Schnorr Signature Scheme with Restricted Signing Capability and Its Application

Chul-Joon Choi Zeen Kim Kwangjo Kim

International Research center for Information Security (IRIS),
Information and Communications University. (ICU),
58-4 Hwa-am Dong, Yuseong Gu, Daejeon, Korea 305-732

{ccjoon, zeenkim, kkj}@icu.ac.kr

Abstract Key exposure can cause serious problem without knowing the loss of secret key. In the current signature scheme, if we are not aware of lost key, then, an impersonator can generate as many signatures as he/she wants. Furthermore this problem can cause great damages to us. Therefore, it is important to construct a signature scheme strong against key exposure problem. But it is an open problem to find a practical signature scheme with resistance to key loss problem.

In this paper, we propose a new signature scheme with limited number of signatures and construct a Schnorr-based proxy signature scheme with limited number of signatures. This scheme would not completely solve the key exposure problem, but can restrict the attacker's forgery. In addition, in proxy signature scheme using proposed scheme, original signer can limit the number of signatures which proxy signer can generate. As a result, we can protect original signer from the misuses of proxy signature.

1 Introduction

In the real world, people are easy to lose their secret keys. Furthermore, we are not aware of this event. In current signature scheme, if we lose the secret key, and if we are not aware of it, an attacker can generate as many signatures as he wants without limit. Furthermore, this kind of problem can cause serious damages to us. Therefore, it is important to construct a signature scheme strong against key exposure problem. But it is not easy to find a practical signature scheme with resistance to key loss problem. The notion of forward secrecy was proposed to solve this kind of problem by Anderson[1]. Bellare and Miner[2] suggested the first practical signature scheme that guarantee forward secrecy. And it extended to intrusion resilient scheme. But this method is not compatible with previous well-known signature scheme. So, it is another burden to the user.

With a different notion, Hwang *et al.* proposed $c-times$ digital signature scheme which restricts the number of messages that can be signed[3]. They used c degree polynomial $f(z)$ for restricting the number of times of the signature. This scheme employs signature scheme that if a signer generates signatures more then threshold value c, then anyone can calculate the signer's secret key using Lagrange interpolation method. And this characteristic is achieved by revealing partial information about secret key using polynomial. This scheme is quite efficient in the sense that it is compatible with DSS (Digital Signature Standard), ElGamal and Schnorr signature scheme, but it has to keep additional information to limit signing capability.

Mambo, Usuda and Okamoto[4] proposed a proxy signature scheme, referred to as MUO scheme, based on discrete logarithm problem. They classified proxy signatures based on delegation type into full delegation, partial delegation and delegation with warrant. They showed various constructions of proxy signature schemes . Kim *et al.*[5] strengthened them by using Schnorr signature and including warrant information. Lee *et al.*[6] proposed a strong proxy signature scheme by classifying proxy signature schemes into strong and weak, according to the undeniable property. They also classified proxy signature schemes into designated and non-designated proxy signatures, according to designation of proxy signer in proxy key issuing stage. But there are still remaining problems related to the misuse of proxy signature in real the world. If a malicious proxy signer abuse his signing ability, then both the malicious proxy signer and the original signer are responsible for the signature.

For example, Bob is a financial agent of Alice. Usually he sings a contract on behalf of Alice. One day, Bob bears a grudge against Alice. Before Alice revokes the proxy and warrant information, Bob signs the forged documents which can do harm to Alice. After that, Bob ran away from Alice. In this case, although Alice can sue Bob, she is not free from responsibilities. To protect the original signer from these kinds of problems, we can apply our scheme to limit the signing ability of proxy signer.

In general, it may be undesirable to limit the capability of signer. But, in some cases this will solve some parts of digital signature scheme. There are several areas to apply this scheme. One is electronic cash and electronic check system. When the bank issues a check to their customers, this would want to restrict the signing ability of their customers according to his cash balance. Another is a proxy signature scheme. In proxy signature scheme, original signer delegates her signing ability to her proxy signer that she may want to restrict her proxy signer's signing ability to prevent the misuse of delegated powers. In this paper, we propose a more efficient scheme when used with Schnorr-based signature.

2 Preliminaries

In this section, we describe the meaning of notations which we use in this paper. Further, we introduce Schnorr signature scheme and LKK proxy signature scheme because our scheme is derived from these two schemes.

2.1 Notation

We denote by $\{0,1\}^*$ the set of all binary strings of finite length. And we require a hash function, $H : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$. we also denote t as a number of pre-selected random secret integers. Additional notations are described as follows:

Definition 1 *Let t be a small integer, and $\mathbf{\Psi}$, $\mathbf{\Omega}$ be sets having the following characteristics,*

a) *A set,*
$\mathbf{\Psi} = \{\alpha_i \mid \alpha_i \in_R \mathbb{Z}_q^*,\ 1 \le i \le t\}$

b) *A set,*
$\mathbf{\Omega} = \{\omega_i \mid \omega_i = H(g^{\alpha_i}||i),\ \alpha_i \in \Psi,\ 1 \le i \le t\}$

2.2 Schnorr Signature Scheme

We describe Schnorr signature scheme to show how we can limit the number of signatures. Schnorr signature scheme employs a subgroup of order q in $\mathbb{Z}_p^*$, where p is some large prime number. The method also requires a hash function $H : \{0,1\}^* \longrightarrow \mathbb{Z}_q$ [7].

Key Generation Algorithm

a) Select prime numbers q and p with the property that q divides $(p-1)$.

b) Select a random integer x such that $1 \le x \le q-1$.

c) Compute $y = g^x \bmod p$.

d) A's public key is (p, q, α, y), and A's secret key is x .

Signature Generation

a) Select a random secret integer k, $1 \le k \le q-1$

b) Compute $r = g^k \bmod p$, $e = H(m||r)$, and $s = x \cdot e + k \bmod q$

c) A's signature for m is the pair $(s,\ e)$.

Signature Verification

a) Compute $v = g^s \cdot y^{-e} \bmod p$, and $\bar{e} = H(m||v)$

b) Accept the signature if and only if $e = \bar{e}$.

2.3 LKK Proxy Signature scheme

We describe the details of LKK scheme [6]. This scheme is strong non-designated proxy signature scheme which can be applicable to mobile agent.

Proxy key issuing

a) Original signer *Alice* selects random number $k_1 \in_R \mathbb{Z}_q^*$.

b) Compute public key $r_1 = g^{k_1} \bmod p$.

c) Compute $\sigma = x_a \cdot e_1 + k_1 \bmod q$, where $e_1 = H(m_w||r_1)$ and m_w does not contain proxy signer's ID but state delegation information.

c) Send $(r_1,\ \sigma,\ m_w)$ to proxy signer *Bob* in a secure manner.

d) *Bob* verifies $g^\sigma = y_a^{\bar{e}_1} \cdot r_1 \bmod p$, where $\bar{e}_1 = H(m_w||r_1)$

e) *Bob* computes proxy key $\sigma_p = \sigma + x_b \bmod q$ and $y_p \equiv g^{\sigma_p} = y_a^{\bar{e}_1} \cdot r_1 \cdot y_b \bmod p$ where $\bar{e}_1 = H(m_w||r_1)$.

Signature Generation

If the message m confirms to m_w.

a) Generate a signature $s = S(\sigma_p, m)$ using proxy secret key σ_p.

b) Send $(s,\ m,\ r_1,\ m_w, y_a, y_b)$ to verifier.

Signature Verification

a) Check $m \in \{m_w\}$ and,

b) Verify whether the output of $V(y_a^{H(m_w||r_1)} \cdot y_b \cdot r_1, m, s)$ is true or not.

3 Proposed Scheme

In this section, we propose Schnorr signature scheme with restricted signing capability and we apply it to proxy signature scheme. We modifiy LKK scheme to meet the requirement of our scheme, and we construct Schnorr-based proxy signature scheme with restricted signing capability.

3.1 Main idea

The main idea of our scheme is that we preselect random integer set $\boldsymbol{\Psi}$ in key generation phase. And we publish corresponding set $\boldsymbol{\Omega}$ at public directory. In verification step, verifier checks whether the hash value of $v = g^s \cdot y^{-e} \bmod p$ is the element of $\boldsymbol{\Omega}$ or not. This means that if the hash value of v is not an element of $\boldsymbol{\Omega}$, the signature is not valid and should be rejected. If the signer uses a random secret value $k \in \boldsymbol{\Psi}$ twice, then, the signer's secret key x is revealed as following,

$$s_1 = x \cdot e_1 + k, \quad e_1 = H(m_1||r) \qquad (1)$$

$$s_2 = x \cdot e_2 + k, \quad e_2 = H(m_2||r) \qquad (2)$$

Equation (1) - (2) is

$$(s_1 - s_2) = x \cdot (e_1 - e_2)$$
$$\Rightarrow x = \frac{(s_1 - s_2)}{e_1 - e_2}$$

The signer can use k only one time to generate signature without revealing of his secret key. In that reason, the signer can generate limited number of signatures.

3.2 Proposed Scheme

Our scheme is a variant of Schnorr signature scheme. So it employs a subgroup of order q in $\mathbb{Z}_p^*$, where p is some large prime number. This scheme also requires a hash function $H : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$ as does in Schnorr signature scheme.

Key Generation

a) Select prime numbers q and p with the property that q divides $(p-1)$.

b) Select a generator g of the unique cyclic group of order q in $\mathbb{Z}_p^*$.

- Select an element $\gamma \in \mathbb{Z}_p^*$ and,
- Compute $g = \gamma^{(p-1)/q} \bmod p$.
- If $g = 1$ then repeat b).

c) Select a random integer x such that $1 \leq x \leq q-1$.

d) Compute $y = g^x \bmod p$.

e) Select random integer set $\boldsymbol{\Psi}$.

f) Compute a set $\boldsymbol{\Omega}$ where $\omega_i = H(g^{\alpha_i}||j)$, and $\alpha_i \in \boldsymbol{\Psi}$.

g) Keep x and $\boldsymbol{\Psi}$ as a secret value, and publish $p,\ q,\ g,\ y$ and a set $\boldsymbol{\Omega}$.

Signature Generation and Verification

1) Signature Generation

   a) Select a random secret value $\alpha_j \in \boldsymbol{\Psi}$.

   b) Compute $r = g^{\alpha_j} \bmod p,\ \ e = H(m||r||j)$, and $s = x \cdot e + \alpha_j \bmod q$.

   c) Alice's signature for m is the pair $(s,\ e,\ j)$

2) Signature Verification

   a) Obtain Alice's authentic public key $\{p,\ q,\ g,\ y\}$ and set $\boldsymbol{\Omega}$.

   b) Compute $\bar{r} = g^s \cdot y^{-e} \bmod p$, and $\bar{e} = H(m||\bar{r}||j)$.

   c) Accept the signature if and only if $e = \bar{e}$ and $H(\bar{r}||j) \in \boldsymbol{\Omega}$.

Our scheme is nothing but a modified version of Schnorr signature scheme that it has same security levels as that of Schnorr signature scheme, except that it pre-selects the random integer set $\boldsymbol{\Psi}$. The pre-selected random integer set $\boldsymbol{\Psi}$ should be kept secretly.

3.3 Application of Proposed Scheme

We demonstrate the proxy signature scheme with limited number of signatures. This scheme restrict the number of signatures in the range of 1 to t where t is the number of pre-selected random secret integers.

Proxy Generation

- Proxy signer *Bob* : (x_p, y_p)

  a) Select random secret integer set Ψ and,

  b) Compute corresponding public integer set $\boldsymbol{\Omega}$.

  c) Send public integer set $\boldsymbol{\Omega}$ to original signer, *Alice*, using public channel.

- Original signer *Alice* : (x_a, y_a)

  a) Select a random integer $k_1 \in_R \mathbb{Z}_q^*$.

  b) Compute public value $h_1 = g^{k_1} \bmod p$.

  c) Compute $\sigma = x_a \cdot e_1 + k_1 \bmod q$, where $e_1 = H(m_w||h_1||\boldsymbol{\Omega})$.

  d) Send $(e_1,\ \sigma,\ m_w)$ to proxy signer, *Bob*, using public channel.

  e) Original signer, *Alice* publishes $\boldsymbol{\Omega}$ at public directory.

Proxy Key Generation

- Proxy signer *Bob* : (x_p, y_p)

  a) Proxy signer, Bob does the following step

     - Compute $\bar{h}_1 = g^{\sigma} \cdot y_1^{-e_1} \bmod p$.
     - Compute $\bar{e}_1 = H(m_w||\bar{h}_1||\boldsymbol{\Omega})$.
     - Accept if and only if $e_1 = \bar{e}_1$.

  b) Compute proxy key $\sigma_p = \sigma + x_p \bmod q$.

Proxy Signature Generation

- Proxy signer *Bob* : (x_p, y_p)

  a) Select $\alpha_j \in \boldsymbol{\Psi}$, and compute $r_j = g^{\alpha_j} \bmod p$.

a) Generate a proxy signature $S_p = \sigma_p \cdot e_2 + \alpha_j \bmod q$, where $e_2 = H(m||r_j||\mathbf{\Omega})$.

c) Send $(S_p,\ e_1,\ e_2,\ m,\ m_w,\ h_1)$ to verifier $Carol$.

Proxy Signature Verification

- Verifier $Carol$:

  a) Compute $\bar{r}_j = g^{S_p} \cdot (y_a^{e_1} \cdot y_p \cdot h_1)^{-e_2}$ mod$p$,

  b) Compute $\bar{e}_2 = H(m||\bar{r}_j||\mathbf{\Omega})$.

  c) Accept if and only if $H(\bar{r}_j||\ j) \in \mathbf{\Omega}$ and $e_2 = \bar{e}_2$. If the check is hold simultaneously, then the signature is valid.

When Carol checks the validity, and if one of them is not satisfied, she rejects the signature. And if the proxy signer generates proxy signatures with same random number, $\alpha_f \in \mathbf{\Psi}$, then the secret key of the proxy signer can be revealed as follows:

$$s_f = \sigma_p \cdot e_f + \alpha_f,\ \ e_f = H(m_f||\omega_f||f) \quad (3)$$

$$s_g = \sigma_p \cdot e_g + \alpha_f,\ \ e_g = H(m_g||\omega_g||g) \quad (4)$$

Equation (3) - (4) is

$$(s_f - s_g) = \sigma_p \cdot (e_f - e_g)$$

$$\Rightarrow \sigma_p = \frac{(s_f - s_g)}{e_f - e_g} \quad (5)$$

And

$$\sigma_p = \sigma + x_p \quad (6)$$

By Eqs. (5) and (6), x_p is revealed.

4 Security Analysis and Efficiency

4.1 Security Analysis

The security of our scheme, Schnorr-based proxy signature scheme with limited number of signatures, is same as LKK scheme, except that it pre-selects random secret values in proxy issuing phase. In common digital signature scheme, pre-selection of random number does not affect the level of security if they are kept securely.

Unforgeability

Our Schnorr-based proxy signature scheme with limited number of signatures is unforgeable. No one, except the proxy signer, *Bob*, can generate a valid proxy key pair under the name of *Bob* because it contains proxy signer's private key x_p. Only the legitimate proxy signer can create a valid proxy signature.

Undeniability

Our scheme is undeniable, because once a proxy signer creates a valid proxy signature, he can not repudiate it, as the proxy key pair can be computed only by himself.

4.2 Efficiency

Our scheme is a modified version of LKK scheme. Especially, we used additional public information, $\mathbf{\Omega}$ whose elements are hash value of the exponent of pre-selected random secret value. Hwang *et al.* proposed the method to generate digital signature scheme with restriction on signing capability. This scheme uses t degree polynomial $f(z)$ to restrict the number of signatures less then t. Therefore, they use additional public information, t degree polynomial, which has t number of coefficients. Hwang's scheme[3] should compute $f(j)$ to generate signature pair, but our scheme needs just one additional hash operation in verification phase. Though, Hwang's scheme has an advantage to extend to ElGamal scheme and DSS scheme, our scheme is more efficient than Hwang's scheme in case of Schnorr-based digital signature scheme.

5 Conclusion

In this paper, we propose a new signature scheme with restricted singing capabilities in terms of limited number of signature generation. And we also propose proxy signature scheme with restricted signing capability which is a modification version of LKK scheme. And we show that if we use this scheme, we can restrict the singing capability of proxy signer. This restriction enables us to protect original signers from the misuses of proxy signatures. And we also show that our scheme is more efficient scheme than Hwang's scheme when used with Schnorr-based signature.

References

[1] R. Anderson, "Cryptography and Security Policy," Invited Talk of the *ACM-CCS'97*, 1997.

[2] M. Bellare and S. Miner, "A forward-secure digital signature scheme," In Proc. of *CRYPTO'99*, LNCS vol. 1666, pp. 431–448. 1999.

[3] J. Hwang, D. Lee and J. Lim, "Digital Signature Scheme with Restriction on Signing Capability," Proc. of *ACISP 2003*, LNCS vol. 2727, pp. 324–335, 2003

[4] M. Mambo, K. Usuda and E. Okamoto, "Proxy Signatures : Delegation of the Power to Sign Messages," IEICE Trans. Fundamentals, vol. E79-A, no. 9, 1996.

[5] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited," In Proc. of *ICICS'97*, LNCS vol. 1334, pp. 223–232, 1997.

[6] B. Lee, H. Kim, and K. Kim, "Secure Mobile Agent using Strong Non-designated Proxy Signature," Proc. of *ACISP 2001*, LNCS vol. 2119, pp. 474–486, 2001.

[7] A. Menezes, P. van Oorschot and S. Vanstone, "Handbook of Applied Cryptography," CRC Press, pp. 451–460, 2002.