Cryptanalysis of Some Proxy Signature Schemes without Certificates *

Wun-She Yap[1], Swee-Huay Heng[2] and Bok-Min Goi[1]

[1] Centre for Cryptography and Information Security (CCIS)
Faculty of Engineering
Multimedia University, 63100 Cyberjaya, Selangor, Malaysia
{wsyap, bmgoi}@mmu.edu.my
[2] Centre for Cryptography and Information Security (CCIS)
Faculty of Information Science and Technology
Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia
shheng@mmu.edu.my

Abstract. The concept of proxy signature was introduced by Mambo *et al.* to delegate signing capability in the digital world. In this paper, we show that three existing proxy signature schemes without certificates, namely, the Qian and Cao identity-based proxy signature (IBPS) scheme, the Guo *et al.* IBPS scheme and the Li *et al.* certificateless proxy signature (CLPS) scheme are insecure against universal forgery. More precisely, we show that any user who has a valid public-private key pair can act as a cheating proxy signer and forge the proxy signature on behalf of the original signer at will, without obtaining the official delegation from the original signer.

Keywords: Proxy signature, identity-based, certificateless, attack

1 Introduction

Proxy Signature and Its Applications. The concept of proxy signature was first introduced by Mambo, Usuda and Okamoto in [18]. A proxy signature scheme involved three entities, namely, the original signer, the proxy signer and the verifier [28]. A proxy signature scheme allows a designated signer called a proxy signer to sign the message on behalf of an original signer. A proxy signature convinces the verifier that the signature is signed by the proxy signer who gets the delegation right from the original signer. Proxy signatures have found various practical applications, particularly in distributed computing where delegation of rights is common. Examples include e-cash systems [19], global distribution network [2], grid computing [7] and mobile agent applications [15, 16], to name a few. To illuminate how to use proxy signatures, we give more explanations on the delegating signing capabilities within organization [29]. If a manager of an

* The authors gratefully acknowledge the Malaysia IRPA grant (04-99-01-00003-EAR) and e-Science fund (01-02-01-SF0032).

organization is on leave, he has to delegate to his assistant manager the capability to sign on behalf of him.

Natural Constructions of Proxy Signature. Proxy signature can be constructed in several ways as stated in [18, 14, 23, 25, 3], according to the delegation type:

1. **Full Delegation**: The most straightforward solution is for the original signer to give its private key to the proxy signer, who can then use it to sign any messages on behalf of the original signer.
2. **Partial Delegation**: In a partial delegation scheme, a proxy signer has a new key called proxy signing key, which is different from the original signer's private key. The proxy signing key is generated by both the original signer and the proxy signer.
3. **Delegation by Certificate/Warrant**: In delegation by warrant, the original signer uses its private key and the signing algorithm of a standard signature to sign a *warrant*, which contains information regarding the particular proxy signer. After receiving the warrant, the proxy signer uses its private key and the signing algorithm of a standard signature to sign messages on behalf of the original signer.
4. **Partial Delegation with Warrant**: Kim *et al.* [14] proposed a partial delegation with warrant proxy signature scheme which enjoys the computational and bandwidth advantages over the proxy signature by warrant and the structure advantage over the proxy signature for partial delegation.

Public Key Cryptography without Certificates. Traditional public key cryptography (TPKC) was introduced by Diffie and Hellman [6] to solve the key distribution problem suffered in symmetric key cryptography. As opposed to the symmetric key cryptography, TPKC involves the use of two different keys, namely a public key and a private key, which are mathematically related to each other. However, TPKC requires the use of certificate in authenticating the public key, which leads to certificate revocation problems. Thus, the design of a secure and efficient cryptographic scheme without certificate becomes the goal of many cryptographers nowadays. Two types of public key cryptography without certificates in focus are identity-based cryptography (IBC) and certificateless public key cryptography (CLPKC).

The concept of IBC was formulated by Shamir [22] to achieve implicit certification. Shamir's original motivation was to simplify certificate management in email systems. In IBC, the public key is effectively replaced by the user's publicly available identity information or any arbitrary string which derived from the user identity (ID), thus certificate can be omitted. However, since all the private keys of the users are generated by a trusted third party (TTP) called private key generator (PKG), the private key escrow problem is inherent in the system. CLPKC [1] is a paradigm which eliminates the usage of certificates in TPKC while solving the inherent key escrow problem in IBC. CLPKC can be seen as a model that is intermediate between TPKC and IBC. In this new paradigm, the user public key is no longer any arbitrary string that identifies the user, rather,

it is similar with the public key used in TPKC. The user private key is computed by using both the partial private key, a key generated by a TTP called Key Generation Centre (KGC), and the user secret value.

Our Contributions. Most of the proxy signature schemes were proposed in the public key infrastructure (PKI) setting. Recently, several proxy signature schemes adapted to IBC [28, 5, 21, 26, 24, 11, 10] and CLPKC [17] have also been proposed.

In this paper, we review three existing partial delegation with warrant proxy signature schemes without certificates, namely, the Qian and Cao identity-based proxy signature (IBPS) scheme [21], the Guo *et al.* IBPS [11] scheme and the Li *et al.* certificateless proxy signature (CLPS) scheme. These three schemes were derived from the provably secure identity-based signature (IBS) schemes [22, 4, 12]. The Qian and Cao IBPS scheme is RSA-based. RSA-based schemes are preferable since it is quite common that companies may have invested in expensive hardware and software implementations of RSA. Meanwhile, the Guo *et al.* IBPS scheme and the Li *et al.* CLPS scheme are constructed by using bilinear pairings, which is an important tool in constructing identity-based and certificateless scheme.

We show that these three schemes did not satisfy the basic security requirement of proxy signature in the ID-based setting and the certificateless setting. More precisely, we show that any user who has a valid public-private key pair can act as a cheating proxy signer and forge the proxy signature on behalf of the original signer at will, without obtaining the official delegation from the original signer.

2 Preliminaries

We review the properties of bilinear pairings below.

Bilinear Pairings: Let $(\mathbb{G}_1, \circ)$ and $(\mathbb{G}_2, \circ)$ denote two cyclic groups of prime order q ($\circ$ denotes a binary operation). A *bilinear map* $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfies the following properties:

1. Bilinearity: For all $P, Q, R \in \mathbb{G}_1$, $e(P \circ Q, R) = e(P, R)e(Q, R)$ and $e(P, Q \circ R) = e(P, Q)e(P, R)$. Thus, for any $a, b \in \mathbb{Z}_q^*$, $e(aP, bP) = e(bP, aP) = \hat{e}(P, P)^{ab}$.
2. Non-degeneracy: $e(P, Q) \neq 1_{\mathbb{G}_2}$ where $1_{\mathbb{G}_2}$ is the identity element of $\mathbb{G}_2$.
3. Computability: There is an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in \mathbb{G}_1$.

3 Cryptanalysis of the Qian and Cao IBPS Scheme

Recently, Qian and Cao proposed an IBPS scheme [21] which was derived from the Shamir IBS scheme [22]. They also proved the Shamir IBS scheme secure against adaptive chosen message attack (CMA) [8] based on the RSA assumption

in the same paper. In this section, we first review the IBPS scheme proposed by Qian and Cao. We then show that this IBPS scheme is universally forgeable.

3.1 The Qian and Cao IBPS Scheme

The Qian and Cao IBPS scheme [21] is defined by the following algorithms:

1. `Setup`: The PKG runs the following steps:
   (a) Compute $n = pq$ where p and q are two large primes.
   (b) Select e at random where $\gcd(e, \phi(n)) = 1$.
   (c) Compute the `master key` d where $ed \equiv 1 \bmod \phi(n)$.
   (d) Choose $h : \{0,1\}^* \rightarrow \mathbb{Z}_{\phi(n)}$ where h is a strong one way function.
   (e) Choose $H : \{0,1\}^* \rightarrow \mathbb{Z}_n$ where H is a cryptographic hash function.

   The PKG keeps d as the `master key` and publicizes the public parameters `params` $= (n, e, h, H)$.
2. `Extract`: The user submits his ID $\in \{0,1\}^*$ to the PKG, the PKG then computes the user private key $D_{ID} = Q_{ID}^d$, where $Q_{ID} = H(ID)$. The user private key must be transmitted to the user through a secure channel. The original signer Alice has her public-private key pair as (Q_{ID_A}, D_{ID_A}), and the proxy signer Bob has his public-private key pair as (Q_{ID_B}, D_{ID_B}).
3. `Proxy Key Generation`: When Alice delegates her signing capability to the proxy signer Bob, Alice performs the following steps:
   (a) Make a warrant m_w which records the delegation policy including limits of authority, valid periods of delegation, the proxy signer ID etc.
   (b) Choose $r_A \in \mathbb{Z}_n$ at random and compute $R_A = r_A^e \bmod n$.
   (c) Compute $S_A = D_{ID_A} \cdot r_A^{h(R_A||m_w)} \bmod n$.
   (d) Send the signature $\sigma_A = (R_A, S_A)$ to the proxy signer Bob.

   After receiving the signature σ_A, Bob checks whether $S_A^e = Q_{ID_A} \cdot R_A^{h(R_A||m_w)} \bmod n$ holds. If not, Bob rejects the signature.
4. `Proxy Signature Generation`: Bob generates the proxy signature as follows:
   (a) Choose $r_B \in \mathbb{Z}_n$ at random and compute $R_B = r_B^e \bmod n$.
   (b) Compute $h = h(R_B||m_w||m)$, where m_w is the warrant and m is the message to be signed.
   (c) Compute $S_B = D_{ID_B} \cdot (r_B \cdot S_A)^{h(R_B||m_w||m)} \bmod n$.

   At last, Bob sends $\sigma_B = (R_A, R_B, S_B)$ to the verifier as a proxy signature on m_w and m for ID_A and ID_B.
5. `Proxy Signature Verification`: After receiving the σ_B, the verifier performs the following steps:
   (a) Check the warrant m_w.
   (b) Compute $Q_{ID_A} = H(ID_A)$ and $Q_{ID_B} = H(ID_B)$.
   (c) Check whether $S_B^e = Q_{ID_B} \cdot (R_B \cdot Q_{ID_A} \cdot R_A^{h(R_A||m_w)})^{h(R_B||m_w||m)} \bmod n$ holds. If not, Bob rejects the signature.

In [21], Qian and Cao showed that the Shamir IBS scheme cannot resist the blinding attack as they claimed that the user private key is the common RSA signature. The forger can pick a random $t \in \mathbb{Z}_n$ and set $\mathrm{ID}_0 = t^e \cdot \mathrm{ID} \bmod n$. The forger then requests the signature from the signer who is willing to sign on ID_0. The forger now simply computes $S = S_0 \cdot t^{-1} \bmod n$ where S and S_0 are respectively the signature for ID and ID_0 on message m. Qian and Cao thereby proposed an improved Shamir IBS scheme by setting the private key $D_{ID} = H(ID)^d$. We note that this improvement had been recommended by Shamir earlier in [22].

3.2 Attack on the Qian and Cao IBPS Scheme

Now, we show that the Qian and Cao IBPS scheme is vulnerable to the forgery attack. This strong attack is the universal forgery against no message attack where no signing oracle is required in the adversarial model. To be more precise, any user who has a valid public-private key pair can act as a cheating proxy signer (which is also considered as a forger here), to sign any message at will on behalf of the original signer, without obtaining any official delegation from the original signer. We describe the efficient algorithm that enables the forger to sign any message on behalf of the original signer. Let A denote the original signer while B denote the cheating proxy signer.

`Proxy Signature Generation`: To sign a message $m \in \{0,1\}^n$, the cheating proxy signer (the forger) who has his own private key D_{ID_B} performs the following steps:

1. Make a warrant m_w.
2. Choose $r_A \in \mathbb{Z}_n$ at random and compute $R_A = r_A^e \bmod n$.
3. Choose $r_B \in \mathbb{Z}_n$ at random and compute $R_B = r_B^e \cdot Q_{ID_A}^{-1} \bmod n$ where $Q_{ID_A} = H(ID_A)$.
4. Compute $h = h(R_B||m_w||m)$, where m_w is the warrant and m is the message to be signed.
5. Compute $S_B = D_{ID_B} \cdot (r_B \cdot r_A^{h(R_A||m_w)})^{h(R_B||m_w||m)} \bmod n$.

The forged proxy signature on message m signed by the cheating proxy signer B on behalf of the original signer A is valid since the verification step is true for the forged proxy signature as follows:

1. Check whether $S_B^e = Q_{ID_B} \cdot (R_B \cdot Q_{ID_A} \cdot R_A^{h(R_A||m_w)})^{h(R_B||m_w||m)} \bmod n$ holds. If not, rejects the signature.

$$\begin{aligned} S_B^e &= D_{ID_B}^e \cdot (r_B^e \cdot r_A^{eh(R_A||m_w)})^{h(R_B||m_w||m)} \\ &= Q_{ID_B} \cdot (r_B^e \cdot R_A^{h(R_A||m_w)})^{h(R_B||m_w||m)} \\ &= Q_{ID_B} \cdot (R_B \cdot Q_{ID_A} \cdot R_A^{h(R_A||m_w)})^{h(R_B||m_w||m)} \end{aligned}$$

   where $r_B^e = R_B \cdot Q_{ID_A}$.

The Qian and Cao IBPS scheme is therefore insecure against the universal forgery since the forger can sign any message he wants on behalf of any original signer.

4 Cryptanalysis of the Guo *et al.* IBPS Scheme

We first review the IBPS schemes proposed by Guo *et al.* [11] which was derived from the Cha and Cheon IBS scheme [4]. In [27], Yoon *et al.* showed that the Cha and Cheon IBS scheme cannot be used in constructing a provably secure identity-based aggregate signature scheme if no further modification is made. In this section, we show that the Guo *et al.* IBPS scheme is insecure against the universal forgery by using the similar approach as in Yoon *et al.*

4.1 The Guo *et al.* IBPS Scheme

The Guo *et al.* IBPS scheme [11] is defined by the following algorithms:

1. `Setup`: The PKG first chooses a security parameter k, it then chooses two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same large prime order $q(|q| = k)$, a generator $P \in \mathbb{G}_1$ and also a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Two one way functions are also necessary: $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$. At last, the PKG chooses $s \in \mathbb{Z}_q^*$ as the system `master key` which is known only by itself. The PKG then computes $P_{pub} = sP$ as the system public key and publicizes the system parameters $\texttt{params} = \{\mathbb{G}_1, \mathbb{G}_2, e, q, k, P, P_{pub}, H_1, H_2\}$.
2. `Extract`: The user submits his ID $\in \{0,1\}^*$ to the PKG, the PKG then computes the user private key $D_{ID} = sQ_{ID}$, where $Q_{ID} = H_1(ID)$. The user private key must be transmitted to the user through a secure channel. The original signer Alice has her public-private key pair as (Q_{ID_A}, D_{ID_A}), and the proxy signer Bob has his public-private key pair as (Q_{ID_B}, D_{ID_B}).
3. `Proxy Key Generation`: To delegate the signing ability to the proxy signer, the original signer Alice first makes a warrant m_w, which consists of the original signer ID, the proxy signer ID, the delegation period T, the proxy signature scope, etc. Then, Alice performs some computations as follows:
   (a) Choose $x_A \in \mathbb{Z}_q^*$ at random and compute $X_{ID_A} = x_A D_{ID_A}$ and $X'_{ID_A} = x_A Q_{ID_A}$.
   (b) Compute $T = e(X'_{ID_A}, P_{pub}) = e(X_{ID_A}, P)$.
   (c) Compute $r = H_2(m_w || T || X'_{ID_A})$.
   (d) Compute $S = (x_A - r) D_{ID_A}$.

   At last, Alice sends (X'_{ID_A}, S, r) and m_w to Bob. When Bob receives the warrant m_w and (X'_{ID_A}, S, r) from Alice, he also makes some computations to check if the triple consists of the original signer's authority. Bob firstly computes:

$$\begin{aligned} T' &= e(S, P) \cdot e(rQ_{ID_A}, P_{pub}) \\ &= e(x_A D_{ID_A}, P) \\ &= e(X_{ID_A}, P) \\ &= e(X'_{ID_A}, P_{pub}) \end{aligned}$$

   Then, he computes $r' = H_2(m_w || T' || X'_{ID_A})$, only if the equations $r' = r$ and $T' = e(X'_{ID_A}, P_{pub})$ are satisfied, so that Bob can confirm that he has got the original signer's authority. The proxy signature key is the combination of (D_{ID_B}, S).

4. `Proxy Signature Generation`: Bob generates the proxy signature as follows:
   (a) Choose $x_B \in \mathbb{Z}_q^*$ at random and compute $U = x_B Q_{ID_B}$.
   (b) Compute $h = H_2(m||m_w||U)$, where m_w is the warrant and m is the message to be signed.
   (c) Compute $V = S + (x_B + h)D_{ID_B}$, where S is the delegation signature from the original signer and D_{ID_B} is Bob's private key.

   At last, Bob sends $(X'_{ID_A}, U, V, m_w, m)$ to the verifier as a proxy signature.
5. `Proxy Signature Verification`: After receiving the $(X'_{ID_A}, U, V, m_w, m)$, the verifier performs the following steps:
   (a) Check the warrant m_w.
   (b) Compute $T'' = e(X'_{ID_A}, P_{pub})$.
   (c) Compute $r' = H_2(m_w||T''||X'_{ID_A})$, where m_w is the warrant.
   (d) Compute $h' = H_2(m||m_w||U)$, where m_w is the warrant and m is the message to be signed.
   (e) Check $e(P, V) = e(P_{pub}, X'_{ID_A} - r'Q_{ID_A} + U + h'Q_{ID_B})$. If it holds, $(X'_{ID_A}, U, V, m_w, m)$ will be accepted, otherwise it will be rejected.

4.2 Attack on the Guo *et al.* IBPS Scheme

Now, we show that the Guo *et al.* IBPS scheme is vulnerable to the forgery attack by using the same approach as in Yoon *et al.* Similar to our previous attack mounted on the Qian and Cao IBPS scheme, this strong attack is again the universal forgery against no message attack where no signing oracle is required in the adversarial model. More precisely, any user who has a valid public-private key pair can act as a cheating proxy signer (which is also considered as a forger here), to sign any message at will on behalf of the original signer, without obtaining any official delegation from the original signer. We describe the efficient algorithm used to sign any message on behalf of the original signer below. Let A denote the original signer while B denote the cheating proxy signer.

`Proxy Signature Generation`: To sign a message $m \in \{0,1\}^n$, the cheating proxy signer (the forger) who has his own private key D_{ID_B} performs the following steps:

1. Select a random $x_A \in \mathbb{Z}_q^*$ and compute $X'_{ID_A} = x_A Q_{ID_A}$.
2. Compute $r = H_2(m_w||T||X'_{ID_A})$ where m_w is selected at random and T is computed as $e(X'_{ID_A}, P_{pub})$.
3. Select a random $x_B \in Z_q^*$ and compute $U = x_B Q_{ID_B} - X'_{ID_A} + rQ_{ID_A}$.
4. Compute $h = H_2(m||m_w||U)$.
5. Compute $V = (x_B + h)D_{ID_B}$.
6. Return $(X'_{ID_A}, U, V, m_w, m)$ as a proxy signature.

The forged proxy signature on message m signed by the cheating proxy signer B on behalf of the original signer A is valid since the verification step is true for the forged proxy signature as follows:

1. Compute $T'' = e(X'_{ID_A}, P_{pub})$.

2. Compute $r' = H_2(m_w||T''||X'_{ID_A})$, where m_w is the warrant.
3. Compute $h' = H_2(m||m_w||U)$, where m_w is the warrant and m is the message to be signed.
4. Accept the proxy signature if $e(P,V) = e(P_{pub}, X'_{ID_A} - r'Q_{ID_A} + U + h'Q_{ID_B})$.

$$\begin{aligned} e(P,V) &= e(P_{pub}, X'_{ID_A} - r'Q_{ID_A} + U + h'Q_{ID_B}) \\ &= e(sP, X'_{ID_A} - r'Q_{ID_A} + x_B Q_{ID_B} - X'_{ID_A} + rQ_{ID_A} + h'Q_{ID_B}) \\ &= e(sP, x_B Q_{ID_B} + h'Q_{ID_B}) \\ &= e(P, (x_B + h')sQ_{ID_B}) \\ &= e(P, (x_B + h)D_{ID_B}) \end{aligned}$$

where $r' = r$ and $h' = h$.

Thus, the Guo *et al.* IBPS scheme is insecure against the universal forgery since the forger can sign any message at will on behalf of any original signer without the cooperation of the original signer at all.

5 Cryptanalysis of the Li *et al.* CLPS Scheme

The Li *et al.* CLPS scheme [17] was derived from the Cha and Cheon IBS scheme [4] and the Hess IBS scheme [12]. It is the only CLPS scheme in the literature.

5.1 The Li *et al.* CLPS Scheme

The Li *et al.* CLPS scheme [17] is defined by the following algorithms:

1. `Setup`: Given a security parameter $k \in Z^+$, the algorithm works as follows:
   (a) Generate the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q and a pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.
   (b) Choose an arbitrary generator $P \in \mathbb{G}_1$.
   (c) Select a random $s \in \mathbb{Z}_q^*$ and set $P_0 = sP$.
   (d) Choose a cryptographic hash function $H_1 : \{0,1\}^* \to \mathbb{G}_1$ and $H_2 : \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{Z}_q^*$.
   The system parameters are `params` $= \langle \mathbb{G}_1, \mathbb{G}_2, e, q, P, P_0, H_1, H_2 \rangle$. The message space is $M = \{0,1\}^*$. The `master key` is $s \in \mathbb{Z}_q^*$.
2. `Set-Partial-Private-Key`: Given `params` and `master-key`, this algorithm works as follows: Compute $Q_{ID_i} = H_1(ID_i) \in \mathbb{G}_1$ and output a partial private key, $D_{ID_i} = sQ_{ID_i} \in \mathbb{G}_1$. Thus, the original signer Alice has her public-private key pair as (Q_{ID_A}, D_{ID_A}), and the proxy signer Bob has his public-private key pair as (Q_{ID_B}, D_{ID_B}).
3. `Set-Secret-Value`: Given `params`, select a random value $x_{ID_i} \in \mathbb{Z}_q^*$ where x_{ID_i} is the secret value.
4. `Set-Private-Key`: Set private key, $S_{ID_i} = x_{ID_i} D_{ID_i}$.
5. `Set-Public-Key`: Given `params` and the secret value $x_{ID_i} \in \mathbb{Z}_q^*$, this algorithm computes $X_{ID_i} = x_{ID_i} P \in \mathbb{G}_1$ and $Y_{ID_i} = x_{ID_i} P_0 \in \mathbb{G}_1$.

6. `Generation of the Proxy Key`: To delegate the signing ability to the proxy signer, the original signer Alice makes a warrant m_w first, which consists of the original signer ID, the proxy signer ID, the delegation period T, the proxy signature scope, etc. Then, Alice makes some computations as follows:
   (a) Choose $r \in \mathbb{Z}_q^*$ at random and compute $U = rQ_{ID_A}$.
   (b) Compute $h_A = H_2(m_w||U)$.
   (c) Compute $V = (r + h_A)S_{ID_A}$.

   At last, Alice sends (U, V) and m_w to Bob. When Bob receives the warrant m_w and (U, V) from Alice, he performs the following steps:
   (a) Check whether $e(X_{ID_A}, P_0) = e(Y_{ID_A}, P)$ holds.
   (b) Compute $h_A = H_2(m_w||U)$.
   (c) Check whether $e(P, V) = e(Y_{ID_A}, U + h_A Q_{ID_A})$ holds.

   Then, the proxy signature key S_P is computed as $S_P = V + S_{ID_B}$.
7. `Proxy Signature Generation`: Bob can generate the proxy signature as follows:
   (a) Choose $a \in \mathbb{Z}_q^*$ at random and compute $R = e(P, P)^a$.
   (b) Compute $h_B = H_2(m||R)$, where m is the message to be signed.
   (c) Compute $S = h_B S_P + aP$.

   At last, Bob sends the proxy signature (R, U, S, m_w, m) to the verifier.
8. `Proxy Signature Verification`: After receiving (R, U, S, m_w, m), the verifier performs the following steps:
   (a) Check whether $e(X_{ID_A}, P_0) = e(Y_{ID_A}, P)$ holds.
   (b) Check whether $e(X_{ID_B}, P_0) = e(Y_{ID_B}, P)$ holds.
   (c) Compute $R' = e(P, S)e(Y_{ID_A}, -h_B(U + h_A Q_{ID_A}))e(Y_{ID_B}, -h_B Q_{ID_B})$, where $h_A = H_2(m_w||U)$ and $h_B = H_2(m||R)$.
   (d) Accept the proxy signature if and only if $h_B = H_2(m||R')$.

5.2 Attack on the Li *et al.* CLPS Scheme

Now, we show that the Li *et al.* certificateless proxy signature scheme is in fact vulnerable to the public key replacement attack against the Type I adversary. 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. This attack is essentially the similar attack mounted by Huang *et al.* [13] against the Al-Riyami and Paterson CLS scheme [1]. More precisely, this strong attack is the universal forgery against no message attack where no signing oracle is required in the Type I adversarial model and the forger can sign any message at will.

We now describe the efficient algorithm used to mount the public key replacement attack against the Li *et al.* CLPS scheme below. This efficient algorithm enables the forger to sign any message at will.
`Sign`: To sign a message m and a warrant m_w on identities ID_A and ID_B, the Type I adversary performs the following steps:

1. Select a random $U, S \in \mathbb{G}_1$ and compute $h_A = H_2(m_w||U)$.
2. Select a random $r \in \mathbb{Z}_q^*$.

3. Compute $R = e(P, S)e(P_0, -(U + h_A Q_{ID_A}))e(rP_0, -Q_{ID_B})$.
4. Compute $h_B = H_2(m, R)$.
5. Set $x_{ID_A} = h_B^{-1} \in \mathbb{Z}_q^*$ and $x_{ID_B} = h_B^{-1} \cdot r \in \mathbb{Z}_q^*$.
6. Compute $X'_{ID_A} = x_{ID_A}P$, $Y'_{ID_A} = x_{ID_A}P_0$, $X'_{ID_B} = x_{ID_B}P$, $Y'_{ID_B} = x_{ID_B}P_0$.
7. Replace the user public key with $\langle X'_{ID_A}, Y'_{ID_A}, X'_{ID_B}, Y'_{ID_B} \rangle$.
8. Return the proxy signature (R, U, S, m_w, m).

The forged signature of message m and warrant m_w on identities ID_A and ID_B is valid the forged signature can be verified as follows:

1. Check whether $e(X'_{ID_A}, P_0) = e(Y'_{ID_A}, P)$ and $e(X'_{ID_B}, P_0) = e(Y'_{ID_B}, P)$ hold. If not, return *Error* and abort the verification. Notice that

$$\begin{aligned} e(X'_{ID_A}, P_0) &= e(x_{ID_A}P, sP) \\ &= e(x_{ID_A}sP, P) \\ &= e(Y'_{ID_A}, P) \end{aligned}$$

$$\begin{aligned} e(X'_{ID_B}, P_0) &= e(x_{ID_B}P, sP) \\ &= e(x_{ID_B}sP, P) \\ &= e(Y'_{ID_B}, P) \end{aligned}$$

2. Compute $R' = e(P, S)e(Y_{ID_A}, -h_B(U + h_A Q_{ID_A}))e(Y_{ID_B}, -h_B Q_{ID_B})$.
3. Accept the signature if and only if $h_B = H_2(m, R')$ holds.

$$\begin{aligned} R' &= e(P, S)e(Y_{ID_A}, -h_B(U + h_A Q_{ID_A}))e(Y_{ID_B}, -h_B Q_{ID_B}) \\ &= e(P, S)e(x_{ID_A}P_0, -h_B(U + h_A Q_{ID_A}))e(x_{ID_B}P_0, -h_B Q_{ID_B}) \\ &= e(P, S)e(h_B^{-1}P_0, -(U + h_A Q_{ID_A}))^{h_B} e(h_B^{-1}rP_0, -Q_{ID_B})^{h_B} \\ &= e(P, S)e(P_0, -(U + h_A Q_{ID_A}))^{h_B \cdot h_B^{-1}} e(rP_0, -Q_{ID_B})^{h_B \cdot h_B^{-1}} \\ &= e(P, S)e(P_0, -(U + h_A Q_{ID_A}))e(rP_0, -Q_{ID_B}) \\ &= R \end{aligned}$$

Since $R' = R$ holds, then $h_B = H_2(M, R')$ holds too.

The public key of ID_B is different from the public key of ID_A since a random r is included.

6 Conclusion

We mounted some attacks on three proxy signature schemes without certificates, they are the Qian and Cao IBPS scheme, the Guo *et al.* IBPS scheme and the Li *et al.* CLPS scheme. From the above security analyses, we may conclude that the security of a proxy signature scheme deriving from a signature scheme is not guaranteed even though the underlying signature scheme is provably secure. Thus, extra caution must be exercised in extracting this kind of scheme.

References

1. S.S. Al-Riyami and K.G. Paterson. Certificateless Public Key Cryptography. *In Proceedings of ASIACRYPT 2003*, LNCS 2894, pp. 452-473, Springer-Verlag, 2003.
2. A. Bakker, M. Steen and A.S. Tanenbaum. A Law-abiding Peer-To-Peer Network for Free-Software Distribution. *In Proceedings of NCA 2001*, pp. 60-67, IEEE, 2001.
3. A. Boldyreva, A. Palacio and B. Warinschi. Secure Proxy Signature Schemes for Delegation of Signing Rights. *Cryptography ePrint Archive*, http://eprint.iacr.org/2003/096.
4. J. Cha and J. Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups. *In Proceedings of PKC 2003*, LNCS 2567, pp. 18-30, Springer-Verlag, 2003.
5. S.S.M. Chow, R.W.C. Liu, L.C.K. Hui and S.M. Yiu. Identity-Based Delegation Network. *In Proceedings of Mycrypt 2005*, LNCS 3715, pp. 99-115, Springer-Verlag, 2005.
6. W. Diffie and M. Hellman. New Directions in Cryptography.. *IEEE Transactions on Information Theory*, 22(6), pp. 644-654, 1976.
7. I. Foster, C. Kesselman, G. Tsudik and S. Tuecke. A Security Atchitecture for Computational Grids. *In Proceedings of CCS 1998*, pp. 83-92, ACM Press, 1998.
8. S. Goldwasser, S. Micali and R. Rivest. A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks. *SIAM Journal of Computing*, vol. 17, no. 2, pp. 281–308, 1988.
9. C. Gu and Y. Zhu. Provable Security of ID-Based Proxy Signature Schemes. *In Proceedings of ICCNMC 2005*, LNCS 3619, pp. 1277-1286, Springer-Verlag, 2005.
10. C. Gu and Y. Zhu. An Efficient ID-Based Proxy Signature Scheme from Pairings. *Cryptology ePrint Archive*, http://eprint.iacr.org/2006/158.
11. S. Guo, Z. Cao and R. Lu. An Efficient ID-Based Multi-Proxy Multi-Signature Scheme. *In Proceedings of IMSCCS 2006*, Volume 2, pp. 81-88, IEEE, 2006.
12. F. Hess. Efficient Identity Based Signature Schemes based on Pairings. *In Proceedings of SAC 2003*, LNCS 2595, pp. 310-324, Springer-Verlag, 2003.
13. X. Huang, W. Susilo, Y. Mu and F. Zhang. On the Security of Certificateless Signature Schemes from Asiacrypt 2003. *In Proceedings of CANS 2005*, LNCS 3810, pp. 13-25, Springer-Verlag, 2005.
14. S. Kim, S. Park and D. Won. Proxy Signatures, Revisited. *In Proceedings of ICICS 1997*, LNCS 1334, pp. 223-232, Springer-Verlag, 1997.
15. B. Lee, H. Kim and K. Kim. Strong Proxy Signature and Its Applications. *In Proceedings of SCIS 2001*, Vol. 2/2, pp. 603-608, 2001.
16. B. Lee, H. Kim and K. Kim. Secure Mobile Agent Using Strong Non-Designated Proxy Signature. *In Proceedings of ACISP 2001*, LNCS 2119, pp. 474-486, Springer-Verlag, 2001.
17. X. Li, K. Chen and L. Sun. Certificateless Signature and Proxy Signature Schemes from Bilinear Pairings. *Lithuanian Mathematical Journal*, Vol 45(1), pp. 76-83, Springer-Verlag, 2005.
18. M. Mambo, K. Usuda and E. Okamoto. Proxy Signatures: Delegation of the Power to Sign Messages. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E79-A, No 9, pp. 1338-1354, 1996.
19. T. Okamoto, M. Tada and E. Okamoto. Extended Proxy Signatures for Smart Cards. *In Proceedings of ISW 1995*, LNCS 1729, pp. 247-258, Springer-Verlag, 1999.

20. D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. *In Proceedings of EUROCRYPT 1996*, LNCS 1070, pp. 387-398, Springer-Verlag, 1996.
21. H. Qian and Z. Cao. A Novel ID-Based Partial Delegation with Warrant Proxy Signature Scheme. *In Proceedings of ISPA 2005*, LNCS 3759, pp. 323-331, Springer-Verlag, 2005.
22. A. Shamir. Identity Based Cryptosystems and Signature Scheme. *In Proceedings of CRYPTO 1984*, LNCS 196, pp. 47-53, Springer-Verlag, 1984.
23. H.−M. Sun and B.−T. Hsieh. On the Security of Some Proxy Signature Schemes. *Cryptology ePrint Archive*, http://eprint.iacr.org/2003/068.
24. Q. Wang and Z. Cao. Efficient ID-Based Proxy Signature and Proxy Signcryption from Bilinear Pairings. *In Proceedings of CIS 2005*, LNAI 3802, pp. 167-172, Springer-Verlag, 2005.
25. G. Wang, F. Bao, J. Zhou and R.H. Deng. Security Analysis of Some Proxy Signatures. *In Proceedings of ICISC 2003*, LNCS 2971, pp. 305-319, Springer-Verlag, 1999.
26. J. Xu, Z. Zhang and D. Feng. ID-Based Proxy Signature Using Bilinear Pairings. *In Proceedings of ISPA 2005*, LNCS 3759, pp. 359-367, Springer-Verlag, 2005.
27. H.J. Yoon, J.H. Cheon and Y. Kim. A New Identity-Based Signature Scheme with Batch Verification. *In Proceedings of ICISC 2004*, LNCS 3506, pp. 233-248, Springer-Verlag, 2004.
28. F. Zhang and K. Kim. Efficient ID-Based Blind Signature and Proxy Signature from Bilinear Pairing. *In Proceedings of ACISP 2003*, LNCS 2727, pp. 312-323, Springer-Verlag, 2003.
29. K. Zhang. Threshold Proxy Signature Schemes. *In Proceedings of ISW 1997*, LNCS 1396, pp. 272-290, Springer-Verlag, 1997.