ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2012, Vol.41, No.1

A VERIFIABLE PROXY SIGNATURE SCHEME BASED ON BILINEAR PAIRINGS WITH IDENTITY-BASED CRYPTOGRAPHIC APPROACHES

Ya-Fen Chang[1], Wei-Liang Tai[2,*], and Chung-Yi Lin[3]

[1] *Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung, Taiwan*
e-mail: cyf@nutc.edu.tw; daniel0326@xuite.net

[2] *Department of Biomedical Informatics, Asia University, Taichung, Taiwan*
e-mail: taiwl@cs.ccu.edu.tw

[3] *Department of Biomedical Informatics, Asia University, Taichung, Taiwan*
e-mail: taiwl@cs.ccu.edu.tw; cyf@cs.ccu.edu.tw

crossref http://dx.doi.org/10.5755/j01.itc.41.1.830

Abstract. Hu and Huang proposed an identity-based proxy signature scheme with bilinear pairings. By this approach, the extra burden of verifying a public key with a certificate can be eliminated, and the length of a digital signature can be 160 bits only. Later, Park et al. pointed out that Hu and Huang's scheme suffers from one serious problem, privacy problem, such that a proxy key is generated by using a designated proxy signer's private key without his agreement. To solve this problem, Park et al. also proposed an improvement. With deep insight into Park et al.'s improvement, two drawbacks are found. First, a designated proxy signer may be fooled. Second, the verification of the proxy key in Park et al.'s scheme will never succeed. To preserve advantages and overcome drawbacks, an enhancement will be proposed in this paper.

Keywords: identity-based cryptosystem; proxy signature; privacy.

1. Introduction

Identity-based cryptosystem (IDC) was first introduced by Shamir in 1984 [19]. IDC provides a simple way to eliminate the extra burden of verifying a public key with a certificate. In IDC, there exists a trustworthy private key generator (*PKG*). Each user has to register his identity with *PKG* in advance. When a user's registration request is accepted, *PKG* will generate the user's private key according to his identity which is the corresponding public key. The benefit of identity-based cryptosystem is that a user's public key can be directly calculated by his identity instead of being extracted from a certificate issued by a certificate authority. Note that a user's private key is computed by a trustworthy party, *PKG*.

The concept of proxy signature was first introduced by Mambo et al. in 1996 [16]. In a proxy signature scheme, there are three entities: an original signer, a proxy signer and a verifier. The original signer can delegate his signing capacity to a designated person who is called a proxy signer. The proxy signer can generate valid signatures on behalf of the original signer. A verifier can determine the original signer and the proxy signer by verification equations. Thus, a valid proxy signature is generated by a proxy signer, and the original signer cannot deny his delegation. That is, a proxy signature scheme must possess the essential security property: non-repudiation. According to Mambo et al.'s statement [16], delegation of proxy signatures can be classified into three types: full delegation, partial delegation and delegation with warrant. Among them, the third type is the most common one.

Thereupon, several types of proxy signature schemes have been proposed [1, 5-8, 12, 13, 20, 21]. Recently, Hu and Huang proposed an ID-based proxy signature scheme, Hu-Huang scheme, with bilinear pairings [10]. With bilinear pairings, the length of a digital signature can be 160 bits only [2, 3]. Short

* Corresponding author

signature schemes possess a great advantage because the needed bandwidth is small. Because of the superior properties of IDS and bilinear pairings, various ID-based applications using bilinear pairings have been proposed [11, 14, 18, 22]. In Hu-Huang scheme, the proxy public key is computed from a warrant. However, Park et al. pointed out that Hu-Huang scheme suffers from one serious problem, privacy problem, such that *PKG* generates a proxy key by using a designated proxy signer's private key without his agreement [17]. To solve this problem, Park et al. also proposed an improvement. With deep insight into Park et al.'s scheme, some drawbacks are observed. First, a designated proxy signer cannot make sure whether the delegation request is indeed sent from the original signer indicated in the warrant. Thus, he may be fooled and execute some computation operations until the verification of the proxy key fails. Via fooling attack, only *PKG* knows who the real original signer is. If no auditing mechanism is employed, the malicious user will never be detected. Second, the verification of the proxy key in Park et al.'s scheme will never succeed.

To preserve its advantages and overcome its drawbacks, an enhancement will be proposed in this paper. Moreover, this enhancement should provide properties which a strong proxy signature scheme should provide. In 2001, Lee et al. [15] defined five properties that a strong proxy signature scheme should provide: (1) strong unforgeability, (2) verifiability, (3) strong identifiability, (4) strong undeniability, and (5) prevention of misuse. Except the above five properties, privacy preservation and request verification should also be preserved to overcome privacy problem and fooling attack, respectively. Thus, details of seven essential properties are listed as follows:

(1) **Strong unforgeability**: A designated proxy signer can generate a valid proxy signature for the original signer while unauthorized parties including the original signer cannot generate a valid proxy signature.
(2) **Verifiability**: While verifying a proxy signature, a verifier can be convinced that a proxy signature of one message is generated under the original signer's authorization.
(3) **Strong identifiability**: Anyone can determine the designated proxy signer's identity from a proxy signature.
(4) **Strong undeniability**: If a proxy signer generates a valid proxy signature on behalf of the original signer, the proxy signer cannot repudiate his proxy signature generation.
(5) **Prevention of misuse**: The proxy key pair cannot be used for other purpose. The responsibility of a proxy signer should be determined explicitly.
(6) **Privacy preservation**: The designated proxy signer's private key will be used for generating a proxy key pair only when he agrees to accept delegation.
(7) **Request verification**: A designated proxy signer can verify the delegation request when receiving it.

The rest of this paper is organized as follows. Preliminaries of bilinear pairings are shown in Section 2. Then, we review Hu-Huang and Park et al.'s schemes and point out drawbacks of Park et al.'s scheme in Section 3. We propose our scheme in Section 4. Section 5 shows security analyses of the proposed scheme. Finally, some conclusions are given in Section 6.

2. Preliminaries

The basic definition and related mathematics of bilinear pairings are briefly described in this section. Let G_1 and G_2 be two cyclic groups of the same large prime order q. G_1 and G_2 are additive and multiplicative groups, respectively. A bilinear paring map e: $G_1 \times G_1 \rightarrow G_2$ is an admissible pairing which satisfies the following properties:

(1) Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_q$.
(2) Non-degenerate: There exist $Q, P \in G_1$ such that $e(Q, P) \neq 1$.
(3) Computable: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

e is symmetric such that $e(P, Q) = e(Q, P)$ for all $P, Q \in G_1$ because e is bilinear and G_1 is a cyclic group. Security of many bilinear-pairing-based cryptographic protocols is based on the hardness of bilinear Diffie-Hellman problem (BDHP). Now we describe some mathematical problems in bilinear pairings as follows:

(1) Discrete logarithm problem (DLP): Given $P, Q \in G_1$, find $n \in Z_q$ such that $P = nQ$ whenever such n exists.
(2) Computational Diffie-Hellman problem (CDHP): Given a triple $(P, aP, bP) \in G_1$ for $a, b \in Z_q$, find the element abP.
(3) Decision Diffie-Hellman problem (DDHP): For a, b, c $\in Z_q$, given P, aP, bP and cP, decide whether $c \equiv ab \pmod q$. DDHP is easy to be solved in polynomial time by verifying $e(aP, bP) = e(P, cP)$.

3. Related works and corresponding analyses

After Hu and Huang proposed a proxy signature scheme, Park et al. indicated that their scheme has a privacy problem and proposed an improvement. However, we find that Park et al.'s scheme cannot work and suffers from some drawbacks. In this section, we first review Hu-Huang scheme, the privacy problem of Hu-Huang scheme and Park et al.'s improvement. Second, we show why Park et al.'s

scheme cannot work and what drawbacks of their scheme are.

3.1. Review of Hu-Huang scheme

We review Hu-Huang scheme in this section. There are three entities in this scheme: an original signer, a proxy signer, and *PKG* (private key generator center). *PKG* is responsible for generating original/proxy signer private key and public key pairs. There are four phases in Hu-Huang's scheme: initialization, key pair generation, proxy key generation, and proxy signature generation and verification. The details are as follows:

3.1.1. Initialization phase.

In this phase, *PKG* selects system parameters and a master key as follows.

Step 1. *PKG* selects q, G_1, G_2, e, and P as defined in the previous section, where P is the generator of G_1.

Step 2. *PKG* selects two one-way hash functions $H_1:\{0,1\}^* \rightarrow G_1$ and $H_2:\{0,1\}^* \rightarrow Z_q$.

Step 3. *PKG* selects a random number $t \in Z_q^*$ and computes $P_{pub}=tP$. Then *PKG* keeps t secretly as the master key and publishes system parameters $\{G_1, G_2, e, q, P, P_{pub}, H_1, H_2\}$.

3.1.2. Key pair generation phase

In this phase, both the original signer and the proxy signer submit their identities ID_o and ID_p to *PKG*. *PKG* generates each user's public and private key pair as follows.

Step 1. *PKG* computes $Q_o=H_1(ID_o)$, $Q_p=H_1(ID_p)$, $S_o=tQ_o$, and $S_p=tQ_p$.

Step 2. *PKG* sends (Q_o, S_o) and (Q_p, S_p) to the original signer and the proxy signer, respectively.

Thereupon, the original signer and the proxy signer's public/private key pairs are denoted by (Q_o, S_o) and (Q_p, S_p), respectively.

3.1.3. Proxy key generation phase

As shown in Figure 1, when an original signer wants to delegate his signing capacity to a proxy signer, the original signer first generates and publishes a warrant W by using Hess's ID-based signature scheme [9]. Then, *PKG* generates the proxy key S_w. The details are as follows:

Step 1. The original signer creates a warrant W which consists of ID_o, ID_p, the message to be signed, and so on. The original signer publishes W, computes $S_1=H_2(W, S_o)$, and sends $\{W, S_1\}$ to *PKG*.

Step 2. *PKG* accepts $\{W, S_1\}$ by checking if $S_1=H_2(W, S_o)$ holds or not. If it holds, *PKG* computes $Q_w=H_1(W)$, $S_w=tQ_w$, and $S_2=S_w+S_p$ and sends $\{W, S_2\}$ to the proxy signer.

Step 3. The proxy signer computes $S_w'=S_2-S_p$ and checks if $e(S_w', P)=e(H_1(W), P_{pub})$. If it holds, the proxy signer accepts $\{W, S_2\}$ and keeps S_w' as the proxy key while Q_w is the proxy public key.

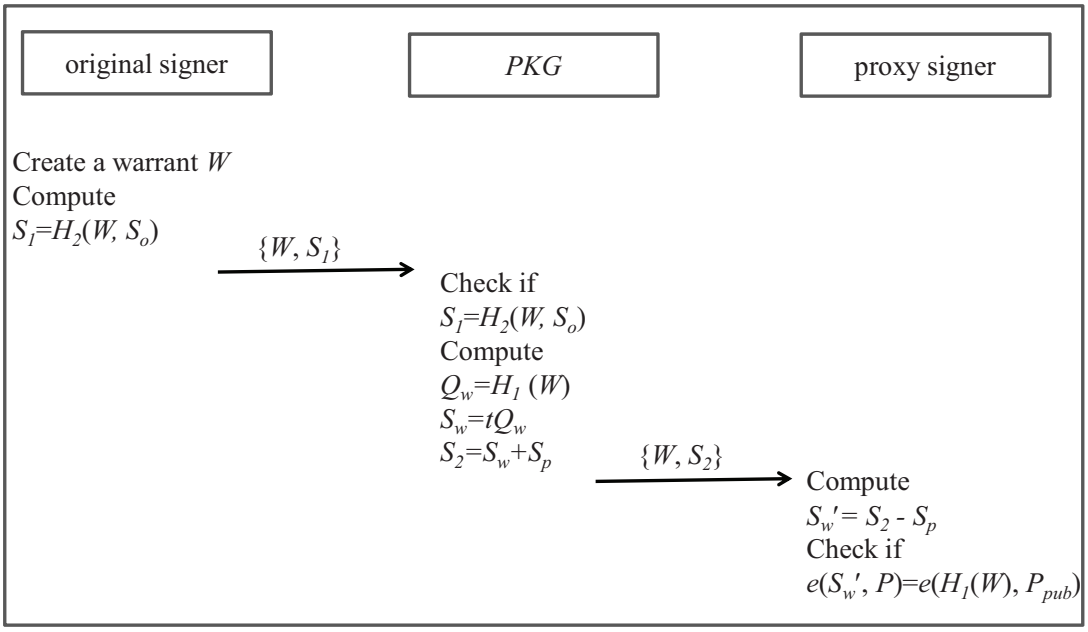

Figure 1. Proxy key generation phase of Hu-Huang scheme

3.1.4. Proxy signature generation and verification phase

As in [4], when the proxy signer wants to sign a message m, he first chooses a random number $r \in Z_q^*$ and computes $U=rQ_w$, $h=H_2(m, U)$, and $V=(r+h)S_w'$. Then a proxy signature (U, V) of the message m is generated. When a verifier gets m and (U, V), he checks if $e(V, P)=e(U+H_2(m, U)Q_w, P_{pub})$, where $Q_w=H_1(W)$. If it holds, the verifier ensures that (U, V) is indeed the proxy signature of m.

3.2. The privacy problem of Hu-Huang's scheme

Park et al. indicated that Hu-Huang's scheme infringes one of key principles of general privacy laws and regulations. It is because the proxy signer's private key is used without his agreement in proxy key generation phase. Park et al. mentioned that a malicious user, an original signer, can get information of a proxy signer's private key. By this approach, the security of proxy signer's private key may be damaged.

3.3. Review of Park et al.'s scheme

To overcome the drawback of Hu-Huang' scheme, Park et al. proposed an improvement. In their improvement, there exist three entities: an original signer, a proxy signer, and *PKG*. There are also four phases in Park et al.'s proxy signature scheme: initialization, key pair generation, proxy key generation, and proxy signature generation and verification. Because initialization, key pair generation, and proxy signature generation and verification phases are the same as those in Hu-Huang scheme, only proxy key generation phase is reviewed.

As shown in Figure 2, when an original signer wants to delegate his signing capacity to a proxy signer, the original signer first generates a warrant W. With the proxy signer's agreement, *PKG* generates the proxy key S_w. The details are as follows:

Proxy key generation phase

Step 1. The original signer creates a warrant W consisting of ID_o, ID_p, the message to be signed, and so on. The original signer makes W public and computes $S_1 = H_2(W, T, S_o)$, where T is a time stamp. He sends a delegation request $\{W, T, S_1\}$ to the designated proxy signer.

Step 2. If the designated proxy signer does not accept the delegation request from the original signer, this protocol is terminated immediately. Otherwise, he computes $S_1' = H_2(S_1, S_p)$ and sends $\{W, T, S_1'\}$ to *PKG*.

Step 3. After getting $\{W, T, S_1'\}$, *PKG* checks if $S_1'=H_2(H_2(W, T, S_o), S_p)$. If it holds, *PKG* accepts $\{W, T, S_1'\}$ and computes $Q_w=H_1(W)$, $S_w=tQ_w$, and $S_2=S_w+S_p$. Then *PKG* sends S_2 to the designated proxy signer.

Step 4. The designated proxy signer computes $S_w' = S_2+S_p$ and checks if $e(S_w', P) = e(H_2(W), P_{pub})$. If it holds, he accepts S_w' and keeps S_w' as a proxy key.

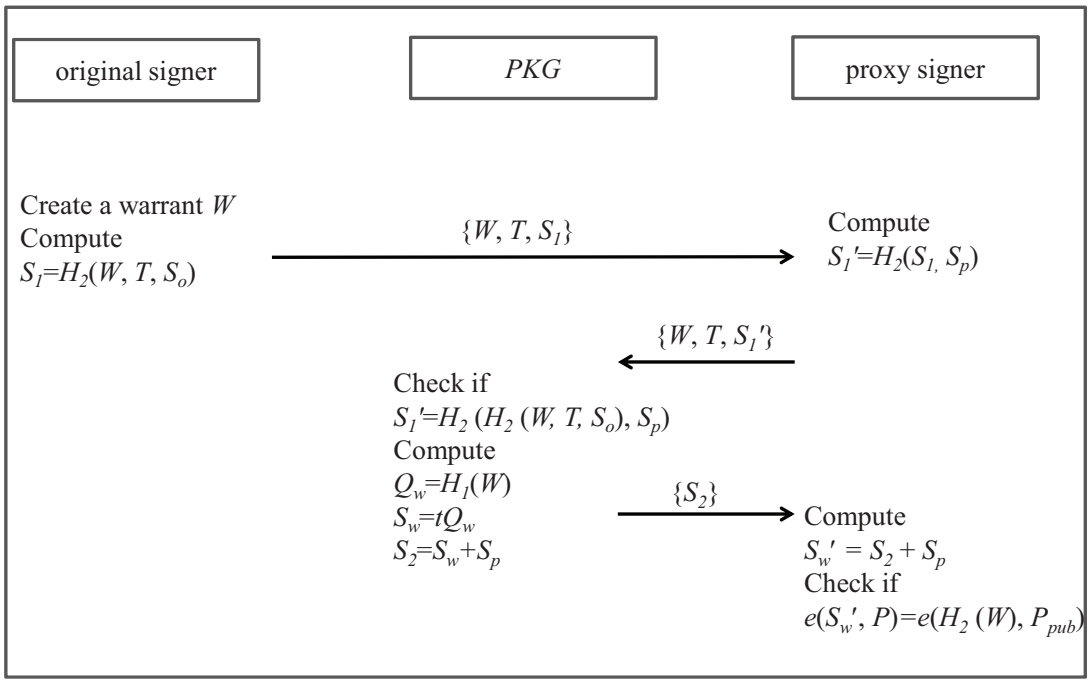

Figure 2. Proxy key generation phase of Park et al.'s scheme

3.4. Flaws of Park et al.'s scheme

After thorough investigation, it is found that there are two flaws in Park et al.'s scheme. First, the designated proxy signer cannot make sure whether the delegation request is indeed sent from the original signer indicated in the warrant. Therefore, he may be fooled and execute some computation operations until the verification of the proxy key fails. Via this fooling attack, if no auditing mechanism is employed, the malicious user will never be detected. This fooling attack is shown in Figure 3, and the details are as follows:

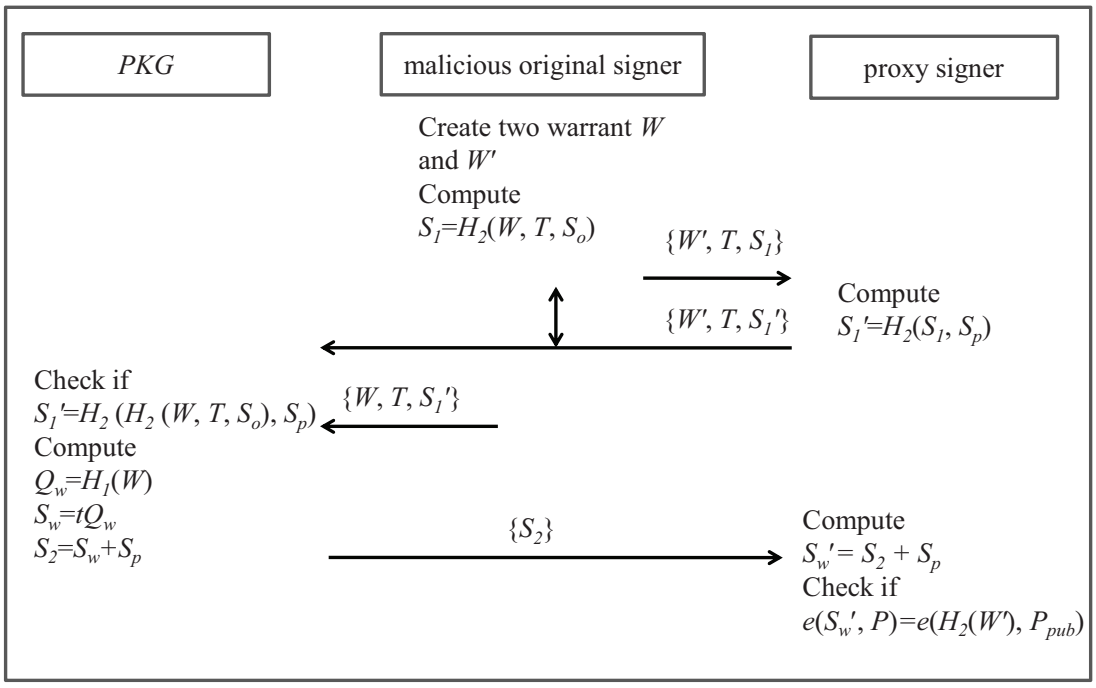

Figure 3. Fooling attack on Park et al.'s scheme

Step 1. When a malicious original signer wants to fool a proxy signer, he creates two warrants W and W'. W consists of ID_o, ID_p, the message to be signed, and so on while W' consists of ID_o', ID_p, the message to be signed, and so on, where ID_o' is another innocent original signer's identity. The malicious original signer computes $S_1=H_2(W, T, S_o)$, where T is a time stamp. He sends a delegation request $\{W', T, S_1\}$ to the designated proxy signer.

Step 2. If the designated proxy signer does not accept the delegation request from the original signer whose identity is ID_o', this protocol is terminated. Otherwise, he computes $S_1' = H_2(S_1, S_p)$ and sends $\{W', T, S_1'\}$ to *PKG*.

Step 3. The malicious original signer intercepts $\{W', T, S_1'\}$ and sends $\{W, T, S_1'\}$ to *PKG*.

Step 4. After getting $\{W, T, S_1'\}$, *PKG* checks if $S_1'=H_2 (H_2 (W, T, S_o), S_p)$. If it holds, *PKG* accepts $\{W, T, S_1'\}$ and computes $Q_w=H_1(W)$, $S_w=tQ_w$, and $S_2=S_w+S_p$. Then *PKG* sends S_2 to the designated proxy signer. Obviously, the verification will succeed, and *PKG* will send S_2 to the designated proxy signer.

Step 5. After getting S_2, the designated proxy signer computes $S_w'= S_2+S_p$ and checks if $e(S_w', P)=e(H_2(W'), P_{pub})$. However, the verification will fail.

Via the above fooling attack, the designated proxy signer will believe that the original signer who sends the delegation request owns an identity ID_o' instead of ID_o. Only *PKG* knows who the real original signer is. If no auditing mechanism is employed, the malicious user will never be detected.

Except the first fooling attack, Park et al.'s scheme has one fatal flaw. Actually, their scheme cannot work accurately because the verification of $e(S_w', P)=e(H_2(W), P_{pub})$ will never hold because $e(S_w', P) = e(S_2+S_p, P) = e(S_w+S_p+S_p, P) = e(tQ_w+2tQ_p, P) = e(tH_1(W)+2tH_1(ID_P), P) = e(H_1(W)+2H_1(ID_P), tP) = e(H_1(W)+2H_1(ID_P), P_{pub}) \neq e(H_2(W), P_{pub})$.

4. The proposed scheme

In this section, an enhancement will be proposed. Different from the previous two schemes, a secure

channel exists between *PKG* and signers for delivering public/private key pairs. In the proposed scheme, the content of an warrant is defined to contain ID_o, ID_p, types of messages to be signed, the valid delegation period, the delegation time stamp T, and the original signer's signature for the above data. In the proposed scheme, a delegated proxy signer can ensure that a delegation request is sent from the original signer mentioned in the warrant. There are four phases in our scheme: initialization, key pair generation, proxy key generation, and proxy signature generation and verification. Because initialization, key pair generation, and proxy signature generation and verification phases are the same as those in Hu-Huang scheme, we introduce the proposed proxy key generation phase only.

The proposed proxy key generation phase

After key pair generation phase, an original signer and a proxy signer have (Q_o, S_o) and (Q_p, S_p), respectively. When the original signer wants to delegate his signing capacity to a proxy signer, he needs to create a warrant and send a delegation request to the designated proxy signer. After the request is verified, *PKG* will help the proxy signer to generate a proxy key. This phase is shown in Figure 4, and the details are as follows:

Step 1. The original signer creates a warrant W, where $W=(W_1 \| W_2)$ .W_1 includes ID_o, ID_p, *type*, *period*, and the delegation time stamp T, where *type* is types of messages to be signed and *period* is the valid delegation period. $W_2 = (U \| V)$, where $r \in Z_q^*$, $U=rQ_o$, $h=H_2(W_1, U)$, and $V=(r+h)S_o$.

Step 2. The original signer computes $S_1=H_2(W, S_o)$ and sends $\{W, S_1\}$ to the proxy signer.

Step 3. First, the proxy signer checks if $T'- T \leqq \Delta T$, where T' is the current time stamp and ΔT is the expected time interval. If it does not hold, this phase terminates; otherwise, the proxy signer verifies the warrant W by checking if $e(V, P)=e(U+H_2(W_1,U)Q_o, P_{pub})$. If it holds, it denotes that the delegation request is indeed sent from the original signer, and the proxy signer computes $S_1' = H_2(S_1, S_p, W, T')$. Finally, the proxy signer sends $\{W, T', S_1'\}$ to *PKG*.

Step 4. After receiving $\{W, T', S_1'\}$, *PKG* checks if $T''- T' \leqq \Delta T$, where T'' is the current time stamp and ΔT is the expected time interval. If it does not hold, this phase terminates; otherwise, *PKG* checks if $S_1'=H_2(H_2(W, S_o), S_p, W, T')$. If it holds, *PKG* accepts the delegation request from the original signer.

Step 5. *PKG* accepts $\{W, T', S_1'\}$ and computes $Q_w=H_1(W)$, $S_w=tQ_w$ and $S_2=S_w+S_p$. Then, *PKG* sends S_2 to the designated proxy signer.

Step 6. The designated proxy signer computes $S_w' = S_2-S_p$ and checks if $e(S_w', P)=e(H_1(W), P_{pub})$. If it holds, he accepts S_w' and keeps S_w' as a proxy key.

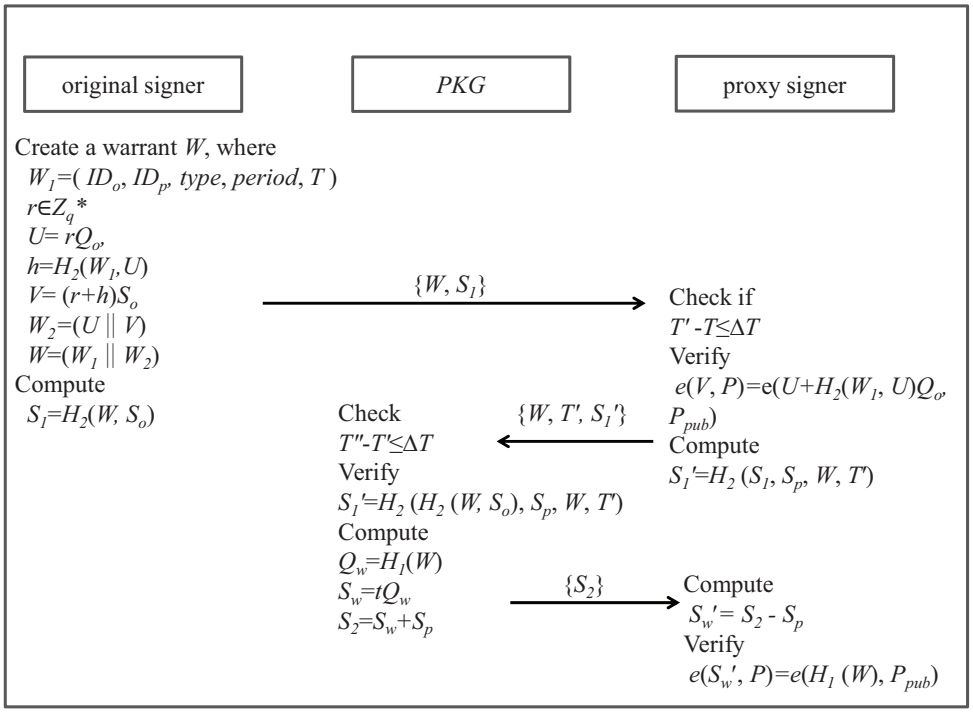

Figure 4. The proposed proxy key generation phase

5. Security analyses

In this section, seven properties, strong unforgeability, verifiability, strong identifiability, strong undeniability, prevention of misuse, privacy preservation and request verification, are first demonstrated to analyze the security of the proposed scheme. At last, the property, correctness, is demonstrated to show the proposed scheme can work accurately. The details are as follows:

Strong unforgeability: A designated proxy signer can generate a valid proxy signature for the original signer while unauthorized parties including the original signer cannot generate a valid proxy signature.

Proof: In the proposed proxy key generation phase, *PKG* computes $Q_w=H_1(W)$, $S_w=tQ_w$, and $S_2=S_w+S_p$, and sends S_2 to the designated proxy signer. After receiving S_2, the designated proxy signer computes $S_w'= S_2-S_p$ and checks if $e(S_w', P)=e(H_1(W), P_{pub})$. If it holds, he accepts S_w' and keeps S_w' as a proxy key. By this approach, only the designated proxy signer can retrieve S_w' with his own private key S_p. Because only the designated proxy signer has S_p, it denotes that only the designate proxy signer can obtain S_w'. Thus, no one except the designated proxy signer can generate a valid key pair under the name of the proxy signer, and strong unforgeability is ensured in the proposed scheme.

Verifiability: While verifying a proxy signature, a verifier can be convinced that a proxy signature of one message is generated under the original signer's authorization.

Proof: In the proposed proxy key generation phase, the original signer creates a warrant W, where $W=(W_1 \| W_2)$ .W_1 includes ID_o, ID_p, *type*, *period*, and the delegation time stamp T, where *type* is types of messages to be signed and *period* is the valid delegation period. $W_2 = (U \| V)$, where $r \in Z_q^*$, $U=rQ_o$, $h=H_2(W_1, U)$, and $V=(r+h)S_o$. This denotes that the original signer has generated a signature W_2 for delegation information W_1. Thus, a warrant W is composed of delegation information and the corresponding signature. Thereupon, *PKG* computes $Q_w=H_1(W)$, $S_w=tQ_w$, and $S_2=S_w+S_p$ and sends S_2 to the designated proxy signer. S_w is a proxy key while Q_w is its corresponding public key. As a result, when a verifier verifies a proxy signature with Q_w, the original signer cannot deny his delegation. Verifiability is ensured.

Strong identifiability: Anyone can determine the designated proxy signer's identity from a proxy signature.

Proof: In proxy signature generation and verification phase, when the proxy signer wants to sign the message m, he first chooses a random number $r \in Z_q^*$ and computes $U=rQ_w$, $h=H_2(m, U)$, and $V=(r+h)S_w'$. Then a proxy signature (U, V) of the message m is generated. When a verifier gets m and (U, V), he checks if $e(V, P)=e(U+H_2(m, U)Q_w, P_{pub})$. If it holds, the verifier ensures that (U, V) is indeed the proxy signature of m. Because $Q_w=H_1(W)$, $W=(W_1 \| W_2)$ and W_1 includes ID_p, anyone can easily determine who the proxy signer is by the warrant W. As a result, strong identifiability is ensured.

Strong undeniability: If a proxy signer generates a valid proxy signature on behalf of the original signer, the proxy signer cannot repudiate his proxy signature generation.

Proof: As mentioned above, only the designated proxy signer can retrieve S_w' with his own private key S_p. Because only the designated proxy signer has S_p, it denotes that only the designate proxy signer can obtain S_w'. Thus, no one except the designated proxy signer can generate a valid key pair under the name of the proxy signer. Consequently, only the proxy signer can generate a valid proxy signature with S_w' because only he knows S_w'. As a result, strong undeniability is ensured in the proposed scheme.

Prevention of misuse: The proxy key pair cannot be used for other purpose. The responsibility of a proxy signer should be determined explicitly.

Proof: In the proposed proxy key generation phase, the original signer creates a warrant W, where $W=(W_1 \| W_2)$. W_1 includes ID_o, ID_p, *type*, *period*, and the delegation time stamp T, where *type* is types of messages to be signed and *period* is the valid delegation period. $W_2 = (U \| V)$, where $r \in Z_q^*$, $U=rQ_o$, $h=H_2(W_1, U)$, and $V=(r+h)S_o$. This denotes that the original signer has generated a signature W_2 for delegation information W_1. As a result, a warrant W includes detailed delegation information. Thus, prevention of misuse is ensured.

Privacy preservation: The designated proxy signer's private key will be used for generating a proxy key pair only when he agrees to accept the delegation.

Proof: In the proposed proxy key generation phase, the proxy signer computes $S_1' = H_2(S_1, S_p, W, T')$ after verifying the warrant. The proxy signer sends $\{W, T', S_1'\}$ to *PKG*. After receiving $\{W, T', S_1'\}$, *PKG* checks if $S_1'=H_2(H_2(W, S_o), S_p, W, T')$. If it holds, *PKG* accepts the delegation request from the original signer. This verification approach ensures (1) the request is indeed sent from the original signer and (2) the proxy signer indeed agrees to accept the delegation. It is because both the original signer and the proxy signer's private keys are involved for verification. Thus, privacy preservation is ensured in the proposed scheme.

Request verification: A designated proxy signer can verify the delegation request when receiving it.

Proof: In the proposed proxy key generation phase, after receiving $\{W, S_1\}$, the proxy signer checks if $T'-T \leqq \Delta T$, where T' is the current time stamp and ΔT is the expected time interval. If it does not hold, this phase terminates; otherwise, the proxy signer verifies the warrant W by checking if $e(V,$

P)=$e(U+H_2(W_1,U)Q_o, P_{pub})$. If it holds, it denotes that the delegation request is indeed sent from the original signer because the original signer has signed the delegation information. Thus, request verification is ensured in the proposed scheme.

Correctness: The proposed scheme can work accurately.

Proof: In the proposed proxy key generation phase, after receiving $\{W, S_1\}$, the proxy signer checks if T'-$T \leqq \Delta T$, where T' is the current time stamp and ΔT is the expected time interval. If it does not hold, this phase terminates; otherwise, the proxy signer verifies the warrant W by checking if $e(V, P)=e(U+H_2(W_1,U)Q_o, P_{pub})$. Because $U=rQ_o$, $h=H_2(W_1, U)$, and $V=(r+h)S_o$, $e(V, P) = e((r+h)S_o, P) = e((r+h)tQ_o, P) = e((r+h)Q_o, tP) = e((r+ H_2(W_1, U))Q_o, tP) = e(rQ_o+H_2(W_1, U)Q_o, tP) = e(U+H_2(W_1, U)Q_o, tP) = e(U+H_2(W_1,U)Q_o, P_{pub})$. Thus, if $\{W, S_1\}$ is indeed sent by the original signer, $e(V, P)=e(U+H_2(W_1,U)Q_o, P_{pub})$. After receiving $\{W, T', S_1'\}$, PKG checks if T''-$T' \leqq \Delta T$, where T'' is the current time stamp and ΔT is the expected time interval. If it does not hold, this phase terminates; otherwise, PKG checks if $S_1' = H_2(H_2(W, S_o), S_p, W, T')$. Because $S_1=H_2(W, S_o)$ and $S_1' = H_2(S_1, S_p, W, T')$, $S_1' = H_2(H_2(W, S_o), S_p, W, T')$ must hold if the delegation request is indeed sent by the original and proxy signers. If PKG accepts $\{W, T', S_1'\}$, PKG computes $Q_w=H_1(W)$, $S_w=tQ_w$ and $S_2=S_w+S_p$, and sends S_2 to the designated proxy signer. The designated proxy signer computes $S_w'= S_2-S_p$ and checks if $e(S_w', P)=e(H_1(W), P_{pub})$. Because $e(S_w', P) = e(S_2-S_p, P) = e(S_w+S_p-S_p, P) = e(S_w', P) = e(tQ_w, P) = e(Q_w, tP) = e(Q_w, P_{pub})=e(H_1(W), P_{pub})$. That is, if S_2 is indeed computed by PKG according to the delegation request, the designated proxy signer can obtain S_w' as a proxy key successfully. Thus, correctness is ensured.

6. Conclusions

Hu and Huang employed bilinear pairings to propose an identity-based proxy signature scheme. Hu-Huang scheme possesses the following two advantages. (1) The extra burden of verifying a public key with a certificate can be eliminated. (2) The length of a digital signature can be 160 bits only. Later, Park et al. pointed out that Hu-Huang scheme suffers from privacy problem. To solve this problem, Park et al. also proposed an improvement. With deep insight into Park et al.'s improvement, we find two drawbacks. First, a designated proxy signer may be fooled. Second, the verification of the proxy key in Park et al.'s scheme will never succeed. To preserve advantages and overcome drawbacks, we have proposed an enhancement. This improvement enables the designated proxy signer to verify if the delegation request is indeed sent from the original signer indicated in a warrant. Thus, the proxy signer will not be fooled to execute lots computation operations. Though the computation load is heavier than that of Park et al.'s scheme, it is necessary to ensure the security of the proposed scheme because such approaches are widely used in many existing protocols such as EAP-LS, EAP-LLS and PEAP for WiFi and https for secure web access. We have shown the proposed scheme achieves the seven essential properties: (1) strong unforgeability, (2) verifiability, (3) strong identifiability, (4) strong undeniability, (5) prevention of misuse, (6) privacy preservation, and (7) request verification. Moreover, the proposed scheme can work accurately according to the property, correctness. Thus, this improvement preserves advantages of Park et al.'s scheme and overcomes its drawbacks. By the possessed advantages, the proposed scheme can be used in environments of limited bandwidth.

References

[1] **A. K. Awasthi, S. Lal.** "Proxy blind signature scheme," *Transaction on Cryptology*, Vol. 2, No. 1, January 2005, pp. 5-11.

[2] **D. Boneh, M. Franklin.** "Identity based encryption from the Weil pairing," *Advances in Cryptology-Cryto'01*, LNCS, Vol. 2139, Springer-Verlag, 2001, pp. 213-229.

[3] **D. Boneh, B. Lynn, H. Shacham.** "Short signatures from the Weil pairing," *Advances in Cryptology-Asiacryp'01*, LNCS, Vol. 2248, Springer-Verlag, 2001, pp. 514-532.

[4] **J. C. Cha, J. H. Cheon.** "An identity-based signature from gap Diffie–Hellman groups," *Proceedings of PKC'03*, LNCS, Vol. 2567, Springer-Verlag, 2003, pp. 18-30.

[5] **S. Chandrasekhar, S. Chakrabarti, M. Singhal, K. L. Calvert.** "Efficient proxy signatures based on trapdoor hash functions," *IET Information Security*, Vol. 4, No. 4, December 2010, pp. 322-332.

[6] **Y. F. Chang, C. C. Chang.** "Efficient multi-proxy multi-signature schemes based on DLP," *International Journal of Computer Science and Network Security*, Vol. 6, No. 2B, February 2006, pp. 152-159.

[7] **Y. F. Chang, C. C. Chang.** "An RSA-based (t,n) threshold proxy signature scheme with freewill identities," *International Journal of Information and Computer Security*, Vol. 1, No. 1/2, 2007, pp. 201-209.

[8] **Y. F. Chang, C. C. Chang.** "Robust t-out-of-n proxy signature based on RSA cryptosystems," *International Journal of Innovative Computing Information and Control*, Vol. 4, No. 2, February 2008, pp. 425-431.

[9] **F. Hess.** "Efficient identity based signature schemes based on pairings," *Proceedings of SAC'02*, LNCS, Vol. 2595, Springer-Verlag, 2002, pp. 310-324.

[10] **X. Hu, S. Huang.** "A novel proxy key generation protocol and its application," *Computer Standards & interfaces*, Vol. 29, 2007, 191-195.

[11] **M. Hölbl, T. Welzer, B. Brumen.** "An improved two-party identity-based authenticated key agreement protocol using pairings ," *Journal of Computer and System Sciences*, Article in Press, Available online 27 January 2011.

[12] J. Kar. Proxy Blind Multi-signature Scheme, LAP Lambert Academic Publishing AG & Co KG, ISBN: 9783844382266, 2011.

[13] Y. S. Kim, J. H. Chang. "Self *proxy signature* scheme," *IJCSNS International Journal of Computer Science and Network Security*, Vol. 7, No. 2, February 2007, pp. 335-338.

[14] J. S. Lee, J. H. Chang, D. H. Lee. "Forgery attacks on Kang et al.'s identity-based strong designated verifier signature scheme and its improvement with security proof," *Computers and Electrical Engineering*, Vol. 36, 2010, pp. 948-954.

[15] B. Lee, H. Kim, K. Won. "Secure mobile agent using strong non-designated proxy signature," *Proceedings of ACISP'01*, LNCS, Vol. 2119, Springer-Verlag, 2001, pp. 474-486.

[16] M. Mambo, K. Usuda, E. Okamoto. "Proxy signature: delegation of the power to sign messages," *IEICE Transactions on Fundamentals*, Vol. E79-A, No. 9, 1996, pp. 1338-1354.

[17] H. Park, S. Lim, I. Yie. "A privacy problem on Hu-Huang's proxy key generation protocol," *Computer Standards & Interfaces,* Vol. 31, 2009, pp. 480-483.

[18] S. S. D. Selvi, S. S. Vivek, S. Gopinath, C. P. Rangan. "Identity based self delegated signature - self proxy signatures," *Proceedings of 2010 Fourth International Conference on Network and System Security*, Melbourne, Australia, 2010, pp.568-573.

[19] A. Shamir. "Identity-based cryptosystems and signature schemes," *Advances in Cryptology – Crypto'84*, LNCS, Vol. 196, Springer-Verlag, 1984, pp.47-53.

[20] K. A. Shim. "Short designated verifier proxy signature," *Computers and Electrical Engineering*, Vol. 37, No. 2, 2011, pp. 180-186.

[21] Q. Xie. "Improvement of a self *proxy signature* scheme," *Applied Mechanics and Materials*, Vol. 40-41, 2011, pp. 643-646.

[22] E. J. Yoon. "An efficient and secure identity-based strong designated verifier signature scheme," *Information Technology and Control*, Vol. 40, No. 4*,* 2011, pp. 323-329.

Received November 2011.