The Journal of Systems and Software 85 (2012) 1409–1417

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Strongly secure certificateless short signatures

Raylin Tso[a,*], Xinyi Huang[b], Willy Susilo[c]

[a] Computer Science Department, National Chengchi University, Wenshan, Taipei 11605, Taiwan
[b] School of Mathematics and Computer Science, Fujian Normal University, China
[c] Centre for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong, Australia

ARTICLE INFO

Article history:
Received 3 May 2011
Received in revised form 4 October 2011
Accepted 5 January 2012
Available online 30 January 2012

Keywords:
Bilinear pairing
Certificateless signature
Random oracle
Short signature
Strongly secure

ABSTRACT

Short certificateless signatures have come into limelight in recent years. On the one hand, the property of certificateless eliminates the certificate management problem in traditional PKI and the key-escrow problem in some ID-based signature schemes. On the other hand, due to the short signature length, short certificateless signatures can be applied to systems where signatures are typed in by human or systems with low-bandwidth channels and/or low-computation power, such as PDAs or cell phones. However, there has been a trade-off between short certificateless signature schemes and their security levels. All existing short certificateless signature schemes can only be proven secure against a normal type adversary rather than a stronger one, who can obtain valid certificateless signatures under public keys replaced by the adversary. In this paper, we solve this open problem by given an efficient strongly secure short certificateless signature scheme. The proposed scheme has the following features. Firstly, it is strongly unforgeable. Secondly, the security can be reduced to the Computational Diffie–Hellman (CDH) assumption – a classic complexity assumption. Lastly, the proposed scheme is provably secure against adversaries with access to a super signing oracle which generates valid certificateless signatures of messages and public keys chosen by the adversary (without providing the corresponding secret values).

1. Introduction

In traditional public key cryptography (PKC), the authentication of a user's public key before using the key is necessary. It ensures that the public key has not been tampered with or replaced by a malicious third party. In serving this goal, the general approach is to use Public Key Infrastructure (PKI) in which a trusted authority, called Certification Authority (CA), issues certificates to bind users and their public keys. However, the certificate management in traditional PKI (e.g., certificate revocation, storage, distribution and verification) is generally considered to be costly to use and manage.

Identity-based (ID-based) cryptography, which was first introduced by Shamir in 1984, is introduced to overcome the aforementioned problem. In an ID-based cryptosystem, users can use their unique identifiers (e.g., names or e-mail addresses) as their public keys. These public keys are publicly known and do not need certificates to ensure their authenticity. Thus, the problems associated with certificates can be eliminated. However, this kind of ID-PKC has an inherent key escrow issue, namely the Private Key Generator (PKG) (which generates private keys for users) has all users' private keys. This requires the full trust on the PKG.

Certificateless cryptography was proposed to solve the key escrow problem inherent in ID-based cryptography on the one hand and to eliminate the use of certificates in the conventional PKI on the other hand. It was first introduced by Al-Riyami and Paterson in 2003. In a certificateless cryptosystem, the third party, which is called KGC, only generates partial private keys for users while users independently generate their own public/secret key pairs. Cryptographic operations in the system are performed successfully only when both the partial key and the secret key are known. This way, the KGC is unable to obtain secret keys of users and the key escrow problem can be overcome in certificateless public key cryptography. Following Al-Riyami and Paterson's pioneering work (Al-Riyami and Paterson, 2003), many certificateless cryptographic schemes (Au et al., 2007; Gorantla and Saxena, 2005; Huang et al., 2005, 2007; Tso et al., 2008, 2011; Yap et al., 2006; Yum and Lee, 2004; Zhang et al., 2006) have been proposed recently. Since there is no certificate to ensure the authenticity of the user's public key in certificateless public key cryptography, it is necessary to assume that an adversary is able to replace the public key with a false key of its choice, which is also known as *key replacement attack*. One assumption in certificateless public key cryptography is that KGC never mounts the key replacement attack.

* Corresponding author. Tel.: +886 2 2939 3091x62328.
E-mail addresses: raylin@cs.nccu.edu.tw (R. Tso), hyhuang81@gmail.com (X. Huang), wsusilo@uow.edu.au (W. Susilo).

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2012.01.016

Short signatures can provide a high security level with relatively short signature length. As an example, BLS (Boneh et al., 2001) short signature has half the size of a DSA signature for a similar level of security. Short signatures have many potential applications in real life. For example, as mentioned in Bellare and Neven (2006), wireless devices such as PDAs, cell phones, RFID chips and sensors, have a short battery life. Communicating even one bit of data uses significantly more power than executing one 32-bit instruction (Barr and Asanovic, 2003). Thus, reducing the number of bits in communication saves power and is important to increase the battery life. Also, in many settings, communication channels are not reliable, and thus the number of bits one send should be kept as few as possible. Recently, short signatures have been investigated intensively and many short signature schemes have been proposed (Boneh and Boyen, 2004; Tso et al., 2009; Zhang et al., 2003). We believe it is worthwhile to introduce the notion of short signatures into certificateless cryptography, and design certificateless signatures with short signature length.

1.1. Motivations

The first certificateless signature scheme with short signature length was introduced by Huang et al. (2007) in ACISP 2007. This notion has come into limelight in recent years (Du and Wen, 2009; Fan et al., 2009; Tso et al., 2008, 2011). On the one hand, the property of "certificateless" overcomes the certificate management problem in traditional-PKI-based signature schemes and eliminate the key-escrow problem in ID-based signature schemes. On the other hand, due to the short signature length, they can be applied in systems where signatures are typed in by human or systems with low-bandwidth channels, such as PDAs or cell phones. However, it appears that there is a trade-off between certificateless short signature schemes and their security levels. It was first mentioned by Tso et al. in 2011 that all known certificateless short signature schemes are provably secure against the adversary who can replace a user's public key PK with PK', but can only obtain valid signatures under PK instead of PK'. While such adversaries captures the essence of key replacement attacks, they are certainly weaker than other types of adversaries defined in Huang et al. (2007) and Zhang et al. (2006) where key replacement attackers are also allowed to obtain valid signatures under the false public key PK'. It remains as an open problem to construct certificateless short signature schemes with a higher security level than the existing ones (Du and Wen, 2009; Fan et al., 2009; Tso et al., 2008, 2011).

1.2. Our contributions

In this paper, we first define a new type of key replacement attacks against certificateless signatures. We then describe a secure short certificateless signature scheme which has the following features:

- The signature size of our scheme is as short as BLS short signature (Boneh et al., 2001).
- When compared with all other CLS schemes (Du and Wen, 2009; Fan et al., 2009; Huang et al., 2007; Tso et al., 2008, 2011) with short signature sizes, our scheme is the only one with provable security against key replacement attackers who can obtain valid signatures under false public keys. Existing short certificateless signature schemes provide security against key replacement attackers who can obtain valid signatures only under the authentic public key.
- The security of our scheme is based on the classic complexity assumption – Computational Diffie–Hellman (CDH) assumption.
- The proposed scheme is strongly unforgeable (Boneh et al., 2006). A signature system is said to be strongly unforgeable if the signature is existentially unforgeable and, given signatures on some message m, the adversary cannot produce a new signature on m. Short signatures are clearly strongly unforgeable since each message has a unique signature.
- The proposed scheme not only enjoys a high security level, but is also very efficient. Signing a message requires only one exponentiation and one multiplication in a multiplicative cyclic group $\mathbb{G}_1$, and signature verification requires only 2 pairing (on-line) operations.

Organization of this paper.

The rest of this paper is organized as follows. In Section 2, we give some preliminaries required throughout this paper and gives a new type of key replacement attacks against certificateless signatures. Section 3 describes the proposed short certificateless signature scheme, whose security analysis is given in Section 4. Section 5 gives the performance comparison of our scheme with other schemes and the conclusion is given in Section 6.

2. Preliminaries

In this section, we briefly review the definition for groups equipped with a bilinear map and some complexity assumptions which will be used for our security proofs. We will also review the definitions of certificateless signatures and their security models.

2.1. Bilinear groups and complexity assumptions

Let $\mathbb{G}_1, \mathbb{G}_2$ be two multiplicative cyclic groups of order p for some large prime p. Our scheme makes use of the *bilinear* map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ between these two groups. The bilinear map should satisfy the following properties:

1. **Bilinear:** A map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is bilinear if $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$ for all $g, h \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^*$.
2. **Non-degenerate:** The map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$. Observe that since $\mathbb{G}_1, \mathbb{G}_2$ are groups of prime order this implies that if g is a generator of $\mathbb{G}_1$, then $\hat{e}(g, g)$ is a generator of $\mathbb{G}_2$.
3. **Computable:** There is an efficient algorithm to compute $\hat{e}(g, h)$ for any $g, h \in \mathbb{G}_1$.

A bilinear map satisfying the above three properties is said to be an *admissible* bilinear map. We can make this map using the Weil pairing or the Tate pairing (Barreto et al., 2002, 2003; Boneh et al., 2001).

Next, we describe the complexity assumptions which are required for the security proof of our scheme.

Definition 1 *(Computational Diffie–Hellman (CDH) problem).* Let $\mathbb{G}$ be a multiplicative cyclic groups of order p for some large prime p. Let also g be a generator of $\mathbb{G}$, the challenger chooses $a, b \in \mathbb{Z}_p^*$ at random and outputs $(g, A = g^a, B = g^b)$. The adversary then attempts to output $g^{ab} \in \mathbb{G}$. An adversary $\mathcal{A}$ is said to have advantage at least ϵ in solving the CDH problem if

$$Pr[\mathcal{A}(g, g^a, g^b) = g^{ab}] \geq \epsilon$$

where the probability is over the randomly chosen a, b and the random bits consumed by $\mathcal{A}$.

The CDH problem is first introduced by Diffie and Hellman (1976). We say the CDH problem is (t, ϵ)-secure if there is no t-time adversary can have advantage at least ϵ in solving the CDH problem.

Definition 2 *(Inverse Computational Diffie–Hellman (InvCDH) problem).* Let $\mathbb{G}$ be a multiplicative cyclic groups of order p for some

large prime p. Let also g be a generator of $\mathbb{G}$, the challenger chooses $a \in \mathbb{Z}_p^*$ at random and outputs (g, g^a). The adversary then attempts to output $g^{a^{-1}} \in \mathbb{G}$. An adversary $\mathcal{A}$ is said to have advantage at least ϵ in solving the InvCDH problem if

$$Pr[\mathcal{A}(g, g^a) = g^{a^{-1}}] \geq \epsilon$$

where the probability is over the randomly chosen a and the random bits consumed by $\mathcal{A}$.

The InvCDH problem is first studied by Pfitzmann and Sadeghi (2000). We say the InvCDH problem is (t, ϵ)-secure if there is no t-time adversary can have advantage at least ϵ in solving the InvCDH problem.

Bao et al. (2003) showed the equivalence of CDH problem and InvCDH problem. In Theorem 8 of Section 4, we will show the reductions of time and success probability from InvCDH problem to CDH problem.

2.2. Certificateless signatures

Following the definition in Al-Riyami and Paterson (2003), a certificateless signature scheme is specified by seven polynomial-time algorithms: **Setup**, **Partial-Private-Key-Extract**, **Set-Secret-Value**, **Set-Private-Key**, **Set-public-Key**, **Sign** and **Verify**.

Setup. This algorithm takes as input a security parameter 1^k and returns the system parameters *params* and the master secret key *msk*. Usually, this algorithm is run by the third party KGC. We assume throughout this paper that *params* are publicly and authentically available, but only the KGC knows *msk*.
Partial-Private-Key-Extract. This algorithm takes the system parameter *params*, the master secret key *msk* and an identity *ID* as input. It returns a partial private key D_{ID}. Usually, this algorithm is run by the KGC and its output is sent to the user with identity *ID* over a confidential and authentic channel.
Set-Secret-Value. This algorithm takes as input the system parameter *params* and an identity *ID*, and outputs a secret value x_{ID}. This algorithm is run by the identity *ID* for itself.
Set-Private-Key. This algorithm takes the system parameter *params*, a partial private key D_{ID} and a secret value x_{ID} of an identity *ID* as input, and outputs the (full) private key SK_{ID}. This algorithm is run by the identity *ID* for itself.
Set-Public-Key. This algorithm takes the system parameter *params*, an identity *ID* and the identity's private key PK_{ID} as input, and outputs the public key PK_{ID} for the identity *ID*.
Sign. This algorithm takes the system parameter *params*, an identity *ID*, the private key SK_{ID} of *ID* and a message *M* as input, and outputs a certificateless signature σ.
Verify. This algorithm takes the system parameter *params*, an identity *ID*, the identity's public key PK_{ID} and a message/signature pair (M, σ) as input, and outputs *true* if the signature is correct, or *false* otherwise.

2.3. Security models

In this section, we model the behavior of the potential adversaries against certificateless signatures, and provide the definition of the security for a certificateless signature scheme secure against such adversaries.

As defined in Al-Riyami and Paterson (2003) and Zhang et al. (2006), there are two types of adversaries with different capabilities:

Type *I* adversary: This type of adversary $\mathcal{A}_I$ models a dishonest user who does not have access to the master key *msk* but has the ability to replace the public key of any entity with a value of his choice.
Type *II* adversary: This type of adversary $\mathcal{A}_{II}$ models a malicious KGC who has access to the master key *msk* but cannot perform public key replacement.

Generally, there are five oracles which can be accessed by adversaries according to game specifications (which will be given shortly).

1. **Create-User:** On input an identity $ID \in \{0, 1\}^*$, if *ID* has already been created, nothing is to be carried out. Otherwise, the oracle runs the algorithms **Private-Key-Extract**, **Set-Secret-Value**, **Set-Public-Key** to obtain the partial private key D_{ID}, secret value x_{ID} and public key PK_{ID}. In this case, *ID* is said to be created. In both cases, PK_{ID} is returned.
2. **Public-Key-Replace:** On input an identity *ID* and a user public key PK'_{ID}, the original user public key of *ID* is replaced with PK'_{ID} if *ID* has been created. Otherwise, no action will be taken.
3. **Secret-Value-Extract:** On input an identity, it returns the corresponding user secret key x_{ID} if *ID* has been created. Otherwise, returns a symbol $\perp$. Note that x_{ID} is the secret value associated with the original public key PK_{ID}. This oracle does not output the secret value associated with the replaced public key PK'_{ID}.
4. **Partial-Private-Key-Extract:** On input an identity *ID*, it returns the partial private key D_{ID} if *ID* has been created. Otherwise, returns a symbol $\perp$.
5. $Sign_{Super}$**:** This oracle takes as input a query (ID, m, PK_{ID}), where *ID* denotes an identity, *m* denotes the message to be signed, and PK_{ID} is the public key which could be the original public key returned from the oracle **Create-User** or a value chosen by the adversary in the public key space.
   - If *ID* has not been created, returns $\perp$.
   - If *ID* has been created, it outputs a signature σ such that *true* = **Verify**$(params, ID, PK_{ID}, m, \sigma)$.

The standard notion of security for a signature scheme is existential unforgeability against adaptive chosen message attacks defined by Goldwasser et al. (1988). We use two games to define the existential unforgeability of a certificateless signature scheme against Type *I* adversary $\mathcal{A}_I$ and Type *II* adversary $\mathcal{A}_{II}$.

Game 1: This game is executed between a challenger $\mathcal{C}$ and an adaptive chosen message and chosen identity adversary $\mathcal{A}_I$.

Setup: The challenger $\mathcal{C}$ runs the algorithm **Setup** of the certificateless signature scheme to obtain both the public parameter *params* and the master secret key *msk*. The adversary $\mathcal{A}_I$ is given *params* but the master secret key *msk* is kept secret by the challenger.
Queries: $\mathcal{A}_I$ adaptively access all oracles defined in Section 2.3 in a polynomial number times.
Forgery: Eventually, $\mathcal{A}_I$ outputs a forgery $(ID^*, PK_{ID^*}, m^*, \sigma^*)$ and wins the game if
1. $true =$ **Verify**$(params, ID^*, PK_{ID^*}, m^*, \sigma^*)$.
2. (ID^*, m^*) has never been submitted to the oracle $Sign_{Super}$.
3. ID^* has never been submitted to the oracle **Partial-Private-Key-Extract** or **Secret-Value-Extract**.

Definition 3. Let $Adv_{\mathcal{A}_I}$ to be the probability of a Type *I* adaptively chosen message and chosen identity adversary $\mathcal{A}_I$ wins in the above game, taken over the coin tosses made by $\mathcal{A}_I$ and the challenger. We say a certificateless signature scheme is secure against Type *I* adversary, if, for all probabilistic polynomial-time (PPT) adversary $\mathcal{A}_I$, the success probability $Adv_{\mathcal{A}_I}$ is negligible.

Game 2: This game is executed between a challenger $\mathcal{C}$ and an adaptive chosen message and chosen identity adversary $\mathcal{A}_{II}$.

Setup: The challenger $\mathcal{C}$ runs the algorithm **Setup** of the certificateless signature scheme to obtain both the public parameter *params* and the master secret key *msk*. The adversary $\mathcal{A}_{II}$ is given both *params* and *msk*.
Queries: $\mathcal{A}_{II}$ adaptively access all oracles defined in Section 2.3 (except the Partial-Private-Key-Extract oracle) in a polynomial number times.
Forgery: Eventually, $\mathcal{A}_{II}$ outputs a forgery $(ID^*, PK_{ID^*}, m^*, \sigma^*)$ and wins the game if the following hold true:
1. $true \leftarrow \text{Verify}(params, ID^*, PK_{ID^*}, m^*, \sigma^*)$.
2. (ID^*, m^*) has never been queried to the oracle $Sign_{Super}$.
3. ID^* has never been submitted to the oracle **Secret-Value-Extract**.

Definition 4. Let $Adv_{\mathcal{A}_{II}}$ be the probability of a Type *II* adaptively chosen message and chosen identity adversary $\mathcal{A}_{II}$ wins in the above game, taken over the coin tosses made by $\mathcal{A}_{II}$ and the challenger. We say a certificateless signature scheme is secure against Type *II* adversary, if for all probabilistic polynomial-time (PPT) adversary $\mathcal{A}_{II}$, the success probability $Adv_{\mathcal{A}_{II}}$ is negligible.

Definition 5. A certificateless signature scheme is existentially unforgeable against adaptive chosen message and chosen identity attacks if it is secure against both Type *I* and Type *II* adversaries.

2.3.1. A comparison with other security models

As one can see, adversaries defined above are very similar to the *super adversaries* defined in Huang et al. (2007), namely adversaries are allowed to obtain valid signatures under any public keys chosen by itself in the public key space. The only difference is that adversaries in our model cannot request the secret value of the challenge user, while this is allowed in the definition in Huang et al. (2007). The super adversaries defined in Huang et al. (2007) are very powerful. However, it may not be a realistic model since it is usually very difficult for an adversary to obtain the secret value of a user. This is because that the secret value and the partial private key of a user are the most two important information for the user so they are usually kept secret in a very secure way. On the other hand, although our new defined adversary model is not as powerful as that in Huang et al. (2007), it is more powerful than those considered in the existing short certificateless signature schemes, where adversaries cannot request the secret value of the challenge user and can only access the following signing oracle:

- $Sign_{Normal}$: This oracle takes as input a query (ID, m), where ID denotes the identity which has been created and m denotes the message. It outputs a signature σ such that $true = \textbf{Verify}(params, ID, PK_{ID}, m, \sigma)$. Here PK_{ID} is the public key returned from the oracle **Create-User**.

Based on social engineering, an adversary may be possible to see some message/signature pairs which are valid under the public key chosen by the adversary. Therefore, our new defined model is a more realistic model than that defined in Huang et al. (2007) and those considered in the existing short certificateless signature schemes. Since all the existing certificateless short signature schemes can only be proven secure against normal adversaries who can only access the signing oracle $Sign_{Normal}$, it is worthwhile to have a new short certificateless signature scheme which is proven secure under a more powerful and realistic adversary model in which an adversary is allowed to access the signing oracle $Sign_{Super}$.

3. Proposed short certificateless signature scheme

In this section, we describe a new construction of certificateless signatures with short signature length. It consists of the following algorithms:

Setup: Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups of a prime order $p \geq 2^k$, where k is the security parameter of the scheme. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be an admissible bilinear pairing. Let $H_0 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$, $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ be two secure cryptographic hash functions. In system initialization, KGC chooses a random number $s \in \mathbb{Z}_p^*$ and an arbitrary generator $g \in \mathbb{G}_1$. It then calculates $P_{pub} = g^s$, and publishes $params = \{\mathbb{G}_1, \mathbb{G}_2, g, \hat{e}, H_0, H_1, P_{pub}\}$. The master secret key $msk = s$ is kept as secret by KGC.
Partial-Private-Key-Extract: Given an entity's identity $ID \in \{0,1\}^*$, KGC sets $Q_{ID} = H_0(ID)$ and computes the entity's partial private key $D_{ID} = Q_{ID}^s$. The partial private key D_{ID} is sent to ID over a confidential and authentic channel.
Set-Secret-Value: The entity ID chooses a random number $x_{ID} \in \mathbb{Z}_p^*$, which is set as the secret value of ID.
Set-Private-Key: The entity ID sets his private key as $SK_{ID} = (D_{ID}, x_{ID})$.
Set-Public-Key: Given SK_{ID}, the entity ID computes the public key $PK_{ID} = (PK_{(ID,1)}, PK_{(ID,2)}) = (D_{ID}^{x_{ID}}, Q_{ID}^{x_{ID}})$.
Sign: To sign a message $m \in \{0,1\}^*$, the entity ID computes the signature $\sigma = D_{ID} \cdot H_1(m||ID||PK_{ID})^{x_{ID}^{-1}}$.
Verify: Given a message/signature pair (m, σ) and ID's public key PK_{ID}, the signature σ is verified by checking the following two equations.

$$\hat{e}(P_{pub}, PK_{(ID,2)}) \stackrel{?}{=} \hat{e}(g, PK_{(ID,1)}) \quad \text{and}$$

$$\hat{e}(\sigma, PK_{(ID,2)}) \stackrel{?}{=} \hat{e}(PK_{(ID,1)} \cdot H_1(m||ID||PK_{ID}), Q_{ID}).$$

If both equations are satisfied, this algorithm outputs *true*. Otherwise, it outputs *false*.
Correctness: If the public key PK_{ID} and the signature σ are generated correctly according to the protocol specifications, then the correctness holds since

$$\hat{e}(P_{pub}, PK_{(ID,2)}) = \hat{e}(g, Q_{ID}^{x_{ID}})^s = \hat{e}(g, D_{ID}^{x_{ID}}) = \hat{e}(g, PK_{(ID,1)}) \quad \text{and}$$

$$\begin{aligned}\hat{e}(\sigma, PK_{(ID,2)}) &= \hat{e}(D_{ID} \cdot H_1(m||ID||PK_{ID})^{x_{ID}^{-1}}, Q_{ID}^{x_{ID}}) \\ &= \hat{e}(D_{ID} \cdot H_1(m||ID||PK_{ID})^{x_{ID}^{-1}}, Q_{ID})^{x_{ID}} \\ &= \hat{e}(D_{ID}^{x_{ID}} \cdot H_1(m||ID||PK_{ID})^{x_{ID}^{-1} x_{ID}}, Q_{ID}) \\ &= \hat{e}(PK_{(ID,1)} \cdot H_1(m||ID||PK_{ID}), Q_{ID}).\end{aligned}$$

3.1. Further observations on our scheme

As one can see, there are two equations in the verification phase. The first equation verifies the validity of the public key $PK_{ID} = (PK_{(ID,1)}, PK_{(ID,2)})$ and the second equation verifies the correctness of the signature σ. The current scheme is well designed as the adversary cannot obtain any advantage by replacing the public key. So, these two equations together can ensure the security of our scheme.

Our proposed scheme belongs to conventional certificateless signatures, which can only achieve Girault's Level-2 security (Girault, 1991; Hu et al., 2007). That is, KGC must be semi-trusted since it can always impersonate a user by generating a valid public-key/secret-value pair. Girault's Level-3 security guarantees that without KGC's assistance, there are no two valid public keys. In

other words, the co-existence of two valid public keys is the proof of the misbehavior of KGC. Based on the idea proposed by Wu et al. in 2009, one can always modify any certificateless signature into a certificate-based signature in order to achieve the Girault's Level-3 security. However, in a certificate-based signature scheme, Girault's Level-3 security is achieved at the sacrifice of a remarkable feature of certificateless cryptosystems. That is, in a certificateless cryptosystem, the public key PK_{ID} of an entity ID can be updated at any time by the entity without changing the partial private key. In other words, one can change the public key by himself/herself, and in particular this process does not need any assistance from KGC. This is the most difference between a certificateless public key cryptography with a certificate-based public key cryptography.

4. Security proofs

In this section, we prove that our scheme is secure against Type *I* and Type *II* adversaries defined in Section 2.3. In our model, an adversary is able to access the **Super Signing Oracle**, $Sign_{Super}$, by which the adversary can obtain valid signatures under the replaced keys without providing the corresponding secret value. This oracle gives adversaries more attack power than those considered in the existing short certificateless signature schemes.

Theorem 6 (Unforgeability against Type *I* adversary). *If there exists a Type I adaptively chosen message and chosen ID adversary $\mathcal{A}_I$ who can ask at most q_C* **Create-User** *queries, q_{KEx}* **Partial-Private-Key-Extract** *queries, q_{VEx}* **Secret-Value-Extract** *queries and q_S super signing queries, $Sign_{Super}$, and can break the proposed scheme in polynomial time with success probability ε, then there exists an algorithm $\mathcal{F}$ which, by using $\mathcal{A}_I$ as a black box, can solve the CDH problem (Definition 1) with probability $Adv_{\mathcal{F}}^{CDH} \geq (1-(1/q_C))^{q_{PKEx}+q_{VEx}}(1-(1/(q_S+1)))^{q_S}(1/(q_C(q_S+1)))\varepsilon$.*

Proof. If there exists an adversary $\mathcal{A}_I$ who can break the unforgeability of the proposed scheme via Type *I* attacks defined in Section 2.3, then we can construct an algorithm $\mathcal{F}$ such that $\mathcal{F}$ can use $\mathcal{A}_I$ as a black-box and solve a random instance of the CDH problem.

Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear groups with bilinear mapping $\hat{e}$. Let g be a generator of $\mathbb{G}_1$ and let a, b be two random numbers of $\mathbb{Z}_p^*$ chosen by the CDH challenger $\mathcal{C}$. $\mathcal{C}$ computes $g_1 = g^a$, $g_2 = g^b$ and sends the challenge (g, g_1, g_2) to $\mathcal{F}$. The purpose of $\mathcal{F}$ is to find g^{ab}, which is the solution to the given instance of CDH problem.

In the proof, $\mathcal{F}$ will act as the challenger of $\mathcal{A}_I$ by simulating the oracles defined in Section 2.3. The hash functions H_0, H_1 in the proof are regarded as random oracles. Due to the ideal randomness of random oracles, we may assume that $\mathcal{A}_I$ is well-behaved in the sense that it always makes an H_1-hash query of $m||ID||PK_{ID}$ before it requests a signature for m under ID's public key PK_{ID}. In addition, it always makes an H_1-hash query of $m^*||ID^*||PK_{ID}^*$ before it outputs the forgery of m^*. It is trivial to modify any adversary-algorithm $\mathcal{A}_I$ to satisfy this property.

Setup: $\mathcal{F}$ starts by computing $P_{pub} = g_1 = g^a$. $\mathcal{F}$ then sends $params = (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, g, P_{pub})$ to $\mathcal{A}_I$.

Query: At any time, $\mathcal{A}_I$ is allowed to access the following oracles in a polynomial number of times. These oracles are all simulated by $\mathcal{F}$.

1. **Create-User:** $\mathcal{A}_I$ can query this oracle by providing an identity ID_i. In response to these queries, $\mathcal{F}$ first chooses a random number $t \in \{1, \ldots, q_C\}$.
   (1) If $i \neq t$, $\mathcal{F}$ chooses $k_i, k_i' \in_R \mathbb{Z}_p^*$ and sets $H_0(ID_i) = g^{k_i'}$, $PK_{(ID_i,1)} = g_1^{k_i k_i'}$, $PK_{(ID_i,2)} = g^{k_i k_i'}$. In this case, the corresponding partial private key of the entity ID_i is $D_{ID_i} = H_0(ID_i)^a = g^{ak_i'} = g_1^{k_i'}$ and the secret value is $x_{ID_i} = k_i$.
   (2) If $i = t$, $\mathcal{F}$ chooses $k_t, k_t' \in_R \mathbb{Z}_p^*$ and sets $H_0(ID_t) = g_2 \cdot g^{k_t'}$, $PK_{(ID_t,1)} = g_1^{k_t}$, $PK_{(ID_t,2)} = g^{k_t}$. In this case, $\mathcal{F}$ will set $D_{ID_t} = x_{ID_t} = \perp$ which means that it cannot compute the secret value and the partial private key of ID_t.
   In both cases, the tuple $(ID_i, H_0(ID_i), D_{ID_i}, x_{ID_i}, PK_{ID_i})$ will be added on the *C-List* which is initially empty. The adversary $\mathcal{A}_I$ is given the response $H_0(ID_i)$ and $PK_{ID_i} = (PK_{(ID_i,1)}, PK_{(ID_i,2)})$.
2. **Partial-Private-Key-Extract:** At any time, $\mathcal{A}_I$ can query this oracle by giving an identity ID_i. $\mathcal{F}$ outputs a symbol $\perp$ if ID_i has not been created. Else, if ID_i has been created and $i \neq t$, $\mathcal{F}$ responds with $D_{ID_i} = g_1^{k_i'}$. Otherwise, $\mathcal{F}$ returns *failure* and terminates the simulation.
3. **Public-Key-Replace:** $\mathcal{A}_I$ can request to replace the public key PK_{ID_i} of an entity ID_i with a new public key PK'_{ID_i} chosen by $\mathcal{A}_I$. In response, $\mathcal{F}$ replaces the original public key PK_{ID_i} with PK'_{ID_i} if ID_i has been created. Otherwise, it outputs $\perp$. Notice that the replacement of a public key does not require the secret value corresponding to the new public key.
4. **Secret-Value-Extract:** Given an identity ID_i chosen by $\mathcal{A}_I$, $\mathcal{F}$ outputs $\perp$ if ID_i has not been created yet. Else, if ID_i has been created and $i \neq t$, $\mathcal{F}$ sends $x_{ID_i} = k_i$ to $\mathcal{A}_I$ as the response. Otherwise, $i = t$ and $\mathcal{F}$ reports *failure* and terminates the simulation.
5. H_1 **queries:** $\mathcal{A}_I$ can query the random oracle H_1 at any time by making a request ω_i of the form $\omega_i = m_j||ID_k||PK'_{ID_k}$. $\mathcal{F}$ returns $\perp$ if ID_k has not been created. Otherwise, for the ith H_1 query
   - If $ID_k \neq ID_t$, $\mathcal{F}$ first checks the *C-List*.
     (1) If $(ID_k, , , , PK'_{ID_k})$ is in the *C-List*, $\mathcal{F}$ picks a random element R_i and sets $H_1(\omega_i) = R_i$.
     (2) Otherwise, $(ID_k, , , , PK'_{ID_k})$ is not in the *C-List*, which means that PK'_{ID_k} is generated by $\mathcal{A}_I$ itself. $\mathcal{F}$ picks $\ell_i \in \mathbb{Z}_p^*$ randomly and sets $H_1(\omega_i) = PK'^{\ell_i}_{(ID_k,2)}/PK'_{(ID_k,1)}$. Here, $PK'_{ID_k} = (PK'_{(ID_k,1)}, PK'_{(ID_k,2)})$.
   - Otherwise, $ID_k = ID_t$, $\mathcal{F}$ first flips a biased-coin which outputs $\zeta_i \in \{0, 1\}$ with $Pr[\zeta_i = 1] = \xi$ and $Pr[\zeta_i = 0] = 1 - \xi$ (the value of ξ will be optimized later).
     (1) If $\zeta_i = 1$, $\mathcal{F}$ picks a random $\ell_i \in \mathbb{Z}_p^*$ and sets $H_1(\omega_i) = PK'^{\ell_i}_{(ID_t,2)}$.
     (2) If $\zeta_i = 0$, $\mathcal{F}$ picks a random $\ell_i \in \mathbb{Z}_p^*$ and sets $H_1(\omega_i) = PK'^{\ell_i}_{(ID_t,2)}/PK'_{(ID_t,1)}$.
   In both cases, $\mathcal{F}$ sends $H_1(\omega_i)$ to $\mathcal{A}_I$ and adds $(m_j, \ell_i, \omega_i, H_1(\omega_i))$ (in the case $ID_k \neq ID_t$) or $(m_j, \ell_i, \omega_i, H_1(\omega_i), \zeta_i)$ (in the case $ID_k = ID_t$) on the H_1-*List* which is initially empty.
6. $Sign_{Super}$**:** For each sign query on an input (m_j, ID_k, PK'_{ID_k}), $\mathcal{F}$ outputs $\perp$ if ID_k has not been created. Otherwise, ID_k has already been created. Since we assume that $\mathcal{A}_I$ is well-behaved, $\mathcal{A}_I$ has already queried the random oracle H_1 on the input $\omega_i = (m_j||ID_k||PK'_{ID_k})$.
   - If $ID_k \neq ID_t$, and
     (1) $(ID_k, H_0(ID_k), D_{ID_k}, x_{ID_k}, PK'_{ID_k})$ is in the *C-List*, $\mathcal{F}$ uses the private key (x_{ID_k}, D_{ID_K}) of ID_k and $H_1(\omega_i)$ on the H_1-*List* to generate the valid signature σ_i for the message m_j and the identity ID_k.
     (2) $(ID_k, , , , PK'_{ID_k})$ is not in the *C-List*, which means that PK'_{ID_k} is generated by $\mathcal{A}_I$ itself, $\mathcal{F}$ sets $\sigma_i = H_0(ID_k)^{\ell_i}$. Note that σ_i is a valid signature on m_j and $PK'_{ID_k} = (PK'_{(ID_k,1)}, PK'_{(ID_k,2)})$ if PK'_{ID_k} is well-generated (which means that $PK'_{(ID_k,1)} = PK'^{a}_{(ID_k,2)}$ as they satisfy the equation $\hat{e}(PK'_{(ID_k,1)}, g) = \hat{e}(PK'_{(ID_k,2)}, P_{pub})$). Assume $PK'_{(ID_k,2)} =$

$H_0(ID_k)^r$ for some unknown r. The correctness of σ_i can be verified according to the following equation:

$$\sigma_i = D_{ID_k} \cdot H_1(\omega_i)^{r^{-1}} = H_0(ID_k)^a \cdot H_1(\omega_i)^{r^{-1}}$$
$$= (H_0(ID_k)^{ar} \cdot H_1(\omega_i))^{r^{-1}} = (PK'^{a}_{(ID_k,2)} \cdot H_1(\omega_i))^{r^{-1}}$$
$$= (PK'_{(ID_k,1)} \cdot H_1(\omega_i))^{r^{-1}} = \left(\frac{PK'_{(ID_k,1)} \cdot PK'^{\ell_i}_{(ID_k,2)}}{PK'_{(ID_k,1)}}\right)^{r^{-1}}$$
$$= PK'^{\ell_i r^{-1}}_{(ID_k,2)} = H_0(ID_k)^{r\ell_i r^{-1}} = H_0(ID_k)^{\ell_i}.$$

In both cases, $\mathcal{F}$ returns σ_i to $\mathcal{A}_I$.

- If $ID_i = ID_t$, then, $\mathcal{F}$ first checks the H_1-*List*.
  (1) If $\zeta_i = 1$, $\mathcal{F}$ reports *failure* and terminates the simulation.
  (2) Otherwise, $\zeta_i = 0$ and $\mathcal{F}$ sets $\sigma_i = H_0(ID_t)^{\ell_i}$ and returns σ_i to $\mathcal{F}$. Here ℓ_i is the value on the H_1-*List*. The correctness of σ_i can be verified in the same way as described above.

Forgery: After all queries, $\mathcal{A}_I$ outputs a forgery $(ID^*, PK^*_{ID}, m^*, \sigma^*)$ and wins the game if σ^* is a valid forgery. By assumption, $H_1(m^*||ID^*||PK^*_{ID})$ is on the H_1-*List*, where PK^*_{ID} could be a new public key replaced by $\mathcal{A}_I$ or the original public key generated by the oracle **Create-User**. If $ID^* \neq ID_t$, then $\mathcal{F}$ outputs *failure* and terminates the simulation. Otherwise, $ID^* = ID_t$ and $\mathcal{F}$ will check the H_1-*List*.

(1) If $\zeta^* = 0$, $\mathcal{F}$ outputs *failure* and terminates the simulation.
(2) Otherwise, $\zeta^* = 1$ and $H_1(m^*||ID^*||PK^*_{ID}) = PK^{*\ell_i}_{(ID_t,2)}$. Suppose $PK^*_{(ID_t,2)} = H_0(ID_t)^{r^*}$ for some unknown r^*. Since σ^* is a valid forgery under the public key $PK^*_{ID_t} = (PK^*_{(ID_t,1)}, PK^*_{(ID_t,2)})$, it follows that $\hat{e}(P_{pub}, PK^*_{(ID_t,2)}) = \hat{e}(g, PK^*_{(ID_t,1)})$. Thus, we have $PK^*_{(ID_t,1)} = PK^{*a}_{(ID_t,2)} = H_0(ID_t)^{r^* a}$. Moreover, from the second equation in the verification algorithm, we have $\sigma^* = D_{ID_t} \cdot H_1(m^*||ID^*||PK^*_{ID})^{r^{*-1}} = D_{ID_t} \cdot PK^{*\ell_t r^{*-1}}_{(ID_t,2)} = D_{ID_t} \cdot H_0(ID_t)^{\ell_i}$. As $ID^* = ID_t$, $P_{pub} = g_1 = g^a$ and $H_0(ID_t) = g_2 \cdot g^{k'_t}$, this implies that $D_{ID_t} = H_0(ID_t)^a = g_2^a \cdot g^{ak'_t} = g^{ab} \cdot g_1^{k'_t}$. Therefore, $\mathcal{F}$ can compute D_{ID_t} as $\sigma^*/H_0(ID_t)^{\ell_i}$ and obtain $g^{ab} = D_{ID_t}/g_1^{k'_t}$, which is the solution to the CDH problem.

It remains to compute the probability that $\mathcal{F}$ solves the CDH problem. Actually, $\mathcal{F}$ succeeds if:

Λ_1: $\mathcal{F}$ does not abort during the simulation.
Λ_1: σ^* is a valid forgery on (ID^*, PK^*_{ID}, m^*).
Λ_1: $ID^* = ID_t$ and $\zeta^* = 1$.

The advantage of $\mathcal{F}$ is $Adv^{CDH}_{\mathcal{F}} = Pr[\Lambda_1 \wedge \Lambda_2 \wedge \Lambda_3] = Pr[\Lambda_1] \cdot Pr[\Lambda_2|\Lambda_1] \cdot Pr[\Lambda_3|\Lambda_1 \wedge \Lambda_2]$. If Λ_1 happens, then:

- $\mathcal{F}$ does not output *failure* during the simulation of the oracle **Partial-Private-Key-Extract**. This happens with probability $(1-(1/q_C))^{q_{PKEx}}$.
- $\mathcal{F}$ does not output *failure* during the simulation of the oracle **Secret-Value-Extract**. This happens with probability $(1-(1/q_C))^{q_{VEx}}$.
- $\mathcal{F}$ does not output *failure* during the simulation of $sign_{Super}$ oracle. This happens with probability $(1-(1/q_C)\xi)^{q_S} \geq (1-\xi)^{q_S}$.

Therefore, $Pr[\Lambda_1] \geq (1-(1/q_C))^{q_{PKEx}+q_{VEx}}(1-\xi)^{q_S}$. In addition, $Pr[\Lambda_2|\Lambda_1] = \varepsilon$ and $Pr[\Lambda_3|\Lambda_1 \wedge \Lambda_2] = \xi/q_C$. It follows that $Adv^{CDH}_{\mathcal{F}} \geq (1-(1/q_C))^{q_{PKEx}+q_{VEx}}(1-\xi)^{q_S}(\xi/q_C)\varepsilon$. The function $\xi(1-\xi)^{q_S}$ is maximized at $\xi = 1/(q_S+1)$. Thus,

$$Adv^{CDH}_{\mathcal{F}} \geq \left(1-\frac{1}{q_C}\right)^{q_{PKEx}+q_{VEx}} \left(1-\frac{1}{q_S+1}\right)^{q_S} \frac{1}{q_C(q_S+1)}\varepsilon.$$

This completes the proof of Theorem 6. □

Theorem 7 (Unforgeability against Type *II* adversary). *If there exists a Type II adaptively chosen message and chosen ID adversary $\mathcal{A}_{II}$ who can ask at most q_C* **Create-User** *queries, q_{VEx}* **Secret-Value-Extract** *queries and q_S $Sign_{Strong}$ queries, respectively, and can break the proposed scheme in polynomial time with success probability ε, then there exists an algorithm $\mathcal{F}$ which, by using $\mathcal{A}_{II}$ as a black box, can solve the InvCDH problem 2 with probability $Adv^{Inv}_{\mathcal{F}} \geq (1-(1/q_C))^{q_{VEx}}(1-(1/(q_S+1)))^{q_S}(1/q_C(q_S+1))\varepsilon$.*

Proof. If there exists an adversary $\mathcal{A}_{II}$ who can break the unforgeability of the proposed scheme via Type II attacks, then we can construct an algorithm $\mathcal{F}$ such that $\mathcal{F}$ can use $\mathcal{A}_{II}$ as a black-box and find $g^{a^{-1}}$ for any given pair $(g, g^a) \in \mathbb{G}_1^2$.

Let g be a generator of $\mathbb{G}_1$, and let the bilinear mapping be $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Let $a \in \mathbb{Z}_p^*$ be a random number chosen by the challenger $\mathcal{C}$. At beginning, $\mathcal{C}$ sends the challenge (g, g^a) to $\mathcal{F}$. The purpose of $\mathcal{F}$ is to find $g^{a^{-1}}$.

In the proof, $\mathcal{F}$ will act as the challenger of $\mathcal{A}_{II}$ by simulating the oracles defined in Section 2.3. Hash functions H_0, H_1 are regarded as random oracles in the proof. Under this assumption, we may assume that $\mathcal{A}_{II}$ is well-behaved in the sense that it always makes an H_1-hash request of $m||ID||PK_{ID}$ before it requests a signature for m corresponding to the public key PK_{ID}. In addition, it always makes an H_1-hash request of $m^*||ID^*||PK^*_{ID}$ before it outputs the forgery of m^*. It is trivial to modify any adversary-algorithm $\mathcal{A}_{II}$ to have this property.

Setup: $\mathcal{F}$ sets $P_{pub} = g^s$ where s is randomly chosen in $\mathbb{Z}_p^*$. $\mathcal{F}$ then sends $params = (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, g, P_{pub})$ together with the master secret key s to $\mathcal{A}_{II}$.

Query: At any time, $\mathcal{A}_{II}$ is allowed to access the following oracles in polynomial time. These oracles are all simulated by $\mathcal{F}$. Note that there is no oracle **Partial-Private-Key-Extract** in this proof as $\mathcal{A}_{II}$ has already obtained the master secret key s. (This enables $\mathcal{A}_{II}$ to compute any partial private keys.)

1 **Create-User:** $\mathcal{A}_{II}$ can query this oracle by providing an identity ID_i. In response to these queries, $\mathcal{F}$ first chooses a random number $t \in \{1, \ldots, q_C\}$.
  (1) If $i \neq t$, $\mathcal{F}$ chooses $k_i, k'_i \in_R \mathbb{Z}_p^*$ and sets $H_0(ID_i) = g^{k'_i}$, $PK_{(ID_i,1)} = P_{pub}^{k_i k'_i}, PK_{(ID_i,2)} = g^{k_i k'_i}$. In this case, the corresponding partial private key of the entity ID_i is $D_{ID_i} = H_0(ID_i)^s = g^{sk'_i}$ and the secret value is $x_{ID_i} = k_i$.
  (2) If $i = t$, $\mathcal{F}$ chooses $k'_t \in_R \mathbb{Z}_p^*$ and sets $H_0(ID_t) = g^{k'_t}$, $PK_{(ID_t,1)} = g^{ask'_t}$, $PK_{(ID_t,2)} = g^{ak'_t}$. Note that a is unknown to $\mathcal{F}$. In this case, $\mathcal{F}$ will set $D_{ID_t} = g^{sk'_t}$ and $x_{ID_t} = \perp$ which means that it cannot compute the secret value of ID_t.
  In both cases, $\mathcal{F}$ sends $H_0(ID_i)$ and $PK_{ID_i} = (PK_{(ID_i,1)}, PK_{(ID_i,2)})$ to $\mathcal{A}_{II}$, and adds $(ID_i, H_0(ID_i), D_{ID_i}, x_{ID_i}, PK_{ID_i})$ to *C-List* which is initially empty.
2. **Public-Key-Replace:** $\mathcal{A}_{II}$ can request to replace the public key PK_{ID_i} of an entity ID_i with a new public key PK'_{ID_i} chosen by $\mathcal{A}_I$ itself. Upon receiving such a request, $\mathcal{F}$ will replace the original public key PK_{ID_i} with PK'_{ID_i} if ID_i has been created. Otherwise, $\mathcal{F}$ outputs $\perp$. Notice that the replacement of a public key does not require the secret value corresponding to the new public key.
3. **Secret-Value-Extract:** Given ID_i chosen by $\mathcal{A}_{II}$, $\mathcal{F}$ outputs $\perp$ if ID_i has not been created. Else, if ID_i has been created and $i \neq t$,

$\mathcal{F}$ responds with $x_{ID_i} = k_i$. Otherwise, $i = t$ and $\mathcal{F}$ reports *failure* and terminates the simulation.

4. H_1 **queries:** $\mathcal{A}_{II}$ can query the random oracle H_1 at any time by making a request ω_i of the form $\omega_i = m_j||ID_k||PK'_{ID_k}$. $\mathcal{F}$ returns $\perp$ if ID_k has not been created. Otherwise, for the ith H_1 query
   - If $ID_k \neq ID_t$, $\mathcal{F}$ first checks the *C-List*.
     (1) If $(ID_k, , , , PK'_{ID_k})$ is in the *C-List*, $\mathcal{F}$ picks a random element R_i and sets $H_1(\omega_i) = R_i$.
     (2) Otherwise, $(ID_k, , , PK'_{ID_k})$ is not in the *C-List*, which means that PK'_{ID_k} is generated by $\mathcal{A}_{II}$ itself. $\mathcal{F}$ picks $\ell_i \in \mathbb{Z}_p^*$ randomly and sets $H_1(\omega_i) = PK'^{\ell_i}_{(ID_k,2)}/PK'_{(ID_k,1)}$. Here, $PK'_{ID_k} = (PK'_{(ID_k,1)}, PK'_{(ID_k,2)})$.
   - Otherwise, $ID_k = ID_t$, $\mathcal{F}$ first flips a biased-coin which outputs $\zeta_i \in \{0, 1\}$ with $Pr[\zeta_i = 1] = \xi$ and $Pr[\zeta_i = 0] = 1 - \xi$ (the value of ξ will be optimized later).
     (1) If $\zeta_i = 1$, $\mathcal{F}$ picks a random $\ell_i \in \mathbb{Z}_p^*$ and sets $H_1(\omega_i) = g^{\ell_i}$.
     (2) Otherwise, $\zeta_i = 0$. $\mathcal{F}$ picks a random $\ell_i \in \mathbb{Z}_p^*$ and sets $H_1(\omega_i) = PK'^{\ell_i}_{(ID_t,2)}/PK'_{(ID_t,1)}$.

   In both cases, $\mathcal{F}$ sends $H_1(\omega_i)$ to $\mathcal{A}_{II}$, and adds $(m_j, \ell_i, \omega_i, H_1(\omega_i))$ if $(ID_k \neq ID_t)$ or $(m_j, \ell_i, \omega_i, H_1(\omega_i), \zeta_i)$ (if $ID_k = ID_t$) on the H_1-*List* which is initially empty.
5. *Sign*$_{Super}$**:** For each sign query on (m_j, ID_k, PK'_{ID_k}), $\mathcal{F}$ outputs $\perp$ if ID_k has not been created. Otherwise, ID_k has already been created. Since we assume that $\mathcal{A}_{II}$ is well-behaved, $\mathcal{A}_{II}$ has already queried the random oracle H_1 on the input $\omega_i = (m_j||ID_k||PK'_{ID_k})$.
   - If $ID_j \neq ID_t$, and
     (1) $(ID_k, H_0(ID_k), D_{ID_k}, x_{ID_k}, PK'_{ID_k})$ is in the *C-List*, $\mathcal{F}$ can use the private key (x_{ID_k}, D_{ID_K}) of ID_k and $H_1(\omega_i)$ on the H_1-*List* to generate the valid signature σ_i for the message m_j and the identity ID_k.
     (2) $(ID_k, , , , PK'_{ID_k})$ is not in the *C-List*, which means that PK'_{ID_k} is generated by $\mathcal{A}_{II}$ itself, $\mathcal{F}$ sets $\sigma_i = H_0(ID_k)^{\ell_i}$. Note that σ_i is a valid signature on m_j and $PK'_{ID_k} = (PK'_{(ID_k,1)}, PK'_{(ID_k,2)})$ if PK'_{ID_k} is well-generated (which means that $PK'_{(ID_k,1)} = PK'^{a}_{(ID_k,2)}$ as they satisfy the equation $\hat{e}(PK'_{(ID_k,1)}, g) = \hat{e}(PK'_{(ID_k,2)}, P_{pub})$). Assume $PK'_{(ID_k,2)} = H_0(ID_k)^r$ for some unknown r. The correctness of σ_i can be verified according to the following equation

$$\sigma_i = D_{ID_k} \cdot H_1(\omega_i)^{r^{-1}} = (PK'^{s}_{(ID_k,2)} \cdot H_1(\omega_i))^{r^{-1}}$$

$$= (PK'_{(ID_k,1)} \cdot H_1(\omega_i))^{r^{-1}} = \left(\frac{PK'_{(ID_k,1)} \cdot PK'^{\ell_i}_{(ID_k,2)}}{PK'_{(ID_k,1)}}\right)^{r^{-1}}$$

$$= PK'^{\ell_i r^{-1}}_{(ID_k,2)} = H_0(ID_k)^{\ell_i}.$$

     In both cases, $\mathcal{F}$ returns σ_i to $\mathcal{A}_{II}$.
   - If $ID_i = ID_t$, then $\mathcal{F}$ first checks the H_1-*List*.
     (1) If $\zeta_i = 1$, $\mathcal{F}$ reports *failure* and terminates the simulation.
     (2) Otherwise, $\zeta_i = 0$ and $\mathcal{F}$ sets $\sigma_i = H_0(ID_t)^{\ell_i}$ which is sent to $\mathcal{A}_{II}$. Here ℓ_i is the value on the H_1-*List*. The correctness of σ_i can be verified in the same way as described above.

Forgery: After all queries, $\mathcal{A}_{II}$ outputs a forgery $(ID^*, PK^*_{ID}, m^*, \sigma^*)$ and wins the game if σ^* is a valid forgery.

By assumption, $H_1(m^*||ID^*||PK^*_{ID})$ is on the H_1-*List*, where PK^*_{ID} must be the original public key generated by the oracle **Create-User**. If $ID^* \neq ID_t$, then $\mathcal{F}$ outputs *failure* and terminates the simulation. Otherwise, $ID^* = ID_t$ and $\mathcal{F}$ will check the H_1-*List*.

(1) If $\zeta^* = 0$, $\mathcal{F}$ outputs *failure* and terminates the simulation.
(2) Otherwise, $\zeta^* = 1$ and $H_1(m^*||ID^*||PK^*_{ID}) = g^{*\ell_i}$ is on the H_1-*List*. Since σ^* is a valid forgery and $ID^* = ID_t$, according to the verification algorithm, we have $\sigma^* = D_{ID_t} \cdot H_1(m^*||ID^*||PK^*_{ID})^{a^{-1}} = D_{ID_t} \cdot g^{a^{-1}\ell_i}$. Therefore, $\mathcal{F}$ can compute $g^{a^{-1}}$ as $(\sigma^*/D_{ID_t})^{\ell_i^{-1}}$ where $D_{ID_t} = g^{sk'_t}$ is computed by $\mathcal{F}$ itself at step (2) of the **Create-User** query phase.

It remains to compute the probability that $\mathcal{F}$ can find $g^{a^{-1}}$. $\mathcal{F}$ will succeed if:

Λ_1 : $\mathcal{F}$ does not abort during the simulation.
Λ_1 : σ^* is a valid forgery on (ID^*, PK^*_{ID}, m^*).
Λ_1 : $ID^* = ID_t$ and $\zeta^* = 1$.

Thus, the advantage of $\mathcal{F}$ is $Adv^{Inv}_{\mathcal{F}} = Pr[\Lambda_1 \wedge \Lambda_2 \wedge \Lambda_3] = Pr[\Lambda_1] \cdot Pr[\Lambda_2|\Lambda_1] \cdot Pr[\Lambda_3|\Lambda_1 \wedge \Lambda_2]$. If Λ_1 happens, then

- $\mathcal{F}$ does not output *failure* during the simulation of the oracle **Secret-Value-Extract**. This happens with probability $(1 - (1/q_C))^{q_{VEx}}$.
- $\mathcal{F}$ does not output *failure* during the simulation of *Sign*$_{Super}$ oracle. This happens with probability $(1 - (1/q_C)\xi)^{q_S} \geq (1 - \xi)^{q_S}$.

Therefore, $Pr[\Lambda_1] \geq (1 - (1/q_C))^{q_{VEx}}(1 - \xi)^{q_S}$. In addition, $Pr[\Lambda_2|\Lambda_1] = \varepsilon$ and $Pr[\Lambda_3|\Lambda_1 \wedge \Lambda_2] = \xi/q_C$. It follows that $Adv^{Inv}_{\mathcal{F}} \geq (1 - (1/q_C))^{q_{VEx}}(1 - \xi)^{q_S}(\xi/q_C)\varepsilon$. The function $\xi(1 - \xi)^{q_S}$ is maximized at $\xi = 1/(q_S + 1)$. Thus,

$$Adv^{Inv}_{\mathcal{F}} \geq \left(1 - \frac{1}{q_C}\right)^{q_{VEx}} \left(1 - \frac{1}{q_S + 1}\right)^{q_S} \frac{1}{q_C(q_S + 1)}\varepsilon.$$

This completes the proof of Theorem 7. □

Bao et al. (2003) showed the equivalence of CDH problem and InvCDH problem. In which follows, we will show the reductions of time and success probability from InvCDH problem to CDH problem.

Theorem 8. *Suppose there is an algorithm $\mathcal{F}$ that can output $g^{a^{-1}}$ on input of any given $g, g^a \in \mathbb{G}_1^*$ with probability δ, in expected time bound T. Then there is another algorithm $\mathcal{M}$ which can solve the CDH problem and output g^{ab} on input of any given $g, g^a, g^b \in \mathbb{G}_1^*$ with probability ε^3 in expected time $3T$.*

Proof. From $\mathcal{F}$, we can construct an algorithm $\mathcal{M}$ as follows:

1. $\mathcal{M}$'s input is $g, g^a, g^b \in \mathbb{G}_1^*$.
2. $\mathcal{M}$ runs $\mathcal{F}$ with input g^a, g (g can be used as $g^{a^{-1}a}$). If $\mathcal{F}$ outputs $Y_1 = g^{aa} = g^{a^2}$, then go to the next step.
3. $\mathcal{M}$ runs $\mathcal{F}$ with input g^b, $g = g^{b^{-1}b}$. If $\mathcal{F}$ outputs $Y_2 = g^{bb} = g^{b^2}$, then go to the next step.
4. $\mathcal{M}$ runs $\mathcal{F}$ with input $g^a g^b = g^{a+b}$, $g = g^{(a+b)^{-1}(a+b)}$. If $\mathcal{F}$ outputs $Y_3 = g^{(a+b)^2}$, then go to the next step.
5. $\mathcal{M}$ computes and outputs $Y = (Y_3/(Y_1 \cdot Y_2))^{2^{-1}}$.

Obviously, $Y = (Y_3/(Y_1 \cdot Y_2))^{2^{-1}} = g^{ab}$. Hence, in expected time $3T$, $\mathcal{M}$ can solve the CDH problem with success probability δ^3. □

With Theorems 7 and 8, we can get the conclusion that the proposed scheme is secure against type *II* adaptively chosen message and chosen *ID* adversary under the hardness assumption of CDH Problem in the random oracle model.

5. Comparison among existing certificateless signature schemes with short signature length

In this section, we compare the proposed certificateless short signature scheme with other existing certificateless short signature schemes from the aspect of security, hardness assumption and computation cost required by a signer and a verifier, respectively.

Table 1
Security levels and performance evaluation.

Schemes	Adversary with oracle $Sign_{Super}$	Hardness assumption	Cost sign-phase	Cost verify-phase
Proposed	Secure	CDH	$1E_{G_1}$	$2\hat{e}+1E_{G_1}$
TYH (Tso et al., 2008, 2011)	Insecure	Modified k-CAA	$1E_{G_1}$	$1\hat{e}+1E_{G_1}$
HMS (Huang et al., 2007)	Insecure	CDH	$1E_{G_1}$	$2\hat{e}$
DW (Du and Wen, 2009)	Insecure	k-CAA	$1E_{G_1}$	$1\hat{e}$
FHH (Fan et al., 2009)	Insecure	N/A	$1E_{G_1}$	$1\hat{e}+1E_{G_1}$

We consider the most costly computations only such as pairing and exponentiation operations in $\mathbb{G}_1$. In addition, some pre-computable operations are not included in the comparison. (For example, the computation of $\hat{e}(g,g)$, $\hat{e}(H_0(ID),P_{pub})$ of other schemes and the computation of $\hat{e}(P_{pub},PK_{(ID,2)})$, $\hat{e}(PK_{(ID,1)},g)$ of our scheme.) We denote by $\hat{e}$ a computation of the pairing, E_{G_1} an exponentiation in $\mathbb{G}_1$, and E_{G_2} an exponentiation in $\mathbb{G}_2$.

As shown in Table 1, signature generation in our scheme is almost as efficient as other schemes, but signature verification requires a little more computational cost. However, our scheme is the only certificateless signature scheme with short signature length and is secure against adversaries providing with super signing oracle $Sign_{Super}$ defined in this paper.

To show that all the existing certificateless short signature schemes are not secure against this adversary, assume a strong adversary exists in their schemes. The adversary $\mathcal{F}$ can query a super signing oracle $Sign_{Super}$ with a false public key chosen by itself (and $\mathcal{F}$ also knows the corresponding secret value). Using the valid signature outputted from $Sign_{Super}$ and the secret value, then the partial private key can be extracted. Now, since $\mathcal{F}$ knows both the secret value and the partial private key of the target user, $\mathcal{F}$ can of course forge any signature on behalf of the target user. On the other hand, our scheme is well designed as the adversary cannot obtain any advantage by replacing the public key.

6. Conclusion

In this paper, we proposed an efficient certificateless signature scheme which has a signature-length as short as Boneh et al.'s short signature. Furthermore, it is the only short certificateless signature scheme with provable security against a stronger adversary. The adversary defined in our model is given valid signatures under any public keys (in the public key space) chosen by itself. Our model captures the attack scenario that the adversary can obtain message/signature pairs which are valid under the public key chosen by the adversary without providing the corresponding secret value. This is the strongest one among all attackers considered in short certificateless signature schemes. We believe the new scheme is more suitable for systems with low-bandwidth channels and/or low-computation power but requiring a higher security level.

References

Au, M.H., Chen, J., Liu, Joseph K., Mu, Y., Duncan, Wong, S., Yang, G., 2007. Malicious KGC attacks in certificateless cryptography. In: Proceedings of ASIACCS'07, pp. 302–311.

Al-Riyami, S.S., Paterson, K.G., 2003. Certificateless public key cryptography. Advances in Cryptology-ASISCRYPT'03. Lecture Notes in Computer Science 2894, 452–473.

Bao, F., Deng, R.H., Zhu, H., 2003. Variations of Diffie–Hellman problem. Proceedings of ICICS'03. Lecture Notes in Computer Science 2836, 301–312.

Barr, K., Asanovic, K., 2003. Energy aware lossless data compression. In: Proceedings of the ACM Conference on Mobile Systems, Applications, and Services (MobiSys).

Barreto, P.S.L.M., Kim, H.Y., Lynn, B., Scott, M., 2002. Efficient algorithm for pairing-based cryptosystems. Advances in Cryptology-CRYPTO'02. Lecture Notes in Computer Science 2442, 354–369.

Barreto, P.S.L.M., Lynn, B., Scott, M., 2003. On the selection of pairing-friendly groups. Proceedings of SAC'03. Lecture Notes in Computer Science 3006, 17–25.

Bellare, M., Neven, G., 2006. Multi-signatures in the plain public-key model and a general forking lemma. In: Proceedings of the 13th ACM Conference on Computer and Communication Security, pp. 390–398.

Boneh, D., Boyen, X., 2004. Short signatures without random oracles. Advances in Cryptology-EUROCRYPT'04. Lecture Notes in Computer Science 3027, 56–73.

Boneh, D., Lynn, B., Shacham, H., 2001. Short signatures from the Weil pairing. Advances in Cryptology-ASIACRYPT'01. Lecture Notes in Computer Science 2248, 514–533.

Boneh, D., Shen, E., Waters, B., 2006. Strongly unforgeable signatures based on computational Diffie–Hellman. Proceedings of PKC'06. Lecture Notes in Computer Science 3958, 229–240.

Diffie, W., Hellman, M., 1976. New directions in cryptography. In: IEEE Transactions on Information Theory, IT, vol. 2 (6), pp. 644–654.

Du, H., Wen, Q., 2009. Efficient and provably-secure certificateless short signature scheme from bilinear pairings. International Journal of Computer Standards & Interfaces 31, 390–394.

Fan, C., Hsu, R., Ho, P. Cyptanalysis on Du–Wen certificateless short signature scheme. In Proceedings of JWIS09. Available at http://jwis2009.nsysu.edu.tw/location/paper/Cryptanalysis.

Girault, M., 1991. Self-certified public keys. Advances in Cryptology-EUROCRYPT'91. Lecture Notes in Computer Science 547, 490–497.

Goldwasser, S., Micali, S., Rivest, R.L., 1988. A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal of Computing 17 (2), 281–308.

Gorantla, M.C., Saxena, A., 2005. An efficient certificateless signature scheme. Proceedings of CIS'05. Lecture Notes in Artificial Intelligence 3802 (II), 110–116.

Hu, B.C., Wong, D.S., Zhang, Z., Deng, X., 2007. Certificatelss signature: a new security model and an improved generic construction. Designs, Codes and Cryptography 42 (2), 109–126.

Huang, X., Mu, Y., Susilo, W., Wong, D.S., Wu, W., 2007. Certificateles signature revisited. Proceedings of ACISP'07. Lecture Notes in Computer Science 4586, 308–322.

Huang, X., Susilo, W., Mu, Y., Zhang, F., 2005. On the security of certificateless signature schemes from Asiacrypt 2003. Proceedings of CANS'05. Lecture Notes in Computer Science 3810, 13–25.

Pfitzmann, B., Sadeghi, A., 2000. Anonymous fingerprinting withdirect non-repudiation. Advances in Cryptology-ASISCRYPT'00. Lecture Notes in Computer Science 1976, 401–414.

Shamir, A., 1984. Identity-based cryptosystems and signature schemes. Advances in Cryptology-CRYPTO'84. Lecture Notes in Computer Science 0196, 47–53.

Tso, R., Okamoto, T., Okamoto, E., 2009. Efficient short signatures from pairing. In: Proceedings of ITNG'09, pp. 417–422.

Tso, R., Yi, X., Huang, X., 2008. Efficient and short certificateless signature. Proceedings of CANS'08. Lecture Notes in Computer Science 5339, 64–79.

Tso, R., Yi, X., Huang, X., 2011. Efficient and short certificateless signatures secure against realistic adversaries. The Journal of Supercomputing 55 (2), 173–191.

Wu, W., Mu, Y., Susilo, W., Huang, X., 2009. Certificate-based signature revisited. Journal of Universal Computer Science 15 (8), 1659–1684.

Yap, W.L., Heng, S.H., Goi, B.M., 2006. An efficient certificteless signature. Proceedings of EUC Workshops'06. Lecture Notes in Computer Science 4097, 322–331.

Yum, D.H., Lee, P.J., 2004. Generic construction of certificateless signature. Proceedings of ACISP'04. Lecture Note in Computer Science 3108, 200–211.

Zhang, F., Safavi-Naini, R., Susilo, W., 2003. An efficient signature scheme from bilinear pairings and its applications. Proceedings of PKC'04. Lecture Notes in Computer Science 2947, 277–290.

Zhang, Z., Wong, D.S., Xu, J., Feng, D., 2006. Certificateless public-key signature: security model and efficient construction. Proceedings of ACNS'06. Lecture Notes in Computer Science 3989, 293–308.

Raylin Tso is an assistant professor of Computer Science at National Chengchi University, Taiwan. He received his PhD degree in Systems and Information Engineering from University of Tsukuba, Japan in 2006. His research interests are mainly in cryptography including secret sharing, key agreement, digital signatures, certificateless cryptosystems, etc. Recently, his research activities are focused on certificateless signatures and digital signatures that providing privacy protection.

Xinyi Huang received his PhD in Computer Science (Information Security) in 2009 from the School of Computer Science and Software Engineering, University of Wollongong, Australia. He is now with Fujian Normal University, China. His research interests focus on cryptography and its applications in information systems. He has published more than 50 referred research papers at international conferences and journals.

Willy Susilo received a PhD in Computer Science from University of Wollongong, Australia. He is a Professor at the School of Computer Science and Software Engineering and the co-director of Centre for Computer and Information Security Research (CCISR) at the University of Wollongong.

He is currently holding the prestigious ARC Future Fellow awarded by the Australian Research Council (ARC).His main research interests include cryptography and information security. His main contribution is in the area of digital signature schemes. He has served as a program committee member in dozens of international conferences. He has published numerous publications in the area of digital signature schemes and encryption schemes.