Forward-Secure Proxy Signature Scheme

Ming-Hsin Chang, Tzu-Shin Lin, and Yi-Shiung Yeh

Department of Computer Science and Information Engineering

National Chiao-Tung University

1001, Dashiue Rd., Hsinchu, Taiwan 300, R.O.C.

Tel: 886-3-5731813, Fax: 886-3-5724176

E-mail: **ucc@cht.com.tw, tzushin.csie90g@nctu.edu.tw , ysyeh@csie.nctu.edu.tw**

Abstract

In this paper, we adopt the concept of forward-secure property for proxy signatures. Proxy signature is a kind of signature schemes in which an original signer delegates his signing capability to a designated signer called a proxy signer. Then, the proxy signer creates a digital signature on behalf of the original signer. We describe the problems of exposing of the secure key in proxy signature scheme. The key exposure problem in distributed environments is also a serious problem. To address the weaknesses we propose a proxy signature scheme with forward-secure based on *2^l-root* signature scheme. The proposed scheme with forward-secure property that renews the proxy key in each period and deletes the previous keys to prevent the secret against key exposure and conforms strong proxy signature requirements. We also mention that the forward proxy signature scheme can be applied to limitation on time duration of delegation.

Keyword: Proxy signature, Forward-secure property, Digital signature

1. Introduction

The concept of proxy signature was firstly introduced by Mambo *et al.* [2] in 1996 which is a kind of digital signature schemes that an original signer delegates his/her signing capability to a proxy signer, and then the proxy signer creates a digital signature on behalf of the original signer. In [2], there are three types of proxy signature based on delegation type as full delegation, partial delegation, and delegation by warrant. In full delegation, an original signer gives its private key to a proxy signer. In this case the proxy signature by the proxy signer is indistinguishable from the original signer. In delegation with warrant, the proxy signer is authorized to sign by warrant. In partial delegation, a new secret is computed from the secret of an original signer, and the new secret is given to a proxy signer in a secure way.

Kim et al [3] used them by using Schnorr signature and including warrant information in partial delegation schemes. Lee-Kim [4] extended the concept of proxy signature of a partial delegation to strong proxy scheme. The requirements of strong proxy scheme should conform the requirements as follows [4][5]:

(i) Strong unforgeability: Except the designated proxy signer can create a valid proxy signature on behalf of the original signer.

(ii) Verifiability: From a proxy signature a verifier can be convinced that the original signer delegates her signing capacity to a proxy signer.

(iii) Strong identifiability: From a proxy signature anyone can confirm the identity of the corresponding proxy signer.

(iv) Strong undeniability: Once a proxy signer creates a valid proxy signature, he cannot repudiate his/her signature creation against anyone.

(v) Prevention of misuse: It should be confident that proxy key should be used only for creating proxy signature conforming to delegation information.

However, many of proxy signature schemes can be proven secure under very reasonable assumptions. In many solutions, security guarantees last as long as the secret keys remain

unrevealed. If a secret key is revealed, security is compromised and any signature created by the key is no longer trusted.

Recently, several lectures are [6][7] proposed. Many use distribution of the key across multiple proxy signers with (*k*, *n*) threshold schemes and proactive schemes [11][12]. In (*k*, *n*) threshold schemes, security is assumed if throughout the entire lifetime of the secret the adversary is restricted to comprise less than *k* of the *n* shares. Moreover, since each of the proxy signer with shares may be faced the same attack, the actual risk may not decrease.

To address the problem forward security is a novel approach. The object of forward security on proxy signature scheme is to protect signature security against the risk of key exposure of the proxy key without requiring effort of key distributions. A proxy signer proxy key σ_0 and keeps the corresponding proxy key. During of validation of the public key, the time is divided into periods, numbered $1...t$. The proxy signer renews the proxy key in each period and deletes the previous keys, while the public key stays fixed. At the start the proxy key σ_0 is the proxy key σ_0. At time period j the proxy signer has the proxy key σ_j. During the period j, an attacker gets proxy key σ_j, but he does not break $\sigma_0 \ldots \sigma_{j-1}$ since they has been deleted.

In this paper, we proposed a proxy signature scheme with forward security property. The proposed scheme refers to the concepts of 2^l*-th* root signature scheme and forward-secure scheme [9]. The security is based on assumptions of square root (*SQROOT*) problem and cryptographic one-way hash function.

The rest of this paper is organized as follows. In section 2, we describe description relative schemes briefly. In section 3 we proposed a forward-secure proxy signature scheme. In section 4 the correctness and conformance of the proposed scheme are presented. The discussion on the proposed proxy signature scheme is presented in section 5. In section 6 we deploy our scheme to limitation of proxy time period. Finally, we make a brief conclusion of this paper in section 7.

2 Preliminary

We modify the forward Abdalla-Reyzin's forward-secure digital signature scheme, 2^v*-th* root signature and Lee et al's [4] to make a proxy signature with forward-secure property. We describe mathematic background and relative work as follows.

2.1 Notations and assumptions

Throughout this paper the following parameters and assumptions are the same. We list the used parameters in this paper. The details of description refer to [9].

$H(\ )$: One-way hash function.

v : A secure parameter known by original signer and proxy signer.

p_1 and p_2 : Two primes of approximately equal size with $p_1 = p_2 = 3 \pmod 4$.

N : A k-bits integer with $N = p_1 p_2$ such N is called a *Blum* integer.

Q : A set of non-zero quadratic residues modulo N.

(s_A, u_A) : public key pairs with $u_A = 1/s_A^{2^v} \pmod N$ for signer *A*.

(s_B, u_B) : public key pairs with $u_B = 1/s_B^{2^v} \pmod N$ for signer *B*.

2.2 Mathematic background

The basis of mathematic is an extension of square root (*SQROOT*) problem. We use the parameters as defined in section 2. For $k \in Q$ and We define a v -bit binary string $r = b_1 \ldots b_v$, we define $F_r(s) \equiv s^{2^v} k^r \pmod N$. Anyone who knows N and r can efficiently compute $F_r(s)$ and who knows p_1 and p_2 can efficient compute s for a given $F_r(s)$. On the contrary, if one does know N and r, it is hard to compute s.

2.3 2^v*-th* root signature scheme

In the proposed scheme we use 2^v*-th* root signature scheme to sign a warrant message. Here we introduce briefly.

(Signature generation) To sign a message M a signer picks a random number $k \in Q$ and computes

$r = (k)^{2^v} \pmod N$,

e = $H(M, r)$, and

σ = $k\, s_A^{\,e}$ (mod *N*).

The pair (r, σ) is the signature on the message M.

(Message verification) To verify the validity of a signature a verifier computes

$\sigma \neq 0$ (mod *N*) and

$r' = (\sigma)^{2^v}\, u^r$

checks that following equation holds

$e = H(r', M)$.

3 The proposed scheme

In this section, we shall present a new forward-secure proxy signature scheme. Like a standard proxy signature scheme, it contains four phases - proxy generation, proxy key verification, proxy signature, and proxy signature verification. The proposed scheme adds a proxy key update phase in which a proxy signer updates his/her proxy key and deletes the previous proxy key during time period. Therefore, a forward-secure proxy signature scheme is a key-evolving digital signature scheme.

The sketch of the proposed scheme is illustrated as follows. The proposed scheme involves the following participants: the original signer (Alice), the proxy signer (Bob) and a verifier (Carol). First, Alice creates a signature σ_A on a warrant M_W using *2^v-th* root signature scheme and Bob receives and verifies the signature to create a proxy key. Second, Bob has secret key pair $(j\text{-}1, \sigma_{B_{j-1}})$ in the time period j-1 and wants to get proxy key pairs (j, σ_{B_j}) at time period j by squaring $\sigma_{B_{j-1}}$ for v times. Finally, to sign on message M Bob uses forward-secure signature scheme to create a signature. The valid proxy signature tuple $(\sigma, M, r, M_W, r_A, j)$ on message M includes proxy information M_W and r_A for conforming requirements of proxy signature.

The protocol presents as following steps. The system parameters are same as given previously. Alice and Bob have public key pairs (u_A, s_A) and (u_B, s_B) respectively.

(Proxy generation) – The original signer Alice chooses a random number $k_A \in Z_N^*$, computes

$r_A = (1/k_A)^{v(t+1)}$ (mod *N*), (1)

$e_A = H(M_W, r_A)$, and (2)

$\sigma_A = k_A\, s_A$ (mod *N*) (3)

and sends (σ, M_W, r_A) to the proxy signer Bob in a secure manner.

(Proxy key verification) – Upon receiving (σ_A, M_W, r_A), Bob computes $e_A = H(M_W, r_A)$ and checks that following equation holds

$\sigma^{2^{v(t+1)}} = (1/r_A)\,(1/u_A)^{e_A}$ (mod *N*) (4)

If the checking passes successfully, Bob accepts σ and computes

$\sigma_{B_0} = \sigma_A\,(s_B)^{e_A}$ (mod *N*) (5)

as his proxy key where the index "0" means base state of the proxy key of σ_B, i.e. the beginning at the time period 0.

(Proxy key update) – At the time period j-1, Bob renews his proxy key pair $(j\text{-}1, \sigma_{B_{j-1}})$ to (j, σ_{B_j}) used in next period j, computes

$\sigma_{B_j} = (\sigma_{B_{j-1}})^{2^v}$ (mod *N*) (6)

and deletes $\sigma_{B_{j-1}}$.

(Proxy signature) – To sign a message M at the current period j Bob chooses a random number $k \in Z_N^*$, computes

$r = k^{2^{v(t+1-j)}}$ (mod *N*), (7)

$e = H(M, r, j)$, and

$\sigma = k\,(\sigma_{B_j})^e$ (mod *N*), (8)

The tuple $(\sigma, M, r, M_W, r_A, j)$ is the proxy signature on the message M.

(Verification of proxy signature) – To verify the validity of a signature the verifier Carol computes

$e_A = H(M_W, r_A)$, and

$$r' = \sigma^{2^{v(t+1-j)}} (r_A (u_A u_B)^{e_A})^{\sigma} \pmod{N} \quad (9)$$

and checks the equation $H(M, r', j) = H(M, r, j)$ holds.

Because the scheme provides forward-secure property, the proxy signer renews the proxy key at every time period. The previous proxy signature schemes were not able to renew proxy keys. Therefore, the proposed scheme provides the property of timestamp. From a proxy signature a verifier can check at what time period the proxy signature created.

In the proxy key update phase Bob renews his proxy key $\sigma_{B_j} = (\sigma_{B_{j-1}})^{2^v} \pmod{N}$ and deletes $\sigma_{B_{j-1}}$ at the time period j. Base on the *SQROOT* problem it is computationally infeasible to get σ_{B_j} from $\sigma_{B_{j-1}}$ without knowing p_1 and p_2.

The procedures of creation and verification of proxy signature resemble *2^l-th* root signature scheme, but the creation of signature uses the proxy key at current time period and the verification of proxy signature uses both of Alice's public key and Bob's public key.

4. Correctness and conformance

4.1 Correctness

In the following theorems, we describe that the proposed scheme works correctly.

Theorem 1*: If an original signer delegates its right, a proxy signer can verify the validity of σ_A in Eq. (4).*

Proof:

Computing $e = (M_W, r_A)$ and Raising both side of Eq. (3) by $2^{v(t+1)}$, we have

$$\sigma_A^{2^{v(t+1)}} = (k_A (s_A)^e)^{2^{v(t+1)}} \pmod{N}$$

$$= (k_A)^{2^{v(t+1)}} ((s_A)^e)^{2^{v(t+1)}} \pmod{N}$$

where s_A is the secret key of the original signer.

From Eq. (1)(2), we replace k_A with r_A and the original signer's secret key, and have

$$\sigma_A^{2^{v(t+1)}} = (1/r_A) ((s_A)^e)^{2^{v(t+1)}} \pmod{N}$$

$$= (1/r_A) (1/u_A)^{e_A} \pmod{N} \quad \text{Q.E.D}$$

The original signer delegates her signing capability to the proxy signer using creating a signature on a warrant message M_W by using the *2^{th}-root* signature scheme. Except the original signer others can create the signature to delegates her signing capability.

Theorem 2 : *If the proposed signature $(\sigma, M, r, M_W, r_A, j)$ is valid at the time period j, it will pass Verfication of proxy siganture in section 4.*

Proof:

We compute $e_A = H(M_W, r_A)$ and $e = H(M, r, j)$ form the signature tuple $(\sigma, M, r, M_W, r_A, j)$.

From Eq. (9), let

$$r' = \sigma^{2^{v(t+1-j)}} (r_A (u_A u_B)^{e_A})^{e} \pmod{N}$$

Replace σ with Eq(8)

$$r' = (k(\sigma_{B_j})^e)^{2^{v(t+1-j)}} (r_A (u_A u_B)^{e_A})^{e}$$

(mod *N*)

$$= k^{2^{v(t+1-j)}} ((\sigma_{B_j})^{2^{v(t+1-j)}})^{e} (r_A (u_A u_B)^{e_A})^{e}$$

(mod *N*)

From (6), we substitute B_0 for B_j

$$r' = k^{2^{v(t+1-j)}} ((\sigma_{B_j})^{2^{v(t+1-j)}})^{e} (r_A (u_A u_B)^{e_A})^{e}$$

(mod *N*)

$$= k^{2^{v(t+1-j)}} ((\sigma_{B_0})^{2^{v(t+1)}})^{e} (r_A (u_A u_B)^{e_A})^{e}$$

(mod *N*)

From (3)(5)(7), replace with B_0 by σ_A and s_B

$r' = k^{2^{v(t+1-j)}} ((\sigma_{B_0})^{2^{v(t+1)}})^e (r_A(u_A u_B)^{e_A})^e \pmod N$

$= k^{2^{v(t+1-j)}} ((1/r_A)((1/u_A)(1/u_B))^{e_A})^e (r_A(u_A u_B)^{e_A})^e \pmod N$

$= k^{2^{v(t+1-j)}} \pmod N$

$= r \pmod N$

Therefore, the equation $H(M, r', j) = H(M, r, j)$ Q.E.D

At the time period j the proxy signer has proxy key pair (j, B_j) and signs on a message using forward-secure scheme [9] except the proxy signer uses proxy key instead of secret key and public key combined from r_A, u_A and u_B.

4.2 Conformance

In this section, we discuss that the proposed scheme conforms to the security requirements of strong proxy signature.

(Strong unforgeability) No one except the proxy signer Bob can generate a valid proxy key pair on behalf of Alice because the proxy key σ_{B0} (at basic state with index '0') contains Bob's secret key s_B, which is only known by Bob. It is computationally infeasible to break Bob's secret key s_B form Bob's public key u_B without additional information based on *SQROOT* problem. Hence, only a designated proxy signer can create a valid proxy signature.

(Verifiability) The warrant information M_W, which includes the original signer Alice's assignment material, is implied in the hash value e_A. And, this hash value is used in the verifying process. Therefore, if the proxy signature σ passes the checking successfully, Alice's agreement on the signed message *M* is also verified explicitly.

(Strong identifiability) Identity information of the proxy signer Bob is included in the public key. Anyone can determine the identity of the corresponding proxy signer because the proxy signer's public key u_B is required in order to check a proxy signature in the verification proxy signature phase.

(Strong undeniability) Once a proxy singer created a legal proxy signature at the time period time j before the key is compromised, Bob cannot repudiate it in the future because he is the only person who can compute the proxy key pairs.

(Prevention of misuse) If Bob uses the proxy key pair for other applications that the warrant M_W does not state, he must be responsible for it because no one except him can generate the proxy signature under the name of Bob. Accordingly, illegal proxy transfer is prevented and signing capability of proxy signer is limited.

(Forward-secure property) In proxy update phase, the new secret key $\sigma_{B\,j+1}$ is generated by computing v time squares, (i.e. $(\sigma_{B\,j+1})^{2^v}$). According to the above sections, it is difficult to compute $\sigma_{B\,j+1}$ from the current secret key $\sigma_{B\,j}$ without knowing p_1 and p_2 under *SQROOT* problem. Therefore, even though an adversary breaks into the proxy system and gets the present key pair ($j, \sigma_{B\,j}$), he cannot generate past key pair form (0, σ_{B_0}) to (j-1, $\sigma_{B\,j-1}$).

5 Discussion

The proposed scheme is derived from Abdalla-Reyzin's forward-secure digital scheme in which the security of scheme is based on the hardness of factoring Blum integers. The most significant feature of forward-secure signature is to allow for key exposure attacks. The adversary can obtain the proxy key σ_j at the current time period j. The adversary can forge the current signature successfully, but he will be fail to forge a valid signature for the time period $i < j$. Since it is computationally infeasible to derive $\sigma_0 \ldots \sigma_{j-1}$ based on *SQROOT* problem. i.e. the hardness of factoring Blum integers.

The efficiency of the phases of proxy

generation and proxy key verification is about the *2th-root* signature scheme, but the phases of proxy key update, proxy signature and proxy key verification is the same as [9]. We analysis the number of modular multiplications and modular squarings, required to sign on a document and verify validation. Like the Abdalla-Reyzin's forward security scheme, the proxy signature takes time $v(t+1-j)$ modular squarings and e modular multiplications. Because the proxy signature verification can require to combining the proxy public key, the scheme takes time e_A modular multiplications more than the verification in forward-secure scheme. So the verification of proxy signature about $v(t+1-j)+e$ modular multiplications is needed.

Thus, our scheme has almost the same as the forward-secure scheme in [9]. In general, we take N = 102b bit and the length of hash value is 160 bits. However, because we add the phases of proxy generation and proxy verification scheme, we believe that the result is more efficient than the scheme inherited from [8].

6. Application to limitation on time duration of delegation

Besides protecting proxy signature before key's explosion, the proposed scheme provides a mechanism of the time limitation of signing key.

In [2], an original signer Alice gives her signature parameter secretly to a proxy signer Bob which does not contain key information. The scheme endures the proxy key will be valid throughout the entire lifetime of the key. Kim et al. [3] proposed a warrant to state the information on delegation relationship that may include valid duration of proxy signature. It not provide an efficient mechanism to process time duration.

In our scheme, a proxy signature includes the value j of the time period that it was created. A verifier can verifies the valid period by checking whether the value j of the time period in duration of delegation. The number of period *T* and each period can be presented a proxy time period. For example, an original signer Alice wants to delegate his signing capability to a proxy signer Bob for a month. The time period could be set *T* = 30 and each period has length one day. The proxy public key keeps for one month and the proxy key is updated daily. When a secret key finishes updating (i.e. it has been updated for all the periods.), the proxy key is revoked automatically.

7. Conclusion

Proxy signatures are very helpful tools in which an original signer delegates his signing capability to a proxy signer and then the proxy signer creates a digital signature on behalf of the original signer. However, the key exposure problem in distributed environments is also a serious problem against the security of a strong proxy signature scheme. Consequently, we have proposed a forward-secure proxy signature scheme to avoid the problem of key compromise.

References

[1] A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.

[2] 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, pages1338-1354.

[3] S. Kim, S. Park, and D. Won, "Proxy Signatures, Revisited," Proc. of ICICS'97, Springer-Verlag, Lecture Notes in Computer Science, LNCS 1334, 1997, pages 223-232.

[4] B. Lee, and K. Kim, "Strong proxy signatures", IEICE Trans. Fundamentals, vol. E82-A, no.1 Jan 1999, pages.1-11.

[5] B. Lee, H. Kim and K. Kim, "Strong proxy signature and its applications," Proc. of SCIS 2001, 11B-1, pages 603-608, 2001.

[6] K. Zhang, "Threshold Proxy Signature Schemes," 1997 Information Security workshop, Japan, Sep. 1997, pages 191-199.

[7] H. M. Sun, N. Y. Lee, and T. Hwang," Threshold Proxy Signatures" IEE proceedings – Computers and Digital Techniques, Vol. 146, No. 5, 1999, pages 259-263.

[8] M. Bellare and S. Miner," A Forward-Secure Digital Signature Scheme," Advances in Cryptology { CRYPTO 99 Proceedings, Lecture Notes in Computer Science, Vol. 1666, Springer-Verlag, M. Wiener, ed, 1999,

pages 431-438.} Full version: Theory of Cryptography Library: Record 99-16, September 1999, http://philby.ucsd.edu/cryptolib.html.

[9] M. Abdalla and L. Yeyzin “A new forward-secure digital signature scheme”, ASIACRYPT pages 116 -129, 2000.

[10] H. Ong and C. Schnorr, “Fast Signature Generation with a Fiat Shamir-Like Scheme”, Advances in Cryptology - Eurocrypt '90, Lecture Notes in Computer Science, Vol.473, Springer Verlag, pages 432-440,1991.

[11] A. Herzberg, S. Jarecki, H. Krawczyk,and M. Yung, “Proactive Secret Sharing or: How to Copy With Perpetual Leakage,” CRYPTO ’95, LNCS 963,1995.

[12] A. Shamir, “How to Share a Secret,” CACM Vol. 22, No. 11, 1979.