Provably Secure Threshold Proxy Signature Scheme Based on Factoring

Bing-Chang Chen

***Abstract*—A proxy signature scheme is a method which allows an original signer to delegate his signing authority to a designated person, called a proxy signer. Moreover, the signing authority can be designated to a few proxy signers to scatter the signing rights. A threshold proxy signature scheme allows *t* out of *n* proxy signers to sign messages on behalf of the original signer. In this paper, we propose a new threshold proxy signature scheme based on the factoring problem in which the original signer delegates his signing authority to *n* authorized proxy signers in such a way that any *t* out of *n* proxy signers can generate a proxy signature collectively, while *t*-1 or less proxy signers cannot. Finally, we provide a proof on its security.**

***Index Terms*—proxy signatures, threshold proxy signatures, factoring, non-repudiation, security proof.**

I. Introduction

In recent years, people have been used to deal with something on the Internet for convenience and speed, such as buying or bidding commodities, ordering the tickets, etc. For these commercial activities, authenticity and non-repudiation are the major and indispensable properties. Usually digital signatures are applied to achieve these requirements. A digital signature scheme is a method which allows a signer to create digital signatures of documents, and the generated signatures can be verified by any person, called a verifier. The scenario for a digital signature scheme includes that the signer uses his private key to sign a message and hence generate a valid digital signature for the message, and then the verifier uses the corresponding public key of the signer to verify the correctness of the message-signature pair. Thus, a digital signature scheme is a general way to authenticate messages for a specific signer. However, more situations for digital signatures need to be considered and addressed. Considering the case when the signer cannot sign messages online, e.g., the signer is busy for something or on a business trip, we desire a variant of digital signature scheme to be suitable for such an environment. The concept of the proxy signature scheme was first introduced by Mambo, Usuda, and Okamoto [7], [8] in 1996. There are two parties, the original signer and the proxy signer, in a proxy signature scheme. The original signer can delegate his signing capability to the proxy signer. The proxy signer can sign a message on behalf of the original signer when the original signer is absent.

So far, there have been four types of delegation proposed. These include full delegation, partial delegation, delegation by warrant, and partial delegation with warrant. In full delegation, the original signer gives his secret key to the proxy signer directly. The proxy signer uses the key to create signatures of documents, which are the same as those created by the original signer. Therefore, full delegation is not practical because proxy signatures issued by the original signer and by the proxy signer are indistinguishable. That is, the non-repudiation property cannot be provided in the full delegation. In partial delegation [7], [8], the proxy signature signing key is generated by the proxy signer with the help of the original signer. Only the proxy signer can know the proxy signature signing key which can be used to generate valid proxy signatures; even the original signer cannot. However, this approach suffers from the disadvantage that the proxy signature signing key is transferable because no warrant indicating the identity of the proxy signer is included in the proxy signatures. In delegation by warrant [9], [12], the original signer signs a warrant, which certifies that the signer is actually the legal proxy signer. Delegation by warrant can be implemented by ordinary digital signature schemes without any modification, and it is appropriate for restricting documents to be signed, e.g., a warrant states its valid period. Usually, delegation by warrant incurs more computational cost than the above. In [11], Kim et al. proposed a proxy signature scheme which is a partial delegation with warrant enjoying the computational advantage over the proxy signature by warrant and the structure advantage over the proxy signature for partial delegation. In this paper, we are interested in the partial delegation with warrant because it is more secure and more efficient than other types of delegation. For simplicity, in this paper we call "the partial delegation with warrant" the proxy signature if it doesn't lead to any confusion.

In order to disperse the risk that the proxy signing key kept by a proxy signer is stolen or lost, the original signer may distribute his signing authority to a group of proxy signers he delegates. A (t, n) threshold proxy signature scheme [8], [11], [16], [17], [20] is a scheme which allows any t or more proxy signers from a designated group of n proxy signers to cooperatively sign messages on behalf of the original signer, while t -1 or less proxy signers cannot generate any valid proxy signatures. Each of the proxy signers can generate a partial proxy signature on behalf of the original signer. So far, most of these existing (t, n) threshold proxy signature schemes were also based on the discrete logarithm problem [8], [11], [16], [17], [20]. The only known threshold proxy signature scheme based on the factoring problem was due to Hwang et al. [6]. Unfortunately Hwang et al.'s scheme is insecure against the original signer's forgery in which the original signer can create a valid proxy signature, while the proxy group cannot deny this proxy signature [18].

Manuscript received April 10, 2012; revised May 5, 2012.

The authors are with Department of Information Communication, Southern Taiwan University, Taiwan (e mail: bcchen@mail.stut.edu.tw).

In recent these years, a formal security proof is becoming a requisite for a secure protocol or scheme. The first emergence with formal proof was shown under the random oracle model in [2]. From then, many researchers showed the security of their schemes according to this model. In proxy signatures, the first provably secure scheme was occurred in [1], another distributed proxy signature scheme [7] was also provided by a formal security proof. In this paper, we propose a factoring-based threshold proxy signature scheme and also provide the security proof [2], [3], [4], [5] afterwards.

II. Threshold Proxy Signature Scheme Based on Factoring

A threshold proxy signature scheme [8], [11], [16], [17], [20] allows *t* out of *n* proxy signers to sign messages on behalf of the original signer. In this section, we propose a new threshold proxy signature scheme based on factoring. Besides, we also provide the security proof for the proposed scheme.

In [5], Guillou and Quisquater proposed a signature scheme, GQ scheme in short, in which its security is based on factoring. The security of GQ signature scheme is based on factoring, i.e. RSA [15]. Later, there were two schemes proposed based on GQ signature scheme, one is forward-secure [9] and the other one is signer-base intrusion-resilient signature [10]. In these two schemes, the authors provide a formal proof for their schemes. In this paper, we refer to these papers and propose a threshold proxy signature scheme based on factoring.

A. System Parameters

In this scheme, it needs a trusted third party (TTP) to help the original signer and proxy signers for the generation of system parameters. First, the TTP selects a large composite number n which is composed of two large primes p_1 and p_2. That is, $n = p_1 \cdot p_2$, where p_1, p_2 are randomly chosen by the TTP. Besides, the original signer generates two primes e_1 and e_2 for computing public keys, generating proxy signatures, and verifying proxy signatures. Then, the original signer chooses his secret key $s_{original}$ randomly in z_n^*, and computes the corresponding public key $p_{original} = s_{original}^{e_1 \cdot e_2} \bmod n$. In this scheme, the original signer has to delegate his signing power to *n* proxy signers. Therefore, every proxy signer i(i = 1~n) needs to choose his secret key $s_{proxy(i)}$ randomly in z_n^*, and computes the corresponding public key $p_{proxy(i)} = s_{proxy(i)}^{e_1 \cdot e_2} \bmod n$. In order to designate a user to be a proxy signer, the original signer prepares an appropriate warrant M_w to claim these proxy signers can sign messages collectively on behalf of him. The warrant includes the identities of the proxy signers and original signer, and other useful information such as delegation period. Moreover, the original signer sends $e_1 M_w$ to TTP for requiring the *n* sub-shares of proxy signers. The TTP selects a polynomial function $f(x)$ of degree t-1 which is $f(x) = e_1 M_w + a_1 x + ... + a_{t-1} x^{t-1} \bmod \phi(n)$, where a_1, a_2, ..., a_{t-1} are random numbers. Then he computes $f(i)$ L_i for each proxy signer's i and then sends back to the original signer, where

$$L_i = \prod_{i, j \in T, j \neq i} \frac{-j}{i-j} \bmod \phi(n).$$

The original signer uses $f(i)$ L_i to compute $s_{original}^{f(i) \cdot L_i} \bmod n$ for the proxy signer i and then sends it to him. The warrant M_w is published to announce the signing authority of proxy signers. After receiving the value, the proxy signer i can compute the partial proxy signing key, $s_{sig(i)} = s_{original}^{f(i) \cdot L_i} \cdot s_{proxy}^{e_1 \cdot M_w} \bmod n$.

B. Signing

To sign a message M, the proxy signer needs to perform the operations in the following.

1) Select a random number $x \in z_n^*$, and then computes $y = x^{e_2} \bmod n$.

2) Let $H()$ be a hash function. The dealer computes $\sigma = H(e_2, y, M)$.

3) The dealer sends σ to the proxy signers.

4) The proxy signer i uses his proxy signing key to compute $z_i = s_{sig(i)}^{\sigma}$, and then sends z_i to the dealer.

5) Once the dealer collects t out of n numbers proxy signers sent, he can compute the proxy signature

$$z = x \cdot \prod_{i=1}^{t} z_i \bmod n.$$

6) The proxy signature of the message M is $(M, z, \sigma, e_2, M_w, p_{proxy(i \in T)})$.

C. Verifying

To verify the proxy signature $(M, z, \sigma, e_2, M_w, p_{proxy(i \in T)})$, the verifier first computes $y' = z^{e_2} \cdot 1/(p_{original} \cdot \prod_{i \in T} p_{proxy(i)})^{M_w \cdot \sigma}$, and then check if $\sigma = H(e_2, y', M)$.

In the following we provide the proof of the verification.

$$y' = z^{e_2} \cdot 1/(p_{original}\prod_{i\in T} p_{proxy(i)})^{M_w\sigma} = x^{e_2} \cdot (\prod_{i\in T} z_i)^{e_2} \cdot 1/(p_{original}\prod_{i\in T} p_{proxy(i)})^{M_w\sigma} =$$

$$y \cdot (\prod_{i\in T} s_{sig(i)}^{\sigma})^{e_2} \cdot 1/(p_{original}\prod_{i\in T} p_{proxy(i)})^{M_w\sigma} = y \cdot (\prod_{i\in T} s_{original}^{f(i)\cdot L_i})(\prod_{i\in T} s_{proxy(i)}^{e_1\cdot M_w})^{\sigma e_2} \cdot 1/(p_{original}\prod_{i\in T} p_{proxy(i)})^{M_w\sigma} =$$

$$y \cdot (s_{original}^{\sum_{i\in T} f(i)\cdot L_i})(\prod_{i\in T} s_{proxy(i)}^{e_1\cdot M_w})^{\sigma e_2} \cdot 1/(p_{original}\prod_{i\in T} p_{proxy(i)})^{M_w\sigma} =$$

$$y \cdot (s_{original}^{e_1 M_w})(\prod_{i\in T} s_{proxy(i)}^{e_1\cdot M_w})^{\sigma e_2} \cdot 1/(p_{original}\prod_{i\in T} p_{proxy(i)})^{M_w\sigma} =$$

$$y \cdot (s_{original}^{e_1 M_w})(\prod_{i\in T} s_{proxy(i)}^{e_1\cdot M_w})^{\sigma e_2} \cdot 1/(s_{original}^{e_1 e_2}\prod_{i\in T} s_{proxy(i)}^{e_1\cdot e_2})^{M_w\sigma} =$$

$$y \cdot (s_{original})(\prod_{i\in T} s_{proxy(i)})^{e_1 e_2 M_w\sigma} \cdot 1/(s_{original}\prod_{i\in T} s_{proxy(i)})^{e_1 e_2 M_w\sigma} = y$$

Therefore $H(e_2, y', M) = H(e_2, y, M) = \sigma$

III. Security Analysis

In this section, we provide the security proof as follows.

Lemma 1 *Let G be a* group. *Suppose* e_1, $e_2 \in Z$, *and* $GCD(e_1, e_2) = 1$. *Given* e_1, $e_2 \in G$ *and* $a^{e_1} = b^{e_2}$, *one can compute c such that* $c^{e_2} = a$ *in* $O(log(e_1 + e_2))$ *group and arithmetic operations.*

Proof. Using Euclid's extended gcd algorithm, find out f_1, f_2 such that $e_1f_1 + e_2f_2 = 1$ within $O(log(e_1 + e_2))$ arithmetic operations. Compute $c = a^{f_2} \cdot b^{f_1}$ with $O(log(f_1 + f_2)) = O(log(e_1 + e_2))$ group operations. That is, $c^{e_2} = a^{e_2 f_2} \cdot b^{e_2 f_1} = a^{e_2 f_2} \cdot a^{e_1 f_1} = a$

Theorem 1 If solving the problem of factoring is hard, then the proposed threshold proxy signature scheme is secure in the random oracle model.

Proof. There exists an adversary *A* who wishes to forge a threshold proxy signature with querying the random(hash) oracle *H* and the signature oracle *S*. There is another adversary *B* which is devoted to factor the number *N* and find out β, given $\beta^r = \alpha \bmod n$ where α, n, r are known. In the experiment, *B* uses the threshold proxy signatures of *A*'s forgery to accomplish its task, and furthermore *B* provides the signature and hashing queries of *A*.

In the simulation, the adversary *A* can query on the random oracle and signature oracle which are totally controlled by the adversary *B*. Therefore, *B* needs to maintain these two databases and answer *A*'s query. Initially, *B* sets the random oracle and signature oracle to be empty.

When *A* queries the random oracle on the tuple (e_2, *y*, *M*), *B* first checks whether (e_2, *y*, *M*) was queried before. If it was, *B* returns the corresponding result σ in the database to *A*. Otherwise, *B* chooses a random number σ' and sends back to *A*. Then *B* keeps the pair σ' and (e_2, *y*, *M*) in the database for the future query.

Moreover, *A* can query the signature oracle on a message which is not the message he wants to forge. In this case, we can simulate the behavior of chosen-message attack. When *A* queries the signature oracle on a message *M*, *B* first chooses two random number *z* and σ. *B* prepares a warrant M_w, the verifying key $p_{original}\prod_{i\in T} p_{proxy(i)}$ and e_2, and then computes $y' = z^{e_2} \cdot 1/(p_{original} \cdot \prod_{i\in T} p_{proxy(i)})^{M_w\cdot\sigma}$.

B checks whether the pair σ and (e_2, *y*, *M*) are in the database of random oracle. If they are not, then *B* keeps the pair σ and (e_2, *y*, *M*) in the database for the future query. At last, *B* returns the result signature (M, *z*, σ, e_2, M_w, $p_{proxy(i\in T)}$) to *A*.

If A outputs a forged threshold proxy signature (M, *z*, σ, e_2, M_w, $p_{proxy(i\in T)}$), then the hashing oracle has been queried on (e_2, *y*, *M*), where

$$y = z^{e_2} \cdot p_{sig}^{M_w\cdot\sigma}.$$

B then comes up with a random tape for *F* , remembers it, and runs *F* on that tape. Therefore, *B* needs to maintain two tables, a random oracle *H* and a signature oracle *S*.

Because *B* controls the random oracle, he can answer the signature queries at random. If there is a signature query on message *M*, it is

$$y' = z^{e_2} \cdot 1/(p_{original} \cdot \prod_{i\in T} p_{proxy(i)})^{M_w\cdot\sigma} = z'^{e_2} \cdot 1/(p_{original} \cdot \prod_{i\in T} p_{proxy(i)})^{M_w\cdot\sigma'}$$

Let $1/(p_{original} \cdot \prod_{i\in T} p_{proxy(i)})^{M_w} = P$.

Then, $(z/z')^{e_2} = P^{(\sigma'-\sigma)}$

By Lemma 1, we can find two numbers f_1 and f_2 using the equation $f_1(\sigma'-\sigma) + f_2 e_2 = 1$ by Euclid's extended gcd algorithm, where $GCD((\sigma'-\sigma), e_2) = 1$.

Therefore, *B* can compute the e_2-th root of p_{sig} which is $c = (z/z')^{f_1} \cdot P^{f_2} \bmod n$.

We can check the correctness as below.

$$c^{e_2} = (z/z')^{f_1 e_2} \cdot P^{f_2 e_2} = P^{(\sigma'-\sigma) f_1} \cdot P^{f_2 e_2} = P = 1/(p_{original} \cdot \prod_{i\in T} p_{proxy(i)})^{M_w}$$

IV. Conclusions

Proxy signatures are becoming more and more important in the future. Many people work on the Internet and sign messages (contracts, documents, etc.) in the environment, too. Once they cannot sign an important message personally because they are busy with something, they have to delegate his signing authority to proxy signers on behalf of him. Therefore proxy signature schemes can be used in this situation. In order to disperse the signing authority, the original signer may distribute his signing authority to a group of proxy signers he delegates. A (*t*, *n*) threshold proxy signature scheme is proposed to allows any *t* or more proxy signers from a designated group of *n* proxy signers to cooperatively sign messages on behalf of the original signer. In this paper, we propose a factoring-based threshold proxy signature scheme and also provide the security proof.

References

[1] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure proxy signature schemes for delegation of signing rights," Preprint available at http://eprint.iacr.org/2003/096/.

[2] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," In *First ACM Conference on Computer and Communications Security*, LNCS, Springer-Verlag, 1993.

[3] E. Fujisaki and T. Okamoto, "Statistical zero knowledge protocols to prove modular polynomial relations," *Advances in Cryptology CRYPTO'97*, LNCS 1294, Springer-Verlag, pp. 16—30, 1997.

[4] S. Goldwasser, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal of Computing*, vol. 17, no. 2, pp. 281-308, April 1998.

[5] L. C. Guillou and J. J. Quisquater, "A "paradoxical" identity-based signature scheme resulting from zero-knowledge," *Advances in Cryptology- Crypto* 1988, LNCS, Springer-Verlag, pp. 216-231, 1988.

[6] M. S. Hwang, J. L. Lu, and I. C. Lin, "A Practical (t,n) Threshold Proxy SignatureScheme Based on the RSA Cryptosystem," *IEEE Trans. Knowledge and Data Engineering*, vol. 15, no. 6, pp. 1552-1560, 2003.

[7] J. Herranz and G. Saez, "Revisiting fully distributed proxy signature schemes," Preprint available at http://eprint.iacr.org/2003/197/.

[8] C. L. Hsu, T. S. Wu, and T. C. Wu, "Improvement of threshold proxy signature scheme," *Applied Mathematics and Computation*, vol. 136, pp. 315-321, 2003.

[9] G. Itkis and L. Reyzin, "Foreward-secure signatures with optimal signing and verifying," *Advances in Cryptology- Crypto* 2001, LNCS, Springer-Verlag, pp. 332-354, 2001.

[10] G. Itkis and L. Reyzin, "SiBIR: Signer-based intrusion-resilient signatures," *Advances in Cryptology- Crypto* 2002, LNCS, Springer-Verlag, pp. 499-514, 2002.

[11] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited," ICICS'97, LNCS 1334, Springer- Verlag, pp. 223-232, 1997.

[12] M. Mambo, K. Usuda, and E. Okamoto, "Proxy Signatures: Delegation of the Power to Sign Message," *IEICE Trans. Fundamentals*, vol. E79-A, no. 9, pp. 1338-1353, Sep. 1996.

[13] M. Mambo, K. Usuda, and E. Okamoto, "Proxy Signatures for Delegation Signing Operation," *Proc. Third ACM Conf. on Computer and Communications Security*, pp. 48-57, 1996.

[14] B. C. Neuman, "Proxy-based authorization and accounting for distributed systems," *Proc. 13th International Conference on Distributed Systems*, pp. 283-291, 1993.

[15] R. L. Rivest, A. Shamir, and L. Adleman, "A method of obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, February 1978.

[16] H. M. Sun, "An efficient nonrepudiable threshold proxy signature scheme with known signers," *Computer Comm*., vol. 22, no. 8, pp. 717-722, 1999.

[17] H. M. Sun, N. Y. Lee, and T. Hwang, "Threshold proxy signatures," *IEEE Proceedings-Computers and Digital Techniques*, Vol. 146, No. 5, pp. 259-263, 1999.

[18] H. M. Sun, C. T. Yang, and B. T. Hsieh, "On the Security of a Threshold Proxy Signature Scheme Based on the RSA Cryptosystem," manuscript, private communication, 2004.

[19] V. Varadharajan, P. Allen, and S. Black, "An analysis of the proxy problem in distributed systems," *Proc. 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 255-275, 1991.

[20] K. Zhang, "Threshold Proxy Signature Schemes," *Information Security Workshop*, Japan, pp. 91-199, Sep. 1997.