A Secure Strong Proxy Signature Scheme with Proxy Signer Privacy Protection

Youn-Ho Lee, Seong-Min Hong and Hyunsoo Yoon
Division of Computer Science, Department of Electric Engineering and Computer Science
Korea Advanced Institute of Science and Technology, Taejon, Korea
Email: (yhlee, smhong, hyoon)@camars.kaist.ac.kr

ABSTRACT
A proxy signature scheme enables an entity to delegate its signing rights to another entity. Since the proxy signature concept was proposed by Mambo et al., many variant schemes have been appeared. In this paper, we propose a new proxy signature scheme with a proxy signer's privacy protection. In proxy signer's privacy scenario, the identity of the proxy signer cannot be revealed in a proxy signature unless the alias authority reveals it. The verifier only knows that the proxy signer is certified by the alias authority. The anonymity of the proxy signer gives an additional application such as electronic voting, electronic cash. In 2002, Shum et al. firstly proposed a scheme. But this scheme was broken because it is based on the LKK proxy signature scheme which is known to be insecure. We modify the triple Schnorr scheme and apply the modified result to our scheme.

Keywords: Proxy Signature, Digital Signature, Security, Privacy Protection

1. Introduction
The concept of proxy signature, introduced by Mambo, Usuda and Okamoto in 1996, provides a method for an original signer to delegate signing capabilities to a proxy signer [1]. By the capability of delegating signing rights, the proxy signature scheme can be applied to many practical applications, such as grid computing [2], mobile agent applications [3-4], distributed systems [5-6], global distributed networks [7], mobile communications [8]. Mambo et al. classified the proxy signature scheme according to the type of delegation as full delegation, delegation by certificate, and partial delegation [1].

In 1997, Kim et al. proposed a new type of proxy signature which is called "partial delegation with warrant" and suggested a new proxy signature scheme in this type [9]. The partial delegation with warrant type has advantages comparing to both partial delegation type and delegation by certificate because it needs less computational cost for the proxy signature verification than delegation-by-certificate schemes. Moreover, it has an ability of limiting the signing capability that partial delegation type does not have.

After the first proposal of the partial delegation with warrant type, many extensions of the basic scheme are proposed [8,11-12,14-15]. One of these extensions is the proxy signature scheme with proxy signer's privacy protection [11].

In 2002, Shum et al. firstly proposed a proxy signature scheme with proxy signer's privacy protection using the Lee et al.'s basic proxy signature scheme [4,11]. Unfortunately, Shum et al.'s scheme is analyzed and shown that without proxy signer's agreement, the original signer can generate the proxy signature as if the proxy signer generates it [16]. To the best of our knowledge, there exists no secure proxy signature scheme with preserving the privacy of the proxy signer.

Our contribution is the proposal of a new proxy signature scheme with proxy signer privacy protection. We modify the triple Schnorr proxy signature scheme and apply it to the proxy signer's privacy protection scenario. In most cases, the proxy signature scheme using the Schnorr signature at delegation phase is shown to be insecure because the random commitment which is used for a part of the Schnorr signature as delegation information can be abused by malicious original signer [13,16-17]. So, we concentrate on removing the effect of the random commitment at delegation phase.

The organization of our paper is as follows. In section 2, we present the preliminaries. In section 3, we describe the triple Schnorr proxy signature scheme. In section 4, we present our new proxy signature scheme with proxy signer's privacy protection. In section 5, we consider the security aspects of the proposed scheme. Finally, we give a conclusion in section 6.

2. Preliminary
In this section, we give the basic scenario and security requirement of the proxy signature scheme with proxy signer privacy protection. At next, we present the notations which are used in the paper. We provide some assumptions for our scheme at last of the section.

**Basic Scenario**

In the proxy signature scheme with proxy signer privacy protection, there are five steps [11,16]. These are **Issuing Alias, Delegation, Proxy Signature Generation, Proxy Signature Verification, and Revoking Privacy**. The detailed explanation of each step is as follows.

**Issuing alias:** In this step, the alias authority sends an alias and an anonymous private key to the proxy signer. The alias value represents an anonymous identity of the proxy signer and is used for generating the anonymous public key of the proxy signer. The alias value also can be used for finding the real identity of the proxy signer at privacy revocation phase.

**Delegation:** This step represents how the original signer delegates the signing rights to proxy signer. It is assumed that the anonymous identity of the proxy signer is known publicly. In this step, the original signer generates proxy information that includes a description of signing rights delegated. The proxy signer can verify that the proxy information is generated by the original signer.

**Proxy signature generation:** The proxy signer generates a proxy signing key using the proxy information and the anonymous private key of the proxy signer. After that, the proxy signer performs the proxy signature generation procedure for a message to be signed using the proxy signing key.

**Proxy signature verification:** The verifier who receives the proxy signature performs the proxy signature verification. This step consists of proxy public key generation and proxy signature verification. The proxy public key can be generated with the public keys of the original signer and the alias authority, and the part of the proxy signature. The proxy signature verification procedure is the same as the standard digital signature verification. In this procedure, the proxy public key is used as the public key for the standard digital signature verification.

**Revoking privacy:** If needed, the real identity of proxy signer can be revealed by sending the alias of the proxy signer to the alias authority.

**Security Requirement**

The security requirements of the proxy signature scheme with privacy protection are as follows [11]. Compared with the security requirements of the original proxy signature scheme [1,4], **Proxy privacy and Privacy revocation** are added and **Strong identifiability** is omitted [11].

**Verifiability:** A verifier can be convinced of the original signer’s agreement by the proxy signature.

**Strong unforgeability:** Only the designated proxy signer can generate a valid proxy signature.

**Strong non-repudiation:** A proxy signer cannot repudiate the valid proxy signature which the proxy signer generates.

**Proxy privacy:** The identity of the proxy signer cannot be revealed from the proxy signature without the alias authority’s help.

**Privacy revocation:** The scheme should provide a way to reveal the proxy signer’s identity, which is required in some exceptional cases.

**Notation**

The notations used for our paper are described below.

\[ M, P, T, V \] : original signer, proxy signer, alias authority and verifier respectively.
\[ \{0,1\}^* \] : set of all binary strings of finite length. \( \| \) : concatenation operation.
\[ p, q \] : prime numbers such that \( p > 2q > 1 \).
\[ g \] : generator of order \( q \) in \( Z_p \).
\[ h() \] : a secure hash function such that \( h(\{0,1\}^*) \in \{0,1\}^q \).
\[ g^{s_i} \mod p \] : public key of \( A \).
\[ (x_A, Z_A) \] : warrant which describes the signing rights delegated to the proxy signer.
\[ (Z_t, r_t) \] : Schnorr digital signature generated by \( T \). Later \( s_t \) will be used for anonymous proxy signing key by \( P \).
\[ (m, s_m) \] : information which describes the signing rights delegated to the proxy signer.
\[ (m, x) \] : discrete logarithm problem based signing algorithm. \( m \in \{0,1\} \) is a message to be signed and \( x \in Z_q^* \) is the private key used for signing. We assume that the signature is unforgeable and non-repudiable.
\[ (m, y) \] : discrete logarithm based signature verification algorithm which is matched to \( Sign(m, x) \). If \( Verify(y, m) \) returns 1, then \( y \) is the digital signature by the result of \( Sign(m, x) \).
Basic Assumption
In the rest of our paper, we assume that the prime number \( p, q \) and generator \( g \) are known publicly. The public key is known publicly and anyone can know who the owner of the given public key is. For removing the rouge key attack, we also assume that the owner of the public key has proved the possession of the matching private key at the public key registration step.

3. Triple Schnorr proxy signature scheme

In this section, we present the triple Schnorr scheme \cite{10}. The description of the operations is as follows.

A. Delegation: Following procedure is done for delegating the signing right of \( M \) to \( P \)

1) \( M \) generates a delegation warrant \( m_w \).
2) \( M \) generates a random number \( k \equiv Z_q \).
3) \( M \) computes \( r_q \equiv g^{s_M} \mod p \) and \( s_M \equiv e \mod q \).
4) \( M \) sends \( m_w, y_M, s_M, r_M \) to \( P \).
5) \( P \) verifies the validity of the delegation by checking the equality of \( g^{s_M} \mod p \equiv (y_M)^{(h(0||m_w||y_M||s_M))} \mod p \).

B. Proxy signature generation: \( P \) performs the following procedure for generating a proxy signature for a message \( m \in \{0,1\}^* \)

1) Generate proxy signing key \( x \equiv h(0||m||y_M||y_r||r_M) \mod q \).
2) Generate random number \( r_p \equiv R \equiv Z_q^* \) and Compute \( r_p \equiv g^{r_p} \mod p \).
3) Compute \( s_p \equiv h(1||m||y_M||y_r||r_p||r_M) \mod q \).
4) Output is \( M, m_w, r_M, s_p, r_p, y_M \).

C. Proxy signature verification: If a proxy signature \( (m, m_w, r_M, s_p, r_p, y_M) \) is sent by \( P \), \( V \) performs the following procedure for verifying the proxy signature

1) Generate proxy public key \( y_r \equiv y_M^{h(0||m||y_M||s_M)} \mod p \).
2) Check the equality of the following equation: \( g^{r_p} \equiv y_r^{h(1||m||y_M||y_r||r_p||r_M)} \mod p \).

4. Proposed Scheme

In this section, we present a new proxy signature scheme with proxy signer privacy protection. Our scheme can be viewed as an enhancement of the triple Schnorr proxy signature scheme. More specifically, we inserted “Issuing alias” stage before Delegation stage of the triple Schnorr proxy scheme and appropriately modified the other stages. Note that the security of triple Schnorr scheme can be reducible to that of the Schnorr digital signature scheme because the proxy public key is generated to the form of \( y_r \) (in Step 3-C-1)) which is used to the Schnorr signature verification; many proxy signature schemes which do not use this form for proxy public key have been found to be insecure [13,16-17]. Because the proxy public key in our scheme has the form of \( y^{h(0||m||y_M||s_M)} \mod p \) which is the form that the Schnorr digital signature is used twice. This enables us to prove security of our scheme s.t. security of our scheme is reducible to that of Schnorr digital signature (which will be shown in Section 5).

A. Issuing Alias: \( P \) gets his/her anonymous identity and anonymous private key from \( T \) by the following procedure.

1) \( P \) sends his identity \( ID_p \in \{0,1\}^* \) to \( T \).
2) \( T \) generates \( k \equiv R \equiv Z_q^* \) and computes \( P's \) anonymous
identity $h_p \equiv h(k_p, ID_p)$.

3) $T$ generates a random number $k_T \equiv R Z_q^*$. $r_T \equiv g^{k_T} \mod p$, and $s_T \equiv h(h_p, r_T) \mod q$.

4) $T$ sends $h_p, r_T, s_T$ to $P$ with a secure manner.

5) $P$ verifies the validity of received values by checking the following equation:

\[ g^{s_T} \mod p = y_T^{h(h_p, r_T)} \mod p \] Eq. (1)

6) $T$ stores $h_p, ID_p, k_p$ for later usage at the privacy revocation phase.

B. Delegation: $M$ delegates his/her signing rights to $P$ by the following procedure. This step is different from the previous approach because we do not use the Schnorr digital signature for delegation but modify it. Remind that we assume that original signer knows the proxy signer's alias $h_p, r_T$.

1) $M$ generates $m_w$ and random number $k_M \equiv R Z_q^*$. $r_M \equiv g^{k_M} \mod p$ and $s_M \equiv \{h(h_p, r_T) \mod q, h(m_w, r_M) \mod q\} \mod q$.

2) $M$ sends $m_w, r_M, s_M$ to $P$.

3) $P$ checks the validity of received values by verifying the following equation:

\[ g^{s_M} \mod p = y_M^{h(h_p, r_T) \mod q} r_M \mod p \] Eq. (2)

C. Proxy signature generation: After making the proxy signing key, $P$ generates a proxy signature for a given message $Sm$. The detailed procedure is described below.

1) $P$ generates the proxy signing key $x$ by the following:

\[ x \equiv s_M \mod p \]

\[ h(m_w, r_M) \mod q, x \mod p \] Eq. (3)

2) $P$ generates ? ? $\text{Sign}(m, x)$.

3) Output is $\{m, x, r_M, m_w, y_M, h_p, r_T\}$.

D. Proxy signature verification: Given proxy signature $(m, x, r_M, m_w, y_M, h_p, r_T)$. $V$ performs the following procedure for checking its validity.

1) $V$ generates proxy public key by the following:

\[ y ? \equiv (y_M, y_T) \] Eq. (4)

\[ y_T \equiv h(h_p, r_T) \mod p \mod p \] Eq. (5)

\[ g^{s_T} \mod p \equiv h(h_p, r_T) \mod p \] Eq. (6)

2) $V$ performs $\text{Verify}(h_p, y, y_T)$. If it returns 1, the received proxy signature is valid. Otherwise, it is invalid.

E. Revoking privacy: If needed, anyone who has the anonymous alias of a proxy signer can know his/her real identity by the interaction with the alias authority. The procedure is as follows. In this procedure, we assume that $V$ wants to know the real identity of the proxy signer.

1) $V$ sends an anonymous identity $h_p$ to $T$.

2) $T$ finds the matching $(ID_p, k_p)$ pairs using $h_p$ as key information.

3) $T$ sends $ID_p$ to $V$.

5. Security Consideration

In this section, we consider the security of our scheme. We show that our scheme preserves the security requirements described in Section 2. In each security requirement, we discuss it as follows.

Verifiability: For satisfying this requirement, the original signer's agreement should be verified implicitly in proxy signature verification step. In delegation step, the proxy signer verifies that the delegation information is generated by the original signer. However, in proxy signature verification step, verifier does not perform to verify the delegation information but only to verify the proxy signature. In fact, the proxy signature verification step only shows that the proxy signer has the proxy signing key which is matched the proxy public key which the verifier generates. Therefore, it should be proven that the possession of the proxy signing key means the proxy signer has valid delegation information and a valid anonymous private key.

Strong unforgeability: For preserving this requirement, anyone except the legitimate proxy signer cannot forge the proxy signature. Our scheme uses the Schnorr signature in many cases: delegation information, proxy signing key generation, and proxy signature generation/verification. It is widely believed that the Schnorr digital signature scheme is immune to existential forgery against the chosen message attack assuming that the discrete logarithm problem is hard [18]. Therefore, our scheme preserves the Strong unforgeability property.

Strong non-repudiation: This requirement means that only the
proxy signer who owns a valid delegation information and a valid anonymous private key can generate a valid proxy signature. In our scenario, we already assumed that the proxy signing algorithm has non-repudiation property. So, we do not need to prove that the proxy signature itself has non-repudiation property. However, if the proxy signing key can be generated by illegitimate method, the proxy signer can deny generating a proxy signature. Thus, for preserving strong non-repudiation, we only prove that the proxy signer has the valid proxy signing key which is matched to the generated proxy public key by verifier.

**Proxy privacy:** In our scheme, the proxy signer can be identified only by the anonymous alias. If an adversary wants to know the real identity of a proxy signer, they should compromise the alias authority because of the following reason: the anonymous alias is the result of the alias authority’s running hash function which has the input values that are the real identity of the proxy signer and the random number generated by the alias authority. If the adversary knows only the anonymous alias, it can be claimed to that of any proxy signer by manipulating the random value generated by the alias authority. So, if the random value is not known, the anonymous alias does not leak any information about the real identity. Therefore, if the anonymous alias is not compromised, the proxy privacy can be preserved.

**Privacy revocation:** Our scheme has an efficient method for revealing the real identity of the proxy signer. So, the property can be preserved.

By the above discussion, it can be summarized that our scheme should satisfy that a valid proxy signing key can be made without either a valid anonymous private key or valid delegation information from an original signer for preserving all security requirements. We define the forgery of the proxy signing key that there can be a method for generating a valid proxy signing key without either valid anonymous private key or valid delegation information from an original signer. We prove the following theorem to show that the proxy signing key can be unforgeable in our scheme.

Theorem 1. A proxy signing key cannot be forgeable unless the Schonorr digital signature scheme is forgeable.

**Proof.** For proving the theorem, we define the following oracles.

$\exists O_{a}(x_{M}):$ access oracle of $M$’s private key; the input value of this oracle is $(m_{r}, h_{p}, r_{p})$ and it outputs delegation information $(s_{M}, t_{M})$.

$\exists O_{p}(x_{T}):$ access oracle of $T$’s private key; this oracle gets $h_{p}$ as input and outputs an $(s_{p}, r_{p})$. In fact, $h_{p}$ is generated by alias authority. But, from the verifier’s view, it is a random value because anyone cannot verify the correctness of $h_{p}$ except the alias authority. Thus, it can be an effective attack if the adversary can get a valid signature of the alias authority for a random $h_{p}$ without accessing $O_{p}(x_{T})$. Therefore, we set $h_{p}$ as an input value of the oracle.

Adversaries can be defined as follows. We show that $C_{adv}$ can generate a forgery of a Schnorr digital signature if $D_{adv}$ succeeds to forge a proxy signing key.

$\exists C_{adv}:$ $C_{adv}$ intends to forge the Schnorr digital signature for which a private key $x_{pr}$ is used. $C_{adv}$ generates $x_{M} \equiv Z^{*} \mod p$ and sets $x_{M} \equiv x_{pr} \mod x_{M}$. $C_{adv}$ sets $y_{M} \equiv g^{x_{M}} \mod p$, and $y_{T} \equiv g^{x_{T}} \mod p$ (public keys) then sends them to $D_{adv}$. After $D_{adv}$ succeeds to generate a forgery, $C_{adv}$ gets the forgery and tries to forge the Schnorr signature using the forgery.

$\exists D_{adv}:$ $D_{adv}$ intends to forge the proxy signing key. $D_{adv}$ generates a $(m_{r}, h_{p})$ randomly and accesses the oracles $O_{a}(x_{M})$ and $O_{p}(x_{T})$ that get $(m_{r}, h_{p})$ as input values. Using the result of the oracles, $D_{adv}$ generates a valid proxy signing key $x \equiv h(m_{r}, r_{p}) \equiv y_{T} \equiv s_{M} \mod q$. The successful forgery of $D_{adv}$ means that without accessing the oracles using $(m_{r}, h_{p})$ as an input. $D_{adv}$ generates $(skp \equiv Z^{*}, h_{p}, r_{p}, m_{r}, r_{M})$ which preserves the following equation.

$$g^{skp} \equiv \{ (y_{M} \equiv y_{T})^{(\hat{h}_{p}, \hat{r}_{p})} \equiv \hat{y}_{T}^{m_{r}, r_{M}} \equiv \hat{y}_{T}^{(skp \equiv Z^{*}, h_{p}, r_{p}, m_{r}, r_{M})} \mod p \} \quad \text{Eq. (5)}$$

If $D_{adv}$ finds a forgery $(skp, h_{p}, r_{p}, m_{r}, r_{M})$, then $D_{adv}$ sends the forgery to $C_{adv}$. In this case, $C_{adv}$ can get a
Schnorr signature forgery by the following procedure.

1. If the public key is thought as \((y_{\mathcal{u}} ? y_{\mathcal{v}})^{h(\hat{m}_u, \hat{r}_u)} \mod p\), \((skp, \hat{m}_u, \hat{r}_u)\) is an existential forgery of Schnorr signature. Using folking lemma [18], \(C_{\text{adv}}\) can find an another forged signature \((skp', \hat{m}_u, \hat{r}_u)\).

2. Set the result of hash values as \(c_1 \equiv h(\hat{m}_u, \hat{r}_u) \mod p\) and set the forged public key as \(y' = (y_{\mathcal{u}} ? y_{\mathcal{v}})^{h(\hat{m}_u, \hat{r}_u)} \mod p\). With \(c_1, c_2, y'\), \(C_{\text{adv}}\) can get the following equations.

\[
g^y \mod p \equiv y' \mod p \mod p \quad \text{Eq. (6)}
\]

3. By the equations (6) and (7), \(C_{\text{adv}}\) gets \(y' = g^{x_m \cdot (skp' \cdot \hat{m}_u) \mod q} \mod p\). Using this result, \(C_{\text{adv}}\) get the following result.

\[
\hat{h}(\hat{r}_u, \hat{r}_u) \cdot (x_{\mathcal{u}} \cdot x_{\mathcal{v}}) \cdot k_r \equiv (skp' \cdot x_{\mathcal{v}} \cdot (c_1 + c_2) \mod q) \quad \text{Eq. (7)}
\]

4. Let \(s' = (skp' \cdot x_{\mathcal{v}}) \cdot (c_1 + c_2) \mod q\). Since \(x_{adv}\) is a forgery of the Schnorr digital signature. Therefore, \(C_{\text{adv}}\) successfully forge a Schnorr digital signature.

The above procedure shows that if \(D_{\text{adv}}\) exists, there exists a method to forge the Schnorr signature. Up till now, there is no efficient algorithm to break the Schnorr digital signature scheme. Therefore, a proxy signing key in our scheme cannot be forged unless the Schnorr digital signature scheme is forgeable.

6. Conclusion

In this paper, we proposed a new secure strong proxy signature scheme with proxy signer privacy protection. To the best of our knowledge, our scheme is the unique one that supports this functionality. We analyzed security of our scheme under the proxy signature forgery case and the proxy signing key forgery case. In both cases, we proved that our scheme is unforgeable if the Schnorr digital scheme remains unforgeable.

7. Acknowledgement

This work was supported by the KOSEF through the ATrc and the MIC, Korea, under the ITRC support program supervised by the IITA

8. References