ that solves CDH with probability at least ϵ'.

The algorithm $\mathcal{C}$ is given a group $\mathbb{G}_1$ of prime order p with generator g and elements $g^a, g^b \in \mathbb{G}_1$ where a, b are selected uniformly at random from $\mathbb{Z}_p^*$. To be able to use $\mathcal{A}_2$ to output g^{ab}, $\mathcal{C}$ must be able to simulate a challenger for $\mathcal{A}_2$. Such a simulation can be created in the following way:

Initial. $\mathcal{C}$ executes *Setup* like Theorem 1. Specifically, $\mathcal{C}$ selects $\alpha \in_R \mathbb{Z}_p^*$ and assigns $g_1 = g^\alpha$, $g_2 = g^b$.

Attack. $\mathcal{C}$ maintains a list $\mathcal{L}$ which is initially empty and consists of entry $(ID, psk_{ID}, upk_{ID}, usk_{ID})$. $\mathcal{C}$ simulates all oracles as follows:

Request-Public-Key Oracle: Given an identity ID, if $ID = ID^*$, $\mathcal{C}$ selects a secret value $x \leftarrow_R \mathbb{Z}_p$ and computes the corresponding public key as $upk_{ID} = ((g^a)^x, (g^a)^{\alpha \cdot x})$. Otherwise, $\mathcal{C}$ looks up its list $\mathcal{L}$ to find out the corresponding entry. If it does not exist, $\mathcal{C}$ runs *UserKeyGen* and *ExtractPartialKey* algorithms to generate the secret/public key pair and partial private key respectively. It then stores $(ID, psk_{ID}, upk_{ID}, usk_{ID})$ into the list $\mathcal{L}$. In both cases, upk_{ID} is returned.

Reveal-Secret-Key Oracle: Upon receiving a query for a public key of an identity ID, if $ID = ID^*$, $\mathcal{C}$ aborts the simulation. Otherwise, $\mathcal{C}$ looks up $\mathcal{L}$ to find out the corresponding entry. If it does not exist, $\mathcal{C}$ runs *UserKeyGen* to generate a secret and public key pair. It stores the key pair in $\mathcal{L}$ and returns the secret key usk_{ID}.

Sign Oracle: For a given query for a signature of ID on m, $\mathcal{C}$ checks if the identity is equal to ID^*. If yes, it just aborts. Otherwise, $\mathcal{C}$ forges the signature as the same with Theorem 1.

Forgery. $\mathcal{C}$ executes *Forgery* like Theorem 1 and outputs the solution to the CDH problem instance.

The probability analysis is similar to the proof of Type I Adversary except the removal of the partial secret key extract query in Type II Adversary. □

6 Potential applications

The potential applications of our CL-KIS scheme are listed as follows:

Secret handshakes [2, 4, 30]: As a fundamental cryptographic primitive, secret handshake scheme enables the members of a certain group to authenticate each other in a private way. Prior to participation, users become group members by registering with group authorities (GAs) and obtaining membership credentials. After that, one member can prove to the communicating peer that it has a valid organizational credential, yet this proof hides the identity of the issuing organization unless the communicating party also has a valid credential from the same organization. The applications of secret handshake range from online dating in the online social networks to the high-bandwidth digital content protection (HDCP). In case the private key of a member has been leaked, it is natural to assume that the adversary can impersonate this member to authenticate with other members in the same group without being observed, which means that the security of the secret handshakes is broken totally. Taking the disastrous consequences of key exposure into account, the secret handshake scheme with key-exposure resilience has been introduced by integrating the idea of key insulated and secret handshakes [29]. However, the key escrow problem was not considered in [29] in the sense that the malicious PKG can impersonate any user since it can generate the private key for any user in the system. Thus, the proposed CL-KIS scheme can be used to construct secret handshakes scheme featured with key escrow resilience and key exposure resilience. Concretely, the KGC in the proposed CL-KIS scheme is acted by the GA in the secret handshake scheme, who is responsible for the registration of new members and for any subsequent membership revocation. After verifying a user's real identity, GA allocates and sends a list of pseudo-identities along with the corresponding partial private keys to this member secretly. Suppose Alice and Bob are two entities who want to authenticate each other anonymously, they need to send a random message to each other, signed with a certificateless key-insulated signature under the one-time pseudo-identity and the user public key. Alice and Bob can verify the certificateless key-insulated signature on the message, and learn that it definitely came from a legitimate member from the same group. Finally, only GA can open the real identity of a malicious member in a secret handshake instance by looking up the lists of the pseudo-identities corresponding to the real identity.

Multi-receiver authentication[18]: In a large-scale one-to-many (multicast) system such as for TV shopping, a broadcaster (root node) needs to communicate with a huge number of subscribing users (leaf nodes). To impede the pirate users from accessing the TV content free of charge, the broadcaster sends a personal information request along with its signature to the users. After receiving this request, users must transmit personal information to the broadcaster if this request is authenticated. Only when the requested personal information has been received, the broadcaster distributes TV content to this user accordingly. In such a system, the broadcaster has to distribute his new verification key to all subscribing users in an authentic manner to renew his verification key in case the private key of the broadcaster has been compromised. To deal with the key exposure problem without renewing a verification key frequently, the idea of key insulated signature has been employed to design secure provider authentication against key leakage [19, 20]. To enjoy the merits of traditional public key cryptography and ID-PKC altogether, the proposed CL-KIS scheme seems to be a promising approach to address this issue. Specifically, the KGC in the proposed CL-KIS scheme can be instantiated by a semi-trusted commercial organization or government department, who publishes the system parameters and issues the partial private key to the broadcaster.

7 Conclusion

In this paper, we have shown that Wan et al.'s CL-KIS scheme [26] is subjected to malicious-but-passive KGC attack by giving concrete attack. To fix the vulnerabilities in [26], we proposed an improved CL-KIS scheme provably secure in the standard model. As far as we know, this is the first provably secure CL-KIS without random oracles in the literature. Furthermore, we suggest several potential applications that can make use of our CL-KIS scheme, including secret handshakes and one-to-many authentication. We believe our scheme is very useful in many practical

applications, such as secret handshakes and one-to-many authentication.

Acknowledgments This research was supported by the National Natural Science Foundation of China General Projects Grant No. 61272029, 61003230, 61370026 and 61202445, Fundamental Research Funds for the Central Universities under Grant No. ZYGX2013J073, Applied Basic Research Program of Sichuan Province under Grant No. 2014JY0041, and the MOE key Laboratory for Transportation Complex Systems Theory and Technology, School of Traffic and Transportation, Beijing Jiaotong University. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

1. Al-Riyami SS, Paterson KG (2003) Certificateless public key cryptography. In: Advances in Cryptology-ASIACRYPT 2003, LNCS 2849. Springer, Berlin Heidelberg, pp 452–473
2. Ateniese G, Blanton M, Kirsch J (2007) Secret handshakes with dynamic and fuzzy matching. In: Proceedings of the 14^{th} annual network and distributed system security symposium-NDSS, vol 2, pp 159–177
3. Au MH, Chen J, Mu Y et al (2007) Malicious KGC attacks in certificateless cryptography. ACM symposium on Information, computer and communications security (ASIACCS'2007), pp 302–311
4. Balfanz D, Durfee G, Shankar N et al (2003) Secret handshakes from pairing-based key agreements. In: IEEE symposium on security and privacy, pp 180–196
5. Bao F, Deng RH, Zhu H (2003) Variations of Diffie-Hellman problem. In: 5^{th} International conference on information and communication security-ICICS 2003, LNCS 2836. Springer, Berlin Heidelberg, pp 301–312
6. Boneh D, Franklin M (2001) Identity-based encryption from the weil pairing. Advances in Cryptology-CRYPTO 2001, LNCS 2139. Springer, Berlin Heidelberg, pp 213–229
7. Canetti R, Goldreich O, Halevi S (1998) The random oracle methodology, revisited. In: Proceedings 30^{th} annual symposium on theory of computing (STOC'98), pp 209–218
8. Canetti R, Halevi S, Katz J (2003) A forward-secure public-key encryption scheme. Advances in Cryptology-EUROCRYPT 2003, LNCS 2656, pp 255–271
9. Diffie W, Hellman ME (1976) New directions in cryptography. IEEE Trans Inf Theory 22(6):644–654
10. Dodis Y, Katz J, Xu S, Yung M (2002) Strong key-insulated public key cryptosystems. Advances in Cryptology-Eurocrypt' 02, LNCS 2332. Springer, Berlin Heidelberg, pp 65–82
11. Dodis Y, Katz J, Xu S, Yung M (2003) Strong key-insulated signature scheme. In: Proceedings of PKC, LNCS 2567. Springer, Berlin Heidelberg, pp 130–144
12. Du H, Li J, Zhang Y, Li T, Zhang Y (2012) Certificate-based key-insulated signature. In: 3^{rd} International conference on data and knowledge Engineering-ICDKE 2012, LNCS 7696. Springer, Berlin Heidelberg, pp 206–220
13. He D, Chen J, Hu J (2011) An ID-based proxy signature schemes without bilinear pairings. Ann Telecommun 66(11–12): 657–662
14. Itkis G, Reyzin L (2001) Forward-secure signatures with optimal signing and verifying. Advances in Cryptology-CRYPTO' 01, LNCS 2139. Springer, Berlin Heidelberg, pp 499–514
15. Itkis G (2002) Intrusion-resilient signature: generic constructions, or defeating a strong adversary with minimal assumption. In: SCN' 02, LNCS 2576. Springer, Berlin Heidelberg, pp 102–118
16. Li J, Du H, Zhang Y, Li T, Zhang Y (2014) Provably secure certificate-based key-insulated signature scheme. Concurrency and Computation: Practice and Experience 26(8):546–1560
17. Liu JK, Au MH, Susilo W (2007) Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model. In: 2^{nd} ACM symposium on information, computer and communications security (ASIACCS 2007), pp 273–283
18. Miller CK (1999) Multicast networking and applications. Addison Wesley, Reading
19. Ohtake G, Hanaoka G, Ogawa K (2006) Provider authentication for bidirectional broadcasting service with fixed verification key. In: 2008 International symposium on information theory and its applications-ISITA 2006, pp 155–160
20. Ohtake G, Hanaoka G, Ogawa K (2008) An efficient strong key-insulated signature scheme and its application. In: 5^{th} European PKI workshop: theory and Practice-EuroPKI 2008, LNCS 5057. Springer, Berlin Heidelberg, pp 150–165
21. Paterson KG, Schuldt JCN (2006) Efficient identity-based signatures secure in the standard model. In: 11^{th} Australasian conference on information security and privacy (ACISP 2006), LNCS 4058. Springer, Berlin Heidelberg, pp 207–222
22. Shamir A (1984) Identity-based cryptosystems and signature schemes. Advances in Cryptology-CRYPTO 1984, LNCS 196. Springer, Berlin Heidelberg, pp 47–53
23. Shao Z (2012) Verifiably encrypted short signatures from bilinear maps. Ann Telecommun 67(9–10):437–445
24. Shim K-A (2014) On the security of verifiably encrypted signature schemes in a multi-user setting. Ann Telecommun 69(11–12): 585–591
25. Tiwari N, Padhye S, He D (2013) Efficient ID-based multiproxy multisignature without bilinear maps in ROM. Ann Telecommun 68(3–4):231–237
26. Wan Z, Lai X, Weng J et al (2009) Certificateless key-insulated signature without random oracles. J Zhejiang Univ (Sci) A 10(12):1790–1800
27. Waters B (2005) Efficient identity based encryption without random oracles. Advances in Cryptology-EUROCRYPT 2005, LNCS 3494. Springer, Berlin Heidelberg, pp 114–127
28. Weng J, Liu S, Chen K, Li X (2006) Identity-based key-insulated signature with secure key-updates. In: 2^{nd} SKLOIS conference on information security and cryptology (Inscrypt 2006), LNCS 4318, pp 13–26
29. Xiong H, Wu S, Li F, Qin Z (2015) Compact leakage-free ID-based signature scheme with applications to secret handshakes. Wirel Pers Commun 80(4):1671–1685
30. Xu S, Yung M (2004) K-anonymous secret handshakes with reusable credentials. In: Proceedings of the 11^{th} ACM conference on computer and communications security-ACM CCS 2004, pp 158–167
31. Yu J, Kong F, Cheng X et al (2012) Intrusion-resilient identity-based signature: security definition and construction. J Syst Softw 85(2):382–391
32. Zhou Y, Cao Z, Chai Z (2006) Identity based key insulated signature. In: 2^{nd} International conference on information security practice and experience (ISPEC 2006), LNCS 3903, pp 226–234