On the Security of Certificateless Signature Schemes from Asiacrypt 2003*

Xinyi Huang[1], Willy Susilo[2], Yi Mu[2], and Futai Zhang[1,**]

[1] College of Mathematics and Computer Science,
Nanjing Normal University, P.R. China
xinyinjnu@126.com, zhangfutai@njnu.edu.cn
[2] Centre for Information Security Research,
School of Information Technology and Computer Science,
University of Wollongong, Australia
{wsusilo, ymu}@uow.edu.au

Abstract. In traditional digital signature schemes, certificates signed by a trusted party are required to ensure the authenticity of the public key. In Asiacrypt 2003, the concept of certificateless signature scheme was introduced. In the new paradigm, the necessity of certificates has been successfully removed. The security model for certificateless cryptography was also introduced in the same paper. However, as we shall show in this paper, the proposed certificateless signature is insecure in their defined model. We provide an attack that *can successfully forge* a certificateless signature in their model. We also fix this problem by proposing a new scheme.

Keywords: Certificateless Signature, Certificateless Cryptography, Attack Model, Bilinear Pairing.

1 Introduction

In traditional digital signature schemes, the binding between a user and his public key needs to be ensured. A typical way to provide this assurance is by providing certificates that are signed by a trusted third party. In [13], Shamir introduced a new notion called identity-based cryptography (and hence, identity-based signature scheme) where the user's public key is indeed his identity (such as an email, IP address, etc.). This way, the need of certification can be avoided. However, this approach creates a new inherent problem namely the key escrow of a user's private key, since the trusted third party called the Private Key Generator (PKG) must be completely trusted, since he has the knowledge of the user's secret key.

* This work is supported by ARC Discovery Grant DP0557493.

** Partially supported by Ministry of Education of Jiangsu Province Project 03KJA520066 and Open Project of Key Laboratory on Computer Network and Information Security of Ministry of Education of China.

Y.G. Desmedt et al. (Eds.): CANS 2005, LNCS 3810, pp. 13–25, 2005.

To fill the gap between traditional cryptography and identity-based cryptography, Al-Riyami and Paterson proposed a new paradigm called *certificateless cryptography* in [1]. In contrast to traditional cryptography, certificateless cryptography does not require the use of any certificates to ensure the authenticity of public keys. Certificateless cryptography relies on the existence of a trusted third party who has the master-key. In this sense, it is similar to identity-based cryptography. Nevertheless, certificateless cryptography does not suffer from the key escrow property that seems to be inherent in identity-based cryptography. We note that the concept of certificateless cryptography has been around [7, 9, 10, 12], but the first formalization was provided in [1].

Intuitively, the characteristic of certificateless cryptography is as follows. The trusted third party, called the KGC, does not have access to the users' private keys. The KGC only supplies a user with a *partial private key* D_i, which the KGC computes from an identifier ID_i. As in the identity-based cryptography, the partial private key needs to be delivered securely to the user. Then, the user combines his partial private key D_i with some secret information to generate his actual private key S_i. This way, the user's private key is *not* available to the KGC. The user also combines his secret information with the KGC's public parameters to generate his public key P_i. The user's public key P_i needs to be made available to the other participants by transmitting it along with messages, in the case of message signing. Hence, it is no longer an identity-based cryptography, since the public key needs to be provided (but in contrast to the traditional cryptography, the public key does not require any certificate).

Due to the lack of public key authentication, it is important to assume that an adversary can replace the user's public key by a false key of its choice [1]. In order to provide a secure certificateless signature scheme, this type of attacks must not be able to produce signatures that verify with the false public key [1]. An assumption that must be made is that the KGC does not mount a public key replacement attack since he is armed with a partial private key. Hence, we must assume that the KGC, who posses the master-key and hence all partial private keys, is trusted not to replace user's public keys. This way, the level of trust is similar to the trust in a CA in a traditional PKI. We will review the adversarial model defined in [1] in the next section.

Following, the work of [1], there are several certificateless public key encryption proposed (eg. [3, 5, 4, 15]). In [14], a generic construction of certificateless signature from any identity-based signature scheme and a secure public key signature scheme in the sense of [8] was proposed.

Our Contribution

In this paper, we show that the proposed certificateless signature scheme in [1] *does not* satisfy the security requirement of certificateless cryptography, in terms of the defined adversarial model in [1]. To be more precise, we show that an attacker who does *not* posses the master-key but can only do a public key replacement attack, can always successfully forge a signature. We also provide a new scheme that resists against this type of attacks and hence, it satisfies the requirements of certificateless signature schemes as defined in [1].

Organization of the Paper
In the next section, we will review some preliminaries required throughout the paper. In Section 3, we review the proposed certificateless signature scheme in [1]. The security of this scheme was not provided in [1], and therefore, firstly we show that the unforgeability of the scheme in Section 4. Unfortunately, as we will also show in Section 4, the scheme fails to resist against the adversarial model type I as defined in [1]. We will show how to fix this problem in Section 5. Finally, Section 6 concludes the paper.

2 Preliminaries

In this section, we will review some fundamental backgrounds required in this paper, namely bilinear pairing and the certificateless cryptography definition.

2.1 Bilinear Pairing

Let $\mathbb{G}_1$ denote an additive group of prime order q and $\mathbb{G}_2$ be a multiplicative group of the same order. Let P denote a generator in $\mathbb{G}_1$. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear mapping with the following properties:

- The map $\hat{e}$ is bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1, a, b \in \mathbb{Z}_q$.
- The map $\hat{e}$ is non-degenerate: $\hat{e}(P, P) \neq 1_{\mathbb{G}_2}$.
- The map $\hat{e}$ is efficiently computable.

A Bilinear pairing instance generator is defined as a probabilistic polynomial time algorithm $\mathcal{IG}$ that takes as input a security parameter ℓ and returns a uniformly random tuple $param = (q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P)$ of bilinear parameters, including a prime number q of size ℓ, a cyclic additive group $\mathbb{G}_1$ of order q, a multiplicative group $\mathbb{G}_2$ of order q, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and a generator P of $\mathbb{G}_1$. For a group $\mathbb{G}$ of prime order, we denote the set $\mathbb{G}^* = \mathbb{G} \setminus \{\mathcal{O}\}$ where $\mathcal{O}$ is the identity element of the group.

Definition 1. Computational Diffie-Hellman (CDH) problem in $\mathbb{G}_1$. *Given (P, aP, bP), for some $a, b \in \mathbb{Z}_q^*$, compute abP.*

The success probability of any probabilistic polynomial-time algorithm $\mathcal{A}$ in solving CDH problem in $\mathbb{G}_1$ is defined to be

$$\mathtt{Succ}_{\mathcal{A},\mathbb{G}_1}^{CDH} = Pr[\mathcal{A}(P, aP, bP) = abP : a, b \in \mathbb{Z}_q^*]$$

The CDH assumption states that for every probabilistic polynomial-time algorithm $\mathcal{A}$, $\mathtt{Succ}_{\mathcal{A},\mathbb{G}_1}^{CDH}$ is negligible.

2.2 Certificateless Signature Schemes

A certificateless signature scheme is defined by seven algorithms: Setup, Partial-Private-Key-Extract, Set-Secret-Value, Set-Private-Key, Set-Public-Key, Sign and Verify. The description of each algorithm is as follows.

- Setup: The master key and parameter generation algorithm is a probabilistic algorithm that accepts as input a security parameter 1^k and returns a master-key and a parameter list params.
- Partial-Private-Key-Extract: The partial private key issuance algorithm is a deterministic algorithm that accepts as input a user identity ID_i, a parameter list param and a master-key to produce the user's partial private key D_i.
- Set-Secret-Value: The set secret value setup algorithm is a probabilistic algorithm that accepts as input a parameter list param and a user identity ID_i to produce the user's secret value x_i.
- Set-Private-Key: The secret value setup algorithm is a probabilistic algorithm that accepts as input a parameter list param, the user's partial private key D_i and the user's secret value x_i to produce a private signing key S_i.
- Set-Public-Key: The public key generation algorithm is a deterministic algorithm that takes as input a parameter list param, a user identity ID_i and the user's secret value x_i to produce a public key P_i.
- Sign: The signing algorithm is a probabilistic algorithm that accepts a message $M \in \mathcal{M}$, $\mathcal{M}$ is the message space, a user's identity ID_i, a parameter list param and the user's signing key S_i to produce a signature σ.
- Verify: The verification algorithm is a deterministic algorithm that accepts a message M, a signature σ, a parameter list param, the public key P_i and the user's identity ID_i to output `true` if the signature is correct, or $\perp$ otherwise.

2.3 Adversarial Model of Certificateless Signature Schemes

As defined in [1], there are two types of adversary with different capabilities:

Type I Adversary: This type of adversary $\mathcal{A}_I$ does not have access to the master-key, but $\mathcal{A}_I$ has the ability to *replace* the public key of any entity with a value of his choice, because there is no certificate involved in certificateless signature schemes.

Type II Adversary: This type of adversary $\mathcal{A}_{II}$ has access to the master-key but cannot perform public keys replacement.

Nevertheless, no formal security model was presented in neither [1] nor [2]. In this section, firstly we provide a formal definition of existential unforgeability of a certificateless signature (CLS) scheme under both two types of chosen message attack. They are defined using the following game between an adversary $\mathcal{A} \in \{\mathcal{A}_I, \mathcal{A}_{II}\}$ and a challenger $\mathcal{C}$.

Type I Adversary

- Setup: $\mathcal{C}$ runs the algorithm to obtain the system parameter lists params, $\mathcal{C}$ then sends params to the adversary $\mathcal{A}_I$.
- Partial-Private-Key Queries: $\mathcal{A}_I$ can request the Partial-Private-Key of the user whose identity is ID. In respond, $\mathcal{C}$ outputs the Partial-Private-Key D_{ID}.
- Public-Key-Replacement: For any user whose identity is ID, $\mathcal{A}_I$ can choose a new Secret-Value x and compute the new public key (X, Y). $\mathcal{A}_I$ then set (X, Y) as the new public key of this user and submit (x, X, Y, ID) to $\mathcal{C}$. $\mathcal{C}$ will record these replacements which will be used later.

- Sign Queries: $\mathcal{A}_I$ can request user's (whose identity is ID) signature on a message M. In respond, $\mathcal{C}$ outputs a signature σ for a message M which is a valid signature under the public key $\mathcal{A}_I$ has replaced earlier.
- Output: Finally, $\mathcal{A}_I$ outputs a target message/signature pair (M^*, σ^*) of the user whose identity is ID^*. This message/signature pair must satisfy the following requirements:
  1. This signature is valid under the public key (X^*, Y^*) chosen by $\mathcal{A}_I$.
  2. A_I does not request the Partial-Private-Key of this user whose identity is ID^*.
  3. M^* has never been queried during the Sign Queries.

The success probability of an Type I adversary to win the game is defined by

$$Succ_{\mathcal{A}_I}^{EF-CLS-CMA}$$

Definition 2. *A certificateless signature scheme is existential unforgeable against Type I chosen-message attacks iff the probability of success of any polynomially bounded Type I adversary in the above game is negligible. In other words,*

$$Succ_{\mathcal{A}_I}^{EF-CLS-CMA}(k) \leq \epsilon$$

k is the system's security parameter.

Type II Adversary

- Setup: $\mathcal{C}$ runs the algorithm to obtain the system parameter lists params and also the system's master-key:s, $\mathcal{C}$ then sends params and s to the adversary $\mathcal{A}_{II}$.
- Sign Queries: $\mathcal{A}_{II}$ can request user's(whose identity is ID) signature on a message M. In respond, $\mathcal{C}$ outputs a signature σ for a message M.
- Output: Finally, $\mathcal{A}_{II}$ outputs a target message/signature pair (M^*, σ^*) of the user whose identity is ID. This message/signature pair must satisfy the following requirements:
  1. This signature is a valid one, i.e. it passes the verification algorithm.
  2. M^* has never been queried during the Sign Queries.

The success probability of an Type II adversary to win the game is defined by

$$Succ_{\mathcal{A}_{II}}^{EF-CLS-CMA}$$

Definition 3. *A certificateless signature scheme is existential unforgeable against Type II chosen-message attacks iff the probability of success of any polynomially bounded Type II adversary in the above game is negligible. In other words,*

$$Succ_{\mathcal{A}_{II}}^{EF-CLS-CMA}(k) \leq \epsilon$$

k is the system's security parameter.

Definition 4. *[1] A certificateless signature scheme is existential unforgeable against chosen-message attacks iff it is secure against both types of adversaries.*

3 Review of Al-Riyami-Paterson's Certificateless Signature Scheme from Asiacrypt 2003

In this section, we review the certificateless signature scheme from [1]. The certificateless signature scheme is defined as follows.

- Setup: This algorithm runs as follows.
  1. Run $\mathcal{IG}$ on input k to generate $(\mathbb{G}_1, \mathbb{G}_2, \hat{e})$ where $\mathbb{G}_1$ and $\mathbb{G}_2$ are groups of some prime order q ($q \geq 2^k$) and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear pairing.
  2. Select a random generator $P \in \mathbb{G}_1$.
  3. Select a master-key s randomly from $\mathbb{Z}_q^*$ and set $P_0 = sP$.
  4. Select cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ and $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$, where n denote the bit-length of plaintexts [1].

  The system parameters $\mathsf{param} = (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_0, H_1, H_2)$. The master-key is $s \in \mathbb{Z}_q^*$. The message space is $\mathcal{M} = \{0,1\}^n$.
- Partial-Private-Key-Extract: This algorithm accepts an identity $\mathsf{ID}_i \in \{0,1\}^*$ and constructs the partial private key for the user as follows.
  1. Compute $\mathsf{Q}_i = H_1(\mathsf{ID}_i)$.
  2. Output the partial private key $D_i = s\mathsf{Q}_i$.
- Set-Secret-Value: This algorithm takes as input param and the user's identity ID_i, and selects a random $x_i \in \mathbb{Z}_q^*$ and outputs x_i as the user's secret value.
- Set-Private-Key: This algorithm accepts param, a user's partial private key D_i and the user's secret value $x_i \in \mathbb{Z}_q^*$ to transform the partial private key D_i to a full private key S_i by computing $S_i = x_i D_i = x_i s \mathsf{Q}_i$ and output S_i.
- Set-Public-Key: This algorithm accepts param and a user's secret value $x_i \in \mathbb{Z}_q^*$ to produce the user's public key $P_i = (X_i, Y_i)$, where $X_i = x_i P$ and $Y_i = x_i P_0 = x_i s P$.
- Sign: To sign a message $M \in \mathcal{M}$ using the private key S_i, perform the following steps.
  1. Select a random $r \in \mathbb{Z}_q^*$.
  2. Compute $R = \hat{e}(rP, P)$.
  3. Set $v = H_2(M, R)$.
  4. Compute $U = vS_i + rP$.
  5. Output (U, v) as the signature on M.
- Verify: To verify a signature (U, v) on a message $M \in \mathcal{M}$ for an identity ID_i and public key (X_i, Y_i), perform the following steps.
  1. Verify whether $\hat{e}(X_i, P_0) \overset{?}{=} \hat{e}(Y_i, P)$ holds with equality. If not, then output $\perp$ and abort.
  2. Compute $R = \hat{e}(U, P)\hat{e}(\mathsf{Q}_i, -Y_i)^v$.
  3. Verify whether $v \overset{?}{=} H_2(M, R)$ holds with equality. If it does, output `true`. Otherwise, output $\perp$.

4 Security Analysis of Al-Riyami-Paterson's Certificateless Signature Schemes

A formal security proof for the provided certificateless public key encryption scheme in [1] has already provided in [1]. Unfortunately, the security proof for their certificateless signature scheme is not provided in the same paper. As we shall show in this section, the scheme in [1] does not resist against type I adversary, defined in the same paper. We will show how to fix this problem in section 5.

4.1 An Attack on Al-Riyami-Paterson's Scheme Using Type I Adversary

As defined in [1], a certificateless signature scheme is existentially unforgeable iff it resists against type I and type II adversaries. Recall that type I adversary does not possess the knowledge of the master-key, s, but the adversary can perform public key replacement, i.e. replacing the public key with its choice. We will show that the scheme in [1] does not resist against type I adversary since the adversary can successfully forge a user's signature on a message of its choice. The attack is as follows.

Without losing generality, we only define the Sign and Verify algorithms in this section. The rest of the algorithms are the same as the original scheme defined in [1]. Recall that the Sign algorithm will be performed by an attacker who can replace the user's public key. The attack is successful, iff the signature verification with respect to the replaced public key is correct.

Sign: To sign an arbitrary message $M \in \mathcal{M}$, the adversary performs the following.

1. Select a random $U \in \mathbb{G}_1$.
2. Compute $R = \hat{e}(U, P)\hat{e}(\mathsf{Q}_i, -P_0)$, where $\mathsf{Q}_i = H_1(\mathsf{ID}_i)$ and ID_i denotes a valid user's identity.
3. Compute $v = H_2(M, R)$.
4. Let $x_i = v^{-1} \pmod q$.
5. Compute $X_i = x_i P$ and $Y_i = x_i P_0$.
6. Replace the user's public key with (X_i, Y_i).
7. Publish (U, v) as the user's signature on a message M.

The attack is said to be *successful*, iff the verification of the signature on a message returns `true`. This is justified as follows.

Verify: To verify a signature (U, v) on a message M, using the public key (X_i, Y_i) for an identity ID_i, anyone can perform the verification algorithm as defined in [1]. As we shall see below, the verification will return `true`.

1. Verify whether $\hat{e}(X_i, P_0) \stackrel{?}{=} \hat{e}(Y_i, P)$ holds. This verification will pass because

$$\begin{aligned}\hat{e}(X_i, P_0) &= \hat{e}(x_i P, sP)\\ &= \hat{e}(x_i sP, P)\\ &= \hat{e}(Y_i, P)\end{aligned}$$

2. Compute $R' = \hat{e}(U,P)\hat{e}(\mathsf{Q}_i, -Y_i)^v$.
3. Verify whether $v \overset{?}{=} H_2(M, R')$ holds. This verification will pass because

$$\begin{aligned} R' &= \hat{e}(U,P)\hat{e}(\mathsf{Q}_i, -Y_i)^v \\ &= \hat{e}(U,P)\hat{e}(\mathsf{Q}_i, -v \cdot x_i \cdot P_0) \\ &= \hat{e}(U,P)\hat{e}(\mathsf{Q}_i, -v \cdot v^{-1} \cdot P_0) \\ &= \hat{e}(U,P)\hat{e}(\mathsf{Q}_i, -P_0) \\ &= R \end{aligned}$$

Since $R' = R$ holds, then $v \overset{?}{=} H_2(M,R)$ will hold with equality. □

Theorem 1. *The Al-Riyami-Paterson's certificateless signature scheme is universally forgeable against type I adversary.*

Remarks: We note that this attack is a strong attack that belongs to the *no-message attack* classes, where *no* signing oracle is required, in the adversarial model type I. The authors of [1] revised their Asiacrypt 2003 paper in [2], but the signature scheme in their revised version is the same as the Asiacrypt version in [1].

4.2 Security of Al-Riyami-Paterson's Certificateless Signature Scheme Against Type II Adversary

Fortunately, as we shall show in this section, the proposed scheme is secure against type II adversary. This is shown in the following theorem.

Theorem 2. *The certificateless signature scheme proposed in [1] is unforgeable against the type II adversary in the random oracle [6] model under the CDH assumption in $\mathbb{G}_1$.*

Proof (sketch). Let $\mathcal{A}$ be our type II adversary. Recall that $\mathcal{A}$ has access to the master-key, s, but cannot perform any public key replacement. Having the access to s, $\mathcal{A}$ can forge any message-signature pair for any user. We will show how to build algorithm $\mathcal{B}$ that will solve the CDH problem using $\mathcal{A}$'s capability as follows.

We model the hash function H_2 as a random oracle and hence, we will need to keep a list of the oracle queries that have been made. The purpose of algorithm $\mathcal{B}$ is to compute abP given aP, bP, for some unknown $a, b \in \mathbb{Z}_q^*$. Firstly, $\mathcal{B}$ sets the user's public key $X_i = aP$ and the user's public identity $\mathsf{Q}_i = bP$. Then, $\mathcal{B}$ selects the system parameter param $= (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_0, H_1, H_2)$. Finally, the master-key is $s \in \mathbb{Z}_q^*$ is selected. The public key Y_i can be computed afterwards from $Y_i = sX_i$.

When the simulation is started, $\mathcal{A}$ is provided with param and the master-key, s. The interaction with the hash oracle, H_2, is recorded in the list of oracle queries. Eventually, applying the forking technqie [11], a set of two forged signatures on the same message M will be obtained. When this happens, $\mathcal{B}$ obtains

$$R = \hat{e}(U,P)\hat{e}(\mathsf{Q}_i, -Y_i)^v$$

and

$$R = \hat{e}(U', P)\hat{e}(\mathsf{Q}_i, -Y_i)^{v'}$$

for both signatures $(U, v), (U', v')$ on the same message M. Therefore, $\mathcal{B}$ obtains the following equations

$$\begin{aligned}\hat{e}(U, P)\hat{e}(\mathsf{Q}_i, -Y_i)^{v} &= e(U', P)\hat{e}(\mathsf{Q}_i, -Y_i)^{v'}\\ \hat{e}(U - U', P) &= \hat{e}(\mathsf{Q}_i, -Y_i)^{v'-v}\\ \hat{e}(U - U', P) &= \hat{e}((v - v')\mathsf{Q}_i, x_i sP)\\ \hat{e}(U - U', P) &= \hat{e}((v - v')x_i s\mathsf{Q}_i, P)\end{aligned}$$

From this equation, $\mathcal{B}$ has the following

$$\begin{aligned}U - U' &= (v - v')x_i s\mathsf{Q}_i\\ (v - v')^{-1}s^{-1}(U - U') &= x_i\mathsf{Q}_i\end{aligned}$$

Since $x_i\mathsf{Q}_i$ can be computed from

$$x_i\mathsf{Q}_i = (v - v')^{-1}s^{-1}(U - U')$$

and $\mathcal{B}$ has the knowledge of (v, v', s, U, U'), then $x_i\mathsf{Q}_i$ is computable by $\mathcal{B}$. Note that $x_i\mathsf{Q}_i = x_i bP = abP$ in our setting above, and hence, $\mathcal{B}$ has successfully obtains the solution of CDH. We obtain the contradiction and hence, complete the proof. □

5 A Secure Certificateless Signature Scheme

In this section, we provide a modification to the certificateless signature scheme proposed in [1]. Unlike the scheme in [1], our scheme is secure against type I and II adversaries. Firstly, we provide an intuition why the proposed scheme in [1] fails against type I adversary.

In the scheme in [1], the receiver of the message verifies the validity of user's public key by testing whether the equation

$$\hat{e}(X_i, P_0) \stackrel{?}{=} \hat{e}(Y_i, P)$$

holds with equality. However, this is *not sufficient* to deter against type I adversary. This equality only ensures that $Y_i = sX_i$ holds. The test should also cover a mechanism to make sure that the secret value x_i, chosen by the user, has been used correctly to obtain $S_i = x_i D_i$, for $X_i = x_i P$ and $Y_i = x_i P_0$. This important aspect is neglected in the design of the certificateless signature scheme in [1]. There is no way to check whether x_i in X_i and Y_i is identical to that of x_i in S_i. In this section, we show how to fix this problem.

5.1 A Secure Scheme

Without losing generality, we only describe the Sign and Verify algorithms as the other algorithms are the same as the one defined in [1].

Sign: To sign a message $M \in \mathcal{M}$ using the private key S_i, perform the following steps.

1. Select a random $r \in \mathbb{Z}_q^*$.
2. Compute $R = \hat{e}(rP, P)$.
3. Compute $v = H_2(M, R, \hat{e}(S_i, P))$.
4. Compute $U = vS_i + rP$.
5. Output the signature on a message M as (U, v).

Verify: To verify a signature (U, v) on a message $M \in \mathcal{M}$ for a public key (X_i, Y_i), perform the following steps.

1. Test whether
$$\hat{e}(X_i, P_0) \stackrel{?}{=} \hat{e}(Y_i, P)$$
holds with equality. If not, then output $\perp$ and abort.
2. Compute $R = \hat{e}(U, P)\hat{e}(\mathsf{Q}_i, -Y_i)^v$.
3. Test whether
$$v \stackrel{?}{=} H_2(M, R, \hat{e}(\mathsf{Q}_i, Y_i))$$
holds with equality. If that so, then output `true`. Otherwise, output $\perp$.

Remarks: Intuitively, the scheme is secure against the attack model presented earlier. This is due to the following arguments. In the signature scheme, the value v is the output of the hash on input $(M, R, e(S_i, P))$ which is determined by the message M, a random choice R and $S_i = x_i s Q_i$. In this scheme, the attacker $\mathcal{A}_I$ cannot use v to change the public key of the signer because S_i is determined by the signer's public key. The formal proof is presented as follows.

Theorem 3. *Our scheme is unforgeable against type I adversary in the random oracle model under the CDH assumption in* $\mathbb{G}_1$.

Proof (sketch). Let $\mathcal{B}$ be a CDH attacker. Suppose that $\mathcal{B}$ is given an instance (q, P, aP, bP). Let $\mathcal{A}$ be a forger that breaks the proposed signature scheme under chosen message attack. We show how $\mathcal{B}$ can use $\mathcal{A}$ to solve the CDH problem, i.e. to compute abP.

First, $\mathcal{B}$ sets $P_0 = aP$ where P_0 denotes the KGC's public key and gives (q, P, P_0) to $\mathcal{A}$. $\mathcal{B}$ then simulates the random oracle H_1 as follows. Let q_{H_1} be the maximum number of queries to the random oracle H_1. $\mathcal{B}$ picks $j \in [1, q_{H_1}]$ uniformly at random. Then, whenever $\mathcal{A}$ issues a query denoted ID_i to H_1 where $1 \leq i \leq q_{H_1}$, $\mathcal{B}$ does the following: If $i \neq j$, pick $l_i \in \mathbb{Z}_q^*$, compute l_iP and return $H(\mathsf{ID}_i) = l_iP$ as answer. Else (if $i = j$) return $H(\mathsf{ID}_j) = bP$ as answer.

From now on, we let $\mathsf{ID}_j = \mathsf{ID}^*$ where ID_j is the j-th query to the random oracle H_1 and j is chosen at the beginning of the above simulation of H_1.

Now, let q_{ex} be the maximum number of partial private key extraction queries. Whenever $\mathcal{A}$ issues such a query each of which is denoted ID_i, where $1 \leq i \leq q_{ex}$,

$\mathcal{B}$ does the following: If $\mathsf{ID}_i \neq \mathsf{ID}^*$, find $l_i \in \mathbb{Z}_q^*$ that used to compute $H(\mathsf{ID}_i) = l_iP$ or pick $l_i \in \mathbb{Z}_q^*$ at random (this is the case when ID_i has not been asked to H_1), compute l_iaP and return $D_i = l_iP_0$ as answer. Else (if $i = j$) abort and stop the simulation.

From the above simulation of partial private key extraction and the random oracle H_1, it can be easily seen that the distribution of the simulated private keys are identical to those in the real attack except for the partial private key associated with ID^* as $D_j = l_jP_0 = l_jaP = al_jP = aH(\mathsf{ID}_j)$.

The random oracle H_2 can naturally be simulated. Namely, whenever $\mathcal{A}$ issues a query $(M_i, R_i, \hat{e}(S_i, P))$ to H_2, $\mathcal{B}$ does the following: Pick $v_i \in \mathbb{Z}_q^*$ at random and return it as answer.

Note that at any time during the simulation, $\mathcal{A}$ can generate a private/public key pair and replace the user's public key with its own. We assume that $\mathcal{B}$ *keeps track* of all such private/public key pairs.

Equipped with those private keys and the partial private keys for any $\mathsf{ID}_i \neq \mathsf{ID}^*$, $\mathcal{A}$ is able to create signatures on any message. Hence, assume that $\mathcal{A}$ issues a query $(M_i, (X_i, Y_i))$, where M_i denotes a message and (X_i, Y_i) denotes a public key chosen by $\mathcal{A}$, to the signing oracle whose secret key is associated with ID^*. Upon receiving this, $\mathcal{B}$ creates a signature as follows:

1. Pick $U_i \in \mathbb{G}_1$ and $v_i \in \mathbb{Z}_q^*$ at random.
2. Compute $R_i = \hat{e}(U_i, P)\hat{e}(H_1(\mathsf{ID}^*), -Y_i)^{v_i}$. (Note that $H_1(\mathsf{ID}^*) = bP$).
3. Set $v_i = H_2(M_i, R_i, \hat{e}(H_1(\mathsf{ID}^*), Y_i))$.
4. Return (U_i, v_i) as a signature on M_i.

Notice that the above simulated signature is identically distributed as the one in the real attack.

The next step of the simulation is to apply the 'forking' technique formalized in [11]: Let $(M, (U, v), \mathsf{ID}^*, (X, Y))$ be a forgery that output by $\mathcal{A}$ at the end of the attack. Note here that if $\mathcal{A}$ does not output ID^* as a part of the forgery, $\mathcal{B}$ just aborts the simulation. (The probability that $\mathcal{B}$ does not abort the simulation is $O(1/q_{H_1})$). $\mathcal{B}$ then replays $\mathcal{A}$ with the same random tape but different choice of the hash function H_2 to get another forgery $(M, (U', v'), \mathsf{ID}^*, (X, Y))$. From these two forgeries, $\mathcal{B}$ obtains

$$R = \hat{e}(U, P)\hat{e}(H_1(\mathsf{ID}^*), -Y)^v$$

and

$$R = \hat{e}(U', P)\hat{e}(H_1(\mathsf{ID}^*), -Y)^{v'}.$$

Since (U, v) and (U', v') are valid signatures on M, $\mathcal{B}$ consequently obtains the following:

$$\begin{aligned}
\hat{e}(U, P)\hat{e}(H_1(\mathsf{ID}^*), -Y)^v &= e(U', P)\hat{e}(H_1(\mathsf{ID}^*), -Y)^{v'} \\
\hat{e}(U, P)\hat{e}(bP, -xaP)^v &= e(U', P)\hat{e}(bP, -xaP)^{v'} \\
\hat{e}(U - U', P) &= \hat{e}(bP, -xaP)^{v'-v} \\
\hat{e}(U - U', P) &= \hat{e}((v - v')xP, abP) \\
\hat{e}(U - U', P) &= \hat{e}((v - v')xabP, P)
\end{aligned}$$

From this equation, $\mathcal{B}$ has the following

$$\begin{aligned} U - U' &= (v - v')xabP \\ (v - v')^{-1}(U - U') &= xabP \end{aligned}$$

Recall that B is assumed to keep track of private/public key pairs of $\mathcal{A}$. Hence, the Diffie-Hellman key abP can be obtained by computing $(v - v')^{-1}x^{-1}(U - U') = abP$. Therefore, we complete the proof. □

It is easy to see that our scheme is unforgeable against type II adversary under the same assumption. The proof is very similar to the proof of theorem 2 and hence, it is omitted.

6 Conclusion

In this paper, we reviewed the security of the certificateless signature scheme proposed in [1]. The authors of [1] did not provide a security proof for this scheme. We showed that the scheme does not resist against type I adversary as defined in the adversarial model in [1]. However, we also show that the scheme is unforgeable against type II adversary. We modified the scheme in [1] and proposed a new scheme that resists against both types of adversaries.

Acknowledgement

The authors would like to express their gratitude thanks to Dr. Joonsang Baek for his fruitful discussion and suggestion on the security proof for our paper. We would also like to thank the anonymous referees of International Conference on Cryptology and Network Security (CANS05) for the suggestions to improve this paper.

References

1. S. S. Al-Riyami and K. G. Paterson. Certificateless Public Key Cryptography. *Advances in Cryptography - Asiacrypt 2003, Lecture Notes in Computer Science 2894*, pages 452–473, Springer-Verlag, Berlin, 2003.
2. S. S. Al-Riyami and K. G. Paterson. Certificateless Public Key Cryptography. Cryptology ePrint Archive. Available online: `Http:// eprint.iacr.org/2003/126`.
3. S. S. Al-Riyami and K. G. Paterson. CBE from CLPKE: A Generic Construction and Efficient Schemes. *Public Key Cryptography, PKC 2005, Lecture Notes in Computer Science 3386*, pages 398–415, Springer-Verlag, Berlin, 2005.
4. J. Baek, R. Safavi-Naini and W. Susilo. Certificateless Public Key Encryption without Pairing. *8th Information Security Conference, ISC 2005, Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 2005.
5. Z. Cheng and R. Comley. Efficient Certificateless Public Key Encryption. Cryptology ePrint Archive. Available online: `http://eprint.iacr.org/2005/012`.

6. M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. *ACM CCCS '93*, pp. 62–73, 1993.
7. M. Girault. Self Certified Public Keys. *Advanced in Cryptology - Eurocrypt 1991, Lecture Notes in Computer Science 547*, pp. 490–497, Springer-Verlag, 1992.
8. S. Goldwasser, S. Micali, and R. Rivest. A Secure Digital Signature Scheme. *SIAM Journal on Computing 17*, pages 281 – 308, 1988.
9. E. Okamoto. Key distribution systems based on identification information. *Advances in Cryptology - Crypto 1987, Lecture Notes in Computer Science 293*, pp. 194 – 202, Springer-Verlag, Berlin, 1987.
10. H. Petersen and P. Horster. Self-Certified Keys – Concepts and Applications. *International Conference on Communications and Multimedia Security*, Chapman and Hall, 1997.
11. D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. *Advanced in Cryptology - Eurocrypt 1996, Lecture Notes in Computer Science 1070*, pages 387 – 398, Springer-Verlag, Berlin, 1996.
12. S. Saeednia. Identity-Based and Self-Certified Key-Exchange Protocols. *Information Security and Privacy, ACISP 1997, Lecture Notes in Computer Science 1270*, pp. 303–313, Springer-Verlag, 1997.
13. A. Shamir. Identity-based cryptosystems and signature schemes. *Advances in Cryptology - Crypto '84, Lecture Notes in Computer Science 196*, pages 47–53, Springer-Verlag, Berlin, 1985.
14. D. H. Yum and P. J. Lee. Generic Construction of Certificateless Signature. *Information Security and Privacy, ACISP 2004, Lecture Notes in Computer Science 3108*, pages 200 – 211, Springer-Verlag, Berlin, 2004.
15. D. H. Yum and P. J. Lee. Generic Construction of Certificateless Encryption. *ICCSA 2004, Lecture Notes in Computer Science 3043*, pp. 802–811, Springer-Verlag, Berlin, 2004.