ITC 3/47
Journal of Information Technology and Control
Vol. 47 / No. 3 / 2018
pp. 575-587
DOI 10.5755/j01.itc.47.3.18528

Security Analysis of a Revocable and Strongly Unforgeable Identity-Based Signature Schemes

Received 2017/07/04 | Accepted after revision 2018/07/16

http://dx.doi.org/10.5755/j01.itc.47.3.18528

Security Analysis of a Revocable and Strongly Unforgeable Identity-Based Signature Scheme

Xiaodong Yang

College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, Gansu, China; State Key Laboratory of Cryptology, Beijing 100878, China; e-mail: y200888@163.com

Tingchun Ma, Ping Yang, Faying An, Caifen Wang

College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, Gansu, China; e-mails: matingchun@163.com, nwnuyp@163.com, nwnuafy@163.com, wangcf@nwnu.edu.cn

Corresponding author: y200888@163.com

Revocation functionality is very important for an identity-based signature to revoke users efficiently and securely. Hung *et al.* proposed a revocable identity-based signature (RIBS) scheme in the standard model and proved that it was strongly unforgeable against chosen-message attacks. However, we find that their RIBS scheme is insecure. In this paper, we provide a security analysis of Hung *et al.*'s RIBS scheme by showing concrete attacks. Our analysis shows that Hung *et al.*'s RIBS scheme does not satisfy the requirement of strong unforgeability, and thus, an adversary can forge a legal signature for a previously signed message. We also note serious flaws in their security proofs. The simulator of Hung *et al.*'s security argument cannot correctly answer the signing query in the security model, and the adversary can obtain any valid signature. Furthermore, we demonstrate that Hung *et al.*'s RIBS scheme is vulnerable to signing key exposure attack. To solve these problems, we construct an improved RIBS scheme with strong unforgeability and signing key exposure resistance in the standard model. Compared with previous RIBS schemes without random oracles, our scheme has advantages regarding computational cost and security.

KEYWORDS: Revocable identity-based signature, strong unforgeability, signing key exposure, standard model, bilinear pairing, security.

1. Introduction

Identity-based cryptography avoids public key certificates and simplifies key management in traditional certificated-based cryptosystem [19, 25]. In an identity-based signature (IBS) scheme, each user sets an email address or other identity information as the user's public key, and the corresponding private key

of the user is computed by a trusted private key generator (PKG). Due to its elimination of complicated certificate management, IBS has attracted great attention from researchers. Based on bilinear pairings, many IBS schemes in the random oracle model have been presented in [14, 20, 26, 27], but these IBS schemes could be insecure in the real world when random oracles are instantiated with specific hash functions [5]. Therefore, it is necessary to construct IBS schemes without random oracles in the standard model. The first IBS scheme in the standard model was presented by Paterson and Schuldt [15]. Since then, several IBS schemes without random oracles have appeared in [6, 7, 13].

However, all of the above-mentioned IBS schemes only satisfy existential unforgeability, in which an adversary is unable to forge a valid signature on a message that has not been signed before. Actually, the stronger security property, called strong unforgeability, is required in some practical applications [4]. Strong unforgeability preserves the property of existential unforgeability and prevents an adversary from forging signatures on previously signed messages. There are some transformation methods to convert existentially unforgeable IBS schemes to strongly unforgeable ones [8, 21]. In particular, several efficient strongly unforgeable IBS schemes in the standard model were directly constructed without the use of any transformation, such as [10, 16, 23].

Practical IBS schemes need an efficient revocation mechanism to revoke compromised or unauthorized users. However, the public key in the IBS scheme cannot be revoked directly since the user's identity is the user's public key. To achieve revocation functionality, some methods of effectively revoking the user in an identity-based setting have been proposed [1, 3, 17, 18]. The main idea of these methods is that PKG needs to periodically update the signing key for each non-revoked user. Sun *et al.* [22] presented a revocable identity-based signature (RIBS) scheme, but its security depends on the random oracle model. To avoid random oracles, Tsai *et al.* [24] proposed an RIBS scheme in the standard model, but this scheme only covers existential unforgeability. Naturally, constructing a strongly unforgeable IBS scheme is very interesting. Liu *et al.* [12] presented an RIBS scheme with strong unforgeability in the standard model. Although their scheme reduces the workload of the PKG's key update, the size of a user's secret key is very large. Meanwhile, the signing algorithm in Liu *et al.*'s RIBS scheme is based on the weakly secure Boneh-Boyen scheme [2], so it could not resist attack algorithms as presented in [11]. Hung *et al.* [9] presented a new RIBS scheme without random oracles in which the signing key of each non-revoked user is derived from a fixed secret key issued by the PKG via a secure channel and a dynamic update key sent by the PKG via a public channel. They also claimed that their RIBS scheme was strongly unforgeable under the computational Diffie-Hellma (CDH) assumption. Nevertheless, we find that their conclusion is incorrect.

In this paper, we first show that Hung *et al.*'s RIBS scheme is not strongly unforgeable by providing a concrete attack. Next, we show that the simulator of Hung *et al.*'s security proof cannot generate correct signatures to respond to the adversary's signing queries. We also show that their scheme does not consider a signing key exposure attack and cannot withstand this attack. Furthermore, we propose an improved RIBS scheme that provides strong unforgeability and signing key exposure resistance in the standard model. In addition, the analysis results indicate that our scheme has higher computational performance and security.

2. Preliminaries

In this section, we briefly review some preliminaries and security notions of strongly unforgeable RIBS scheme.

2.1. Bilinear Pairings and Security Assumption

Let p be a large prime, G and G_T be two multiplicative cyclic groups of order p, and g be a generator of G. An efficiently computable map e: $G\times G\rightarrow G_T$ is said to be a bilinear pairing if it has the following properties:

- Bilinearity: for any $a, b\in Z_p$, $e(g^a, g^b)= e(g, g)^{ab}= e(g^b, g^a)$.
- Non-degeneracy: $e(g, g)\neq 1$, where 1 is the identity element of G_T.
- Given a tuple $(g, g^a, g^b)\in G^3$ with unknown $a, b\in Z_p$, the CDH problem in G is to compute g^{ab}.

Definition 1. If no probabilistic polynomial-time al-

gorithm has a non-negligible probability of solving the CDH problem in G, we say that the CDH assumption holds in G.

2.2. Formal Definition and Security Model of RIBS Scheme

A strongly unforgeable RIBS scheme is defined by the following six algorithms:

Setup: On input of a security parameter λ, this algorithm outputs a PKG's master secret key msk and the public parameters pp.

Extract: On input of pp, msk and a user's identity ID, this algorithm outputs an ID's secret key sk_{ID}.

KeyUp: Taking as input pp, msk, a non-revoked user's identity ID and a time period t, this algorithm outputs the user's update key $vk_{ID,t}$.

SKGen: Taking as input pp, sk_{ID} and $vk_{ID,t}$, this algorithm outputs a signing key $dk_{ID,t}$ for a non-revoked user with identity ID.

Sign: On input of pp, a signing key $dk_{ID,t}$ and a message M, this algorithm outputs a signature σ on M.

Verify: Taking as input pp, an identity ID, a time period t, a message M and a signature σ, a verifier accepts σ if σ is a valid signature on M with respect to (ID, t); otherwise, the verifier rejects σ.

Liu *et al.* [12] and Hung *et al.* [9] gave the security model of strongly unforgeable RIBS scheme.

Definition 2. An RIBS scheme is said to be strongly unforgeable against adaptive chosen-message attacks if the probability that an adversary $\mathcal{A}$ wins the following game played with a challenger $\mathcal{C}$ is negligible.

Setup: $\mathcal{C}$ runs the algorithm **Setup** to produce the master secret key msk and the public parameters pp. Then, $\mathcal{C}$ sends pp to $\mathcal{A}$, and keeps msk secretly.

PKG queries: On receiving a secret key query on an identity ID, $\mathcal{C}$ runs the algorithm **Extract**(pp, msk, ID) to generate ID's secret key sk_{ID} and returns it to $\mathcal{A}$.

KeyUp queries: On receiving an update key query on an identity ID and a time period t, $\mathcal{C}$ runs the algorithm **KeyUp**(pp, msk, ID, t) to generate ID's update key $vk_{ID,t}$ and returns it to $\mathcal{A}$.

SKGen queries: On receiving a signing key query on (ID, t), $\mathcal{C}$ returns $\perp$ to $\mathcal{A}$ if ID has been revoked. Otherwise, $\mathcal{C}$ first asks for a secret key query on ID and an update key query on (ID, t) to get a secret key sk_{ID} and an update key $vk_{ID,t}$, respectively. Then, $\mathcal{C}$ runs the algorithm **SKGen**(pp, sk_{ID}, $vk_{ID,t}$) to generate ID's signing key $dk_{ID,t}$ and returns it to $\mathcal{A}$.

Signing queries: On receiving a signature query on a message M for an identity ID and a time period t, $\mathcal{C}$ first issues a **SKGen** query on (ID, t) to obtain a signing key $dk_{ID,t}$. Then, $\mathcal{C}$ runs the algorithm **Sign**(pp, $dk_{ID,t}$, M) to generate a signature σ on M and returns it to $\mathcal{A}$.

Forgery: $\mathcal{A}$ finally outputs a forged signature σ^* on a message M^* for an identity ID^* and a time period t^*. We say $\mathcal{A}$ wins in the above the game if the following conditions hold:

1. σ^* is a valid signature of M^* with regard to (ID^*, t^*).
2. ID^* and (ID^*, t^*) have not appeared in **PKG** queries and **SKGen** queries, respectively,
3. σ^* is not outputted by **Signing** query on (M^*, ID^*, t^*).

3. The Review of Hung et al.'s RIBS Scheme

3.1. Hung *et al.*'s RIBS Scheme

For subsequent convenience, we define some notations used in this paper. We let ID, t and M denote a user's identity, a time period and a message to be signed, respectively. ID^*, t^* and M^* denote a challenged user's identity, time period and message, respectively. We also assume that all identities and messages are bit strings of fixed lengths m and l, respectively. In practice, we can achieve identities and messages that are bit strings of arbitrary length by using two cryptographic hash functions: H_{ID}: $\{0,1\}^* \rightarrow \{0,1\}^m$ and H_M: $\{0,1\}^* \rightarrow \{0,1\}^l$.

The RIBS scheme of Hung *et al.* [9] consists of five algorithms: **Setup, Extract, KeyUp, Sign** and **Verify**. The details of this scheme are described below.

Setup: On input of a security parameter λ, PKG runs this algorithm to produce a master secret key msk and the public parameters pp.

1. Select two cyclic groups G and G_T of prime order $p>2^\lambda$, a generator g of G, a bilinear pairing e: $G\times G \rightarrow G_T$ and two random elements g_2, g_3 from G.
2. Pick two collision-resistant hash functions H_1: $\{0,1\}^* \rightarrow \{0,1\}^n$ and H_2: $\{0,1\}^* \rightarrow Z_p$, where n is the fixed length of the output of H_1. Further select random elements $u_0, u_i (1\le i\le m), v_0, v_j (1\le j\le n), w_0, w_k (1\le k\le l) \in G$.

3 Select two random integers $\alpha, \beta \in Z_p$ and compute g_2^{α}, g_2^{β} and $g_1 = g^{\alpha+\beta}$.

4 Publish the parameters $pp = (G, G_T, p, e, g, g_1, g_2, g_3, u_0, u_1, ..., u_m, v_0, v_1, ..., v_n, w_0, w_1, ..., w_l, H_1, H_2)$ and keep the master secret key $msk = (g_2^{\alpha}, g_2^{\beta})$.

For an identity $ID = (ID_1, ..., ID_m) \in \{0,1\}^m$, a string $T = (T_1, ..., T_n) \in \{0,1\}^n$ and a message $M = (M_1, ..., M_l) \in \{0,1\}^l$, we define the following three functions, which will be used in all subsequent schemes:

$F_{W,1}(ID) = u_0 \prod_{i=1}^{m} u_i^{ID_i}$, $F_{W,2}(T) = v_0 \prod_{j=1}^{n} v_j^{T_j}$ and

$F_{W,3}(M) = w_0 \prod_{k=1}^{l} w_k^{M_k}$.

Extract: Given a user's identity *ID*, the PKG runs the extraction algorithm to generate *ID*'s secret key sk_{ID}.

Randomly select $r_s \in Z_p$ and compute $sk_{ID} = (sk_1, sk_2) = (g_2^{\alpha} F_{W,1}(ID)^{r_s}, g^{r_s})$.

Send sk_{ID} to the user via a secure channel.

KeyUp: Given a time period *t* and a non-revoked user's identity *ID*, the PKG runs the key update algorithm to generate the user's update key $vk_{ID,t}$.

Compute $T = H_1(ID, t) = (T_1, ..., T_n) \in \{0,1\}^n$.

Select a random exponent $r_t \in Z_p$ and compute $vk_{ID,t} = (vk_1, vk_2) = (g_2^{\beta} F_{W,2}(T)^{r_t}, g^{r_t})$.

Send $vk_{ID,t}$ to the user via a public channel.

After receiving $sk_{ID} = (sk_1, sk_2)$ and $vk_{ID,t} = (vk_1, vk_2)$, a non-revoked user with identity *ID* computes $dk_{ID,t} = (dk_1, dk_2, dk_3) = (sk_1 \cdot vk_1, sk_2, vk_2)$ as the signing key at time period *t*.

Sign: For a message *M*, a time period *t* and a signing key $dk_{ID,t} = (dk_1, dk_2, dk_3)$, a signer with identity *ID* runs the signature generation algorithm to create a signature σ on *M*.

Select a random integer $r_m \in Z_p$ and compute $h = H_2(M, g^{r_m})$.

Compute $\sigma_1 = (dk_1)^h F_{W,3}(M)^{r_m}$, $\sigma_2 = (dk_2)^h$, $\sigma_3 = (dk_3)^h$ and $\sigma_4 = g^{r_m}$.

Output a signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ on *M*.

Verify: Given an identity *ID*, a time period *t*, a message *M* and a signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$, a verifier runs the signature verification algorithm to check the validity of σ.

1 Compute $T = H_1(ID, t)$ and $h = H_2(M, \sigma_4)$.

2 Verify the following equation

$$e(\sigma_1, g) = e(g_2, g_1)^h \cdot e(F_{W,1}(ID), \sigma_2) \cdot e(F_{W,2}(T), \sigma_3) \cdot e(F_{W,3}(M), \sigma_4).$$

If this equation holds, σ is a valid signature and the verifier accepts σ. Otherwise, the verifier rejects σ.

3.2. Hung *et al.*'s Security Proof

Hung *et al.* [9] proved that their RIBS scheme was strongly unforgeable against two types of adversaries. The first type of attacker is an external adversary who can request all queries except for the challenged user's secret key query. The second type of attacker is a revoked user who can request all queries except for the challenged user's update key query in the target time period. In this subsection, to save space, we mainly describe the simulator in Hung *et al.*'s security proof, which uses a forgery of an external adversary to solve the CDH problem.

Suppose that an external adversary $\mathcal{A}_1$ makes q_E secret key queries, q_U update key queries and q_S signing queries and outputs a forged signature for Hung *et al.*'s RIBS scheme. By using $\mathcal{A}_1$, a simulator $\mathcal{C}$ solves the CDH problem. Specifically, $\mathcal{C}$ is given a random instance $(g, g^a, g^b) \in G^3$ of the CDH problem, and $\mathcal{C}$'s goal is to calculate g^{ab}. The details of the interaction between $\mathcal{C}$ and $\mathcal{A}_1$ are described as follows.

Setup: $\mathcal{C}$ sets $l_v = 2(q_E + q_S)$ and $l_m = 2q_S$ satisfying $l_v(m+1) < p$ and $l_m(l+1) < p$. $\mathcal{C}$ picks three random integers $\beta \in Z_p$, $k_v (0 \le k_v \le m)$ and $k_m (0 \le k_m \le l)$, and sets $g_2 = g^b$ and $g_1 = g^a g^{\beta}$. This indicates that the master secret key $msk = (g_2^a, g_2^{\beta})$, but a is unknown to $\mathcal{C}$. Moreover, $\mathcal{C}$ chooses two collision-resistant hash functions H_1: $\{0,1\}^* \to \{0,1\}^n$ and H_2: $\{0,1\}^* \to Z_p$. Then, $\mathcal{C}$ selects random elements $x_0, x_1, ..., x_m \in Z_{l_v}$, $c_0, c_1, ..., c_l \in Z_{l_m}$, $y_0, y_1, ..., y_m$, $t_0, t_1, ..., t_n, z_0, z_1, ..., z_l \in Z_p$ and computes $v_0 = g^{t_0}$, $v_j = g^{t_j} (1 \le j \le n)$, $u_0 = g_2^{-l_v k_v + x_0} g^{y_0}$, $u_i = g_2^{x_i} g^{y_i} (1 \le i \le m)$, $w_0 = g_2^{-l_m k_m + c_0} g^{z_0}$ and $w_k = g_2^{c_k} g^{z_k} (1 \le k \le l)$. Finally, $\mathcal{C}$ sends the public parameters $pp = (G, G_T, p, e, g, g_1, g_2, g_3, u_0, u_1, ..., u_m, v_0, v_1, ..., v_n, w_0, w_1, ..., w_l, H_1, H_2)$ to $\mathcal{A}_1$.

To simplify the following description, given an identity *ID*, a string *T* and a message *M*, we define the fol-

lowing five functions

$$F(ID) = -l_v k_v + x_0 + \sum_{i=1}^{m} x_i ID_i,$$
$$J(ID) = y_0 + \sum_{i=1}^{m} y_i ID_i,\ E(T) = t_0 + \sum_{j=1}^{n} t_j ID_j,$$
$$K(M) = -l_m k_m + c_0 + \sum_{k=1}^{l} c_k M_k,$$
$$L(M) = z_0 + \sum_{k=1}^{l} z_k M_k.$$

PKG queries: On receiving a secret key query on ID, C computes $F(ID)$ and $J(ID)$. If $F(ID) = 0 \bmod p$, C exits the simulation. Otherwise, C randomly selects $r_s \in Z_p$, computes

$$sk_{ID} = (sk_1, sk_2) = ((g^a)^{\frac{-J(ID)}{F(ID)}} F_{W,1}(ID)^{r_s}, (g^a)^{\frac{-1}{F(ID)}} g^{r_s}),$$

and returns ID's secret key sk_{ID} to $\mathcal{A}_1$.

KeyUp queries: On receiving an update key query on (ID, t), C computes $T = H_1(ID, t)$. Then, C randomly selects $r_t \in Z_p$, uses the secret value β to compute $vk_{ID,t} = (vk_1, vk_2) = (g_2^{\beta} F_{W,2}(T)^{r_t}, g^{r_t})$, and returns the update key $vk_{ID,t}$ to $\mathcal{A}_1$.

Signing queries: On receiving a signature query on (M, ID, t), C answers this query in the following manner.

1. If $F(ID) \neq 0 \bmod p$, C computes a secret key sk_{ID} and an update key $vk_{ID,t}$ by the secret key simulation and the update key simulation, respectively. Then, C gets a signing key $dk_{ID,t}$ at time period t by combining sk_{ID} and $vk_{ID,t}$, runs the algorithm **Sign** to produce a signature σ on M, and returns σ to $\mathcal{A}_1$.
2. If $F(ID) = 0 \bmod p$, C computes $K(M)$ and $L(M)$. Based on $K(M) = 0 \bmod p$, it can be divided into two subcases.
   - If $K(M) = 0 \bmod p$, C aborts the simulation.
   - If $K(M) \neq 0 \bmod p$, C randomly selects r_s, $r_t, r_m \in Z_p$, and computes $T = H_1(ID, t)$ and $h = H_2(M, g^{r_m})$. Next, C computes $\sigma_1 = ((g_2^{\beta}) F_{W,1}(ID)^{r_s} F_{W,2}(T)^{r_t})^h (g^a)^{\frac{-L(M)h}{K(M)}} F_{W,3}(M)^{r_m}$, $\sigma_2 = g^{r_s h}$, $\sigma_3 = g^{r_t h}$ and $\sigma_4 = (g^a)^{\frac{-h}{K(M)}} g^{r_m}$, and returns a signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ on M to $\mathcal{A}_1$.

Forgery: $\mathcal{A}_1$ eventually outputs a forged signature $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*)$ with regard to (M^*, ID^*, t^*). If $F(ID^*) \neq 0 \bmod p$ or $K(M^*) \neq 0 \bmod p$, C aborts. Otherwise, C computes $T^* = H_1(ID^*, t^*)$ and $h^* = H_2(M^*, \sigma_4^*)$, and then outputs the CDH value

$$g^{ab} = \frac{(\sigma_1^*)^{1/h^*}}{g_2^{\beta} (\sigma_2^*)^{J(ID^*)/h^*} (\sigma_3^*)^{E(T^*)/h^*} (\sigma_4^*)^{L(M^*)/h^*}}.$$

4. Cryptanalysis of Hung *et al.*'s RIBS Scheme

4.1. Hung *et al.*'s RIBS Vulnerability Against Strong Unforgeability

In [9], Hung *et al.* demonstrated that their RIBS scheme satisfies strong unforgeability in the standard model. However, we find that their conclusion is not true. Concretely, given a message-signature pair (M, σ), there exists an adversary $\mathcal{A}$ that can always succeed in forging a new valid signature on the same message M as shown below.

$\mathcal{A}$ intercepts a valid signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ on a message M of an identity ID at time period t, where $T = H_1(ID, t)$, $h = H_2(M, \sigma_4)$, $\sigma_2 = g^{r_s h}$, $\sigma_3 = g^{r_t h}$, $\sigma_1 = g_2^{(\alpha+\beta)h} F_{W,1}(ID)^{r_s h} F_{W,2}(T)^{r_t h} F_{W,3}(M)^{r_m}$ and $\sigma_4 = g^{r_m}$.

$\mathcal{A}$ randomly selects $r_t' \in Z_p$, computes $\sigma_1' = \sigma_1 F_{W,2}(T)^{r_t'}$ and $\sigma_3' = \sigma_3 g^{r_t'} = g^{r_t h + r_t'}$, and sets $\sigma_2' = \sigma_2$, $\sigma_4' = \sigma_4$.

$\mathcal{A}$ outputs a signature $\sigma' = (\sigma_1', \sigma_2', \sigma_3', \sigma_4')$ on M.

It is obvious that σ' is a legal signature of ID on M at time period t since σ' satisfies the following verification equation in Hung *et al.*'s RIBS scheme:

$$\begin{aligned} e(\sigma_1', g) &= e(\sigma_1 F_{W,2}(T)^{r_t'}, g) \\ &= e(g_2^{(\alpha+\beta)h} F_{W,1}(ID)^{r_s h} F_{W,2}(T)^{r_t h} F_{W,3}(M)^{r_m} F_{W,2}(T)^{r_t'}, g) \\ &= e(g_2^{(\alpha+\beta)h} F_{W,1}(ID)^{r_s h} F_{W,2}(T)^{r_t h + r_t'} F_{W,3}(M)^{r_m}, g) \\ &= e(g_2^{(\alpha+\beta)h}, g) \cdot e(F_{W,1}(ID)^{r_s h}, g) \cdot e(F_{W,2}(T)^{r_t h + r_t'}, g) \cdot e(F_{W,3}(M)^{r_m}, g) \\ &= e(g_2, g_1)^h e(F_{W,1}(ID), \sigma_2') e(F_{W,2}(T), \sigma_3') e(F_{W,3}(M), \sigma_4'). \end{aligned}$$

From the above attack, we know that the adversary $\mathcal{A}$ does not make secret queries on ID or update key que-

ries on (*ID, t*), and σ' is not the output of the signing queries on (*M, ID, t*). Therefore, Hung *et al.*'s RIBS scheme does not satisfy the requirement of strong unforgeability. The main reason for the inability to resist the above attack is that the previous signature σ and the forged signature σ' have the same hash value h. Hence, the adversary can easily re-randomize existing signatures to forge new signatures on the same messages.

4.2. Analysis of Hung *et al.*'s Security Proof

In Subsection 3.2, we have reviewed the security proof of Hung *et al.*'s RIBS scheme against an external adversary. Here, we show that there are serious flaws in their proofs.

When an external adversary $\mathcal{A}_1$ issues a signature query on (*M, ID, t*), the simulator C of Hung *et al.* randomly selects $r_s, r_t, r_m \in Z_p$ and computes $T = H_1(ID,t)$ and $h = H_2(M, g^{r_m})$. If $F(ID) = 0 \bmod p$ and $K(M) \neq 0 \bmod p$, C computes a signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ on *M*, where $\sigma_2 = g^{r_s h}$, $\sigma_3 = g^{r_t h}$, $\sigma_4 = (g^a)^{\frac{-h}{K(M)}} g^{r_m} = g^{r_m - \frac{ah}{K(M)}} = g^{r_m'}$ and $\sigma_1 = ((g_2^{\beta})F_{W,1}(ID)^{r_s} F_{W,2}(T)^{r_t})^h (g^a)^{\frac{-L(M)h}{K(M)}} F_{W,3}(M)^{r_m}$.

From the above simulation, we can see that the hash value h in σ_1 is calculated by M and g^{r_m}, just as $h = H_2(M, g^{r_m})$. However, in accordance with the verification algorithm **Verify** of Hung *et al.*'s RIBS scheme, the hash value h' in the verification equation is calculated by M and σ_4, just as $h' = H_2(M, \sigma_4)$. Because of $r_m' = r_m - \frac{ah}{K(M)}$, we have $r_m' \neq r_m$. Furthermore, we get $h' = H_2(M, \sigma_4) = H_2(M, g^{r_m'}) \neq H_2(M, g^{r_m}) = h$ since H_2 is collision-resistant. Then, we have

$$e(\sigma_1, g) = e(g_2, g_1)^h \cdot e(F_{W,1}(ID), \sigma_2) \cdot e(F_{W,2}(T), \sigma_3)$$
$$\cdot e(F_{W,3}(M), \sigma_4) \neq e(g_2, g_1)^{h'} \cdot e(F_{W,1}(ID), \sigma_2)$$
$$\cdot e(F_{W,2}(T), \sigma_3) \cdot e(F_{W,3}(M), \sigma_4).$$

Due to $h' \neq h$, the simulated signature σ computed by C cannot satisfy the verification equation of Hung *et al.*'s RIBS scheme. Therefore, the simulator C of Hung *et al.* is unable to correctly answer the signing queries defined in the security model, and the adversary $\mathcal{A}_1$ cannot obtain any valid signature for $F(ID) = 0 \bmod p$ and $K(M) \neq 0 \bmod p$. Moreover, based on the wrong simulation, it is not correct that the security of Hung *et al.*'s RIBS scheme is directly reduced to the hardness of the CDH assumption by using $\mathcal{A}_1$ against their scheme. Thus, their conclusion of Theorem 1 in [9] is wrong. Note that a similar security analysis applies to an internal adversary. Furthermore, we could conclude that Theorem 2 in [9] is also wrong.

4.3. Hung *et al.*'s RIBS Vulnerability Against Signing Key Exposure

Hung *et al.*'s RIBS scheme does not consider a realistic threat known as signing key exposure [1,3,12]. That is, if a RIBS scheme is secure against signing key exposure attack, the exposure of a user's signing key in the current time period does not reveal any information on subsequent signing keys of the user in other time periods. More concretely, let t^* be a target time period and ID^* be a target user's identity who is not revoked at times *t* and t^*. An adversary is allowed to get a signing key $dk_{ID^*,t}$ and two update keys $vk_{ID^*,t}$ and vk_{ID^*,t^*} at time *t* and t^*, respectively. Then, the adversary could not obtain a secret key sk_{ID^*} from $dk_{ID^*,t}$ and $vk_{ID^*,t}$, and could not get a signing key dk_{ID^*,t^*} at time t^* from $dk_{ID^*,t}$ and vk_{ID^*,t^*}.

In this subsection, we show that Hung *et al.*'s RIBS scheme is unable to withstand signing key exposure attack as follows.

1. The adversary $\mathcal{A}$ obtains a compromised signing key $dk_{ID^*,t}$ on time period *t*. In Hung *et al.*'s RIBS scheme, $dk_{ID^*,t} = (dk_1, dk_2, dk_3) = (sk_1^* \cdot vk_1, sk_2^*, vk_2)$ is computed by a secret key $sk_{ID^*} = (sk_1^*, sk_2^*)$ and an update key $vk_{ID^*,t} = (vk_1, vk_2)$ at time period *t*. Note that $\mathcal{A}$ can get $vk_{ID^*,t}$ and vk_{ID^*,t^*}, since all update keys are transmitted by a public channel.
2. $\mathcal{A}$ who has $dk_{ID^*,t} = (dk_1, dk_2, dk_3)$ and $vk_{ID^*,t} = (vk_1, vk_2)$ can easily recover the target user's secret key $sk_{ID^*} = (sk_1^*, sk_2^*)$ such that $sk_1^* = dk_1 / vk_1$ and $sk_2^* = dk_2$.
3. $\mathcal{A}$ computes the signing key $dk_{ID^*,t^*} = (dk_1^*, dk_2^*, dk_3^*) = (sk_1^* \cdot vk_1^*, sk_2^*, vk_2^*)$ from $sk_{ID^*} = (sk_1^*, sk_2^*)$ and $vk_{ID^*,t^*} = (vk_1^*, vk_2^*)$ at time period t^*.

Therefore, Hung *et al.*'s RIBS scheme is vulnerable to signing key exposure.

5. An improved RIBS Scheme with Strong Unforgeability

5.1. Our Construction

Based on Hung *et al.*'s RIBS scheme [9], we construct a secure RIBS scheme that is strongly unforgeable in the standard model. To resist attacks presented in Section 4, we add a probabilistic signing key algorithm (**SKGen**) and modify the signing algorithm in our improved scheme. Our IBS scheme is described as follows.

The algorithms **Setup, Extract** and **KeyUp** are the same as those of Hung *et al.*'s RIBS scheme described in Section 3.1.

SKGen: After receiving a secret key $sk_{ID} = (sk_1, sk_2)$ and an update key $vk_{ID,t} = (vk_1, vk_2)$ at time period t from the PKG, a non-revoked user with identity ID outputs a signing key via the following steps:

Select two random integers $r, s \in Z_p$ and compute $T = H_1(ID, t)$.

Compute the signing key $dk_{ID,t} = (dk_1, dk_2, dk_3) = (sk_1 \cdot F_{W,1}(ID)^s \cdot vk_1 \cdot F_{W,2}(T)^r, sk_2 g^s, vk_2 g^r)$.

Sign: Given a message M and a signing key $dk_{ID,t} = (dk_1, dk_2, dk_3)$ on time period t, a signer with identity ID performs the following steps:

1. Select a random exponent $r_m \in Z_p$ and compute $\sigma_4 = g^{r_m}$.
2. Set $\sigma_2 = dk_2$ and $\sigma_3 = dk_3$.
3. Compute $h = H_2(ID, M, t, \sigma_2, \sigma_3, \sigma_4)$ and $\sigma_1 = dk_1 \cdot (F_{W,3}(M) g_3^h)^{r_m}$.
4. Output a signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ on M.

Verify: Given a signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ of an identity ID on a message M at time period t, a verifier checks the validity of σ as follows:

1. Compute $T = H_1(ID, t)$ and $h = H_2(ID, M, t, \sigma_2, \sigma_3, \sigma_4)$.
2. Verify the following equation

$$e(\sigma_1, g) = e(g_2, g_1) \cdot e(F_{W,1}(ID), \sigma_2) \cdot e(F_{W,2}(T), \sigma_3) \cdot e(F_{W,3}(M) g_3^h, \sigma_4).$$

If this equation holds, the verifier accepts σ. Otherwise, the verifier rejects σ.

Note that we introduce an algorithm **SKGen** to re-randomize the signing key in our scheme, but the signing key in Hung *et al.*'s RIBS scheme has the same random numbers used in the secret key and the update key. For a signing key $dk_{ID,t} = (dk_1, dk_2, dk_3) = (g_2^{\alpha+\beta} F_{W,1}(ID)^{r_s+s} F_{W,2}(T)^{r_t+r}, g^{r_s+s}, g^{r_t+r})$ and an update key $vk_{ID,t} = (vk_1, vk_2) = (g_2^{\beta} F_{W,2}(T)^{r_t}, g^{r_t})$ at time period t, it is infeasible to recover the secret key $sk_{ID} = (sk_1, sk_2) = (g_2^{\alpha} F_{W,1}(ID)^{r_s}, g^{r_s})$ from $dk_{ID,t}$ and $vk_{ID,t}$. Hence, our scheme is secure against signing key exposure attack. In the signing algorithm of our scheme, the hash value $h = H_2(ID, M, t, dk_2, dk_3, g^{r_m})$ is hashed by a signer's identity ID, a message M, a time period t, a part $(\sigma_2, \sigma_3) = (dk_2, dk_3)$ of a signing key $dk_{ID,t}$ and an element for the randomness of M. Due to the collision resistance of H_2, it is difficult to re-randomize our RIBS scheme to forge a new signature without a signing key. Therefore, our scheme is resistant to the attack presented in Section 4.1.

5.2. Security Proof

In this subsection, we reduce the security of our RIBS scheme to the CDH assumption. Our proof approach is similar to that of Hung *et al.*'s security proof [9]. To simplify the security analysis, we also classify attacks into two categories: type-1 adversary and type-2 adversary. The type-1 adversary $\mathcal{A}_1$ models an external attacker who cannot request the secret key query on ID^* and the signing key query on (ID^*, t^*). The type-2 adversary $\mathcal{A}_2$ models an internal attacker (or a revoked user) who cannot issue the update key and signing key queries on (ID^*, t^*). By presenting the following lemmas, we prove that our RIBS scheme is strongly unforgeable against two types of adversaries.

Lemma 1. *If there exists a type-1 adversary $\mathcal{A}_1$ breaking strong unforgeability of our RIBS scheme, then the CDH problem can be solved.*

Proof. Suppose that $\mathcal{A}_1$ forges a valid signature for our RIBS scheme after making q_E secret key queries, q_U update key queries, q_K signing key queries and q_S signing queries. Then, we are able to construct a simulator $\mathcal{C}$ that solves the CDH problem by using $\mathcal{A}_1$. $\mathcal{C}$ is given a random instance $(g, g^a, g^b) \in G^3$ of the CDH problem, and the goal of $\mathcal{C}$ is to calculate g^{ab}.

Setup: Let $l_v = 2(q_1 + q_S)$ and $l_m = 2q_S$ satisfying $l_v(m+1) < p$ and $l_m(l+1) < p$, where $q_1 = \max\{q_E, q_K\}$. $\mathcal{C}$ randomly selects $d \in Z_p$ and computes

$g_3 = g^d$. C sets other parameters $(G, G_T, p, g, g_1, g_2, u_0, u_1, ..., u_m, v_0, v_1, ..., v_n, w_0, w_1, ..., w_l, H_1, H_2)$ as described in Section 3.2, where $g_1 = g^a g^\beta$ and $g_2 = g^b$. Note that the master secret key $msk = (g_2^a, g_2^\beta)$, but a is unknown to C. Meanwhile, C sends the public parameters pp to $\mathcal{A}_1$.

To make the expression simpler, we also define five functions: $F(ID)$, $J(ID)$, $E(T)$, $K(M)$ and $L(M)$. Accordingly, we have the following equations $F_{W,1}(ID) = g_2^{F(ID)} g^{J(ID)}$, $F_{W,2}(T) = g^{E(T)}$ and $F_{W,3}(M) = g_2^{K(M)} g^{L(M)}$.

Please refer to Section 3.2 for the detailed description of these functions.

PKG queries: C maintains an initially empty list L_{sk} that consists of tuples in the form of (ID, r_s). When $\mathcal{A}_1$ requests a secret key query on ID, C computes $F(ID)$ and $J(ID)$. If $F(ID) = 0 \bmod p$, C terminates the simulation. If $F(ID) \neq 0 \bmod p$, C first recalls r_s from L_{sk} if there is a tuple (ID, r_s) in L_{sk}. Otherwise, C randomly selects $r_s \in Z_p$ and adds (ID, r_s) to L_{sk}. Then, C computes $sk_{ID} = (sk_1, sk_2) = ((g^a)^{\frac{-J(ID)}{F(ID)}} F_{W,1}(ID)^{r_s}, (g^a)^{\frac{-1}{F(ID)}} g^{r_s})$ and returns ID's secret key sk_{ID} to $\mathcal{A}_1$.

KeyUp queries: C maintains an initially empty list L_{vk} that consists of tuples in the form of (ID, t, r_t). When $\mathcal{A}_1$ requests an update key query on (ID, t), C computes $T = H_1(ID, t)$. Next, C recalls r_t from L_{vk} if there is a tuple (ID, t, r_t) in L_{vk}. Otherwise, C randomly selects $r_t \in Z_p$ and adds (ID, t, r_t) to L_{vk}. Then, C uses the secret value β to compute $vk_{ID,t} = (vk_1, vk_2) = (g_2^\beta F_{W,2}(T)^{r_t}, g^{r_t})$ and returns an ID's update key $vk_{ID,t}$ to $\mathcal{A}_1$.

SKGen queries: C maintains an initially empty list L_{dk} that consists of tuples in the form of (ID, t, r, s). On receiving a signing key query on (ID, t), C returns $\perp$ to $\mathcal{A}_1$ if ID has been revoked. Otherwise, C simulates the signing key generation algorithm by executing the following steps:

1. Recall (r, s) from L_{dk} if there is a tuple (ID, t, r, s) in L_{dk}. Otherwise, select random elements $r, s \in Z_p$ and add (ID, t, r, s) to L_{dk}.
2. Ask for a secret key query on ID and an update key query on (ID, t) to get a secret key $sk_{ID} = (sk_1, sk_2)$ and an update key $vk_{ID,t} = (vk_1, vk_2)$, respectively.
3. Compute $T = H_1(ID, t)$ and $dk_{ID,t} = (dk_1, dk_2, dk_3) = (sk_1 \cdot F_{W,1}(ID)^s \cdot vk_1 \cdot F_{W,2}(T)^r, sk_2 \cdot g^s, vk_2 \cdot g^r)$.
4. Return a signing key $dk_{ID,t} = (dk_1, dk_2, dk_3)$ to $\mathcal{A}_1$.

Note that C cannot compute a secret key and aborts simulation in case of $F(ID) = 0 \bmod p$.

Signing queries: On receiving a signature query on (M, ID, t), C answers this query in the following way.

Case 1: If $F(ID) \neq 0 \bmod p$, C makes the signing key query on (ID, t) to get a signing key $dk_{ID,t}$, and then runs the algorithm **Sign** to produce a signature σ on M.

Case 2: If $F(ID) = 0 \bmod p$, C computes $K(M)$ and $L(M)$, and considers the following two subcases:

- *Case* 2.1: If $K(M) = 0 \bmod p$, C aborts the simulation.
- *Case* 2.1: If $K(M) \neq 0 \bmod p$, C recalls $r_s, r_t, (r, s)$ from L_{sk}, L_{vk} and L_{dk}, respectively, if there were defined. Otherwise, C selects random elements $r_s, r_t, r, s \in Z_p$ and adds (ID, r_s), (ID, t, r_t), (ID, t, r, s) to L_{sk}, L_{vk} and L_{dk}, respectively. Then, C randomly selects $r_m \in Z_p$, and computes $T = H_1(ID, t)$, $\sigma_2 = g^{r_s + s}$, $\sigma_3 = g^{r_t + r}$, $\sigma_4 = (g^a)^{\frac{-1}{K(M)}} g^{r_m}$, $h = H_2(ID, M, t, \sigma_2, \sigma_3, \sigma_4)$ and $\sigma_1 = F_{W,1}(ID)^{r_s + s} \cdot (g_2^\beta) F_{W,2}(T)^{r_t + r} \cdot (g^a)^{\frac{-L(M) - hd}{K(M)}} \cdot (F_{W,3}(M) g_3^h)^{r_m}$. Finally, C returns a signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ on M to $\mathcal{A}_1$.

Let $r_s' = r_s + s$, $r_t' = r_t + r$ and $r_m' = r_m - \frac{a}{K(M)}$, we have $\sigma_2 = g^{r_s + s} = g^{r_s'}$, $\sigma_3 = g^{r_t + r} = g^{r_t'}$, $\sigma_4 = (g^a)^{\frac{-1}{K(M)}} g^{r_m} = g^{r_m - \frac{a}{K(M)}} = g^{r_m'}$, $h = H_2(ID, M, t, \sigma_2, \sigma_3, \sigma_4) = H_2(ID, M, t, g^{r_s'}, g^{r_t'}, g^{r_m'})$ and

$$
\begin{aligned}
&\sigma_1 = F_{W,1}(ID)^{r_s + s} \cdot (g_2^\beta) F_{W,2}(T)^{r_t + r} \cdot (g^a)^{\frac{-L(M) - hd}{K(M)}} \\
&\cdot (F_{W,3}(M) g_3^h)^{r_m} \\
&= (g_2^\beta) \cdot F_{W,1}(ID)^{r_s'} \cdot F_{W,2}(T)^{r_t'} \cdot g_2^a \cdot g_2^{-a} \cdot g^{\frac{-aL(M)}{K(M)}} \\
&\cdot g^{\frac{-ahd}{K(M)}} \cdot (F_{W,3}(M) g_3^h)^{r_m} \\
&= (g_2^{a+\beta}) F_{W,1}(ID)^{r_s'} F_{W,2}(T)^{r_t'} (g_2^{K(M)} g^{L(M)} (g^d)^h)^{\frac{-a}{K(M)}} \\
&\cdot (F_{W,3}(M) g_3^h)^{r_m} \\
&= (g_2^{a+\beta}) \cdot F_{W,1}(ID)^{r_s'} \cdot F_{W,2}(T)^{r_t'} \cdot (F_{W,3}(M) g_3^h)^{\frac{-a}{K(M)}} \\
&\cdot (F_{W,3}(M) g_3^h)^{r_m} \\
&= g_2^{a+\beta} F_{W,1}(ID)^{r_s'} F_{W,2}(T)^{r_t'} (F_{W,3}(M) g_3^h)^{r_m'}.
\end{aligned}
$$

It is easy to verify that the signature $\sigma=(\sigma_1,\sigma_2,\sigma_3,\sigma_4)$ generated by the simulator C is valid since

$$
\begin{aligned}
&e(\sigma_1,g)=e(g_2^{a+\beta}F_{W,1}(ID)^{r_s'}F_{W,2}(T)^{r_t'}(F_{W,3}(M)g_3^h)^{r_m'},g)\\
&=e(g_2,g^{a+\beta})\cdot e(F_{W,1}(ID),g^{r_s'})\cdot e(F_{W,2}(T),g^{r_t'})\\
&\cdot e(F_{W,3}(M)g_3^h,g^{r_m'})\\
&=e(g_2,g_1)\cdot e(F_{W,1}(ID),\sigma_2)\cdot e(F_{W,2}(T),\sigma_3)\\
&\cdot e(F_{W,3}(M)g_3^h,\sigma_4).
\end{aligned}
$$

Forgery: $\mathcal{A}_1$ eventually outputs a forged signature $\sigma^*=(\sigma_1^*,\sigma_2^*,\sigma_3^*,\sigma_4^*)$ with regard to (M^*,ID^*,t^*), where ID^* and (ID^*,t^*) have not appeared in the secret key and signing key queries, and σ^* is not generated by signing queries on (M^*,ID^*,t^*). If $F(ID^*)\neq 0 \bmod p$ or $K(M^*)\neq 0 \bmod p$, C aborts the simulation. Otherwise, C computes $T^*=H_1(ID^*,t^*)$ and $h^*=H_2(ID^*,M^*,t^*,\sigma_2^*,\sigma_3^*,\sigma_4^*)$, and uses the secret value β to calculate the CDH value g^{ab} in the following manner:

$$
\begin{aligned}
&\frac{\sigma_1^*}{g_2^{\beta}(\sigma_2^*)^{J(ID^*)}(\sigma_3^*)^{E(T^*)}(\sigma_4^*)^{L(M^*)}(\sigma_4^*)^{h^*d}}\\
&=\frac{g_2^{a+\beta}F_{W,1}(ID^*)^{r_s^*}F_{W,2}(T^*)^{r_t^*}(F_{W,3}(M^*)g_3^{h^*})^{r_m^*}}{g_2^{\beta}(g^{r_s^*})^{J(ID^*)}(g^{r_t^*})^{E(T^*)}(g^{r_m^*})^{L(M^*)}(g^{r_m^*})^{h^*d}}\\
&=\frac{g_2^a(g_2^{F(ID^*)}g^{J(ID^*)})^{r_s^*}(g^{E(T^*)})^{r_t^*}(g_2^{K(M^*)}g^{L(M^*)}g_3^{h^*})^{r_m^*}}{(g^{J(ID^*)})^{r_s^*}(g^{E(T^*)})^{r_t^*}(g^{L(M^*)})^{r_m^*}(g_3^{h^*})^{r_m^*}}\\
&=g_2^a\ (\text{since } F(ID^*)=K(M^*)=0 \bmod p)\\
&=g^{ab}.
\end{aligned}
$$

We now analyse the probability that C does not abort in the above simulation. If C completes the entire simulation, the following events must occur.

1. All secret key and signing key queries on an identity ID satisfy $F(ID)\neq 0 \bmod p$.
2. All signing queries on an identity ID and a message M satisfy $F(ID)\neq 0 \bmod p$ or $K(M)\neq 0 \bmod p$.
3. The forged signature σ^* of identity ID^* on message M^* satisfies $F(ID^*)=0 \bmod p$ and $K(M^*)=0 \bmod p$.

To simplify the analysis, we define the events as follows:

$$
\begin{aligned}
&A_i: F(ID_i)\neq 0 \bmod p \text{ for } i=1,\ldots,q_1+q_S,\\
&A^*: F(ID^*)=0 \bmod p,\\
&B_j: K(M_j)\neq 0 \bmod p \text{ for } j=1,\ldots,q_S,\\
&B^*: K(M^*)=0 \bmod p.
\end{aligned}
$$

Thus, the probability that C does not abort is

$$
\begin{aligned}
&\Pr[\neg abort]\geq \Pr[\bigcap_{i=1}^{q_1+q_S}A_i\cap A^*\cap\bigcap_{j=1}^{q_S}B_j\cap B^*]\\
&=\Pr[\bigcap_{i=1}^{q_1+q_S}A_i\cap A^*]\Pr[\bigcap_{j=1}^{q_S}B_j\cap B^*]\\
&=\Pr[A^*]\Pr[\bigcap_{i=1}^{q_1+q_S}A_i|A^*]\Pr[B^*]\Pr[\bigcap_{j=1}^{q_S}B_j|B^*].
\end{aligned}
$$

Since $l_m(l+1)<p$, we have that

$$
\begin{aligned}
&\Pr[A^*]=\Pr[F(ID^*)=0 \bmod p]\\
&\geq \Pr[F(ID^*)=0 \bmod p\cap F(ID^*)=0 \bmod l_v]\\
&=\Pr[F(ID^*)=0 \bmod l_v]\cdot
\end{aligned}
$$

$$
\begin{aligned}
&\Pr[F(ID^*)=0 \bmod p\mid F(ID^*)=0 \bmod l_v]\\
&=\frac{1}{l_v}\frac{1}{m+1},
\end{aligned}
$$

$$
\begin{aligned}
&\Pr[\bigcap_{i=1}^{q_1+q_S}A_i|A^*]=1-\Pr[\bigcup_{i=1}^{q_1+q_S}\neg A_i\mid A^*]\\
&\geq 1-\sum_{i=1}^{q_1+q_S}\Pr[\neg A_i\mid A^*]\\
&=1-\frac{q_1+q_S}{l_v}.
\end{aligned}
$$

Since $l_v(m+1)<p$, we have that

$$
\begin{aligned}
&\Pr[B^*]=\Pr[K(M^*)=0 \bmod p]\\
&\geq \Pr[K(M^*)=0 \bmod p\cap K(M^*)=0 \bmod l_m]\\
&=\Pr[K(M^*)=0 \bmod l_m]\cdot
\end{aligned}
$$

$$\Pr[K(M^*)=0 \bmod p \mid F(M^*)=0 \bmod l_m]$$
$$=\frac{1}{l_m}\frac{1}{l+1},$$

$$\Pr[\bigcap_{j=1}^{q_S} B_j \mid B^*]=1-\Pr[\bigcup_{j=1}^{q_S} \neg B_j \mid B^*]$$
$$\geq 1-\sum_{j=1}^{q_S}\Pr[\neg B_j \mid B^*]$$
$$=1-\frac{q_S}{l_m}.$$

As $l_v=2(q_1+q_S)$ and $l_m=2q_S$, we can obtain the resulting probability

$$\Pr[\neg abort]=\frac{1}{l_v}\frac{1}{m+1}\left(1-\frac{q_1+q_S}{l_v}\right)\frac{1}{l_m}\frac{1}{l+1}\left(1-\frac{q_S}{l_m}\right)$$
$$=\frac{1}{2(q_1+q_S)}\frac{1}{m+1}\left(1-\frac{q_1+q_S}{2(q_1+q_S)}\right)\frac{1}{2q_S}\frac{1}{l+1}\left(1-\frac{q_S}{2q_S}\right)$$
$$=\frac{1}{16(m+1)(l+1)q_S(q_1+q_S)}.$$

Therefore, if $\mathcal{A}_1$ breaks the strong unforgeability of our IBS scheme with probability ε, then C can solve the CDH problem with probability at least $\frac{\varepsilon}{16q_S(q_1+q_S)(m+1)(l+1)}$.

Lemma 2. *If there exists a type-2 adversary $\mathcal{A}_2$ breaking strong unforgeability of our RIBS scheme, then the CDH problem can be solved.*

Proof. Suppose that $\mathcal{A}_2$ makes q_E secret key queries, q_U update key queries, q_K signing key queries and q_S signing queries, and forges a valid signature for our RIBS scheme with probability ε. Then, we show that a simulator C can solve the CDH problem by using $\mathcal{A}_2$. C is given $(g, g^a, g^b) \in G^3$, and C's goal is to output g^{ab}.

Setup: C sets $l_t=2(q_2+q_S)$ and $l_m=2q_S$ satisfying $l_t(n+1)<p$ and $l_m(l+1)<p$, where $q_2=\max\{q_U, q_E\}$. C randomly selects $d, \chi \in Z_p$, $k_t(0 \leq k_t \leq n)$ and $k_m(0 \leq k_m \leq l)$. Additionally, C selects random elements $x_0', x_1', ..., x_m', t_0', t_1', ..., t_n' \in Z_p$, $y_0', y_1', ..., y_n' \in Z_{l_t}$, and sets $g_1=g^{\chi}g^a=g^{\chi+a}$, $g_2=g^b$, $u_0=g^{x_0'}$, $u_i=g^{x_i'}(1 \leq i \leq m)$, $v_0=g_2^{-l_t k_t+y_0'}g^{t_0'}$, $u_j=g_2^{y_j'}g^{t_j'}(1 \leq j \leq n)$ and $g_3=g^d$. Note that the master secret key $msk=(g_2^{\chi}, g_2^a)$, but a is unknown to C. Then, C constructs other parameters w_0, $w_k(1 \leq k \leq l)$, H_1 and H_2 in the same manner as in Lemma 1. Finally, C sends public parameters pp to the adversary $\mathcal{A}_2$.

Given an identity ID, a string T and a message M, we also define the following five functions:

$$E_2(ID)=x_0'+\sum_{i=1}^{m}x_i'ID_i,$$
$$F_2(T)=-l_t k_t+y_0'+\sum_{j=1}^{n}y_j'T_j,$$
$$J_2(T)=t_0'+\sum_{j=1}^{n}t_j'T_j,$$
$$K(M)=-l_m k_m+c_0+\sum_{k=1}^{l}c_k M_k,$$
$$L(M)=z_0+\sum_{k=1}^{l}z_k M_k.$$

Consequently, we have three equations $F_{W,1}(ID)=g^{E_2(ID)}$, $F_{W,2}(T)=g_2^{F_2(T)}g^{J_2(T)}$ and $F_{W,3}(M)=g_2^{K(M)}g^{L(M)}$.

PKG queries: C maintains an initially empty list L_{sk} in the same manner as **PKG queries** in Lemma 1. On receiving a secret key query on ID, C recalls r_s from L_{sk}. Then, C computes $sk_{ID}=(sk_1, sk_2)=(g_2^{\chi}F_{W,1}(ID)^{r_s}, g^{r_s})$, and returns sk_{ID} to $\mathcal{A}_2$.

KeyUp queries: C maintains an initially empty list L_{vk} in the same manner as **KeyUp queries** in Lemma 1. On receiving an update key query on (ID, t), C computes $T=H_1(ID,t)$, $F_2(T)$ and $J_2(T)$. If $F_2(T)=0 \bmod p$, C aborts. If $F_2(T) \neq 0 \bmod p$, C recalls r_t from L_{vk}, computes $vk_{ID,t}=(vk_1, vk_2)=((g^a)^{\frac{-J_2(T)}{F_2(T)}} \cdot F_{W,2}(T)^{r_t}, (g^a)^{\frac{-1}{F_2(T)}}g^{r_t})$, and returns $vk_{ID,t}$ to $\mathcal{A}_2$.

SKGen queries: On receiving a signing key query on (ID, t), C performs the same as **SKGen queries** in Lemma 1 to return a signing key $dk_{ID,t}$ to $\mathcal{A}_2$. Similarly, C aborts the simulation when C cannot generate an update key with $F_2(T)=0 \bmod p$.

Signing queries: On receiving a signature query on (M, ID, t), C computes $T=H_1(ID,t)$, $F_2(T)$ and $J_2(T)$, and then responds to this query as follows.

Case 1: If $F_2(T) \neq 0 \bmod p$, C performs the same as in *Case* 1 in Lemma 1 and returns a signature σ on M to $\mathcal{A}_2$.

Case 2: If $F_2(T) = 0 \bmod p$, C computes $K(M)$ and $L(M)$, and considers the following two subcases:

- *Case* 2.1: If $K(M) = 0 \bmod p$, C terminates the simulation.
- *Case* 2.2: If $K(M) \neq 0 \bmod p$, C returns a signature σ on M to $\mathcal{A}_2$ as in *Case* 2.2 in Lemma 1.

Forgery: $\mathcal{A}_2$ eventually outputs a forged signature $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*)$ with regard to (M^*, ID^*, t^*), where (ID^*, t^*) has not appeared in the update key and signing key queries, and σ^* is not generated by signing queries on (M^*, ID^*, t^*). C computes $T^* = H_1(ID^*, t^*)$ and $h^* = H_2(ID^*, M^*, t^*, \sigma_2^*, \sigma_3^*, \sigma_4^*)$. If $F_2(T^*) \neq 0 \bmod p$ or $K(M^*) \neq 0 \bmod p$, C terminates simulation. Otherwise, C uses the secret value χ to calculate the CDH value g^{ab} as below:

$$\frac{\sigma_1^*}{g_2^{\chi}(\sigma_2^*)^{E_2(ID^*)}(\sigma_3^*)^{J_2(T^*)}(\sigma_4^*)^{L(M^*)}(\sigma_4^*)^{h^* d}}$$

$$= \frac{g_2^{\chi+a} F_{W,1}(ID^*)^{r_s^*} F_{W,2}(T^*)^{r_t^*} (F_{W,3}(M^*) g_3^{h^*})^{r_m^*}}{g_2^{\chi}(g^{r_s^*})^{E_2(ID^*)}(g^{r_t^*})^{J_2(T^*)}(g^{r_m^*})^{L(M^*)}(g^{r_m^*})^{d h^*}}$$

$$= \frac{g_2^{a}(g^{E_2(ID^*)})^{r_s^*}(g_2^{F_2(T^*)} g^{J_2(T^*)})^{r_t^*}(g_2^{K(M^*)} g^{L(M^*)} g_3^{h^*})^{r_m^*}}{(g^{E_2(ID^*)})^{r_s^*}(g^{J_2(T^*)})^{r_t^*}(g^{L(M^*)})^{r_m^*}(g_3^{h^*})^{r_m^*}}$$

$$= g_2^{a} \text{ (since } F_2(T^*) = K(M^*) = 0 \bmod p)$$

$$= g^{ab}.$$

The probability analysis is similar to Lemma 1. The probability that C completes the simulation not aborting is at least $\frac{1}{16 q_S (q_2 + q_S)(n+1)(l+1)}$, and C can solve the CDH problem with probability at least $\frac{\varepsilon}{16 q_S (q_2 + q_S)(n+1)(l+1)}$.

Combining Lemma 1 and Lemma 2, we can obtain the following theorem.

Theorem 1. *In the standard model, our RIBS scheme is strongly unforgeable against adaptive chosen-message attacks under the CDH assumption.*

5.3. Comparison

We give a comparison between our scheme and the previous RIBS schemes in the standard model. In Table 1, $|G|$ denotes the bit-length of an element in group G, and T_P and T_E denote a pairing operation and an exponentiation operation, respectively. We do not take into account relatively efficient operations, such as hash operations, multiplication, etc.

As shown in Table 1, among the three strong unforgeable IBS schemes [10,16,23], the signature length and signature verification cost of Sato *et al.*'s IBS scheme are the largest, while Tsai *et al.*'s IBS scheme has the largest signature cost. However, none of three schemes [10,16,23] support key revocation mechanism.

Three RIBS schemes have the same signature size, but our RIBS scheme is more efficient than Liu *et al.*'s scheme [12] and Hung *et al.*'s scheme [9] in terms of computational costs of the signing and verifying phases. It is noteworthy that Hung *et al.*'s scheme is vulnerable to signing key exposure and that our scheme and Liu *et al.*'s scheme can resist signing key exposure attack. However, there are some flaws in Liu *et al.*'s security proof as presented in [11], and the size of each user's secret key increases logarithmically with the number of total users involved in Liu *et al.*'s

Table 1
Comparisons of computational costs and security with previous RIBS schemes

RIBS scheme	Signature size	Sign	Verify	Signing key exposure resistance	Security
Sato *et al.*'s scheme [16]	$5\|G\|$	$3T_E$	$6T_P$	No	Yes
Kwon's scheme [10]	$3\|G\|$	$3T_E$	$T_E + 4T_P$	No	Yes
Tsai *et al.*'s scheme [23]	$3\|G\|$	$4T_E$	$T_E + 4T_P$	No	Yes
Liu *et al.*'s scheme [12]	$4\|G\|$	$3T_E$	$2T_E + 4T_P$	Yes	No
Hung *et al.*'s scheme [9]	$4\|G\|$	$5T_E$	$T_E + 4T_P$	No	No
Our scheme	$4\|G\|$	$3T_E$	$T_E + 4T_P$	Yes	Yes

scheme. In addition, Hung *et al.*'s scheme does not satisfy strong unforgeability, but our scheme is strongly unforgeable in the standard model.

6. Conclusions

In this paper, we revisit Hung *et al.*'s RIBS scheme and its security proof [9]. Unfortunately, we find that their scheme does not possess strong unforgeability and that its security proof has serious flaws. Moreover, their scheme is insecure against signing key exposure attack. To resolve these problems, we propose an improved RIBS scheme in the standard model. The analysis results show that our RIBS scheme satisfies strong unforgeability and signing key exposure resistance.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61662069, China Postdoctoral Science Foundation under Grant No. 2017M610817, Science and Technology Project of Lanzhou City of China under Grant No. 2013-4-22, Foundation for Excellent Young Teachers by Northwest Normal University under Grant No. NWNU-LKQN-14-7.

References

1. Boldyreva, A., Goyal, V., Kumar, V. Identity-Based Encryption with Efficient Revocation. Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS 2008), Alexandria, Virginia, USA, October 27-31, 2008, 417-426. https://doi.org/10.1145/1455770.145 5823
2. Boneh, D., Boyen, X. Efficient Selective-ID Secure Identity -Based Encryption without Random Oracles. Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'04), Interlaken, Switzerland, May 2-6, 2004, 223-238. https://doi.org/10. 1007/978-3-540-24676-3_14
3. Boneh, D., Franklin M. Identity-Based Encryption from the Weil Pairing. SIAM Journal on Computing, 2003, 32(3), 586-615. https://doi.org/10.1137/S00975 39701398521
4. Boneh, D., Shen, E., Waters, B. Strongly Unforgeable Signatures Based on Computational Diffie-Hellman. Proceedings of International Workshop on Public Key Cryptography (PKC 2006), New York, USA, April 24-26, 2006, 229-240. https://doi.org/10.1007/ 11745853_15
5. Canetti, R., Goldreich, O., Halevi, S. The Random Oracle Methodology, Revisited. Journal of the ACM, 2004, 51(4), 557-594. https://doi.org/10.1145/1008731.1008734
6. Hu, X. M., Wang, J., Xu, H. J., Yang, Y., Xu, X. An Improved Efficient Identity-Based Proxy Signature in the Standard Model. International Journal of Computer Mathematics, 2017, 94(1), 22-38. https://doi.org/10.1080/ 00207160.2015.1086759
7. Hu, X., Zhang, X., Wang, J., Xu, H., Tan, W., Yang, Y. Secure and Efficient Identity-Based Proxy Signature Scheme in the Standard Model Based on Computational Diffie–Hellman Problem. Arabian Journal for Science and Engineering, 2017, 42(2), 639-649. https://doi.org/10.1007/s13369-016-2280-6
8. Huang, Q., Wong, D. S., Li, J., Zhao, Y. M. Generic Transformation from Weakly to Strongly Unforgeable Signatures. Journal of Computer Science and Technology, 2008, 23(2), 240-252. https://doi.org/10.1007/s11390-008-9126-y
9. Hung, Y. H., Tsai, T. T., Tseng, Y. M., Huang, S. S. Strongly Secure Revocable ID-Based Signature without Random Oracles. Information Technology and Control, 2014, 43(3), 264-276. http://dx.doi.org/ 10.5755/j01.itc.43.3.5718
10. Kwon, S. An Identity-Based Strongly Unforgeable Signature Without Random Oracles from Bilinear Pairings. Information Sciences, 2014, 276, 1-9. https://doi.org/10.1016/j.ins.2014.02.041
11. Lee, K., Lee, D. H. Security Analysis of an Identity-Based Strongly Unforgeable Signature Scheme. Information Sciences, 2014, 286, 29-34. https://doi. org/10.1016/j.ins. 2014.07.022
12. Liu, Z., Zhang, X., Hu, Y., Takagi, T. Revocable and Strongly Unforgeable Identity-Based Signature Scheme in the Standard Model. Security and Communication Networks, 2016, 9(14), 2422-2433. https://doi.org/10.1002/ sec.1513
13. Narayan, S., Parampalli, U. Efficient Identity-Based Signatures in the Standard Model. IET Information Security, 2008, 2(4), 108-118. https://doi.org/10.1049/iet-ifs:20070135

14. Paterson, K. G. ID-Based Signatures from Pairings on Elliptic Curves. Electronics Letters, 2002, 38(18), 1025-1026. https://doi.org/10.1049/el:20020682

15. Paterson, K. G., Schuldt, J. C. N. Efficient Identity-Based Signatures Secure in the Standard Model. Proceedings of Australasian Conference on Information Security and Privacy (ACISP 2006), Melbourne, Australia, July 3-5, 2006, 207-222. https://doi.org/10.1007/11780656_18

16. Sato, C., Okamoto, T., Okamoto, E. Strongly Unforgeable ID-Based Signatures Without Random Oracles. International Journal of Applied Cryptography, 2010, 2(1), 35-45. https://doi.org/10. 1504/IJACT.2010.033797

17. Seo, J. H., Emura, K. Revocable Identity-Based Encryption Revisited: Security Model and Construction. Proceedings of Public-Key Cryptography (PKC 2013), Nara, Japan, February 26-March 1, 2013, 216-234. https://doi.org/10.1007/978-3-642-36362-7_14

18. Seo, J. H., Emura, K. Revocable Identity-Based Cryptosystem Revisited: Security Models and Constructions. IEEE Transactions on Information Forensics and Security, 2014, 9(7), 1193-1205. https://doi. org/10.1109/TIFS.2014. 2327758

19. Shamir, A. Identity-Based Cryptosystems and Signature Schemes. Proceedings of Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT 1984), Paris, France, April 9-11, 1984, 47-53. https://doi.org/10.1007/3-540-39568-7_5

20. Shim, K. A. An ID-Based Aggregate Signature Scheme with Constant Pairing Computations. Journal of Systems and Software, 2010, 83(10), 1873-1880. https://doi.org/10. 1016/j.jss.2010.05.071

21. Steinfeld, R., Pieprzyk, J., Wang, H. How to Strengthen any Weakly Unforgeable Signature into a Strongly Unforgeable Signature. Proceedings of Cryptographers' Track at the RSA Conference (CT-RSA 2007), San Francisco, USA, February 5-9, 2007, 357-371. https://doi.org/10.1007/11967 668_23

22. Sun, Y., Zhang, F., Shen, L., Deng, R. Revocable Identity-Based Signature Without Pairing. Proceedings of Intelligent Networking and Collaborative Systems (INCOS 2013), Thessaloniki, Greece, September 9-11, 2013, 363-365. https://doi.org/10.1109/INCoS.2013.68

23. Tsai, T. T., Tseng, Y. M., Huang, S. S. Efficient Strongly Unforgeable ID-Based Signature without Random Oracles. Informatica, 2014, 25(3), 505-521. https://doi.org/10.15388/ Informatica.2014.26

24. Tsai, T. T., Tseng, Y. M., Wu, T. Y. Provably Secure Revocable ID-Based Signature in the Standard Model. Security and Communication Networks, 2013, 6(10), 1250-1260. https://doi.org/10.1002/sec.696

25. Wu, L., Zhang, Y., Ren, Y., He, D. Efficient Certificate-Based Signature Scheme for Electronic Commerce Security Using Bilinear Pairing. Internet Technology Journal, 2017, 18(5), 1159-1166. https://doi.org/10.6138/JIT.2017.18.5.20 160602

26. Yi, X. An Identity-Based Signature Scheme from the Weil Pairing. IEEE Communications Letters, 2003, 7(2), 76-78. https://doi.org/10.1109/LCOMM.2002.808397

27. Zhang, J., Mao, J. A Novel ID-Based Designated Verifier Signature Scheme. Information Sciences, 2008, 178(3), 766-773. https://doi.org/10.1016/j.ins. 2007.07.005