Certificateless Signature with Strong Unforgeability in the Standard Model

Ying-Hao HUNG, Sen-Shan HUANG, Yuh-Min TSENG*, Tung-Tso TSAI

Department of Mathematics, National Changhua University of Education
Jin-De Campus, Chang-Hua City 500, Taiwan

e-mail: ymtseng@cc.ncue.edu.tw

Received: November 2014; accepted: January 2015

Abstract. Certificateless public-key systems (CL-PKS) were introduced to simultaneously solve two critical problems in public-key systems. One is the key escrow problem in ID-based public-key systems and the other is to eliminate the presence of certificates in conventional public-key systems. In the last decade, several certificateless signature (CLS) schemes have been proposed in the random oracle model. These CLS schemes possess existential unforgeability against adaptive chosen-message attacks, and only few of them possess strong unforgeability. A CLS scheme with strong unforgeability plays an important role in the construction of certificateless cryptographic schemes. Unfortunately, all the existing CLS schemes in the standard model (without random oracles) have been shown insecure to provide existential unforgeability under a generally adopted security model. In the article, we propose a strongly secure CLS scheme in the standard model under the generally adopted security model. Our scheme possesses not only existential unforgeability but also strong unforgeability, and turns out to be the first strongly secure CLS scheme in the standard model. Under the collision resistant hash (CRH) and computational Diffie–Hellman (CDH) assumptions, we prove that our CLS scheme possesses strong unforgeability against both Type I (outsiders) and Type II (key generation center) adversaries.

Key words: certificateless signature, strong unforgeability, random oracle.

1. Introduction

Shamir (1984) proposed a prominent opinion for public-key cryptography, called identity (ID)-based public-key cryptography (ID-PKC), to simplify public-key management. The first practical ID-based cryptographic scheme (i.e., ID-based encryption) with bilinear maps is constructed by Boneh and Franklin (2001). Afterward, the design of ID-based cryptographic mechanisms has undergone quite rapid progress, and enormous literatures have been presented such as (Waters, 2005; Tseng and Tsai, 2012; Tseng et al., 2014; Tsai et al., 2012, 2014b, 2014c). In ID-PKC setting, the public key of a user is the combination of her/his name, e-mail address, social security number, IP address or other identity information while the private key of the user is generated and issued securely by a trusted
third party called private key generator (PKG). ID-PKC eliminates certificate management needed in conventional public-key cryptography. However, ID-PKC suffers from the key escrow problem in the sense that the PKG knows the private key of every user so that the PKG can decrypt ciphertexts or sign messages on behalf of any user.

To resolve the key escrow problem in ID-PKC, Al-Riyami and Paterson (2003) devised a new paradigm called certificateless public-key cryptography (CL-PKC). In CL-PKC setting, a semi-trusted third party, called key generation center (KGC), generates the partial private keys of users. The full private key of a user consists of a partial private key generated by the KGC and a secret key chosen randomly by the user. Meanwhile, the public key of the user is generated by using the secret key, and is published. The KGC has no access to the full private key of any user since the secret key is generated randomly by the user herself/himself. Hence, the key escrow problem is resolved. Subsequently, enormous CL-PKC schemes have been proposed such as certificateless public-key encryption (CL-PKE) (L Libert and Quisquater, 2006; Dent, 2008; Yang and Tan, 2011) and certificateless signature (CLS) (Yum and Lee, 2004; Huang et al., 2005, 2007; Hu et al., 2006; Zhang and Zhang, 2008; He et al., 2012; Tso et al., 2012; Tsai et al., 2014a).

1.1. Related Work

Al-Riyami and Paterson (2003) presented a security model for CL-PKC. The model has two types of adversaries: Type I (outsiders) and Type II (KGC) adversaries. Type I adversary represents a malicious outsider and Type II adversary represents an honest-but-curious KGC. Al-Riyami and Paterson also proposed the first concrete CLS scheme in the random oracle model (Bellare and Rogaway, 1993) but did not present the security notions for CLS schemes. In 2005, Huang et al. (2005) pointed out that Al-Riyami and Paterson’s CLS scheme is insecure against Type I adversary, and presented security notions for CLS schemes. Hu et al. (2006) enhanced the security notions of Huang et al. (2005) to permit adversaries more query capabilities. Since then, Hu et al.’s security model is generally adopted to formalize the security notions for CLS schemes. To improve the performance of signing and verifying, several CLS schemes (Gorantla and Saxena, 2005; Cao et al., 2006; Zhang and Zhang, 2008; Zhang and Mao, 2007) were constructed and analyzed. For reducing communication cost, Huang et al. (2007) proposed a certificateless short signature scheme, but Shim (2009) proved that their scheme is insecure against key replacement attacks. Cheng et al. (2013) wrote a survey article on security models for CLS schemes and presented eight potential security models according to activities and behaviors of adversaries. In particular, strong unforgeability is included in some of the eight potential security models.

The security proofs of these CLS schemes mentioned above must rely on the usage of the random oracle model (Bellare and Rogaway, 1993). However, when random oracles in real implementation are adopted with some hash functions such as SHA-1, these CLS schemes could be insecure. To overcome this problem, Liu et al. (2007) proposed the first CLS scheme without random oracles based on the ID-based signature proposed by
Paterson and Schuldt (2006). Unfortunately, Xiong et al. (2008) pointed out that the Liu et al.’s scheme is insecure against the attacks of Type II adversary, and proposed an improved scheme. In addition, Yuan et al. (2009) also proposed a new CLS scheme in the standard model. Later, Xia et al. (2012) presented that both schemes of Xiong et al. and Yuan et al. are vulnerable to key replacement attacks. Quite recently, Yu et al. (2012) proposed a new CLS scheme in the standard model. However, Cheng et al. (2013) presented that Yu et al.’s scheme is still insecure against both the KGC and the key replacement attacks under the generally adopted security model of Hu et al. (2006).

1.2. Contributions

The CLS schemes in the standard model mentioned above have been shown to be insecure under Hu et al.’s security model. In addition, these schemes did not concern with strongly unforgeable property. A signature scheme is said to be strongly unforgeable (Boneh et al., 2006) if the signature is existentially unforgeable and, given a signature on some message \( m \), an adversary cannot generate a new signature on \( m \). Indeed, CLS schemes with strong unforgeability are important for constructing certificateless cryptographic schemes such as chosen-ciphertext secure certificateless cryptosystems, certificateless signcryption certificateless group signatures and so forth. In the article, we propose a strongly secure CLS scheme in the standard model. Our scheme possesses not only existential unforgeability but also strong unforgeability, while retaining efficiency when compared with previously proposed CLS schemes in the standard model. Our scheme turns out to be the first CLS scheme with strong unforgeability in the standard model. Under the collision resistant hash (CRH) and computational Diffie–Hellman (CDH) assumptions, we prove that our CLS scheme possesses strong unforgeability against both Type I (outsiders) and Type II (KGC) adversaries.

1.3. Organization

The remainder of the article is organized as follows. Preliminaries are given in Section 2. In Section 3, we present the framework and security notions for strongly secure CLS schemes. Our concrete scheme is given in Section 4. In Section 5, we analyze the security of our scheme. Comparisons are presented in Section 6. Conclusions are given in Section 7.

2. Preliminaries

In the section, we briefly review fundamental facts of bilinear pairings and two related security assumptions. A bilinear pairing is a map \( \hat{e} : G_1 \times G_1 \to G_2 \), where \( G_1 \) and \( G_2 \) are two multiplicative cyclic groups of large prime order \( p \). Additionally, a bilinear pairing is an admissible bilinear map if it possesses three properties, namely, bilinear, non-degeneracy and computable (Boneh and Franklin, 2001; Tsai et al., 2014b). In the following, we first present a mathematical problem and its corresponding security assumption.
**Definition 1.** (Computational Diffie–Hellman (CDH) problem and assumption). Given a cyclic multiplicative group $G_1$ of large prime order $p$ with generator $g$ and $g^a, g^b \in G_1$ with unknown $a, b \in \mathbb{Z}_p^*$, the computational Diffie–Hellman (CDH) problem in $G_1$ is to obtain $g^{ab}$. We say that the $(\epsilon, t)$-CDH assumption holds in the group $G_1$ if no probabilistic polynomial-time (PPT) adversary $A$ running in time at most $t$ can solve the CDH problem in $G_1$ with probability at least $\epsilon$. The advantage of $A$ is denoted as $\Pr[A(g, g^a, g^b) = g^{ab}]$, where the advantage is based on the random instances chosen by $A$.

In our scheme, we use collision-resistant hash (CRH) functions to construct our strongly secure CLS scheme, in which the CRH functions can be easily obtained based on the CDH assumption (Boneh et al., 2006). So, the usage of the CRH functions does not strengthen the security assumption of our scheme.

**Definition 2.** (Collision-resistant hashing (CRH) assumption.) Let $H_k : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a family of CRH functions, where $k$ is an index and $n$ is a fixed bit length. We say that the $(\epsilon, t)$-CRH assumption holds if no PPT adversary $A$ running in time at most $t$ can break the collision resistance of $H_k$ with probability at least $\epsilon$. The advantage of $A$ is denoted as $\Pr[A(k) = (m_1, m_2) : m_1 \neq m_2, H_k(m_1) = H_k(m_2)]$.

3. **Framework and Adversarial Model of CLS with Strong Unforgeability**

3.1. **Framework**

We present the framework of CLS schemes with strong unforgeability (or called strongly secure CLS schemes), which is identical to that of the CLS schemes in Hu et al. (2006), Yu et al. (2012), Cheng et al. (2013). A strongly secure CLS scheme consists of two kinds of entities, namely, users and key generation center (KGC). A strongly secure CLS is specified by five algorithms, namely, the system setup, the partial private key extract, the user key generation, the signing and the verifying algorithms.

- **System setup:** On input a security parameter, the KGC runs this algorithm to return the master secret key and public parameters $PP$. $PP$ is available for all the other algorithms.

- **Partial private key extract:** This algorithm, run by the KGC, takes as input the master secret key and a user’s identity ID, and returns the user’s partial private key $D_{ID}$ to the user by way of a secure channel.

- **User key generation:** This algorithm, run by a user, takes as input the user’s identity ID, and outputs the secret key $SK_{ID}$ and the public key $PK_{ID}$. Note that the full private key of a user consists of a partial private key $D_{ID}$ and a secret key $SK_{ID}$.

- **Signing:** This algorithm, run by a user (signer), takes as input the user’s partial private key $D_{ID}$, secret key $SK_{ID}$ and a message $M$, and returns a signature $\sigma$.

- **Verifying:** This algorithm, run by a user (verifier), takes as input a signature $\sigma$, a message $M$, a user identity ID with the public key $PK_{ID}$, the algorithm outputs either “accept” or “reject”.
3.2. Adversarial Model

Based on the security models in Huang et al. (2005), Hu et al. (2006), Cheng et al. (2013), we present the security notions for strongly secure CLS schemes. We present two types of adversaries, namely, Type I and Type II adversaries. A Type I adversary acts as a dishonest user (outsider) who can replace the public key of any entity with a value of her/his choice, but has no access to the master secret key. A Type II adversary represents an honest-but-curious KGC that owns the master secret key, but cannot perform public key replacement. The security notions for strongly secure CLS schemes are modeled using the following games (Games I and II) between a challenger \( B \) and two types of adversaries.

**Game 1 (for Type I Adversary \( A \), Outsider)**

- **Setup.** The challenger \( B \) takes a security parameter \( \psi \) and runs the system setup algorithm to produce the master secret key and public parameters \( PP \). \( PP \) is given to \( A \) and the master secret key is kept by \( B \).
- **Queries.** The Type I adversary \( A \) performs the following queries adaptively:
  - **Public key retrieve (ID).** When \( A \) requests the public key of an entity \( ID \), the challenger \( B \) runs the user key generation algorithm to obtain the public key \( PK_{ID} \) and returns it to \( A \).
  - **Public key replace (ID, PK'_{ID}).** The adversary \( A \) replaces the public key of a user with identity \( ID \) by \( PK'_{ID} \). \( B \) records this replacement.
  - **Partial private key extract (ID).** When \( A \) requests the partial private key of an entity \( ID \), \( B \) runs the partial private key extract algorithm to obtain \( D_{ID} \) and returns it to \( A \).
  - **Secret key extract (ID).** When \( A \) requests the secret key of an entity \( ID \), \( B \) runs the user key generation algorithm to obtain the secret key \( SK_{ID} \) and returns it to \( A \). Here, \( B \) returns the symbol \( \perp \) if the identity \( ID \) has already appeared in the public key replace query.
  - **Signing (ID, M).** When \( A \) requests a signature on the message \( M \) for an entity \( ID \), \( B \) uses the current partial private key \( D_{ID} \) and secret key \( SK_{ID} \) to run the signing algorithm to obtain a signature on the message \( M \). Note that, no matter whether the public key of the identity \( ID \) has not been replaced or not, \( B \) then returns to \( A \).
- **Forgery.** \( A \) generates a signature tuple \((ID^*, M^*, \sigma^*)\). We say that \( A \) win the game if the following conditions holds:
  1. \((ID^*, M^*, \sigma^*)\) can pass the verifying algorithm.
  2. \((ID^*, M^*, \sigma^*)\) has never appeared during the signing query.
  3. \(ID^*\) has never been submitted in the partial private key extract query.

**Game 2 (for Type II Adversary, KGC)**

- **Setup.** The challenger \( B \) takes a security parameter \( \psi \) and runs the system setup algorithm to produce the master secret key and public parameters \( PP \). The master secret key and \( PP \) are given to the adversary \( A \).
4. Strongly Secure CLS Scheme

In this section, we present a concrete CLS scheme with strong unforgeability in the standard model that consists of the following algorithms:

- **Setup:** Given a security parameter \( \psi \), the KGC selects two cyclic groups \( \mathbb{G}_1 \) and \( \mathbb{G}_2 \) of a prime order \( p > 2^{\psi} \). Let \( g \) be a generator of \( \mathbb{G}_1 \) and \( \hat{\phi} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2 \) be an admissible bilinear map. The KGC selects \( \alpha \in \mathbb{Z}_p^* \) and \( g_2 \in \mathbb{G}_1 \) at random, computes \( g_1 = g^\alpha \in \mathbb{G}_1 \), and sets the master secret key as \( g_2^\alpha \). The KGC also selects five collision-resistant hash functions \( H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^m \), \( H_2, H_3 : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \{0, 1\}^\theta \), \( H_4 : \{0, 1\}^* \rightarrow \{0, 1\}^l \) and \( H_5 : \{0, 1\}^* \rightarrow \mathbb{Z}_p^* \), where \( m, n, l \) are fixed lengths. Furthermore, the KGC randomly selects \( u_i, u', s_j, s', t_j, t', w_k \in \mathbb{G}_1 \) and four vectors \( \vec{u} = (u_i), \vec{s} = (s_j), \vec{t} = (t_j) \) and \( \vec{w} = (w_k) \) of length \( m, n, n \) and \( l \), respectively. The KGC publishes \( PP = (\mathbb{G}_1, \mathbb{G}_2, \hat{\phi}, H_1, H_2, H_3, H_4, H_5, g, g_1, g_2, u', \vec{u}, \vec{s}', \vec{t}', \vec{w}') \).

- **Partial private key extract:** Let \( \vec{v} = H_1(ID) = (v_1, v_2, \ldots, v_m) \) be a bit string of length \( m \) representing a user’s identity \( ID \in \{0, 1\}^* \). To construct the user’s partial private key \( D_{ID} \), the KGC selects a random value \( r_v \in \mathbb{Z}_p^* \) and computes \( D_{ID} = (D_1, D_2) = (g_2^{v_1} U^{v_1}, g_2^{v_m}), \) where \( U = u' \prod_{i=1}^m u_i^{v_i} \). The KGC transmits \( D_{ID} \) to the user by way of a secure channel.

- **User key generation:** On input a user’s identity \( ID \), this algorithm randomly chooses two secret values \( \theta_1, \theta_2 \in \mathbb{Z}_p^* \), and computes the user’s public key \( PK_{ID} = (PK_1, PK_2) = (g^\theta_1, g^\theta_2), \) \( \vec{v} = H_2(PK_1, PK_2) = (v_{u_1}, \ldots, v_{u_n}) \) and \( \vec{v} = \)
\(H_3(PK_1, PK_2) = (v_1, \ldots, v_n)\), where \(v_3\) and \(v_7\) are two bit strings of length \(n\). Finally, the user’s secret key is \(SK_{ID} = g_2^{s_1} S^{\theta_1} T^{\theta_2}\), where \(S = s_1 \prod_{j=1}^{\tau} t_j^{\delta_j}\) and \(T = t' \prod_{j=1}^{\tau} t_j^{\delta}j\).

- **Sign:** Given a user’s \(DID\), \(SK_{ID}\) and \(M\), the signer selects a random number \(r_m \in \mathbb{Z}_p\), and computes \(\tilde{v} \tilde{n} = H_4(M) = (v_{m1}, v_{m2}, \ldots, v_{m})\) and \(h = H_5(M || g^{\tilde{m}})\). A signature \(\sigma\) on the message \(M\) is constructed by computing

\[
\sigma = (\sigma_1, \sigma_2, \sigma_3) = (D_1^h(SK_{ID})^h W^{\tilde{m}}, D_2^h, g^{\tilde{m}}),
\]

where \(W = (w_i \prod_{k=1}^{l} w_k^{v_3})\).

- **Verifying:** Given a signature \(\sigma = (\sigma_1, \sigma_2, \sigma_3)\), the message \(M\), identity \(ID\) and its associated public key \(PK_{ID} = (PK_1, PK_2)\), a verifier accepts the signature if the following equality holds:

\[
\hat{\epsilon}(g, \sigma_1) = \hat{\epsilon}(g_1, g_2^h) \hat{\epsilon}(\sigma_2, U) \hat{\epsilon}(PK_1, g_2 S)^h \hat{\epsilon}(PK_2, T)^h \hat{\epsilon}(\sigma_3, W).
\]

**Correctness.** We present that the equality in the verifying algorithm is correct as follows:

\[
\hat{\epsilon}(g, \sigma_1) = \hat{\epsilon}(g, D_1^h(SK_{ID})^h W^{\tilde{m}})
= \hat{\epsilon}(g, g_2^h U^{hr_2} g_2 S^{\theta_1 h} T^{\theta_2 h} W^{\tilde{m}})
= \hat{\epsilon}(g, g_2^h) \hat{\epsilon}(g, U^{hr_2}) \hat{\epsilon}(g, g_2 S^{\theta_1 h}) \hat{\epsilon}(g, T^{\theta_2 h}) \hat{\epsilon}(g, W^{\tilde{m}})
= \hat{\epsilon}(g_1, g_2^h) \hat{\epsilon}(g, hr_2, U) \hat{\epsilon}(g_2 S)^h \hat{\epsilon}(g_2, T)^h \hat{\epsilon}(g^{\tilde{m}}, W)
= \hat{\epsilon}(g_1, g_2^h) \hat{\epsilon}(\sigma_2, U) \hat{\epsilon}(PK_1, g_2 S)^h \hat{\epsilon}(PK_2, T)^h \hat{\epsilon}(\sigma_3, W).
\]

5. **Security Analysis**

In this section, we establish two theorems to prove that our CLS scheme possesses strong unforgeability against adaptive chosen-message attacks under the CRH and CDH assumptions for both Type I (in Game 1) and Type II (in Game 2) adversaries defined in Section 3.

**Theorem 1.** Under the CDH and CRH assumptions, our CLS scheme is strongly secure against Type I adversary. Concretely, suppose that a Type I adversary \(A\) with an advantage \(\epsilon\) can break our CLS scheme within a running time \(\tau\). In the meantime, \(A\) can make at most \(q_E\) partial private key extract queries, \(q_S\) signing queries and \(q_K\) public key replace and secret key extract queries combined. Then there exists an algorithm \(B\) who has an advantage

\[
\epsilon' \geq \frac{\epsilon}{16q_K q_S (q_E + q_S)(m + 1)(n + 1)(l + 1)}
\]

to violate the CDH assumption or a advantage \(\epsilon'' \geq \frac{\epsilon}{\tau} \) to violate the CRH assumption within a running time \(\tau' = \tau + O(m q_E + n q_S + (m + n + l) q_S) \tau_1 + O(q_E + q_S + q_K) \tau_2\).
where $\tau_1$ and $\tau_2$ are the computational costs of a scalar multiplication and an exponentiation in $\mathbb{G}_1$, respectively.

**Proof.** Suppose that a Type I adversary $A$ may forge a valid signature to our CLS scheme, then we can establish an algorithm $B$ to resolve the CDH problem or find a collision pair for the CRH assumption. We assume that $B$ is given an instance of the CDH problem with $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, g, g^\alpha, g^\beta \rangle$. The algorithm $B$ simulates the challenger in Game 1 to respond $A$ as follows.

**Setup.** The challenger $B$ chooses five CRH functions $H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^n$, $H_2, H_3 : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \{0, 1\}^n$, $H_4 : \{0, 1\}^* \rightarrow \{0, 1\}^l$ and $H_5 : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$, where $m, n$ and $l$ are fixed lengths. The adopted CRH functions do not act as random oracles in the following proof. $B$ sets $l_0 = 2(q_E + q_S)$. $l_s = q_E$ and $l_m = 2q_S$, and choose three random integers $k_v, k_s$ and $k_m$, where $0 \leq k_v \leq m$, $0 \leq k_s \leq n$ and $0 \leq k_m \leq l$. For the given values of $q_E, q_S, m, n$ and $l$, the following inequalities $l_0(m + 1) < p, l_s(n + 1) < p$ and $l_m(l + 1) < p$ must hold. $B$ selects the following random integers: $x', x_1, \ldots, x_m \in \mathbb{Z}_p$, $y', y_1, \ldots, y_m \in \mathbb{Z}_p$, $r', r_1, \ldots, r_n \in \mathbb{Z}_t$, $z', z_1, \ldots, z_n \in \mathbb{Z}_p$, $c', c_1, \ldots, c_l \in \mathbb{Z}_m$, and $d', d_1, \ldots, d_l \in \mathbb{Z}_p$. As in our scheme, we have $\vec{v} = H_1(ID) = (v_1, \ldots, v_m)$ for an identity $ID$, $\vec{w} = H_2(PK_1, PK_2) = (w_1, \ldots, w_m)$ and $\vec{w} = H_3(PK_1, PK_2) = (w_1, \ldots, w_m)$ for a public key $PK_0 = (PK_1, PK_2)$, and $\vec{m} = H_4(M) = (m_1, \ldots, m_l)$ for a message $M$. We then construct six functions $F, J, Q, E, K$ and $L$ as follows:

\[
F(\vec{v}) = -l_vk_v + x' + \sum_{i=1}^m v_ix_i, \quad J(\vec{v}) = y' + \sum_{i=1}^m v_iy_i,
\]

\[
Q(\vec{w}) = -l_sk_s + r' + \sum_{j=1}^n ws_jr_j, \quad E(\vec{v}l) = z' + \sum_{j=1}^n vt_je_j,
\]

\[
K(\vec{m}) = -l_mk_m + c' + \sum_{k=1}^l \sum_{j=1}^l vm_kck, \quad L(\vec{m}) = d' + \sum_{k=1}^l \sum_{j=1}^l vm_kdk.
\]

$B$ constructs public parameters $PP$ by computing $g_1 = g^a, g_2 = g^b; u' = g_2^{-l_vk_v + x'}g^{y'}, u_1 = g_2^{s_1}g^{\vec{w}}$ for $1 \leq i \leq m; s_1 = g_2^{-l_vk_v + x'}; s_j = g_2^{v_j}$ for $1 \leq j \leq n; t' = g^{y'}, t_1 = g^{y'}$ for $1 \leq i \leq n; w' = g_2^{-l_mk_m + c'}g^{d'}; w_k = g_2^{s_1}g^{\vec{w}}$ for $1 \leq k \leq l$. $B$ publishes $PP = \langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, \hat{H}_1, \hat{H}_2, \hat{H}_3, \hat{H}_4, g, g_1, g_2, u', \hat{u}, s', \vec{t}', \vec{i}, w', \vec{w} \rangle$. For reducing complexity of the cumbersome notations mentioned above, we also conclude with four relations which will be frequently used in the sequel, namely,

\[
U = u' \prod_{i=1}^m u_i^{v_i} = g_2 F(\vec{v}) \cdot g_2 J(\vec{v}), \quad S = s' \prod_{j=1}^n s_j^{v_j} = g_2 Q(\vec{w})^{-1},
\]

\[
T = t' \prod_{j=1}^l t_j^{v_j} = g_2 E(\vec{v}l), \quad W = u' \prod_{k=1}^l u_k^{v_k} = g_2 K(\vec{m}) \cdot L(\vec{m})^{-1}.
\]
Queries. To avoid collision and consistently respond to queries, \( B \) maintains a list \( L \) of tuples \( (ID, \theta_1, \theta_2, PK_ID, SK_ID) \), which is initially empty. The challenger \( B \) responds to \( A \)’s queries in an adaptive manner as follows:

- **Public key retrieve (ID):** When \( A \) makes this query on \( ID \), the challenger \( B \) does as follows:
  1. If the list \( L \) contains \( ID \), \( B \) returns the corresponding \( PK_{ID} \) to \( A \).
  2. Otherwise, \( B \) chooses two secret values \( \theta_1, \theta_2 \in \mathbb{Z}_p \) and computes the public key \( PK_{ID} = (PK_1, PK_2) = (g^{\theta_1}, g^{\theta_2}), \overline{\theta} = H_2(PK_1, PK_2) = (v_{s1}, \ldots, v_{sn}) \), \( \overline{vt} = H_3(PK_1, PK_2) = (vt_1, \ldots, vt_n) \) and the secret key \( SK_{ID} = g^{\theta_1}H^{\theta_2} \). \( B \) then adds the tuple \( (ID, \theta_1, \theta_2, PK_{ID}, SK_{ID}) \) to \( L \) and returns \( PK_{ID} \) to \( A \).

- **Public key replace (ID, PK’_{ID}):** When \( A \) makes this query on \( ID \), \( B \) looks up \( L \) for the tuple \( (ID, \theta_1, \theta_2, PK_{ID}, SK_{ID}) \). If the list \( L \) contains \( ID \), \( B \) replaces \( PK_{ID} \) with \( PK’_{ID} \). Otherwise, \( B \) adds the tuple \( (ID, \bot, \bot, PK’_{ID}, \bot) \) in \( L \).

- **Partial private key extract (ID):** \( B \) computes \( \overline{v} = H_1(ID), F(\overline{v}) \) and \( J(\overline{v}) \). If \( F(\overline{v}) \neq 0 \mod p \), \( B \) selects a random value \( r_v \in \mathbb{Z}_p \) and responds with the partial private key \( DK_{ID} \) computed by

\[
D_{KD} = (D_1, D_2) = \left( (g^a)^{\frac{\overline{v}}{\overline{m}}}{U}^{r_v}, (g^a)^{\frac{1}{\overline{m}}}{g}^{r_v} \right).
\]

It is convinced that \( D_{KD} \) is a valid partial private key since

\[
D_1 = \left( (g^a)^{\frac{J(\overline{v})}{\overline{m}}} {U}^{r_v} = \left( (g^a)^{\frac{J(\overline{v})}{\overline{m}}} g^{F(\overline{v})} g^{J(\overline{v})}\right)^{r_v} = g_2^a (g_2^{a F(\overline{v})} g^{J(\overline{v})})^{r_v} = \left( g_2^a g^{F(\overline{v})} g^{J(\overline{v})}\right)^{r_v} \right. \]

and

\[
D_2 = \left( (g^a)^{\frac{1}{\overline{m}}} g^{r_v} = g^{r_v - \frac{a}{\overline{m}}} = g^{r_v} \right.
\]

where \( r_v' = r_v - \frac{a}{\overline{m}} \). Otherwise, if \( F(\overline{v}) = 0 \mod p \), \( B \) aborts.

- **Secret key extract (ID):** When \( A \) makes this query on \( ID \), \( B \) looks up \( L \) for the tuple \( (ID, \theta_1, \theta_2, PK_{ID}, SK_{ID}) \). If the list \( L \) contains \( ID \), \( B \) returns \( SK_{ID} \) to \( A \). Otherwise, \( B \) runs the user key generation algorithm to generate \( \theta_1, \theta_2, PK_{ID} \) and \( SK_{ID} \). \( B \) adds the tuple \( (ID, \theta_1, \theta_2, PK_{ID}, SK_{ID}) \) to \( L \) and returns \( SK_{ID} \) to \( A \).

- **Signing (ID, M):** When \( A \) makes this query on \( (ID, M) \), \( B \) first computes \( \overline{v} = H_1(ID), \overline{\theta} = H_2(PK_1, PK_2) = (v_{s1}, v_{s2}, \ldots, v_{sn}), \overline{vt} = H_3(PK_1, PK_2) = (vt_1, vt_2, \ldots, vt_n), \overline{vm} = H_4(ID) = (vm_1, vm_2, \ldots, vm_l) \) and \( h = H_5(M || g'^{\overline{m}}) \). \( B \) then computes \( F(\overline{v}), J(\overline{v}), Q(\overline{\theta}), E(\overline{vt}), K(\overline{vm}) \) and \( L(\overline{vm}) \). If \( K(\overline{vm}) = 0 \mod p \), \( B \) reports failure and terminates. Otherwise, if \( K(\overline{vm}) \neq 0 \), \( B \) considers two cases as follows.
Case 1: Assume that $ID$ has been replaced with $PK_{ID} = (PK_1, PK_2)$. If $F(v) \neq 0 \bmod l_v$, $B$ can compute the partial private key $D_ID = (D_1, D_2)$ as in the partial private key extract query, and $B$ then creates a signature

$$
\sigma = (\sigma_1, \sigma_2, \sigma_3)
= \left( D_1^h (PK_1) - \frac{hL(v)}{k(v,m)} \right) \left( g^{B(v)} \right) \left( g^{hE(v)} \right) \left( g^{L(v,m)} \right)^{r_m} E(v), D_2^h (PK_1) - \frac{hQ(v)}{k(v,m)} g^{r_m}.
$$

Let $r'_m = r_m - \frac{hQ(v)}{k(v,m)}$. Then $\sigma$ is a valid signature since

$$
\begin{align*}
\sigma_1 &= D_1^h \left( g^{B(v)} \right) \left( g^{hE(v)} \right) \left( g^{L(v,m)} \right)^{r_m} E(v), \\
&= D_1^h S_2^h \left( g^{hQ(v)} \right) \left( g^{hE(v)} \right) \left( g^{L(v,m)} \right)^{r_m} E(v), \\
&= D_1^h S_2^h g^{hQ(v)} W^{r_m}, \\
\sigma_2 &= D_2^h, \\
\sigma_3 &= \left( \frac{hL(v)}{k(v,m)} \right) \left( g^{r_m} \right) E(v),
\end{align*}
$$

On the other hand, if $F(v) = 0 \bmod p$, $B$ selects two values $r_v, r_m \in \mathbb{Z}_p^*$ at random and responds with the signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$, where

$$
\begin{align*}
\sigma_1 &= \left( g^{h} \right) \left( g^{hE(v)} \right) \left( g^{L(v,m)} \right)^{r_m} E(v), \\
\sigma_2 &= g^{h r_v}, \\
\sigma_3 &= g^{r_m}.
\end{align*}
$$

Let $r'_m = r_m - \frac{hQ(v)}{k(v,m)}$, it is obvious that $\sigma$ is a valid signature since

$$
\begin{align*}
\sigma_1 &= \left( g^{h} \right) \left( g^{hE(v)} \right) \left( g^{L(v,m)} \right)^{r_m} E(v), \\
&= S_2^h U^{h r_v} \left( g^{hE(v)} \right) \left( g^{L(v,m)} \right)^{r_m} E(v), \\
&= S_2^h U^{h r_v} g^{hQ(v)} W^{r_m}, \\
\sigma_3 &= \left( g^{h} \right) \left( g^{r_m} \right) g^{r_m} = g^{r_m}.
\end{align*}
$$
Case 2: Let us consider the case that $ID$ has not appeared in the public key replace query. If $F(\bar{v}) \neq 0$, $B$ can compute the partial private key $D_{ID} = (D_1, D_2)$ as in the partial key extract query, and accesses the list $L$ to obtain the secret key $SK_{ID}$. $B$ randomly selects a value $r_m \in \mathbb{Z}_p^*$ and returns the signature

$$\sigma = (\sigma_1, \sigma_2, \sigma_3) = (D_1^h(SK_{ID})^h W_{\sigma_m}, D_2^h, g^{\sigma_m}).$$

If $F(\bar{v}) = 0$ mod $p$, then $B$ randomly selects two values $r_v, r_m \in \mathbb{Z}_p^*$ and returns the signature

$$\sigma = (\sigma_1, \sigma_2, \sigma_3) = ((g^a)^{\frac{hL(L(\bar{v})^h)}{k(\bar{v})}}, U_{\sigma_m} g^{br_v}(SK_{ID})^h W_{\sigma_m}, g^{br_v}, (g_1^1)^{\frac{h}{k(\bar{v})}} g^{\sigma_m}).$$

Forgery. Suppose that $A$ forges a valid signature $\sigma^* = (\sigma_1, \sigma_2, \sigma_3)$ for $ID^*$ on $M^*$, where $ID^*$ and $M^*$ are the target identity and message, respectively. We discuss two cases.

Case 1: If $(ID^*, M^*)$ does not appear in the signing query, $B$ accesses the list $L$ to get $PK_{ID^*} = (PK_1^*, PK_2^*)$ and computes $\bar{v}^* = H_1(ID^*), \bar{m}^* = H_2(PK_1^*, PK_2^*), \bar{m} = H_3(\bar{m}^*), h = H_5(M || \sigma_3), F(\bar{v}^*), J(\bar{v}^*), Q(\bar{m}^*), E(\bar{m}^*), L(\bar{m}^*)$ and $K(\bar{v}^*)$. If $F(\bar{v}^*) \neq 0$ mod $p$, $Q(\bar{m}^*) \neq 0$ mod $p$ or $K(\bar{v}^*) \neq 0$, mod $p$, then $B$ aborts. Otherwise, if $F(\bar{v}^*) = Q(\bar{m}^*) = K(\bar{v}^*) = 0$ mod $p$, $B$ computes $g^{ab}$ as follows.

$$\begin{align*}
\sigma_1^{h_1} &= (\sigma_2 J(\bar{v}^*))^{h_1} (PK_2^*)^{(\sigma_3 L(\bar{m}^*))^{h_1}} \\
&= g_2^a U_{\sigma_m} g_2^h S_{\sigma_1} g_2^h T_{\sigma_2} W_{\sigma_m}^{h_1} \\
&= g_2^a (g_2^{J(\bar{v}^*)} g_2^h S_{\sigma_1} g_2^h T_{\sigma_2} W_{\sigma_m}^{h_1}) \\
&= g_2^a (g_2^{J(\bar{v}^*)} g_2^h S_{\sigma_1} g_2^h T_{\sigma_2} W_{\sigma_m}^{h_1}) \\
&= g_2^a = g^{ab}.
\end{align*}$$

This solves the CDH problem.

Case 2: If $(ID^*, M^*)$ has appeared in the signing query, $A$ owned a previously queried signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ of $ID^*$ on $M^*$. If $\sigma_3 \neq \sigma_3^*$, the challenge $B$ can output $g^{ab}$ as in Case 1. Otherwise, if $\sigma_2 = \sigma_2^*$, then $g^{br_v} = g^{hr_v}$ and so $h^* = h$. Namely, $H_5(M || g^{r_v}) = H_5(M || g^{w_m})$, where $\sigma_3^* = g^{w_m}$ and $\sigma_3 = g^{r_m}$. This causes a collision of $H_5$ which violates the CRH assumption.
In the following, we analyze the probabilities of the events that the challenger $B$ does not abort. In the partial key extract query, if $F(\bar{v}) \neq 0 \mod p$, $B$ may respond to queries without aborting. In the signing query, if $K(\bar{v}h) \neq 0 \mod p$, $B$ may respond to queries without aborting. In the forgery phase, if $F(\bar{v}^*) = Q(\bar{v}h^*) = 0 \mod p$, $B$ completes the simulation without aborting. We denote that $q_l$ represents the number of the identities queried in partial private key extract and signing queries not involving $ID^*$. Meanwhile, $q_M$ represents the number of the messages queried in the signing involving $ID^*$. It is obvious that we have $q_1 < q_E + q_S$ and $q_M < q_S$. Here, we define several events as follows: $X_i : F(\bar{v}) \neq 0 \mod l_i$, $X^* : F(\bar{v}^*) = 0 \mod p$, $Y_k : K(\bar{v}h) \neq 0 \mod l_m$, $Y^* : K(\bar{v}h^*) = 0 \mod p$, $Z^* : Q(\bar{v}h^*) = 0 \mod p$, where $1 < i \leq q_l$ and $1 < k \leq q_M$. Hence, the probabilities of $B$ not aborting in Case 1 and Case 2, respectively, are

$$\Pr[\neg \text{abortCase1}] \geq \Pr \left[ \bigwedge_{i=1}^{q_1} X_i \land X^* \land \bigwedge_{k=1}^{q_M} Y_k \land Y^* \land Z^* \right]$$

$$= \Pr[X^*] \cdot \Pr \left[ \bigwedge_{i=1}^{q_1} X_i \land X^* \right] \cdot \Pr[Y^*] \cdot \Pr \left[ \bigwedge_{k=1}^{q_M} Y_k \right].$$

and

$$\Pr[\neg \text{abortCase2}] \geq \Pr \left[ \bigwedge_{i=1}^{q_1} X_i \land \bigwedge_{k=1}^{q_M} Y_k \right] \geq \Pr \left[ \bigwedge_{i=1}^{q_1} X_i \right] \cdot \Pr \left[ \bigwedge_{k=1}^{q_M} Y_k \right].$$

Since $l_v(m+1) < p$, $r_v(n+1) < p$ and $l_m(l+1) < p$, we have that $F(\bar{v}) = 0 \mod p$ implies $F(\bar{v}) = 0 \mod l_v$, $Q(\bar{v}h) = 0 \mod p$ implies $Q(\bar{v}h) = 0 \mod l_v$ and $K(\bar{v}h) = 0 \mod l_m$. Furthermore, $F(\bar{v}) = 0 \mod l_v$, $Q(\bar{v}h) = 0 \mod l_v$ and $K(\bar{v}h) = 0 \mod l_m$, there will be a unique choice of $k_v$ with $0 \leq k_v \leq n$, $k_s$ with $0 \leq k_s \leq n$ and $k_m$ with $0 \leq k_m \leq l$ such that $F(\bar{v}) = 0 \mod p$, $Q(\bar{v}h) = 0 \mod p$ and $K(\bar{v}h) = 0 \mod p$. Since $k_v, x^*, x_1, \ldots, x_m, k_s, r^*, r_1, \ldots, r_n, k_m$ and $c^*, c_1, \ldots, c_l$ are randomly chosen, we have the probabilities of the events $X^*$, $Y^*$ and $Z^*$ as follows.

$$\Pr[X^*] = \Pr[F(\bar{v}^*) = 0 \mod p]$$

$$= \Pr[F(\bar{v}^*) = 0 \mod p \land F(\bar{v}^*) = 0 \mod l_v]$$

$$= \Pr[F(\bar{v}^*) = 0 \mod l_v] \cdot \Pr[F(\bar{v}^*) = 0 \mod p | F(\bar{v}^*) = 0 \mod l_v]$$

$$= \frac{1}{l_v} \frac{1}{m+1}.$$

$$\Pr[Y^*] = \Pr[K(\bar{v}h^*) = 0 \mod p]$$

$$= \Pr[K(\bar{v}h^*) = 0 \mod p \land K(\bar{v}h^*) = 0 \mod l_m]$$

$$= \Pr[K(\bar{v}h^*) = 0 \mod l_m] \cdot \Pr[K(\bar{v}h^*) = 0 \mod p | K(\bar{v}h^*) = 0 \mod l_m]$$

$$= \frac{1}{l_m} \frac{1}{l+1}.$$

$$\Pr[Z^*] = \Pr[Q(\bar{v}h^*) = 0 \mod p] \geq \frac{1}{l_v} \frac{1}{n+1}.$$
We then have that
\[
\Pr\left[ \bigwedge_{i=1}^{q_l} X_i | X^* \right] = 1 - \Pr\left[ \bigvee_{i=1}^{q_l} \neg X_i | X^* \right] \geq 1 - \sum_{i=1}^{q_l} \Pr[\neg X_i | X^*] \\
= 1 - \frac{q_l}{l_v} \geq 1 - \frac{q_E + q_S}{l_v}
\]

and
\[
\Pr\left[ \bigwedge_{k=1}^{q_M} Y_k | Y^* \right] = 1 - \Pr\left[ \bigvee_{k=1}^{q_M} \neg Y_k | Y^* \right] \geq 1 - \sum_{k=1}^{q_M} \Pr[\neg Y_k | Y^*] \\
= 1 - \frac{q_M}{l_m} \geq 1 - \frac{q_S}{l_m}.
\]

We also have
\[
\Pr\left[ \bigwedge_{i=1}^{q_l} X_i \right] = \Pr\left[ \bigwedge_{i=1}^{q_l} X_i | X^* \right] \text{ and } \Pr\left[ \bigwedge_{k=1}^{q_M} Y_k \right] = \Pr\left[ \bigwedge_{k=1}^{q_M} Y_k | Y^* \right].
\]

by independency, hence we can obtain that
\[
\Pr\left[ \bigwedge_{i=1}^{q_l} X_i \wedge X^* \right] = \Pr[X^*] \cdot \Pr\left[ \bigwedge_{i=1}^{q_l} X_i | X^* \right] \geq \left( \frac{1}{l_v} \cdot \frac{1}{m+1} \right) \left( 1 - \frac{q_E + q_S}{l_v} \right)
\]

and
\[
\Pr\left[ \bigwedge_{k=1}^{q_M} Y_k \wedge Y^* \right] = \Pr[Y^*] \cdot \Pr\left[ \bigwedge_{k=1}^{q_M} Y_k | Y^* \right] \geq \left( \frac{1}{l_m} \cdot \frac{1}{l+1} \right) \left( 1 - \frac{q_S}{l_m} \right).
\]

As mentioned earlier, we have set \( l_v = 2(q_E + q_S) \), \( l_s = q_K \) and \( l_m = q_S \). Hence, the probabilities of \( B \) not aborting in Case 1 and Case 2, respectively,
\[
\Pr[\neg \text{abortCase1}] \geq \Pr\left[ \bigwedge_{i=1}^{q_l} X_i \wedge X^* \wedge \bigwedge_{k=1}^{q_M} Y_k \wedge Y^* \wedge Z^* \right] \\
= \Pr[X^*] \cdot \Pr\left[ \bigwedge_{i=1}^{q_l} X_i | X^* \right] \cdot \Pr[Y^*] \cdot \Pr\left[ \bigwedge_{k=1}^{q_M} Y_k | Y^* \right] \cdot \Pr[Z^*] \\
\geq \frac{1}{4(q_E + q_S)(m + 1)4q_S(l + 1)q_K(n + 1)} \\
= \frac{1}{16q_Kq_S(q_E + q_S)(m + 1)(n + 1)(l + 1)}
\]
and

$$\Pr[\neg\text{abortCase2}] \geq \Pr \left[ \bigwedge_{i=1}^{q_1} X_i \wedge \bigwedge_{k=1}^{q_M} Y_k \right] = \Pr \left[ \bigwedge_{i=1}^{q_1} X_i \right] \cdot \Pr \left[ \bigwedge_{k=1}^{q_M} Y_k \right] \geq \frac{1}{4}.$$ 

Hence, if $A$ with an advantage $\epsilon$ can break the proposed CLS scheme, $B$ has an advantage

$$\epsilon' \geq \frac{\epsilon}{16q_K q_S (q_E + q_S)(m + 1)(n + 1)(l + 1)}$$

to violate the CDH assumption or a advantage $\epsilon'' \geq \frac{\epsilon}{4}$ to violate the CRH assumption.

According to the descriptions above, $B$ requires $O(m)$ scalar multiplications and $O(1)$ exponentiations in the private partial key extract queries. In both public key replace and secret key extract queries, $O(n)$ scalar multiplications and $O(1)$ exponentiations are required. In the signing queries, $B$ requires $O(m + n + l)$ scalar multiplications and $O(1)$ exponentiations. So, the total running time required for $B$ is $t' = \tau + O(nq_E + nq_K + (m + n + l)q_S)\tau_1 + O(q_E + q_K + q_S)\tau_2$, where $\tau, \tau_1$ and $\tau_2$ are $A$’s running time, the computational costs of a scalar multiplication and an exponentiation, respectively.

**Theorem 2.** Under the CDH and CRH assumptions, our CLS scheme is strongly secure against Type II adversary. Concretely, suppose that a Type II adversary $A$ with an advantage $\epsilon$ can break our CLS scheme within a running time $\tau$. In the meantime, $A$ can make at most $q_K$ secret key extract queries and $q_S$ signing queries. Then there exists an algorithm $B$ who has an advantage

$$\epsilon' \geq \frac{\epsilon}{4q_S q_K (l + 1)}$$

to violate the CDH assumption or a advantage $\epsilon'' \geq \frac{\epsilon}{4}$ to violate the CRH assumption within a running time $t' = \tau + O(nq_K + (m + n + l)q_S)\tau_1 + O(q_K + q_S)\tau_2$, where $\tau_1$ and $\tau_2$ are the computational costs of a scalar multiplication and an exponentiation in $\mathbb{G}_1$, respectively.

**Proof.** Suppose that a Type II adversary $A$ may forge a valid signature to our CLS scheme, then we can establish an algorithm $B$ to resolve the CDH problem or find a collision pair for the CRH assumption. We assume that $B$ is given an instance of the CDH problem with $(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, g, g^a, g^b)$. The algorithm $B$ simulates the challenger in Game 2 to respond $A$ as follows.

**Setup.** As the Setup phase in Theorem 1, the challenger $B$ first chooses five CRH functions. $B$ then sets $l_m = 2q_S$, and selects a random integer $k_m$, where $0 \leq k_m \leq l$. We suppose that $l_m(l + 1) < p$ for the given values of $q_S$ and $l$. $B$ selects the following random integers: $y', y_1, \ldots, y_m \in \mathbb{Z}_p, r', r_1, \ldots, r_n \in \mathbb{Z}_p, c', z_1, \ldots, z_n \in \mathbb{Z}_p, c_1, \ldots, c_1, c_1, \ldots, c_1 \in \mathbb{Z}_m$, and $d', a_1, \ldots, d_l \in \mathbb{Z}_p$, and the following vectors $\vec{u} = (u_1), \vec{s} = (s_j), \vec{t} = (t_j)$ and
\(\vec{w} = (w_k)\) of the length \(m, n, n, l,\) respectively. Then, as the Setup phase in Theorem 1, \(B\) constructs \(\vec{v} = H_1(ID) = (v_1, \ldots, v_m),\ v^B = H_2(PK_1, PK_2) = (v_1, \ldots, v_n)\) and \(\vec{v} = H_3(PK_1, PK_2) = (v_1, \ldots, v_n),\) and \(\vec{w} = H_4(M) = (v_m, \ldots, v_l)\). We construct five functions \(J, Q, E, K\) and \(L\) as follows:

\[
J(\vec{v}) = y' + \sum_{i=1}^{m} v_i y_i, \quad Q(\vec{v}^B) = t' + \sum_{j=1}^{n} v_s j r_j, \quad E(\vec{v}^B) = z' + \sum_{j=1}^{n} v_t j z_j, \quad K(\vec{w}) = -l_m k_m + c' + \sum_{k=1}^{l} v_m k c_k, \quad L(\vec{v}) = d' + \sum_{k=1}^{l} v_m k d_k.
\]

The challenger \(B\) chooses a random value \(\alpha \in \mathbb{Z}_p\), and sets \(g_1 = g_\alpha^a\) and \(g_2 = g_\beta^b\). Furthermore, \(B\) computes and sends the master secret key \(g_\alpha^a\) to the adversary \(A\). \(B\) constructs public parameters \(PP\) by computing \(g_1 = g_\alpha^a, g_2 = g_\beta^b, u' = g'_\gamma, u_i = g_i^\gamma\) for \(1 \leq i \leq m; s' = g_2^s, s_j = g_j^s\) for \(1 \leq j \leq n; t' = g_\gamma^t, t_j = g_j^t\) for \(1 \leq j \leq n; w' = g_2^{l_m k_m + c'}, g'^d, w_k = g_2^{l_m k_m + c'} g'^d\) for \(1 \leq k \leq l\). \(B\) publishes \(PP = (G_1, G_2, \vec{e}, H_1, H_2, H_3, H_4, g, g_1, g_2, u', \vec{s}, \vec{s}' , \vec{t}', \vec{t}, \vec{w}', \vec{w})\). For the cumbersome notations defined above, as the Setup phase in Theorem 1, we also have four notations \(U, S, T\) and \(W\).

**Queries.** To avoid collision and consistently respond to queries, the challenger \(B\) maintains an initially empty list \(L\) of tuples \((ID, \theta_1, \theta_2, PK_ID, SK_ID)\). Moreover, \(B\) chooses a target identity \(ID\) and a random value \(\theta_2 \in \mathbb{Z}_p^\star, B\) computes the public key \(PK_ID = (g_\alpha^a, g_\beta^b)\). \(B\) adds \((ID, \bot, \theta_2, PK_ID, \bot)\) in \(L\). \(B\) responds to \(A\)'s queries in an adaptive manner as follows:

- **Public key retrieve (ID):** As the public key retrieve query in Theorem 1, \(B\) responds to \(A\)'s queries.

- **Secret key extract (ID):** Upon receiving a query on \(ID, B\) responds to the query as follows:

  1. If \(ID = ID', B\) reports failure and terminates.
  2. If \(ID \neq ID', B\) accesses the tuple \((ID, \theta_1, \theta_2, PK_ID, SK_ID)\) in the list \(L\). If \(ID\) appears in \(L, B\) returns \(SK_ID\) to \(A\). If \(ID\) does not appear in \(L, B\) runs the user key generation algorithm to generate \(\theta_1, \theta_2, PK_ID, SK_ID\) in \(L\) and returns \(SK_ID\) to \(A\).

- **Signing (ID, M):** When \(A\) makes this query on \((ID, M), B\) first computes \(\vec{v} = H_1(ID), \ v^B = H_2(PK_1, PK_2) = (v_1, v_2, \ldots, v_n), \ \vec{v} = H_3(PK_1, PK_2) = (v_1, v_2, \ldots, v_n), \ vec{w} = H_4(ID) = (v_m, v_n, \ldots, v_l)\) and \(h = H_5(M \parallel g^m)\). \(B\) then computes \(J(\vec{v}), Q(\vec{v}^B), E(\vec{v}^B), K(\vec{w})\) and \(L(\vec{w})\). If \(K(\vec{w}) \neq 0 \mod p, B\) considers the following two cases.

**Case 1:** If \(ID = ID', B\) performs the partial private key extract algorithm to obtain the partial private key \(D_{ID'} = (D_1, D_2). B\) randomly selects a value \(r_m \in \mathbb{Z}_p\) and computes the signature.
\[ \sigma = (\sigma_1, \sigma_2, \sigma_3) \]
\[ = \left( D_1^h \left( PK_1 \right) \frac{a_1(1 + Q(1)))}{K(\gamma)} \right) (PK_2)^h E(\gamma) W^{r_m}, \ D_2^h , \ (PK_1) \frac{a_1(1 + Q(1)))}{K(\gamma)} g^{r_m}. \]
Let \( r'_m = r_m - \frac{ab(1 + Q(1)))}{K(\gamma)} \). Then \( \sigma \) is a valid signature since
\[ \sigma_1 = D_1^h \left( g^a \right) \frac{a_1(1 + Q(1)))}{K(\gamma)} \left( g^{E(\gamma)} \right) \left( g_2^h K(\gamma) \right) L_2(\gamma) W^{r_m} \]
\[ = D_1^h \left( g^{a_2(1 + Q(1)))} \right) \left( g^{E(\gamma)} \right) \left( g_2^h K(\gamma) \right) L_2(\gamma) W^{r_m} \]
\[ = D_2^h \left( g^{a_2(1 + Q(1)))} \right) \left( g^{E(\gamma)} \right) \left( g_2^h K(\gamma) \right) L_2(\gamma) W^{r_m} \]
\[ = D_2^h \left( g^{a_2(1 + Q(1)))} \right) \left( g^{E(\gamma)} \right) \left( g_2^h K(\gamma) \right) L_2(\gamma) W^{r_m} \]
\[ \sigma_2 = D_2^h \]
\[ \sigma_3 = (g^a) \frac{a_1(1 + Q(1)))}{K(\gamma)} g^{r_m} = g^{r_m}. \]

**Case 2:** If \( ID \neq ID' \), \( B \) performs the *partial private key extract* algorithm to get the partial private key \( D_{ID} = (D_1, D_2) \), and accesses the list \( L \) to obtain the secret key \( SK_{ID} \). \( B \) randomly selects a value \( r_m \in \mathbb{Z}_p \) and computes the signature
\[ \sigma = (\sigma_1, \sigma_2, \sigma_3) = \left( D_1^h (SK_{ID})^h W^{r_m}, D_2^h, g^{r_m} \right). \]

**Forgery.** Suppose that the adversary \( A \) generates a valid signature \( \sigma^* = (\sigma_1, \sigma_2, \sigma_3) \) for \( ID^* \) on \( M^* \), where \( ID^* \) and \( M^* \) are the target identity and message, respectively. We discuss two cases.

**Case 1:** If \((ID^*, M^*)\) does not appear in the signing query. If \( ID^* \neq ID' \), \( B \) reports failure and terminates. If \( ID^* = ID' \), \( B \) accesses the list \( L \) to obtain \( PK_{ID^*} = (PK_1, PK_2) \), and computes \( \overrightarrow{v} = H_1(ID^*), \overrightarrow{v}^* = H_2(PK_1, PK_2), \overrightarrow{v}^{\prime} = H_3(PK_1, PK_2), \overrightarrow{v}^* = H_4(M^*), \]
\( h = H_5(M \parallel \sigma_3), J(\overrightarrow{v}), Q(\overrightarrow{v}^*) \), \( E(\overrightarrow{v}^*) \), \( L(\overrightarrow{v}^*) \) and \( K(\overrightarrow{v}^*) \). If \( K(\overrightarrow{v}^*) \neq 0 \), \( B \) aborts. If \( K(\overrightarrow{v}^*) = 0 \mod p \), \( B \) computes \((g^{ah})^1 + Q(\overrightarrow{v}^*) \) as follows.
\[ V = \frac{\sigma_1^{h^{-1}}}{g_2^a (g_2^a)^{\frac{1}{h^{-1}}} (PK_2)^{h^{-1}} (\sigma_3 L(\overrightarrow{v}^*))^{h^{-1}}} \]
\[ = g_2^a U^{r_m} g_2^a S^{h^{-1}} W^{r_m h^{-1}} \]
\[ = g_2^a g_2^a J(\overrightarrow{v}) g_2^a E(\overrightarrow{v}) g_2^a L(\overrightarrow{v}^*) h^{-1} \]
\[ = g_2^a (g_2^a)^{h^{-1}} g_2^a (g_2^a)^{\frac{1}{h^{-1}}} (g_2^a)^{h^{-1}} (g_2^a)^{h^{-1}} g_2^a (g_2^a)^{h^{-1}} g_2^a L(\overrightarrow{v}^*) h^{-1} \]
By computing \( V^{(1+Q(\overline{V}^S))^{-1}} \), we obtain the value \( g^{ab} \). This solves the CDH problem.

**Case 2:** If \( (ID^*, M^*) \) has appeared in the \textit{signing} query, \( A \) owned a previously queried signature \( \sigma = (\sigma_1, \sigma_2, \sigma_3) \) of \( ID^* \) on \( M^* \). If \( \sigma_2 \neq \sigma_2^* \), the challenge \( B \) is able to output \( g^{ab} \) as in Case 1. Otherwise, if \( \sigma_2 = \sigma_2^* \), \( g^{hrv} = g^{hrv^*} \) and so \( h^* = h \). Namely, \( H_3(\overline{M} || g^{r+}) = H_3(M || g^{r+}) \), where \( \sigma_3^* = g^{r+} \) and \( \sigma_3 = g^{r+} \). This causes a collision of \( H_3 \) which violates the CRH assumption.

In the following, we analyze the probabilities of the events that the challenger \( B \) does not abort. In the \textit{signing} query, if \( K(\overline{m}) \neq 0 \) mod \( p \), \( B \) may respond to queries without aborting. In the \textit{forgery} phase, if \( ID^* = ID \) and \( K(\overline{m}^*) = 0 \) mod \( p \), \( B \) completes the simulation without aborting. We denote that \( q_M \) represents the number of the messages in \textit{signing} queries not involving \( ID^* \). It is obvious that we have \( q_M < q_S \). Here, we define several events as follows: \( Y_1 : K(\overline{m}) \neq 0 \) mod \( l_m \); \( Y^* : K(\overline{m}^*) = 0 \) mod \( p \); \( Z^* : ID^* = ID \), where \( 1 < k < q_M \). Hence, the probabilities of the challenger \( B \) not aborting in Case 1 and Case 2, respectively, are:

\[
Pr[\neg \text{abortCase1}] \geq Pr\left[ \bigwedge_{k=1}^{q_M} Y_k \land Y^* \land Z^* \right] = Pr[\neg Y^*] \cdot Pr\left[ \bigwedge_{k=1}^{q_M} Y_k | Y^* \right] \cdot Pr[Z^*]
\]

and

\[
Pr[\neg \text{abortCase2}] \geq Pr\left[ \bigwedge_{k=1}^{q_M} Y_k \right].
\]

By the assumption in the setup, we have that \( 0 \leq l_m k_m \leq l_m l < p \) and \( 0 \leq c' + \sum_{k=1}^{q_M} v m_k c_k < l_m (l + 1) < p \). where \( 0 \leq k_m \leq l \) and \( c', c_1, \ldots, c_l \in \mathbb{Z}_{l_m} \). This implies that \( -p \leq l_m k_m + c' + \sum_{k=1}^{q_M} v m_k c_k < p \), i.e. \(-p < K(\overline{m}) < p \). Thus, \( K(\overline{m}) = 0 \) mod \( p \) implies \( k(\overline{m}) = 0 \) mod \( l_m \). With similar to the probability analysis in Theorem 1, we have:

\[
Pr\left[ \bigwedge_{k=1}^{q_M} Y_k \right] = Pr[\neg Y^*] \cdot Pr\left[ \bigwedge_{k=1}^{q_M} Y_k | Y^* \right] \geq \left( \frac{1}{l_m l + 1} \right) \left( 1 - \frac{q_M}{l_m} \right).
\]

By \( l_m = 2q_S \), the probabilities of \( B \) not aborting in Case 1 and Case 2 are, respectively.
Pr[¬abortCase1] ≥ Pr\left[ Y^* \wedge \bigwedge_{k=1}^{\text{qm}} Y_k \wedge Z^* \right]
= Pr[Y^*] \cdot Pr\left[ \bigwedge_{k=1}^{\text{qm}} Y_k | Y^* \right] \cdot Pr[Z^*]
\geq \frac{1}{4qSq_K(l + 1)}

and

Pr[¬abortCase2] ≥ Pr\left[ \bigwedge_{k=1}^{\text{qm}} Y_k \right] \geq \frac{1}{2}.

Hence, if \( A \) with an advantage \( \epsilon \) can break the proposed CLS scheme, \( B \) has an advantage
\[
\epsilon' \geq \frac{\epsilon}{4qSq_K(l + 1)}
\]
to violate the CDH assumption or a probability \( \epsilon'' \geq \frac{1}{2} \) to violate the CRH assumption.

According to the descriptions above, \( B \) requires \( O(n) \) scalar multiplications and \( O(1) \) exponentiations in secret key extract queries. In the signing queries, \( B \) requires \( O(m + n + l) \) scalar multiplications and \( O(1) \) exponentiations. Therefore, the total time required for \( B \) is \( \tau' = \tau + O((nq_K + (m + n + l)q_s)\tau_1 + O(q_K + q_S)\tau_2) \), where \( \tau, \tau_1 \) and \( \tau_2 \) are \( A \)'s running time, the computational costs of a scalar multiplication and an exponentiation, respectively.

\( \Box \)

6. Comparisons

To analyze and compare the computational cost, we consider two time-consuming operations \( T_p \) and \( T_e \), respectively, denote the time of executing a bilinear pairing operation \( \hat{e} : G_1 \times G_1 \rightarrow G_2 \) and the time of executing an exponentiation operation in \( G_1 \) or \( G_2 \). Table 1 lists the comparisons among the schemes of Liu et al. (2007), Yuan et al. (2009), Yu et al. (2012) and ours in terms of computational cost, security assumption and security property. All the signing phases of the schemes above require no pairing operation to sign a message. For the verifying phase, Yu et al. (2012) require five pairing operations but their scheme has been shown insecure against the Type I adversary.

For security analysis, the schemes of Liu et al. (2007), Yuan et al. (2009) and Yu et al. (2012) have been shown insecure against the Type I adversary or Type II adversary. For the security assumption, our scheme is based on the CDH and CRH assumptions, but the others are based on non-pairing-based generalized bilinear Diffie–Hellman (NGBDH) (Liu...
Certificateless Signature with Strong Unforgeability

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational cost</td>
<td>$5T_e$</td>
<td>$9T_e$</td>
<td>$6T_e$</td>
<td>$5T_e$</td>
</tr>
<tr>
<td>for signing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computational cost</td>
<td>$6T_p$</td>
<td>$6T_p$</td>
<td>$5T_p + T_e$</td>
<td>$6T_p + 3T_e$</td>
</tr>
<tr>
<td>for verifying</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Against Type I adversary</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Against Type II adversary</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Security assumption</td>
<td>NGBDH</td>
<td>AC-DH</td>
<td>NGBDH</td>
<td>CDH</td>
</tr>
<tr>
<td>Many-DH</td>
<td></td>
<td>2-Many-DH</td>
<td>Many-DH</td>
<td>CRH</td>
</tr>
<tr>
<td>Security property</td>
<td>Existential</td>
<td>Existential</td>
<td>Existential</td>
<td>Strong</td>
</tr>
<tr>
<td>unforgeability</td>
<td>unforgeability</td>
<td>unforgeability</td>
<td>unforgeability</td>
<td></td>
</tr>
</tbody>
</table>

et al., 2007), many Diffie–Hellman (Many-DH) assumptions (Lysyanskaya, 2002), augmented computational Diffie–Hellman (AC-DH) and 2-many Diffie–Hellman (2-Many-DH) assumptions (Yuan et al., 2009). For Type I and Type II adversaries, we emphasize that our scheme possesses strong unforgeability and is the first secure CLS scheme in the standard model.

7. Conclusions

Strongly secure CLS schemes are important for constructing certificateless cryptographic schemes such as chosen-ciphertext secure certificateless cryptosystems and certificateless group signatures. In this article, we proposed the first strongly secure CLS scheme in the standard model. Comparisons with previously proposed schemes were made to demonstrate the advantages of our scheme in terms of security property while retaining efficiency. For security analysis, under the CDH and CRH assumptions, we demonstrate that the proposed CLS scheme possesses strong unforgeability against adaptive chosen-message attacks under a generally adopted security model.

Acknowledgments. This research was partially supported by the Ministry of Science and Technology, Taiwan, R.O.C., under contract No. MOST103-2221-E-022-MY2.

References

Y.-H. Hung et al.


Y.-H. Hung received the BS degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 1999. He received the MS degree from the Department of Applied Mathematics, National Hsinchu University of Education, Taiwan, in 2008. He is currently a PhD candidate in the Department of Mathematics, National Changhua University of Education, Taiwan. His research interests include applied cryptography and pairing-based cryptography.

S.-S. Huang is currently a Professor in the Department of Mathematics, National Changhua University of Education, Taiwan. His research interests include number theory, cryptography, and network security. He received his PhD from the University of Illinois at Urbana-Champaign in 1997 under the supervision of Professor Bruce C. Berndt.

Y.-M. Tseng is currently a Professor in the Department of Mathematics, National Changhua University of Education, Taiwan. He is a member of IEEE Computer Society, IEEE Communications Society and the Chinese Cryptology and Information Security Association (CCISA). In 2006, his paper received the Wilkes Award from The British Computer Society. He has published over one hundred scientific journal and conference papers on various research areas of cryptography, security and computer network. His research interests include cryptography, network security, computer network and mobile communications. He serves as an editor of several international journals.

T.-T. Tsai is currently a senior engineer in software system develop department of innovation Digital System Business Group (iDSBG) of HON HAL Precision IND. CO., LTD, Taiwan. His research interests include applied cryptography and pairing-based cryptography. He received his PhD from the University of the Department of Mathematics, National Changhua University of Education, Taiwan, in 2014.
Standartinio modelio visiškai nesuklastojamas parašas be sertifikato

Ying-Hao HUNG, Sen-Shan HUANG, Yuh-Min TSENG, Tung-Tso TSAI

Viešojo rakto šifravimo sistemos be sertifikatų (CL-PKS) buvo pasiūlytos dvių svarbių problemų sprendimui: pirmą, pašalinto asmens, kuris žino slaptąją raktą, problema ir, antrą, sertifikavimo eliminavimas įprastojo viešojo raktų sistemose. Pasiūlytas kelios sertifikatų (CLS) schema, kurios yra daliniškai nepažeidžiamos prieš adaptiavusias pasirinktų pranešimų atakas ir tik kai kurios iš jų yra visiškai nepažeidžiamos. Šio straipsnio autoriai siūlo saugią standartinio šifravimo modelio CLS schemą, kuria yra nepažeidžiama prieš adaptiavusias pasirinktų pranešimų atakas. Kai maišos funkcija yra kolizijoms atspari (CRF) ir galioja Diffie–Hellmano prielaidos (CDF), tuomet pasiūlyta schema yra visiškai nesuklastojama esant I tipo (pašalinto asmens) ir II tipo (raktų generavimo centro) grėsmėms.