Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 854135, 11 pages
http://dx.doi.org/10.1155/2014/854135

Research Article

Strongly Secure Certificateless Signature Scheme Supporting Batch Verification

Chun-I Fan,[1] Pei-Hsiu Ho,[2] and Yi-Feng Tseng[1]

[1] *Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan*
[2] *Network Benchmarking Lab, Hsinchu 30010, Taiwan*

Correspondence should be addressed to Chun-I Fan; cifan@faculty.nsysu.edu.tw

Received 14 November 2013; Revised 7 March 2014; Accepted 13 March 2014; Published 16 April 2014

Academic Editor: Jian Guo Zhou

We propose a strongly secure certificateless signature scheme supporting batch verification, which makes it possible for a verifier to verify a set of signatures more efficiently than verifying them one by one. In an identity-based digital signature scheme, private key generator (PKG) knows each user's signing key, so it can generate a signature which is indistinguishable from the signature generated by the user. This is a serious problem because the property of signature nonrepudiation will not be achieved. In our proposed scheme, it is impossible for PKG to produce a signature which is indistinguishable from any signature produced by a user. Compared with existing signature schemes with batch verification, although our proposed scheme is not the most efficient one, it achieves Girault's level-3 security, while the others have Girault's level-1 or level-2 security only. We also formally prove that the proposed scheme is unforgeable and satisfies Girault's level-3 security based on hard problems.

1. Introduction

In traditional certificate-based public-key cryptosystems, a user's public key is produced and is not related to her/his identity. Therefore, the key needs to be certificated by some certification authority (CA) with respect to the user's identity. Anyone who wants to use the public key must verify the validity of the corresponding certificate for the key first. Considering implementation, the management of public key certificates requires a large amount of computation cost and storage.

To reduce the cost of certificate management, Shamir [1] proposed identity based public key cryptography (ID-PKC) in 1984. In ID-PKC, a user's public key can be an arbitrary bit string which can represent the user's identity, such as her/his email address or telephone number. And the user's corresponding private key is computed by a trusted party, called private key generator (PKG) [2].

An inherent problem of ID-PKC is the key escrow problem. That is, the private key of a user is known to PKG. PKG can act as any user to decrypt any ciphertext or generate a signature on any message. To solve this problem, Al-Riyami and Paterson [3] proposed certificateless public key cryptography (CL-PKC) in 2003. In CL-PKC, a user's secret key is a combination of the secret key, computed by PKG using its master secret key, and a user-chosen secret. Thus, PKG cannot know the complete secret key of the user.

In 2003, a certificateless public-key signature scheme [3] was proposed, but it suffered from the key replacement attack [4]. After that, several certificateless signature schemes [5–18] were introduced recently. In 2007, Hu et al. [19] proposed a new security model and an improved generic construction for certificateless signatures. It showed that certificateless signatures should satisfy the property of Girault's level-3 security [20]. If a certificateless signature scheme meets the above property, the framework of CL-PKC will be with the same security level as that of the traditional certificate-based public key cryptosystems.

In 1989, batch cryptography was first introduced by Fiat [21]. In a signature scheme with batch verification, the cost for verifying n signatures is less than verifying them one by one. After [21], some results [22–24] about batch verification based on RSA or DLP have been proposed. In 2004, Yoon et al. proposed an ID-based signature scheme with batch verification [25], but their security proof does not meet the definition of batch verification [26]. After that, some ID-based and group signature schemes with batch verification were proposed in [27–34], respectively, but only [29] is a certificateless signature scheme. Besides, the randomization technique [26, 35–39] for the security of signatures with batch verification was introduced. The technique can withstand the attack that an attacker cheats a verifier to accept invalid signatures.

In this paper, we will design a certificateless signature scheme with efficient batch verification. The computation cost of our scheme in batch verification is only three pairings and it is independent of the number of individual signatures which will be verified. Furthermore, our scheme satisfies Girault's level-3 security. Compared to certificateless, ID-based, and group signature schemes with batch verification [7, 11, 14, 16, 27–34], our scheme achieves Girault's level-3 security, while [27–34] meet Girault's level-1 security and [7, 11, 14, 16, 29] satisfy Girault's level-2 security only. Compared to [5, 13], which also achieve Girault's level-3 security, our scheme is more efficient in verification because of the batch property.

The rest of this paper is organized as follows. In Section 2, we introduce some preliminaries about mathematical backgrounds and definitions. In Section 3, we present the proposed scheme. In Section 4, we provide formal security proofs for our scheme. We compare the proposed scheme with [5, 7, 11, 13, 14, 16, 27–34] in Section 5. Finally, a concluding remark is given in Section 6.

2. Preliminaries

In this section, we review the properties of bilinear groups and some related hard problems.

Let G_1 and G_2 be two cyclic groups of prime order q. Let P be a randomly chosen generator of G_1 and let e be a bilinear mapping such that $e: G_1 \times G_1 \rightarrow G_2$, which satisfies the following properties.

(1) Bilinearity: for all $P, Q, R \in G_1$, $e(P+Q, R) = e(P, R)e(Q, R)$ and $e(P, Q+R) = e(P, Q)e(P, R)$.

(2) Nondegeneracy: there exists $P, Q \in G_1$ such that $e(P, Q) \neq 1$.

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

Definition 1 (batch verification of signatures [35]). Let l be the security parameter. Suppose that $(Gen, Sign, Verify)$ is a signature scheme, $n \in$ polynomial(l), and $(pk_1, sk_1), \ldots, (pk_n, sk_n)$ are generated independently according to $Gen(1^l)$ where pk_i and sk_i are a user i's public key and secret key, respectively. Then, we call probabilistic *Batch* a batch verification algorithm when the following conditions hold.

(i) If $Verify(\sigma_i, m_i, pk_i) = 1$ for all $i \in [1, n]$, then $Batch((\sigma_1, m_1, pk_1), \ldots, (\sigma_n, m_n, pk_n)) = 1$ where m_i and σ_i are a message and a signature, respectively.

(ii) If $Verify(\sigma_i, m_i, pk_i) = 0$ for some $i \in [1, n]$, then $Batch((\sigma_1, m_1, pk_1), \ldots, (\sigma_n, m_n, pk_n)) = 1$ with probability negligible in k, taken over the randomness of *Batch*.

Definition 2 (a certificateless signature scheme with batch verification). A certificateless signature scheme with batch verification consists of the following algorithms.

(i) *Setup*: PKG randomly chooses a secret key and computes the public key T_{pub} by using the secret key. Then, it publishes T_{pub} and other system parameters.

(ii) *KeyGen*: a user ID_i first randomly chooses a secret key x_i, computes corresponding public key P_i and then sends P_i to PKG. After receiving P_i, PKG outputs a partial private key D_i to a legal user with identity ID_i.

(iii) *Signing*(P_i, D_i, x_i, m): this algorithm gets a user's public key P_i, the user's partial private key D_i, the user's secret key x_i, and a message m and then it outputs a signature σ on m.

(iv) $Verifying(\sigma, m, ID_i, P_i)$: this algorithm gets a signature σ on a message m, a signer's identity ID_i, and a signer's public key P_i. It then outputs True or False.

(v) $Batch\_Verify((\sigma_1, m_1, ID_1, P_1), \ldots, (\sigma_n, m_n, ID_n, P_n))$: this algorithm gets n signatures $\sigma_1, \ldots, \sigma_n$ on message $m_1, \ldots, m_n$, the signers' identities $ID_1, \ldots, ID_n$, and the signers' public keys $P_1, \ldots, P_n$, respectively. This algorithm outputs True or False.

Definition 3 (Girault's security [19, 20]). Girault proposed three security levels to classify the levels of trust to PKG. The three levels are described as follows.

(i) Level 1: PKG knows the secret of any user.

(ii) Level 2: PKG cannot find out all the information of a user's secret. However, PKG can generate a contradictory public key (or a contradictory certificate) and impersonate the user to generate signatures with respect to the contradictory public key.

(iii) Level 3: PKG cannot find out all the information of a user's secret nor generate a contradictory public key. PKG can only generate a valid public key (or a valid certificate).

Definition 4 (the computational Diffie-Hellman (CDH) problem). Let G_1 be a cyclic group of order q and let P be a generator of G_1. Given $\langle G_1, q, P, aP, bP \rangle$ for some $a, b \in Z_q^*$, compute abP.

3. Our Proposed Scheme

In this section, we propose an efficient certificateless signature scheme with batch verification based on [15]. G_1 is an additive group and G_2 is a multiplicative group. Let G_1 and G_2 be two cyclic groups of prime order q. Let P be a randomly chosen generator of G_1 and e a bilinear mapping such that

$e : G_1 \times G_1 \rightarrow G_2$. The details of our scheme are described as follows.

Setup. PKG performs the following operations.

(1) Choose an integer $\lambda \in Z_q^*$ and $\theta \in \{0,1\}^*$ randomly, and set $T_{\text{pub}} = \lambda P$.

(2) Choose four cryptographic one-way hash functions, $H_0 : \{0,1\}^* \rightarrow G_1$, $H_1 : G_1 \rightarrow G_1$, $H_2 : \{0,1\}^* \times G_1 \times G_1 \rightarrow Z_q^*$, $H_3 : \{0,1\}^* \times G_1 \times G_1 \rightarrow Z_q^*$, and $H_4 : \{0,1\}^* \rightarrow G_1$.

(3) Publish the system parameters $\{G_1, G_2, e, q, P, \theta, T_{\text{pub}}, H_0, H_1, H_2, H_3, H_4\}$ and keep the master key λ secret.

Key Generating Phase

(1) A user with identity ID_i randomly chooses $x_i \in Z_q^*$ and computes $P_i = x_iP$, where x_i and P_i are called the secret key and the public key, respectively, of user ID_i.

(2) The user sends (ID_i, P_i) to PKG.

(3) PKG gets $Q_i = H_0(ID_i)$ and $\Gamma_i = H_1(P_i)$.

(4) PKG computes $D_{i_0} = \lambda Q_i$ and $D_{i_1} = \lambda\Gamma_i$ and sends (D_{i_0}, D_{i_1}) to the user via a secret channel. The pair (D_{i_0}, D_{i_1}) is called the partial private key of user ID_i. The private key of user ID_i consists of x_i and (D_{i_0}, D_{i_1}).

Signing Phase. Assume that a signer ID_i wants to sign a message $m \in \{0,1\}^*$. The signer does the following works.

(1) Choose r and $\alpha \in Z_q^*$ randomly.

(2) Compute $U_1 = r(Q_i + \Gamma_i)$ and $U_2 = \alpha x_i P$.

(3) Compute $h_2 = H_2(m, U_1, U_2)$, $h_3 = H_3(m, U_2, U_1)$, and $W = H_4(\theta)$.

(4) Compute $V = (r + h_2)(D_{i_0} + D_{i_1}) + (\alpha + h_3)x_iW$.

(5) The signature on m is $\sigma = (V, U_1, U_2, P_i)$.

Verifying Phase. To verify a signature (V, U_1, U_2, P_i) on message m, a verifier should do the following works.

(1) Compute $Q_i = H_0(ID_i)$, $\Gamma_i = H_1(P_i)$, $h_2 = H_2(m, U_1, U_2)$, $h_3 = H_3(m, U_2, U_1)$, and $W = H_4(\theta)$.

(2) Verify if $e(P, V) = e(T_{\text{pub}}, U_1 + h_2(Q_i + \Gamma_i))e(W, U_2 + h_3P_i)$.

If it is true, the verifier accepts the signature; otherwise, the verifier rejects it.

Batch Verifying Phase. To verify n signatures $\sigma_1 = (V_1, U_{1_1}, U_{2_1}, P_1), \ldots, \sigma_n = (V_n, U_{1_n}, U_{2_n}, P_n)$ of the n signers $ID_1, \ldots, ID_n$ on message $m_1, \ldots, m_n$, respectively, a verifier performs the following works.

(1) Choose $w_1, \ldots, w_n \in Z_q^*$ randomly.

(2) Compute $Q_i = H_0(ID_i)$, $\Gamma_i = H_1(P_i)$, $h_{2_i} = H_2(m_i, U_{1_i}, U_{2_i})$, $h_{3_i} = H_3(m_i, U_{2_i}, U_{1_i})$ for $i = 1, \ldots, n$, and $W = H_4(\theta)$.

(3) Verify if $e(P, \sum_{i=1}^n w_iV_i) = e(T_{\text{pub}}, \sum_{i=1}^n w_iU_{1_i} + w_ih_{2_i}(Q_i + \Gamma_i))e(W, \sum_{i=1}^n w_iU_{2_i} + w_ih_{3_i}P_i)$.

If it is true, the verifier accepts the n signatures.

Correctness. Consider

$$
\begin{aligned}
& e\left(P, \sum_{i=1}^{n} w_i V_i\right) \\
&= e\left(P, \sum_{i=1}^{n} w_i \left(\left(r_i + h_{2_i}\right)\left(D_{i_0} + D_{i_1}\right) + \left(\alpha_i + h_{3_i}\right) x_i W\right)\right) \\
&= e\left(P, \sum_{i=1}^{n} w_i \left(r_i \left(D_{i_0} + D_{i_1}\right) + h_{2_i}\left(D_{i_0} + D_{i_1}\right)\right)\right) \\
&\quad \times e\left(P, \sum_{i=1}^{n} w_i \left(\alpha_i x_i W + h_{3_i} x_i W\right)\right) \\
&= e\left(\lambda P, \sum_{i=1}^{n} w_i \left(r_i \left(Q_i + \Gamma_i\right) + h_{2_i}\left(Q_i + \Gamma_i\right)\right)\right) \\
&\quad \times e\left(W, \sum_{i=1}^{n} w_i \left(\alpha_i x_i P + h_{3_i} x_i P\right)\right) \\
&= e\left(T_{\text{pub}}, \sum_{i=1}^{n} w_i \left(U_{1_i} + h_{2_i}\left(Q_i + \Gamma_i\right)\right)\right) \\
&\quad \times e\left(W, \sum_{i=1}^{n} w_i \left(U_{2_i} + h_{3_i} P_i\right)\right).
\end{aligned}
\tag{1}
$$

4. Security Models and Formal Proofs

4.1. Security Models. A simulator B simulates an environment such that an adversary E can query signatures from B. If E can forge a signature, B can use the output from E to solve a hard problem.

We classify adversary E into three types. The adversary of type I cannot access the master secret key and query the partial private key of target ID. The adversary of type II can access the master secret key but cannot query target ID's secret key nor replace her/his public key. The adversary of type III is to simulate the environment for the proof of that a user cannot produce a signature with a new public-secret key pair, which is different from his own one, without the corresponding partial private key generated from the master secret key.

We define the capability of E which can be captured by the following queries.

(i) $H_0(ID_i)$: if E inputs a user's identity ID_i to H_0, B will output a randomly chosen $Q_i \in G_1$ as the user's public key.

(ii) $H_1(P_i)$: when E inputs a public key $P_i \in G_1$ to H_1, B outputs a randomly chosen $\Gamma_i \in G_1$.

(iii) $H_2(m, U_1, U_2)$: if E inputs a message $m \in \{0,1\}^*$ and $U_1, U_2 \in G_1$, B will output an integer randomly chosen in Z_q^*.

(iv) $H_3(m, U_2, U_1)$: if E inputs a message $m \in \{0,1\}^*, U_2 \in G_1$, and $U_1 \in G_1$, B will output a random integer in Z_q^*.

(v) $H_4(\varrho)$: if E inputs a string $\varrho \in \{0,1\}^*$, B will output an element randomly chosen in G_1.

(vi) *Public_Key*(ID_i): if E inputs a user's identity, ID_i, then B will output the user's public key P_i.

(vii) *Partial_Private_Key*(ID_i, P_i): if E inputs a user's identity ID_i and the user's public key P_i, B will output the partial private key (D_{i_0}, D_{i_1}).

(viii) *Secret_Key*(ID_i): if E inputs a user's identity, ID_i, B will output the secret key x_i of user ID_i to E.

(ix) *Public_Key_Replacement*(ID_i, P_i'): when E inputs a user's identity ID_i and the user's new public key P_i', B will replace P_i with P_i'. The new partial private key can be obtained by querying *Partial_Private_Key*(ID_i).

(x) *Sign*(ID_i, m): if E inputs a user's identity ID_i and a message m, B will output a user ID_i's signature σ on m to E.

Definition 5 (the CDH assumption). We say that the (t, ε)-CDH assumption holds in G_1 if no polynomial-time algorithm within running time t can solve the CDH problem with probability at least ε.

Definition 6 (the unforgeability game I). Let E_1 be a polynomial-time attacker of type I. E_1 interacts with a challenger B in the following game.

(i) Step 1: B runs the setup algorithm of a certificateless signature scheme with batch verification. B publishes the public parameters.

(ii) Step 2: E_1 queries *Partial_Private_Key*, *Public_Key*, *Public_Key_Replacement*, *Secret_Key*, *Sign*, H_0, H_1, H_2, H_3, and H_4 in an arbitrary sequence.

(iii) Step 3: E_1 outputs n signatures $\sigma_1, \ldots, \sigma_n$ on $m_1, \ldots, m_n$ corresponding to the signers $ID_1, \ldots, ID_n$ with the public keys $P_1, \ldots, P_n$, respectively.

E_1 wins the game if

(1) *Batch_Verify*$((\sigma_1, m_1, ID_1, P_1), \ldots, (\sigma_n, m_n, ID_n, P_n))$ = True;

(2) there exists $\sigma_y \in \{\sigma_1, \ldots, \sigma_n\}$ whose (ID_y, m_y) has not been queried to *Sign* oracle;

(3) *Partial_Private_Key*(ID_y, P_y) has never been queried.

The scheme is $(t, \varepsilon,$ I)-unforgeable if no polynomial-time attacker E_1, with running time at most t, can win the unforgeability game I with probability at least ε.

Definition 7 (the unforgeability game II). Let E_2 be a polynomial-time attacker of type II. E_2 interacts with a challenger B in the following game.

(i) Step 1: B runs the setup algorithm of a certificateless signature scheme with batch verification. B publishes the public parameters and sends the master secret key to E_2.

(ii) Step 2: E_2 queries *Partial_Private_Key*, *Public_Key*, *Public_Key_Replacement*, *Secret_Key*, *Sign*, H_0, H_1, H_2, H_3, and H_4 in an arbitrary sequence.

(iii) Step 3: E_2 outputs n signatures $\sigma_1, \ldots, \sigma_n$ on $m_1, \ldots, m_n$ corresponding to the signers $ID_1, \ldots, ID_n$ with the public keys $P_1, \ldots, P_n$, respectively.

E_2 wins the game if

(1) *Batch_Verify*$((\sigma_1, m_1, ID_1, P_1), \ldots, (\sigma_n, m_n, ID_n, P_n))$ = True;

(2) there exists $\sigma_y \in \{\sigma_1, \ldots, \sigma_n\}$ whose (ID_y, m_y) has not been queried to *Sign* oracle;

(3) Neither *Secret_Key*(ID_y) nor *Public_Key_Replacement*$(ID_y, \cdot, \cdot)$ has been queried.

The scheme is $(t, \varepsilon, \text{II})$-unforgeable if no polynomial-time attacker E_2, with running time at most t, can win the unforgeability game II with probability at least ε.

Definition 8 (the unforgeability game III [19]). Let E_3 be a polynomial-time attacker of type III. E_3 interacts with a challenger B in the following game.

(I) Step 1: B runs the setup algorithm of a certificateless signature scheme with batch verification. B publishes the public parameters.

(II) Step 2: E_3 queries *Partial_Private_Key*, *Public_Key*, *Public_Key_Replacement*, *Secret_Key*, H_0, H_1, H_2, H_3, and H_4 in an arbitrary sequence.

(III) Step 3: E_3 outputs a signature $\sigma^* = (V^*, U_1^*, U_2^*, P_y^*)$ on a message m^* for the user ID_y.

E_3 wins the game if

(1) *Verifying*$(\sigma^*, m^*, ID_y, P_y^*)$ = True;

(2) User ID_y has been created, that is, *Public_Key*(ID_y), *Partial_Private_Key* (ID_y, P_y), and *Secret_Key*(ID_y) have been queried;

(3) P_y^* is different from all of the public keys P_y's returned by *Public_Key*(ID_y) or used to query *Public_Key_Replacement*.

The scheme is $(t, \varepsilon, \text{III})$-unforgeable if no polynomial-time attacker E_3, with running time at most t, can win the unforgeability game III with probability at least ε.

4.2. Formal Proofs. In this section, we will prove that our scheme is unforgeable based on the CDH assumption.

Lemma 9 (the forking lemma [40]). *Let (s, m, c) be a valid signature-message triple of a signature scheme and h the hashed value of (m, c) where m is a plaintext message, c is a string, and s is called the signature part of the triple. Let A be a probabilistic polynomial-time Turning machine. Given only the public data of the signature scheme as input, if A can find, with nonnegligible probability, a valid signature-message triple (s, m, c) with h, then, with nonnegligible probability, a replay of this machine, with the same random tape and a different value returned by the random oracle, outputs two valid signature-message triples (s, m, c) with h and (s', m, c) with h' such that $h \neq h'$.*

Lemma 10 (the splitting lemma [40]). *Let $A \subset X \times Y$ such that $\Pr[(x, y) \in A] \geq \varepsilon$. For any $\alpha < \varepsilon$, define $B = \{(x, y) \in X \times Y \mid \Pr_{y' \in Y}[(x, y') \in A] \geq \varepsilon - \alpha\}$ and then the following statements hold:*

(1) $\Pr[B] \geq \alpha$,

(2) $\forall (x, y) \in B$, $\Pr_{y' \in Y}[(x, y') \in A] \geq \varepsilon - \alpha$,

(3) $\Pr[B \mid A] \geq \alpha/\varepsilon$.

Theorem 11. *Given n 5-tuples $(V_i, U_{1_i}, U_{2_i}, P_i, m_i)$'s, if $e(P, \sum_{i=1}^{n} w_i V_i) = e(T_{pub}, \sum_{i=1}^{n} w_i U_{1_i} + w_i h_{2_i}(Q_i + \Gamma_i)) e(W, \sum_{i=1}^{n} w_i U_{2_i} + w_i h_{3_i} P_i)$ where w_i is randomly chosen in Z_q^*, $h_{2_i} = H_2(m_i, U_{1_i}, U_{2_i})$, and $h_{3_i} = H_3(m_i, U_{2_i}, U_{1_i})$ for each i, the probability of that $e(P, V_i) \neq e(T_{pub}, U_{1_i} + h_{2_i}(Q_i + \Gamma_i)) e(W, U_{2_i} + h_{3_i} P_i)$ for some $i \in \{1, \dots, n\}$ is $1/2^{|q|}$.*

Proof. The proof is based on [35, 37]. If $e(P, V_i) \neq e(T_{\text{pub}}, U_{1_i} + h_{2_i}(Q_i + \Gamma_i)) e(W, U_{2_i} + h_{3_i} P_i)$ for some i, we have that $V_i \neq (r_i + h_{2_i})(\lambda Q_i + \lambda \Gamma_i) + (\alpha_i + h_{3_i}) x_i W$ for some i. Thus, there exists $c_i \neq 0 \pmod{q}$ such that $V_i = (r_i + h_{2_i})(\lambda Q_i + \lambda \Gamma_i) + (\alpha_i + h_{3_i}) x_i W + c_i P$ for some i.

Let $V_j = (r_j + h_{2_j})(\lambda Q_j + \lambda \Gamma_j) + (\alpha_j + h_{3_j}) x_j W + c_j P$ and $c_j \in \{0, 1, \dots, q-1\}, \forall j \in \{1, \dots, n\} - \{i\}$. As $e(P, \sum_{i=1}^{n} w_i V_i) = e(T_{\text{pub}}, \sum_{i=1}^{n} w_i U_{1_i} + w_i h_{2_i}(Q_i + \Gamma_i)) e(W, \sum_{i=1}^{n} w_i U_{2_i} + w_i h_{3_i} P_i)$, $w_1 c_1 + w_2 c_2 + w_3 c_3 + \cdots + w_n c_n \equiv 0 \pmod{q}$ and thus $w_i = c_i^{-1}(w_1 c_1 + w_2 c_2 + \cdots + w_{i-1} c_{i-1} + w_{i+1} c_{i+1} + \cdots + w_n c_n) \bmod q$. Since w_i is randomly chosen in Z_q^*, the probability of $w_i = c_i^{-1}(w_1 c_1 + w_2 c_2 + \cdots + w_{i-1} c_{i-1} + w_{i+1} c_{i+1} + \cdots + w_n c_n) \bmod q$ is $1/2^{|q|}$, which is negligible. □

Theorem 12. *The proposed scheme is $(t, q_{sign}, q_{ppk}, q_{sk}, q_{pkr}, q_{pk}, q_{h_0}, q_{h_1}, q_{h_2}, q_{h_3}, q_{h_4}, \varepsilon, I)$-unforgeable assuming that the (t', ε')-CDH assumption holds in G_1 where $\varepsilon' \geq (\varepsilon/(2q_{h_0}))(1 - 1/2^{|q|})^2$, $t' \approx t + q_{sign}\mathcal{O}(t_{sign}) + q_{ppk}\mathcal{O}(t_{ppk}) + q_{sk}\mathcal{O}(t_{sk}) + q_{pkr}\mathcal{O}(t_{pkr}) + q_{pk}\mathcal{O}(t_{pk}) + q_{h_0}\mathcal{O}(t_{h_0}) + q_{h_1}\mathcal{O}(t_{h_1}) + q_{h_2}\mathcal{O}(t_{h_2}) + q_{h_3}\mathcal{O}(t_{h_3}) + q_{h_4}\mathcal{O}(t_{h_4}) + \mathcal{O}(1)$, q_{sign}, q_{ppk}, q_{sk}, q_{pkr}, q_{pk}, q_{h_0}, q_{h_1}, q_{h_2}, q_{h_3}, and q_{h_4} being the numbers of queries to Sign, Partial_Private_Key, Secret_Key, Public_Key_Replacement, Public_Key, H_0, H_1, H_2, H_3, and H_4, respectively, and t_{sign}, t_{ppk}, t_{sk}, t_{pkr}, t_{pk}, t_{h_0}, t_{h_1}, t_{h_2}, t_{h_3}, and t_{h_4} being the computing time of the queries to Sign, Partial_Private_Key, Secret_Key, Public_Key_Replacement, Public_Key, H_0, H_1, H_2, H_3, and H_4, respectively.*

Proof. Assume that a polynomial-time attacker E_1 wins the game of Definition 6 with probability being at least ε within running time t. A simulator B is given an instance of the CDH problem $\langle G_1, q, P, aP, bP \rangle$, and B's goal is to output the value of abP. We will construct B which plays the game in Definition 6 with E_1 and outputs the value of abP.

Setup. B sets $T_{\text{pub}} = bP$ and chooses $\theta \in \{0, 1\}^*$ randomly. Then, B publishes $\{G_1, G_2, e, q, P, \theta, T_{\text{pub}}, H_0, H_1, H_2, H_3, H_4\}$.

B can respond to the queries from E_1 as follows.

(i) H_0 query: B constructs a list, H_0-list, and chooses an identity ID_π randomly. When E_1 queries $H_0(ID_i)$ to B, B checks whether ID_i is in H_0-list or not. If ID_i does not exist in H_0-list, then there are the following two cases. Case 1: if $ID_i = ID_\pi$, B sets $Q_i = aP = H_0(ID_i)$ and stores (ID_i, aP) in H_0-list. Case 2: if $ID_i \neq ID_\pi$, B sets $Q_i = k_i P = H_0(ID_i)$ where k_i is randomly chosen in Z_q^* and stores $(ID_i, k_i P, k_i)$ in H_0-list. However, if ID_i exists in H_0-list, B gets its mapping value, $Q_i = k_i P$ or aP. Finally, B returns Q_i.

(ii) H_1 query: B constructs a list, H_1-list. If E_1 queries $H_1(P_i)$ to B where $P_i \in G_1$, B checks whether P_i is in H_1-list or not. If it does not exist in H_1-list, B randomly chooses $\rho_i \in Z_q^*$ and records $(P_i, \rho_i P, \rho_i)$ in H_1-list; else, B gets its mapping value, $\rho_i P$, from H_1-list. Then, B returns $\rho_i P$ to E_1.

(iii) H_2 query: B constructs H_2-list. If E_1 queries $H_2(m, U_1, U_2)$ to B, B checks whether (m, U_1, U_2) is in H_2-list or not. If not, B randomly chooses $h_2 \in Z_q^*$ and records $((m, U_1, U_2), h_2)$ in H_2-list; else, B gets its mapping value, h_2, from H_2-list. Then, B returns h_2 to E_1.

(iv) H_3 query: B constructs H_3-list. When E_1 queries $H_3(m, U_2, U_1)$ to B, B checks whether (m, U_2, U_1) is in H_3-list or not. If not, B randomly chooses $h_3 \in Z_q^*$ and records $((m, U_2, U_1), h_3)$ in H_3-list; otherwise, B gets its mapping value, h_3, from H_3-list. Then, B responds h_3 to E_1.

(v) H_4 query: B constructs H_4-list. If E_1 queries H_4 with a string ϱ to B, B checks whether ϱ is in H_4-list or not. If ϱ does not exist in H_4-list, then there are the following two conditions. Condition 1: if $\varrho = \theta$, B sets $W = \beta P = H_4(\varrho)$ where β is randomly chosen in Z_q^* and stores $(\varrho, \beta P, \beta)$ in H_4-list. Condition 2: if $\varrho \neq \theta$, B sets $W = \Phi = H_4(\varrho)$ where Φ is randomly chosen in G_1 and stores (ϱ, Φ) in H_4-list. However, if ϱ exists in H_4-list, B gets its mapping value, $W = \Phi$ or βP. Finally, B returns W.

(vi) *Public_Key* query: B constructs a list, pk-list. When E_1 queries *Public_Key*(ID_i) to B, B looks up pk-list. If ID_i is not found in pk-list, B randomly chooses $x_i \in$

Z_q^*, computes $P_i = x_i P$, and stores (ID_i, P_i, x_i) in *pk*-list; otherwise, B gets P_i from *pk*-list. Finally, B returns P_i.

(vii) *Partial_Private_Key* query: if E_1 queries *Partial_Private_Key*(ID_i, P_i') to B, B looks up H_0-list, *pk*-list, and H_1-list. If $ID_i = ID_\pi$, B returns "failure." If ID_i is not found in H_0-list, B queries $H_0(ID_i)$ and gets k_i from H_0-list; else, B retrieves k_i from H_0-list. If ID_i is not in *pk*-list, B queries *Public_Key*(ID_i) and obtains P_i; otherwise, B catches P_i from *pk*-list. If $P_i \neq P_i'$, B returns "failure." Then, if P_i is not found in H_1-list, B queries $H_1(P_i)$ and gets ρ_i; else, B retrieves ρ_i from H_1-list. Finally, B returns $(D_{i_0} = k_i bP, D_{i_1} = \rho_i bP)$.

(viii) *Secret_Key* query: if E_1 queries *Secret_Key*(ID_i) to B, B looks up *pk*-list. If ID_i is not found in *pk*-list, B queries *Public_Key*(ID_i) and gets x_i; if there is a record (ID_i, P_i, x_i), then B retrieves x_i from *pk*-list and returns x_i.

(ix) *Public_Key_Replacement* query: when E_1 queries *Public_Key_Replacement*(ID_i, P_i') to B, B looks up *pk*-list. If ID_i is not found in *pk*-list, B queries *Public_Key*(ID_i). Then, B replaces the record (ID_i, P_i, x_i) with $(ID_i, P_i', \perp)$ in *pk*-list.

(x) *Sign* query: when E_1 queries *Sign*(ID_i, m) to B, if $ID_i = ID_\pi$, B does the following works.

(1) Choose z, α, and $h_2 \in Z_q^*$ randomly;

(2) compute $U_1 = zP - h_2 aP$ and $U_2 = \alpha P$;

(3) set $H_2(m, U_1, U_2) = h_2$;

(4) compute $h_3 = H_3(m, U_2, U_1)$ and $W = H_4(\theta)$;

(5) compute $V = zbP + h_2\rho_i bP + \alpha\beta P + h_3\beta P_i$;

(6) form $\sigma = (V, U_1, U_2, P_i)$.

Thus, the signature on m is $\sigma = (V, U_1, U_2, P_i)$ and it satisfies the verifying formula in Section 3.

If $ID_i \neq ID_\pi$, B can return a signature on m to E_1 because B can compute all secrets of user ID_i. Finally, suppose that E_1 outputs, with probability at least ε, n signatures $\sigma_1, \ldots, \sigma_n$ on $m_1, \ldots, m_n$ of the signers $ID_1, \ldots, ID_n$, respectively, such that

(1) *Batch_Verify*$((\sigma_1, m_1, ID_1, P_1), \ldots, (\sigma_n, m_n, ID_n, P_n))$ = True;

(2) there exists $\sigma_y \in \{\sigma_1, \ldots, \sigma_n\}$ which is not the output from *Sign*(ID_y, m_y);

(3) *Partial_Private_Key*(ID_y, P_y) has never been queried.

From Lemma 9, we fork the sequence of signatures one time and get $\sigma_1', \ldots, \sigma_n'$ on $m_1, \ldots, m_n$ by setting $h_{2_y}' \neq h_{2_y}$. Thus, we randomly choose w_i's and obtain the following two equations:

$$\begin{aligned} e\left(P, \sum_{i=1}^{n} w_i V_i\right) &= e\left(T_{\text{pub}}, \sum_{i=1}^{n} w_i U_{1_i} + w_i h_{2_i}\left(Q_i + \Gamma_i\right)\right) \\ &\quad \times e\left(W, \sum_{i=1}^{n} w_i U_{2_i} + w_i h_{3_i} P_i\right), \\ e\left(P, \sum_{i=1}^{n} w_i V_i'\right) &= e\left(T_{\text{pub}}, \sum_{i=1}^{n} w_i U_{1_i}' + w_i h_{2_i}'\left(Q_i + \Gamma_i\right)\right) \\ &\quad \times e\left(W, \sum_{i=1}^{n} w_i U_{2_i}' + w_i h_{3_i}' P_i\right) \end{aligned} \tag{2}$$

with probability being at least $\varepsilon/2$ by Lemma 10, where $V_y \neq V_y'$, $U_{1_y} = U_{1_y}'$, $h_{2y} \neq h_{2_y}'$, $U_{2_y} = U_{2_y}'$, and $h_{3y} = h_{3_y}'$.

We assume that $ID_y = ID_\pi$. By Theorem 11, we have that

$$\begin{aligned} e\left(P, V_y\right) &= e\left(T_{\text{pub}}, U_{1_y} + h_{2_y}\left(Q_y + \Gamma_y\right)\right) \\ &\quad \times e\left(W, U_{2_y} + h_{3_y} P_y\right), \\ e\left(P, V_y'\right) &= e\left(T_{\text{pub}}, U_{1_y} + h_{2_y}'\left(Q_y + \Gamma_y\right)\right) \\ &\quad \times e\left(W, U_{2_y} + h_{3_y} P_y\right) \end{aligned} \tag{3}$$

with probability being at least $(\varepsilon/(2q_{h_0}))\ (1-1/2^{|q|})^2$. Thus, we can compute $(h_{2_y} - h_{2_y}')^{-1}((V_y - V_y') - \rho_y T_{\text{pub}})$, which is abP, to solve the CDH problem with $\varepsilon' \geq (\varepsilon/(2q_{h_0}))(1 - 1/2^{|q|})^2$ and $t' \approx t + q_{\text{sign}}\mathcal{O}(t_{\text{sign}}) + q_{ppk}\mathcal{O}(t_{ppk}) + q_{sk}\mathcal{O}(t_{sk}) + q_{pkr}\mathcal{O}(t_{pkr}) + q_{pk}\mathcal{O}(t_{pk}) + q_{h_0}\mathcal{O}(t_{h_0}) + q_{h_1}\mathcal{O}(t_{h_1}) + q_{h_2}\mathcal{O}(t_{h_2}) + q_{h_3}\mathcal{O}(t_{h_3}) + q_{h_4}\mathcal{O}(t_{h_4}) + \mathcal{O}(1)$. □

Theorem 13. *The proposed scheme is $(t, q_{sign}, q_{ppk}, q_{sk}, q_{pkr}, q_{pk}, q_{h_0}, q_{h_1}, q_{h_2}, q_{h_3}, q_{h_4}, \varepsilon, II)$-unforgeable assuming that the (t', ε')-CDH assumption holds in G_1 where $\varepsilon' \geq (\varepsilon/(2q_{pk}))(1 - 1/2^{|q|})^2, t' \approx t + q_{sign}\mathcal{O}(t_{sign}) + q_{ppk}\mathcal{O}(t_{ppk}) + q_{sk}\mathcal{O}(t_{sk}) + q_{pkr}\mathcal{O}(t_{pkr}) + q_{pk}\mathcal{O}(t_{pk}) + q_{h_0}\mathcal{O}(t_{h_0}) + q_{h_1}\mathcal{O}(t_{h_1}) + q_{h_2}\mathcal{O}(t_{h_2}) + q_{h_3}\mathcal{O}(t_{h_3}) + q_{h_4}\mathcal{O}(t_{h_4}) + \mathcal{O}(1)$, q_{sign}, q_{ppk}, q_{sk}, q_{pkr}, q_{pk}, q_{h_0}, q_{h_1}, q_{h_2}, q_{h_3}, and q_{h_4} are the numbers of queries to Sign, Partial_Private_Key, Secret_Key, Public_Key_Replacement, Public_Key, H_0, H_1, H_2, H_3, and H_4, respectively, and t_{sign}, t_{ppk}, t_{sk}, t_{pkr}, t_{pk}, t_{h_0}, t_{h_1}, t_{h_2}, t_{h_3}, and t_{h_4} are the computing time of the queries to Sign, Partial_Private_Key, Secret_Key, Public_Key_Replacement, Public_Key, H_0, H_1, H_2, H_3, and H_4, respectively.*

Proof. Assume that a polynomial-time attacker E_2 wins the game of Definition 7 with probability at least ε within running time t. A simulator B is given an instance of the CDH problem $\langle G_1, q, P, aP, bP\rangle$, and B's goal is to output the value of abP. We will construct B which plays the game in Definition 7 with E_2 and outputs the value of abP.

Setup. B sets $T_{\text{pub}} = \lambda P$ where λ is randomly chosen in Z_q^* and chooses $\theta \in \{0,1\}^*$ randomly. Then, B publishes $\{G_1, G_2, e, q, P, \theta, T_{\text{pub}}, H_0, H_1, H_2, H_3, H_4\}$ and sends λ to E_2. B can respond to the queries from E_2 as follows.

(i) H_0 query: B constructs a list, H_0-list. When E_2 queries $H_0(ID_i)$ to B, B checks whether ID_i is in H_0-list or not. If ID_i does not exist in H_0-list, B sets $Q_i = k_iP = H_0(ID_i)$ where k_i is a random integer in Z_q^* and records $(ID_i,\ k_iP,\ k_i)$ in H_0-list; else, B gets its mapping value, k_iP, from H_0-list. Then, B returns $Q_i = k_iP$ to E_2.

(ii) H_4 query: B constructs H_4-list. If E_2 queries H_4 with a string ϱ to B, B checks whether ϱ is in H_4-list or not. If ϱ does not exist in H_4-list, then there are the following two conditions. Condition 1: if $\varrho = \theta$, B sets $W = bP = H_4(\varrho)$ and stores (ϱ, bP) in H_4-list. Condition 2: if $\varrho \neq \theta$, B sets $W = \Phi = H_4(\varrho)$ where Φ is randomly chosen in G_1 and stores (ϱ, Φ) in H_4-list. However, if ϱ exists in H_4-list, B gets its mapping value, $W = \Phi$ or bP. Finally, B returns W.

(iii) The simulations for H_1, H_2, H_3, and *Public_Key_Replacement* are the same as those in the proof of Theorem 12

(iv) *Public_Key* query: B constructs a list, pk-list, and chooses an identity ID_π randomly. When E_2 queries *Public_Key*(ID_i) to B, B looks up pk-list. If ID_i does not exist in pk-list, then there are the following two conditions. Condition 1: if $ID_i = ID_\pi$, B sets $P_i = aP$ and stores (ID_i, aP) in pk-list. Condition 2: if $ID_i \neq ID_\pi$, B sets $P_i = x_iP$ where x_i is randomly chosen in Z_q^* and stores (ID_i, x_iP, x_i) in pk-list. However, if ID_i exists in pk-list, B gets its mapping value, $P_i = x_iP$ or aP. Finally, B returns P_i.

(v) *Partial_Private_Key* query: when E_2 queries *Partial_Private_Key*(ID_i, P_i') to B, B looks up H_0-list, pk-list, and H_1-list. If ID_i is not found in H_0-list, B queries $H_0(ID_i)$ and gets k_i from H_0-list; else, B retrieves k_i from H_0-list. If ID_i is not in pk-list, B queries *Public_Key*(ID_i) and obtains P_i; otherwise, B catches P_i from pk-list. If $P_i \neq P_i'$, B returns "failure." Then, if P_i is not found in H_1-list, B queries $H_1(P_i)$ and gets ρ_i; else, B retrieves ρ_i from H_1-list. Finally, B returns $(D_{i_0} = \lambda k_iP, D_{i_1} = \lambda\rho_iP)$.

(vi) *Secret_Key* query: when E_2 queries *Secret_Key*(ID_i) to B, B looks up pk-list. If $ID_i = ID_\pi$, B returns "failure." If ID_i is not found in pk-list, B queries *Public_Key*(ID_i) and gets x_i from pk-list; else, B retrieves x_i from pk-list. Finally, B returns x_i.

(vii) *Sign* query: when E_2 queries *Sign*(ID_i, m) to B, if $ID_i = ID_\pi$, B performs the following works.

(1) choose z, α, and $h_3 \in Z_q^*$ randomly;

(2) compute $U_1 = zP$ and $U_2 = \alpha P - h_3aP$;

(3) compute $h_2 = H_2(m, U_1, U_2)$ and $W = H_4(\theta)$;

(4) set $H_3(m, U_2, U_1) = h_3$;

(5) compute $V = z\lambda P + h_2k_i\lambda P + h_2\rho_i\lambda P + \alpha W$;

(6) form $\sigma = (V, U_1, U_2, P_i)$.

Therefore, the signature on m is $\sigma = (V, U_1, U_2, P_i)$ and it meets the verifying formula in Section 3.

If $ID_i \neq ID_\pi$, B can return a signature on m to E_2 since B knows all secrets of user ID_i. Finally, E_2 outputs, with probability being at least ε, n signatures $\sigma_1, \ldots, \sigma_n$ on $m_1, \ldots, m_n$ of the signers $ID_1, \ldots, ID_n$, respectively, such that

(1) *Batch_Verify*$((\sigma_1, m_1, ID_1, P_1), \ldots, (\sigma_n, m_n, ID_n, P_n))$ = True;

(2) there exists $\sigma_y \in \{\sigma_1, \ldots, \sigma_n\}$ which is not the output from *Sign*(ID_y, m_y);

(3) neither *Secret_Key*(ID_y) nor *Public_Key_Replacement*$(ID_y, \cdot, \cdot)$ has been queried.

From Lemma 9, we fork the sequence of signatures one time and get $\sigma_1', \ldots, \sigma_n'$ on $m_1, \ldots, m_n$ by setting $h_{3_y}' \neq h_{3_y}$. Thus, we randomly choose w_i's and obtain the following two equations:

$$\begin{aligned} & e\left(P, \sum_{i=1}^{n} w_iV_i\right) \\ & \quad = e\left(T_{\text{pub}}, \sum_{i=1}^{n} w_iU_{1_i} + w_ih_{2_i}\left(Q_i + \Gamma_i\right)\right) \\ & \quad \times e\left(W, \sum_{i=1}^{n} w_iU_{2_i} + w_ih_{3_i}P_i\right), \\ & e\left(P, \sum_{i=1}^{n} w_iV_i'\right) \\ & \quad = e\left(T_{\text{pub}}, \sum_{i=1}^{n} w_iU_{1_i}' + w_ih_{2_i}'\left(Q_i + \Gamma_i\right)\right) \\ & \quad \times e\left(W, \sum_{i=1}^{n} w_iU_{2_i}' + w_ih_{3_i}'P_i\right) \end{aligned} \tag{4}$$

with probability being at least $\varepsilon/2$ by Lemma 10, where $V_y \neq V_y'$, $U_{1_y} = U_{1_y}'$, $h_{2y} = h_{2_y}'$, $U_{2_y} = U_{2_y}'$, and $h_{3y} \neq h_{3_y}'$.

We assume that $ID_y = ID_\pi$. By Theorem 11, we have that

$$\begin{aligned} e\left(P, V_y\right) &= e\left(T_{\text{pub}}, U_{1_y} + h_{2_y}\left(Q_y + \Gamma_y\right)\right) \\ & \quad \times e\left(W, U_{2_y} + h_{3_y}P_y\right), \\ e\left(P, V_y'\right) &= e\left(T_{\text{pub}}, U_{1_y} + h_{2_y}\left(Q_y + \Gamma_y\right)\right) \\ & \quad \times e\left(W, U_{2_y} + h_{3_y}'P_y\right) \end{aligned} \tag{5}$$

with probability being at least $(\varepsilon/(2q_{pk}))(1-1/2^{|q|})^2$. Thus, we can compute $(h_{3_y}-h'_{3_y})^{-1}(V_y-V'_y)$, which is abP, to solve the CDH problem with $\varepsilon' \geq (\varepsilon/(2q_{pk}))(1-1/2^{|q|})^2$ and $t' \approx t + q_{\text{sign}}\mathcal{O}(t_{\text{sign}}) + q_{ppk}\mathcal{O}(t_{ppk}) + q_{sk}\mathcal{O}(t_{sk}) + q_{pkr}\mathcal{O}(t_{pkr}) + q_{pk}\mathcal{O}(t_{pk}) + q_{h_0}\mathcal{O}(t_{h_0}) + q_{h_1}\mathcal{O}(t_{h_1}) + q_{h_2}\mathcal{O}(t_{h_2}) + q_{h_3}\mathcal{O}(t_{h_3}) + q_{h_4}\mathcal{O}(t_{h_4}) + \mathcal{O}(1)$. □

Theorem 14. *The proposed scheme is* $(t, q_{ppk}, q_{sk}, q_{pkr}, q_{pk}, q_{h_0}, q_{h_1}, q_{h_2}, q_{h_3}, q_{h_4}, \varepsilon, \text{III})$*-unforgeable assuming that the* (t', ε')*-CDH assumption holds in* G_1 *with* $\varepsilon' \geq \varepsilon/(2q_{h_1-\max})$, $t' \approx t + q_{ppk}\mathcal{O}(t_{ppk}) + q_{sk}\mathcal{O}(t_{sk}) + q_{pkr}\mathcal{O}(t_{pkr}) + q_{pk}\mathcal{O}(t_{pk}) + q_{h_0}\mathcal{O}(t_{h_0}) + q_{h_1}\mathcal{O}(t_{h_1}) + q_{h_2}\mathcal{O}(t_{h_2}) + q_{h_3}\mathcal{O}(t_{h_3}) + q_{h_4}\mathcal{O}(t_{h_4}) + \mathcal{O}(1)$, $(2q_{h_1-\max})$ *being the possibly maximal number of queries to* H_1, q_{ppk}, q_{sk}, q_{pkr}, q_{pk}, q_{h_0}, q_{h_1}, q_{h_2}, q_{h_3}, *and* q_{h_4} *being the numbers of queries to Partial_Private_Key, Secret_Key, Public_Key_Replacement, Public_Key,* H_0, H_1, H_2, H_3, *and* H_4, *respectively, and* t_{sign}, t_{ppk}, t_{sk}, t_{pkr}, t_{pk}, t_{h_0}, t_{h_1}, t_{h_2}, t_{h_3}, *and* t_{h_4} *being the computing time of the queries to Sign, Partial_Private_Key, Secret_Key, Public_Key_Replacement, Public_Key,* H_0, H_1, H_2, H_3, *and* H_4, *respectively.*

Proof. Assume that a polynomial-time attacker E_3 wins the game of Definition 8 with probability being at least ε within running time t. A simulator B is given an instance of the CDH problem $\langle G_1, q, P, aP, bP\rangle$, and B's goal is to output abP. We will construct B which plays the game in Definition 8 with E_3 and outputs abP.

Setup. B sets $T_{\text{pub}} = bP$ and chooses $\theta \in \{0,1\}^*$ randomly. Then, B publishes $\{G_1, G_2, e, q, P, \theta, T_{\text{pub}}, H_0, H_1, H_2, H_3, H_4\}$. B can respond to the queries from E_3 as follows.

The simulation for H_0 is identical to that in the proof of Theorem 13. Consider the following.

(i) H_1 query: B constructs a list, H_1-list, and then chooses $\pi \in \{1, \ldots, q_{h_1-\max}\}$ at random and sets an index $j = 0$. When E_3 queries $H_1(P_i)$ to B, B checks whether P_i is in H_1-list or not. If P_i does not exist in H_1-list, B computes $j = j + 1$ and there are the following two cases. Case 1: if $j = \pi$, B sets $H_1(P_i) = aP$ and stores (P_i, aP) in H_1-list. Case 2: if $j \neq \pi$, B sets $H_1(P_i) = \rho_i P$ where ρ_i is randomly chosen in Z_q^* and stores $(P_i, \rho_i P, \rho_i)$ in H_1-list. Besides, if P_i exists in H_1-list, B gets its mapping value, $\rho_i P$ or aP. Finally, B returns $\rho_i P$ or aP.

(ii) H_4 query: B constructs H_4-list. If E_3 queries H_4 with a string ϱ to B, B checks whether ϱ is in H_4-list or not. If ϱ does not exist in H_4-list, then there are the following two conditions. Condition 1: if $\varrho = \theta$, B sets $W = \beta P = H_4(\varrho)$ where β is randomly chosen in Z_q^* and stores $(\varrho, \beta P, \beta)$ in H_4-list. Condition 2: if $\varrho \neq \theta$, B sets $W = \Phi = H_4(\varrho)$ where Φ is randomly chosen in G_1 and stores (ϱ, Φ) in H_4-list. However, if ϱ exists in H_4-list, B gets its mapping value, $W = \Phi$ or βP. Finally, B returns W.

(iii) *Public_Key* query: B constructs a list, pk-list. When E_3 queries *Public_Key*(ID_i) to B, B looks up pk-list. If ID_i is not found in pk-list, B does the following works. B computes $P_i = x_i P$ where x_i is randomly chosen in Z_q^*. B queries H_1 with P_i. If $H_1(P_i) = aP$, B returns "failure"; otherwise, B stores (ID_i, P_i, x_i) in pk-list. Besides, if ID_i is found in pk-list, B gets P_i from pk-list. Finally, B returns P_i to E_3.

(iv) *Partial_Private_Key* query: when E_3 queries *Partial_Private_Key*(ID_i, P_i') to B, B looks up H_0-list, pk-list, and H_1-list. If ID_i is not found in H_0-list, B queries $H_0(ID_i)$ and gets k_i from H_0-list; else, B retrieves k_i from H_0-list. If ID_i is not found in pk-list, B queries *Public_Key* with ID_i and obtains P_i from pk-list; otherwise, B catches P_i from pk-list. If $P_i \neq P_i'$, B returns "failure." Then, B gets ρ_i from H_1-list. Finally, B returns $(D_{i_0} = k_i bP, D_{i_1} = \rho_i bP)$.

(v) H_2, H_3, *Secret_Key*, and *Public_Key_Replacement* queries are the same as those in the proof of Theorem 12. Finally, E_3 outputs, with probability at least ε, a signature $\sigma^* = (V^*, U_1^*, U_2^*, P_y^*)$ on a message m^* for the user ID_y such that

(1) *Verifying*$(\sigma^*, m^*, ID_y, P_y^*) = \text{True}$;

(2) user ID_y has been created;

(3) P_y^* is different from all of the public keys P_y^*'s returned by *Public_Key*(ID_y) or used to query *Public_Key_Replacement*.

From Lemma 9, we fork the sequence of signatures one time and get $\sigma' = (V', U_1', U_2', P_y^*)$ on m^* by setting $h_2' \neq h_2^*$. Thus, we obtain the following two equations:

$$e(P, V^*) = e\left(T_{\text{pub}}, U_1^* + h_2^*\left(Q_y + \Gamma_y\right)\right) e\left(W, U_2^* + h_3^* P_y^*\right)$$
$$e\left(P, V'\right) = e\left(T_{\text{pub}}, U_1' + h_2'\left(Q_y + \Gamma_y\right)\right) e\left(W, U_2' + h_3' P_y^*\right) \tag{6}$$

with a probability at least $\varepsilon/2$ by Lemma 10, where $V^* \neq V'$, $U_1^* = U_1'$, $h_2^* \neq h_2'$, $U_2^* = U_2'$, and $h_3^* = h_3'$.

If $H_1(P_y^*) = aP$, B can get $abP = (h_2^* - h_2')^{-1}(V^* - V') - k_y T_{\text{pub}}$ and solve the CDH problem.

The success probability is $\varepsilon' \geq \varepsilon/(2q_{h_1-\max})$ with the computing time $t' \approx t + q_{\text{sign}}\mathcal{O}(t_{\text{sign}}) + q_{ppk}\mathcal{O}(t_{ppk}) + q_{sk}\mathcal{O}(t_{sk}) + q_{pkr}\mathcal{O}(t_{pkr}) + q_{pk}\mathcal{O}(t_{pk}) + q_{h_0}\mathcal{O}(t_{h_0}) + q_{h_1}\mathcal{O}(t_{h_1}) + q_{h_2}\mathcal{O}(t_{h_2}) + q_{h_3}\mathcal{O}(t_{h_3}) + q_{h_4}\mathcal{O}(t_{h_4}) + \mathcal{O}(1)$.

By Theorem 14, a user cannot produce a signature with a new public-secret key pair, which is different from his own one, without the corresponding partial private key being generated from the master secret key. Therefore, if there exist two valid signatures of a user with different public key, the user can prove to anyone that PKG is misbehaving, which means that our scheme achieves Girault's level-3 security. □

5. Discussions

The comparisons between our scheme and [5, 7, 11, 13, 14, 16, 27–34] are shown in Table 1. Although our proposed scheme

Table 1: The comparisons between [5, 7, 11, 13, 14, 16, 27–34] and our scheme.

	Signing phase	Verification phase	Security level	Formally proved	Security model
[5]	nT_s	$2nT_e + 2nT_a$ $\approx 2400nt_m$	Girault's level-3	Yes	ROM
[7]	$2nT_s + 2nT_h + nT_a$	$(2n+1)T_p + nT_a$ $+nT_s + 2nT_h$ $\approx (2475n + 1200)t_m$	Girault's level-2	Yes	ROM
[27]	$2nT_s$	$2T_p + 3nT_s + T_a$ $\approx (87n + 480)t_m$	Girault's level-1	Yes	ROM
[28] Cha-based	$2nT_s$	$2T_p + 2nT_s + nT_a$ $+nt_m$ $\approx (59n + 480)t_m$	Girault's level-1	No	ROM
[28] Waters-based	$(z+2)T_s + (z+1)T_a$	$3T_p + 2n(z+1)T_s$ $+n(2z+3)T_a$ $\approx (58n(z+1) + 720)t_m$	Girault's level-1	No	STD
[28] Hess-based	$nT_p + 3nT_s + nT_a$	$2T_p + 3nT_s$ $+nt_m$ $\approx (88n + 480)t_m$	Girault's level-1	No	ROM
[29] Geng-based	$3nT_s + nT_a + nt_m$	$3T_p + 4nT_s$ $+(n-1)T_a + 2nt_m$ $\approx (118n + 720)t_m$	Girault's level-2	No	ROM
[32]	$2nT_p + 11nT_s + 4n(T_a + t_m)$	$2T_p + 13nT_s$ $+8T_a$ $\approx (377n + 480)t_m$	Girault's level-1	No	ROM
[11]	$2nT_s$	$(n+1)T_p$ $\approx (1200n + 1200)t_m$	Girault's level-2	No	—
[13]	nt_e	$(n+1)T_e + nt_e$ $\approx (1200n + 1440)t_m$	Girault's level-3	Yes	ROM
[30]	$(k+4)nT_s + (k+2)nT_a$	$5T_p + 4nT_s$ $+4(n-1)T_a$ $\approx (116n + 1200)t_m$	Girault's level-1	No	—
[14]	$5nT_s + nT_a$	$(n+1)T_p + 3nT_a + 2nT_s$ $\approx (1258n + 1200)t_m$	Girault's level-2	Yes	ROM
[33]	$2nT_p + 13nT_s + 4nT_a + 8nt_m$	$2T_p + 14nT_s$ $+7nT_a + 2nt_m$ $\approx (408n + 480)t_m$	Girault's level-1	No	ROM
[34]	$2nT_p + 10nT_s + 6nT_a + 7nt_m$	$2T_p + 13nT_s$ $+n(T_a + t_m)$ $\approx (378n + 480)t_m$	Girault's level-1	No	—
[16]	$3nt_m + 5nt_e$	$(3n+1)T_p + nt_e$ $\approx (3601n + 1200)t_m$	Girault's level-2	Yes	STD
[31]	$n(T_s + T_a)$	$3T_p + nT_s$ $+nT_h + 3nT_a$ $\approx (52n + 720)t_m$	Girault's level-1	No	—
Ours	$4nT_s + nT_h + 3nT_a$	$3T_p + 3nT_s$ $+2nT_h + nT_a$ $\approx (133n + 720)t_m$	Girault's level-3	Yes	ROM

According to [41–43], $T_p \approx 5t_e$, $T_s \approx 29t_m$, $T_h \approx 23t_m$, $T_a \approx 0.12t_m$, and $t_e \approx 240t_m$.

z: the number of ℓ-bit chunks in Waters scheme; k: the number of registered users in Qin scheme; n: the number of individual signatures; T_p: the time cost of a pairing operation; T_s: the time cost of a scalar multiplication in G_1; T_h: the time cost of a map-to-point hash operation; T_a: the time cost of a point addition operation; t_m: the time cost of a modular multiplication in Z_p; t_h: the time cost of a hash operation; ROM: random oracle model; STD: standard model.

is not the most efficient, it satisfies the property of Girault's level-3 security with formal proofs.

6. Conclusions

In this paper, we have proposed a certificateless signature scheme with fast batch verification and it satisfies Girault's level-3 security, where almost all existing signatures for batch verification reach Girault's level-1 security and only one reaches Girault's level-2 security. Finally, we have formally demonstrated that the proposed scheme is unforgeable and achieves Girault's level-3 security based on the CDH problem.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Science Council of Taiwan under Grant NSC 102-2219-E-110-002, NSYSU-KMU Joint Research Project (NSYSUKMU 2013-I001), and Aim for the Top University Plan of the National Sun Yat-sen University and Ministry of Education, Taiwan. A partial result of this research has been presented at the Eighth Asia Joint Conference on Information Security, Seoul, Republic of Korea, July 25-26, 2013.

References

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology-CRYPTO*, pp. 47–53, 1985.

[2] B. Libert and J. Quisquater, "What is possible with identity based cryptography for PKIs and what still must be improved," in *Proceedings of the European PKI Workshop (EuroPKI '04)*, pp. 57–70, 2004.

[3] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology-ASIACRYPT*, pp. 452–473, 2003.

[4] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the Security of Certificateless Signature Schemes from Asiacrypt 2003," in *Proceedings of the International Conference on Cryptology and Network Security (CANS '05)*, pp. 13–25, 2005.

[5] Y. Chen, G. Horng, and C. Liu, "Strong non-repudiation based on certificateless short signatures," *IET Infomation Security*, vol. 7, no. 3, pp. 253–263, 2012.

[6] K. Y. Choi, J. H. Park, and D. H. Lee, "A new provably secure certificateless short signature scheme," *Computers and Mathematics with Applications*, vol. 61, no. 7, pp. 1760–1768, 2011.

[7] L. Cheng and Q. Wen, "A secure and efficient certificateless short signature scheme," *Journal of Engineering Science and Technology Review*, vol. 6, no. 2, pp. 35–44, 2011.

[8] H. Du and Q. Wen, "Efficient and provably-secure certificateless short signature scheme from bilinear pairings," *Computer Standards and Interfaces*, vol. 31, no. 2, pp. 390–394, 2009.

[9] M. Gorantla and A. Saxena, "An efficient certificateless signature scheme," in *Proceedings of International Conference on Computational Intelligence and Security (CIS '05)*, pp. 110–116, 2005.

[10] X. Li, K. Chen, and L. Sun, "Certificateless signature and proxy signature schemes from bilinear pairings," *Lithuanian Mathematical Journal*, vol. 45, no. 1, pp. 76–83, 2005.

[11] F. Li and P. Liu, "An efficient certificateless signature scheme from bilinear parings," in *Proceedings of the International Conference on Network Computing and Information Security (NCIS '11)*, pp. 35–37, May 2011.

[12] R. Tso, X. Huang, and W. Susilo, "Strongly secure certificateless short signatures," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1409–1417, 2012.

[13] R. Tso, C. Kim, and X. Yi, "Certificateless message recovery signatures providing Girault's level-3 security," *Journal of Shanghai Jiaotong University (Science)*, vol. 16, no. 5, pp. 577–585, 2011.

[14] Z. Wan, "Certificateless directed signature scheme," in *Proceedings of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '11)*, pp. 1–4, September 2011.

[15] W. Yap, S. Heng, and B. Goi, "An efficient certificateless signature scheme," in *Proceedings of International Conference on Emerging Directions in Embedded and Ubiquitous Computing (EUC '06)*, pp. 322–331, 2006.

[16] Y. Yuan and C. Wang, "A secure certificateless signature scheme in the standard model," *Journal of Computational Information Systems*, vol. 9, no. 11, pp. 4353–4362, 2013.

[17] Z. Zhang, D. Wong, J. Xu, and D. Feng, "Certificateless public-key signature: security model and efficient construction," in *Proceedings of International Conference on Applied Cryptography and Network Securit (ACNS '06)*, pp. 293–308, 2006.

[18] L. Zhang, Q. Wu, J. Domingo-Ferrer, and B. Qin, "New efficient certificateless signature scheme," in *Proceedings of International Conference on Emerging Directions in Embedded and Ubiquitous Computing (EUC '07)*, pp. 692–703, 2007.

[19] B. C. Hu, D. S. Wong, Z. Zhang, and X. Deng, "Certificateless signature: a new security model and an improved generic construction," *Designs, Codes, and Cryptography*, vol. 42, no. 2, pp. 109–126, 2007.

[20] M. Girault, "Self-certified public keys," in *Advances in Cryptology-EUROCRYPT*, pp. 490–497, 1991.

[21] A. Fiat, "Batch RSA," in *Advances in cryptology-CRYPTO*, pp. 175–185, 1990.

[22] C. Boyd and C. Pavlovski, "Attacking and repairing batch verification schemes," in *Advances in cryptology-ASIACRYPT*, pp. 58–71, 2000.

[23] L. Harn, "Batch verifying multiple RSA digital signatures," *Electronics Letters*, vol. 34, no. 12, pp. 1219–1220, 1998.

[24] D. Naccache, D. M'Raihi, S. Vaudenay, and D. Raphaeli, "Can D. S. A. be improved? Complexity trade-offs with the digital signature standard," in *Advances in Cryptology-EUROCRYPT*, pp. 77–85, 1994.

[25] H. Yoon, J. H. Cheon, and Y. Kim, "Batch verifications with ID-based signatures," in *Proceedings of International Conference on Information Security and Cryptology (ICISC '04)*, pp. 233–248, 2004.

[26] M. Bellare and J. Garay, "Fast batch verification for modular exponentiation and digital signatures," in *Advances In Cryptology-EUROCRYPT*, pp. 236–250, 1998.

[27] C. Shi, D. Pu, and W. C. Choong, "An efficient identity-based signature scheme with batch verifications," in *Proceedings of the 1st International Conference on Scalable information systems (INFOSCALE '06)*, vol. 22, pp. 1–6, June 2006.

[28] A. Ferrara, M. Green, S. Hobenberger, and M. Pedersen, "Practical short signature batch verification," in *Proceedings of the The Cryptographers' Track at the RSA Conference on Topics in Cryptology (CT-RSA '09)*, pp. 309–324, 2009.

[29] M. Geng and F. Zhang, "Batch verification for certificateless signature schemes," in *Proceedings of the International Conference on Computational Intelligence and Security (CIS '09)*, pp. 288–292, December 2009.

[30] X. Qin, S. Zhang, and L. Jia, "Research on pairing-based batch verification," in *Proceedings of the International Conference on Communications and Mobile Computing (CMC '10)*, pp. 46–50, April 2010.

[31] C. Zhang, P. Ho, and J. Tapolcai, "On batch verification with group testing for vehicular communications," *Wireless Networks*, vol. 17, no. 8, pp. 1851–1865, 2011.

[32] K. Kim, I. Yie, S. Lim, and D. Nyang, "Batch verification and finding invalid signatures in a group signature scheme," *International Journal of Network Security*, vol. 13, no. 2, pp. 61–70, 2011.

[33] A. Wasef and X. Shen, "Efficient group signature scheme supporting batch verification for securing vehicular networks," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, pp. 1–5, May 2010.

[34] L. Wei, J. Liu, and T. Zhu, "On a group signature scheme supporting batch verification for vehicular networks," in *Proceedings of the 3rd International Conference on Multimedia Information Networking and Security (MINES '11)*, pp. 436–440, November 2011.

[35] J. Camenisch, S. Hohenberger, and M. Pedersen, "Batch verification of short signature," in *Proceedings of the Advances in Cryptology-(EUROCRYPT '07)*, pp. 246–263, 2007.

[36] T. Cao, D. Lin, and R. Xue, "Security analysis of some batch verifying signatures from pairings," *International Journal of Network Security*, vol. 3, no. 2, pp. 138–143, 2006.

[37] F. Guo, Y. Mu, and Z. Chen, "Efficient batch verification of short signatures for a single-signer setting without random oracles," in *Proceedings of the International Workshop on Security (IWSEC '08)*, pp. 49–63, 2008.

[38] M. Hwang, C. Lee, and Y. Tang, "Two simple batch verifying multiple digital signatures," in *Proceedings of International Conference on Information and Communications Security (ICICS '01)*, pp. 233–237, 2001.

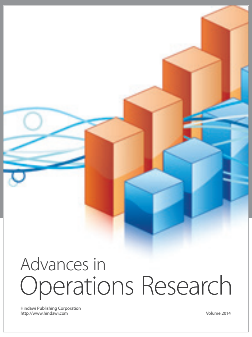
[39] S. Yen and C. Laih, "Improved digital signature suitable for batch verification," *IEEE Transactions on Computers*, vol. 44, no. 7, pp. 957–959, 1995.

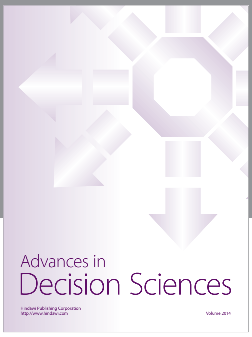
[40] D. Pointcheval and J. Stern, "Security proofs for signature schemes," in *Advances in Cryptology-EUROCRYPT*, pp. 387–398, 1996.

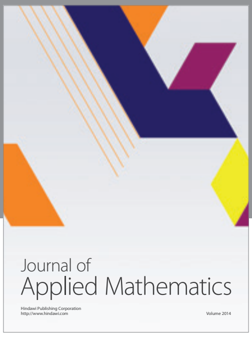
[41] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes, and Cryptography*, vol. 19, no. 2-3, pp. 173–193, 2000.

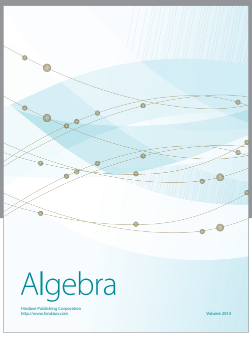
[42] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, LLC, Boca Raton, Fla, USA, 1997.

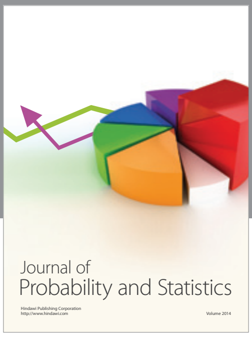
[43] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Securing mobile ad hoc networks with certificateless public keys," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 4, pp. 386–399, 2006.


Advances in Operations Research


Advances in Decision Sciences

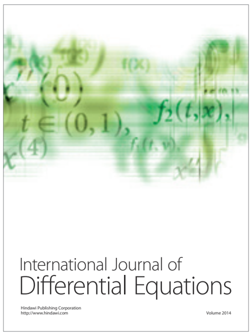

Journal of Applied Mathematics

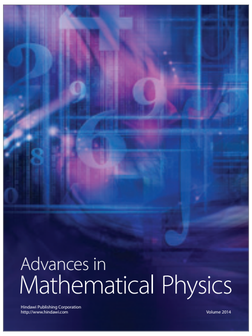

Algebra

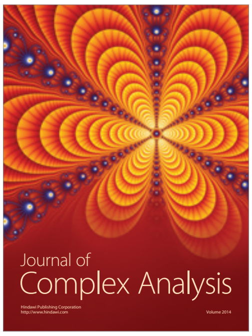

Journal of Probability and Statistics

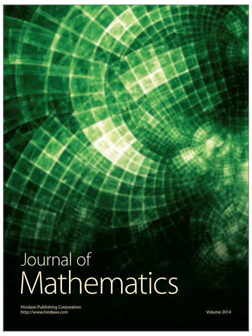
The Scientific World Journal

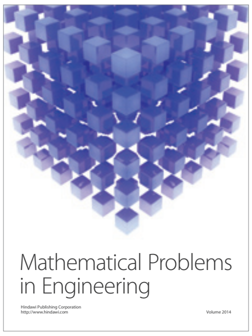
International Journal of Combinatorics

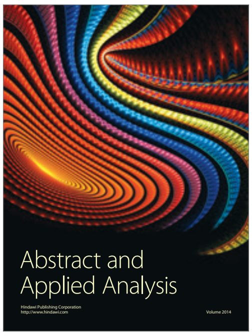

International Journal of Differential Equations

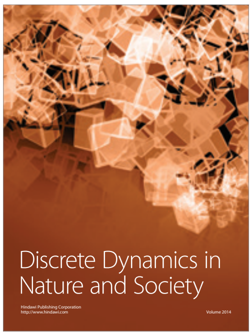

Advances in Mathematical Physics

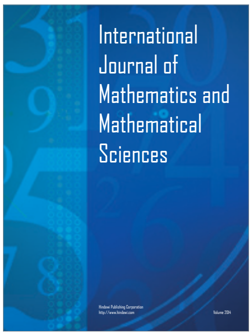

Journal of Complex Analysis


Journal of Mathematics


Mathematical Problems in Engineering

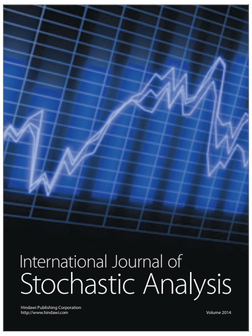

Abstract and Applied Analysis


Discrete Dynamics in Nature and Society


International Journal of Mathematics and Mathematical Sciences

Journal of Discrete Mathematics

Journal of Function Spaces


International Journal of Stochastic Analysis

Journal of Optimization