9-2004

Proxy Signature Scheme with Multiple Original Signers for Wireless E-Commerce Applications

Guilin WANG
Institute for Infocomm Research

Feng BAO
Singapore Management University, fbao@smu.edu.sg

Jianying ZHOU
Institute for Infocomm Research

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

DOI: https://doi.org/10.1109/VETECF.2004.1404663

Citation

WANG, Guilin; BAO, Feng; ZHOU, Jianying; and DENG, Robert H.. Proxy Signature Scheme with Multiple Original Signers for Wireless E-Commerce Applications. (2004). *Proceedings of the 2004 IEEE 60th Vehicular Technology Conference: 26-29 September 2004, Los Angeles, California*. 3249-3253. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/531

Proxy Signature Scheme with Multiple Original Signers for Wireless E-Commerce Applications

Guilin Wang, Feng Bao, Jianying Zhou, and Robert H. Deng

Infocomm Security Department, Institute for Infocomm Research (I^{2}R)
21 Heng Mui Keng Terrace, Singapore 119613
{glwang,baofeng,jyzhou,deng}@i2r.a-star.edu.sg

***Abstract*— In a proxy signature scheme, a user delegates his/her signing capability to another user in such a way that the latter can sign messages on behalf of the former. In this paper, we propose an efficient and secure proxy signature scheme with multiple original signers. Our scheme is suitable for wireless electronic commerce applications, since the overheads of computation and communication are low. As an example, we present an electronic air ticket booking scheme for wireless customers.**

Keywords: proxy signature, digital signature, e-commerce, wireless communications.

I. Introduction

In a proxy signature scheme, one user Alice, called *original signer*, delegates her signing capability to another user Bob, called *proxy signer*. After that, the proxy signer Bob can sign messages on behalf of the original signer Alice. Upon receiving a proxy signature on some message, a verifier not only can validate its correctness by a given verification procedure, but also be convinced of the original signer's agreement on the signed message. Proxy signature schemes have been suggested for use in a number of applications, including mobile agent, mobile communications, electronic commerce, and distributed shared object systems etc. [2], [6], [12], [17].

Mambo, Usuda, and Okamoto firstly introduced the concept of proxy signatures and proposed several constructions in [9], [10]. Based on the delegation type, they classified proxy signatures into *full delegation*, *partial delegation*, and *delegation by warrant* schemes. In a full delegation, Alice's secret key is given to Bob directly so Bob has the same signing capability as Alice. For most of real-world settings, such schemes are obviously impractical and insecure. In a partial delegation scheme, a proxy signer possesses a new key, called *proxy secret key*, which is different from Alice's secret key. So, proxy signatures generated by using proxy secret key are different from Alice's standard signatures. However, in such schemes the range of messages a proxy signer can sign is not limited. This weakness is eliminated in delegation by warrant schemes by adding a *warrant* that specifies the types of messages to be delegated, the identities of Alice and Bob, the delegation period, etc.

According to another criterion – whether the original signer knows the proxy secret key, proxy signatures can be classified into *proxy-unprotected* and *proxy-protected* schemes. That is, in a proxy-protected scheme only proxy signer can generate proxy signatures, while in a proxy-unprotected scheme either proxy signer or original signer can generate proxy signatures since both of them know the proxy secret key. In many practical applications, proxy-protected schemes are required to avoid potential disputes between the original signer and proxy signer. Aiming to distinguish the rights and responsibilities of both parties clearly, the proxy-protected schemes with partial delegation by warrant have attracted much more investigations than others. Sometimes, this special type of schemes is refereed to proxy signature scheme for simplicity.

Following Mambo et al.'s first work in [9], [10], a number of new schemes and improvements have been proposed [5], [20], [6], [7], [4]; however, most of them do not fully meet the desired security requirements (see Section 2.2). In [5], Kim et al. introduced the concept of partial delegation by warrant, and proposed a threshold proxy signature, in which the original signer's signing ability is shared among a delegated group of n proxy singers such that only t or more of them can generate proxy signatures cooperatively. However, Sun et al. [16] pointed out that the discrete log based threshold proxy signatures proposed in [5], [20] are insecure, and Wang et al. [18] analyzed the RSA-based threshold proxy signature by Hwang et al. [4]. In [6], Lee, Kim and Kim constructed mobile agents for electronic commerce applications from non-designated proxy signature, in which a warrant does not specify the identity of a proxy signer so any possible proxy signer may respond this delegation and become a proxy signer. In [7], Lee, Cheon, and Kim investigated whether a secure channel for delivery of a signed warrant is necessary in existing schemes. Their results show that if secure channels are not provided, the schemes in [9], [6] are insecure. To avoid the usage of secure channels and overcome some other weaknesses, some improved schemes are presented. But Wang et al. [17] showed that all of the schemes and improvements proposed in [6], [7] are insecure by demonstrating several kinds of attacks. Boldyreva et al. [2] presented a formal model for proxy signature, i.e., the existential unforgeablity against adaptive chosen-message attacks [3].

Note that Wang et al.'s attacks in [17] mainly result from the fact that a valid proxy key pair colud be forged by an adversary, including the original signer and the proxy signer. Wang proposed a new proxy signature scheme in [19], which is based on the two-party Schnorr signature scheme of Nicolosi et al. [11]. This scheme is provably secure and as efficient as the schemes in [6], [7], [2]. Furthermore, Wang extended this basic scheme into two versions of designated-verifier proxy signatures so that the validity of a proxy signature can be checked only by the designated proxy signer.

In this paper, based on the standard proxy signature scheme proposed in [19], we propose an *efficient* and *secure* proxy signature scheme with multiple original signers. Our scheme is suitable for wireless electronic commerce applications, since the overheads of computation and communication are low. As an example, we present an electronic air ticket booking scheme for wireless customers.

The rest of this paper is organized as follows. Section 2 introduces the computational assumption and security requirements for proxy signatures. Section 3 briefly reviews Wang's proxy signature scheme based on two-party Schnorr signature. Then, in Section 4, we extend this scheme into a proxy signature scheme with multiple original signers, and analyze its security and efficiency. Finally, Section 5 discusses some potential applications of a proxy signature scheme with multiple original signers.

II. Preliminaries

A. Computational Assumption

The proxy signature schemes in this paper are based on the following computational assumption.

Assumption 1: Discrete Logarithm (DL) assumption. *Let $G_q = \langle g \rangle$ be a cyclic multiplicative group generated by g of order q. Then, on inputs $(g, g^x) \in G_q^2$ where $x \in_R \mathbb{Z}_q$ is a random (unknown) number, there is no probabilistic polynomial-time algorithm that outputs the value of x with non-negligible probability.*

The DL assumption is widely believed to be true for many cyclic groups, such as the multiplicative subgroup $G_q = \langle g \rangle$ of the finite field $\mathbb{Z}_p$, where p is a large prime and q is a prime factor of $p-1$. In practice, $|p| = 1024$ and $|q| = 160$ are considered to be suitable for most current security applications.

B. Definitions

Definition 1. A **proxy signature scheme** is usually comprised of the following procedures [19]:

- **System Setup**: On input of a security parameter l, this probabilistic algorithm outputs two secret/public key pairs (x_A, y_A) and (x_B, y_B) for the original signer Alice and the proxy signer Bob. Note that those key pairs may be used in a standard signature scheme at the same time.
- **Proxy Key Generation**: The original signer Alice and the proxy signer Bob execute this interactive randomized algorithm to generate a proxy key pair (x_P, y_P) for Bob, such that only Bob knows the value of x_P, while y_P is public or publicly recoverable.
- **Proxy Signature Generation**: The proxy signer Bob runs this (possibly probabilistic) algorithm to generate a proxy signature σ for a message m by using the proxy secret key x_P.
- **Proxy Signature Verification**: A verifier runs this deterministic algorithm to check whether an alleged proxy signature σ for a message m is valid with respect to a specific original signer and a proxy signer.

The security requirements for proxy signature are first specified in [9], [10], and later are kept almost the same beside being enhanced in [6], and formalized in [2].

Definition 2. A **secure** proxy signature scheme should satisfy the following requirements:

- **Verifiability**: From the proxy signature, a verifier can be convinced of the original signer's agreement on the signed message.
- **Identifiability**: Anyone can determine the identities of the corresponding original signer and proxy signer from a proxy signature.
- **Unforgeability**: Only the designated proxy signer can create a valid proxy signature on behalf of the original signer. In other words, the original signer and other third parties who are not designated as proxy signers cannot create a valid proxy signature.
- **Undeniability**: Once a proxy signer creates a valid proxy signature on behalf of an original signer, he cannot repudiate the signature creation against anyone else.
- **Prevention of misuse**: The proxy signer cannot use the proxy secret key for purposes other than generating valid proxy signatures. In case of misuse, the responsibility of the proxy signer should be determined explicitly.

III. Proxy Signature Scheme Based on Two-Party Schnorr Signature

In this section, we review the proxy signature scheme proposed in [19], which is based on the provably secure two-party Schnorr signature scheme of Nicolosi et al. [11]. The basic idea is that Nicolosi et al.'s scheme is used to generate a proxy key pair (x_P, y_P) such that

$$g^{x_P} = y_P = (y_A \cdot y_B)^{h(m_w, r_P)} \cdot r_P \bmod p, \qquad (1)$$

where r_P is a public value, and m_w is a warrant which specifies the related information about a proxy delegation. In fact, (r_P, x_P) is exactly a two-party Schnorr signature on message m_w. The point is that *only* Bob knows the value of x_P, but (r_P, x_P) can *only* be generated by Alice and Bob *jointly*. Therefore, x_P can be used as the proxy secret key to generate proxy signatures according to a standard DLP-based signature scheme. At the same time, a verifier can validate proxy signatures after recovering the public proxy key y_P from Eq. (1).

System Setup: Let p and q be two large primes such that $q | (p-1)$, and $g \in \mathbb{Z}_p^*$ a generator of order q. Denote $G_q = \langle g \rangle$. The discrete logarithm assumption holds in G_q. Hereafter, we call three such integers (p, q, g) a *DLP-triple*. Let $h(\cdot)$

and $h'(\cdot)$ be two secure cryptographic hash functions. It is assumed that Alice and Bob have agreed on a warrant m_w before generating a proxy key pair for Bob. In addition, Alice has her key pair $(x_A, y_A = g^{x_A} \bmod p)$, and Bob has his key pair $(x_B, y_B = g^{x_B} \bmod p)$ [1].

Proxy Key Generation: To generate a proxy key pair (x_P, y_P) for the proxy signer Bob, Alice and Bob execute the following interactive protocol jointly.

(1) Alice picks a random number $k_A \in_R \mathbb{Z}_q^*$, computes $r_A = g^{k_A} \bmod p$ and $c = h'(r_A)$, and then sends c to Bob.

(2) Similarly, Bob chooses a random number $k_B \in_R \mathbb{Z}_q^*$, computes $r_B = g^{k_B} \bmod p$, and replies Alice with (c, r_B).

(3) When (c, r_B) is received, Alice checks whether $r_B^q \equiv 1 \bmod p$. If the validation goes through, she computes $r_P = r_A \cdot r_B \bmod p$, $s_A = k_A + x_A \cdot h(m_w, r_P) \bmod q$, and sends the pair (r_A, s_A) to Bob.

(4) Upon receiving (r_A, s_A), Bob first computes $r_P = r_A \cdot r_B \bmod p$, and then checks whether $r_A^q \equiv 1 \bmod p$, $c \equiv h'(r_A)$, and $g^{s_A} \equiv y_A^{h(m_w, r_P)} \cdot r_A \bmod p$. If all validations pass, he calculates $s_B = k_B + x_B \cdot h(m_w, r_P) \bmod q$, and finally sets his proxy key pair (x_P, y_P) by

$$x_P = s_A + s_B \bmod q, \quad \text{and} \quad y_P = g^{x_P} \bmod p. \quad (2)$$

It is easy to know that the above defined proxy key pair (x_P, y_P) satisfies Eq. (1), i.e., (r_P, x_P) is a standard Schnorr signature [15] on the warrant m_w with respect to the public key $y_A \cdot y_B \bmod p$.

Proxy Signature Generation: To generate a proxy signature on a message m that conforms to the warrant m_w, the proxy signer Bob performs the same operations as in the standard Schnorr signature scheme [15]. That is, he first selects a random number $k \in \mathbb{Z}_q^*$, computes $r = g^k \bmod p$ and $s = k + x_P \cdot h(m, m_w, r) \bmod q$. The resulting proxy signature on message m is $\sigma = (m_w, r_P, r, s)$.

Proxy Signature Verification: To verify the validity of σ, a verifier operates as follows.

(1) Check whether the message m conforms to the warrant m_w. If not, stop. Otherwise, continue.

(2) Check whether Alice and Bob are specified as the original signer and the proxy signer in the warrant m_w, respectively.

(3) Recover the proxy public key y_P by computing $y_P = (y_A \cdot y_B)^{h(m_w, r_P)} \cdot r_P \bmod p$.

(4) Accept the proxy signature σ if the following equation holds:

$$g^s = y_P^{h(m, m_w, r)} \cdot r \bmod p. \quad (3)$$

In the above scheme, Nicolosi et al.'s two-party Schnorr signature scheme [11] is used to generate proxy key pair (x_P, y_P). Therefore, from the provable security of Nicolosi et al.'s scheme, it is known that a valid proxy key pair (x_P, y_P) (defined by Eq. (1)) can *only* be generated by the original signer Alice and the proxy signer Bob *jointly*. In other words, anybody (including Alice and Bob) cannot generate a valid proxy key pair independently. On the other hand, without a valid proxy key pair anybody cannot generate a proxy signature such that Eq. (3) is satisfied. This is because the proxy signature generation algorithm is exactly the Schnorr scheme [15], which is also provably secure [13] in the random oracle model [1]. Therefore, Wang concluded that the above proxy signature scheme is secure, as stated in Proposition 1.

Proposition 1 [19]. *Based on the results in [13] and [11], the above proxy signature scheme is secure in the random oracle model, under the assumption that the discrete log problem in the multiplicative subgroup $\langle g \rangle$ is intractable.*

IV. Proxy Signature Scheme with Multiple Original Signers

In this section, we present a new proxy signature scheme with multiple original signers. The basic idea is that we first extend the provably secure two-party Schnorr signature scheme [11] into $(n+1)$-party setting, and then use the resulting scheme to generate a proxy key pair. We assume that the proxy signer Bob has public key y_B, and each of n original signers has certified public key y_j, $1 \le j \le n$. For simplicity, we denote the proxy signer Bob as the user U_0 and let $x_0 = x_B$, $y_0 = y_B$.

A. Description of the Proposed Scheme

System Setup: (p, q, g) is a DLP-triple, and $(x_i, y_i = g^{x_i} \bmod p)$ is user U_i's certified key pair, for each $i \in \{0, 1, \cdots, n\}$. The warrant m_w specifies the types of messages to be delegated, the delegation period, all identities of n original signers and the proxy signer Bob (i.e., user U_0), and other related information. Other notations are the same as in previous section.

Proxy Key Generation: To generate a proxy key pair (x_P, y_P) for Bob, n original signers and the proxy signer Bob execute the following interactive protocol cooperatively.

(1) Each user U_i $(0 \le i \le n)$ picks a random number $k_i \in_R \mathbb{Z}_q^*$, computes $r_i = g^{k_i} \bmod p$ and $c_i = h'(r_i)$, and then broadcasts c_i.

(2) Each user U_i reveals the value of r_i, $0 \le i \le n$. Then, each user checks whether all r_i's are correct, i.e., $r_i^q \equiv 1 \bmod p$ and $c_i \equiv h'(r_i)$, for each $0 \le i \le n$. If all validations go through, continue; otherwise, stop.

(3) Each original singer U_j $(1 \le j \le n)$ computes $r_P = r_0 r_1 \cdots r_n \bmod p$, $s_j = k_j + x_j \cdot h(m_w, r_P) \bmod q$, and sends the pair (r_j, s_j) to Bob.

(4) Upon receiving (r_j, s_j), Bob first computes $r_P = r_0 r_1 \cdots r_n \bmod p$, and then checks whether $g^{s_j} \equiv y_j^{h(m_w, r_P)} \cdot r_j \bmod p$, $1 \le j \le n$. If all validations pass, he calculates $s_B = k_0 + x_B \cdot h(m_w, r_P) \bmod q$, and

[1] Here, we assume that each user's key pair is certified by a certification authority (CA). That is, when a user registers a public key with the CA, he/she has to show the knowledge of the corresponding secret key. Actually, this is a recommended practice for issuing public key certificates, and can be used to prevent rogue-key attacks [2].

finally sets his proxy key pair (x_P, y_P) by

$$x_P = s_B + s_1 + \cdots + s_n \bmod q, \quad y_P = g^{x_P} \bmod p. \quad (4)$$

In fact, (r_P, x_P) is exactly an $(n+1)$-party Schnorr signature on message m_w with respect to the public key $y_B y_1 \cdots y_n$. The point is that *only* the proxy signer Bob knows the proxy secret key x_P, but (r_P, x_P) needs to be generated by the n original signers and the proxy signer Bob *jointly*.

Proxy Signature Generation: To generate a proxy signature on a message m that conforms to the warrant m_w, the proxy signer Bob performs the same operations as in the standard Schnorr signature scheme [15]. That is, he first selects a random number $k \in \mathbb{Z}_q^*$, computes $r = g^k \bmod p$ and $s = k + x_P \cdot h(m, m_w, r) \bmod q$. The resulting proxy signature on message m is $\sigma = (m_w, r_P, r, s)$.

Proxy Signature Verification: To verify the validity of σ, a verifier operates as follows.

(1) Check whether the message m conforms to the warrant m_w. If not, stop. Otherwise, continue.

(2) Check whether the delegation period expires. If not, continue.

(3) Check whether each user j ($j \in \{1, 2, \cdots, n\}$) is specified as the original signer, and Bob is specified as the proxy signer in the warrant m_w.

(4) Recover the proxy public key y_P by computing

$$y_P = (y_B y_1 \cdots y_n)^{h(m_w, r_P)} \cdot r_P \bmod p. \qquad (5)$$

(5) Accept the proxy signature σ if the following equation holds:

$$g^s = y_P^{h(m, m_w, r)} \cdot r \bmod p. \qquad (6)$$

B. Discussion

It is not difficult to directly verify the correctness of the above proxy signature scheme with multiple original signers. We now discuss its efficiency and security.

The proposed scheme is efficient. Firstly, note that the procedure of proxy key pair generation needs to be executed *only once* for a sufficiently long period, for example, one year, though it is complicated a little. Secondly, to generate a proxy signature, only one modular exponentiation is needed. Thirdly, to enhance the performance equations (5) and (6) can be checked together as a single equation. That is, a verifier only needs to check the following equation:

$$g^s \cdot (y_B y_1 \cdots y_n)^{-h_1 \cdot h_2} \cdot r_P^{-h_2} \equiv r \bmod p, \qquad (7)$$

where $h_1 = h(m_w, r_P)$ and $h_2 = h(m, m_w, r)$. So the left side of Eq. (7) can be carried out in 1.25 modular exponentiations by means of an exponent array (pages 618 of [8]). Moreover, note that a modular exponentiation with exponent of size 160 bits requires about 240 modular multiplications. Therefore, a proxy signature can be generated and verified by 240 and 300 modular multiplications, respectively.

About the security of the proposed scheme, we have the following proposition.

Proposition 2. *Under the discrete logarithm assumption, our proxy signature scheme with multiple original signers is secure in the random oracle model.*

Proof: Firstly, we claim that in our scheme even n users collude together, they cannot forge a valid proxy key pair (x_P, y_P) satisfying Eq. (5) and $y_P = g^{x_P} \bmod p$. If this is not the fact, i.e., there is an adversary $\mathcal{A}$ that can forge a valid proxy key pair in the scheme with multiple original signers, then from $\mathcal{A}$ we can construct a new adversary $\mathcal{A}'$ that forges a valid proxy key pair in the basic proxy signature scheme. More specifically, $\mathcal{A}'$ can be constructed as follows. For a given public key pair (y_A, y_B), we first choose an index $i \in_R [1, n]$ and a number $a \in_R \mathbb{Z}_q^*$ at random, and set $x_i = x_A + a \bmod q$, $y_i = y_A \cdot g^a \bmod p$. Then, for each index $1 \leq j \leq n$ and $j \neq i$, we select random numbers $x_j \in_R \mathbb{Z}_q^*$ such that $-a = \sum_{1 \leq j \leq n, j \neq i} x_j \bmod q$, and set $y_j = g^{x_j} \bmod p$. So we have $y_A y_B = y_B y_1 \cdots y_n \bmod p$. Finally, we feed on the adversary $\mathcal{A}$ by input $(y_B, y_1, \cdots, y_n)$. Consequently, when $\mathcal{A}$ outputs a valid proxy key pair (x_P, y_P), (x_P, y_P) is also a valid proxy key pair in the basic scheme with respect to the original signer Alice and the proxy singer Bob. In addition, any adversary (including the original signers) cannot forge valid proxy signatures in the name of the proxy signer Bob, since the provably secure Schnorr signature is used to generate proxy signatures. Therefore, the proposed proxy signature scheme with multiple original signers is unforgeable. At the same time, other security requirements are also met, since the discussion in [6] can be adapted to our scheme.

V. Applications

We now present an application of the above proxy signature scheme with multiple original signers as follows. Assume customer Cindy gets her electronic air ticket from a ticket agent Bob. To guarantee the integrity, authenticity and non-repudiation of e-tickets, some signatures on e-tickets are necessarily to be attached. Usually, Bob's signature is not sufficient to meet those security requirements since Cindy may not have much confidence in the credit of the agent Bob. At the same time, requiring signatures from both agent Bob and the air company is not convenient in practice. The reason is that to reduce cost, an air company may unlikely provide on-line service for every booked or paid e-ticket. In such a case, we can exploit a proxy signature scheme with multiple original signers. That is, multiple air companies first delegate their powers of issuing air tickets to a number of e-ticket agents. After that, each designated agent can issue e-tickets to costumers on behalf of those air companies. When the costumer Cindy gets an e-ticket companied with valid proxy signature from an agent Bob, she believes that not only Bob is certified by those air companies, but also Bob and those air companies will take the corresponding responsibilities if there is any dispute on the e-ticket. Furthermore, it is very convenient for the the e-ticket agent Bob since he only needs to store one proxy secret key for all air companies. In addition, a customer can verify the proxy signature on an e-ticket by using a wireless device which has lower computational power, such as a mobile phone. The reason is that in our scheme, to generate and verify a proxy signature only 240 and 300 modular multiplications are needed, as we discussed earlier.

It is also possible to use proxy signatures with multiple original signers in some other applications, such as mobile agent in e-commerce settings.

References

[1] M. Bellare and P. Rogaway, "Random Oracles Are Practical: A Paradigm for Designing Efficient Protocols," *Proc. 1st ACM Conference on Computer and Communications Security* (*CCS'93*), pp. 62-73, ACM Press, 1993.

[2] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure Proxy Signature Schemes for Delegation of Signing Rights," manuscript. Available at `http://eprint.iacr.org/2003/096/`.

[3] 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 1988.

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

[5] S. Kim, S. Park, and D. Won, "Proxy Signatures, Revisited," *Proc. Information and Communications Security* (*ICICS'97*), LNCS 1334, Springer-Verlag, pp. 223-232, 1997.

[6] B. Lee, H. Kim, and K. Kim, "Secure Mobile Agent Using Strong Non-Designated Proxy Signature," *Proc. Information Security and Privacy* (*ACISP'01*), LNCS 2119, Springer-Verlag, pp. 474-486, 2001.

[7] J.-Y. Lee, J. H. Cheon, and S. Kim, "An Analysis of Proxy Signatures: Is a Secure Channel Necessary?" *Proc. Topics in Cryptology* (*CT-RSA 2003*), LNCS 2612, Springer-Verlag, pp. 68-79, 2003.

[8] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

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

[10] M. Mambo, K. Usuda, E. Okamoto, "Proxy Signatures for Delegating Signing Operation," *Proc. 3rd ACM Conference on Computer and Communications Security* (*CCS'96*), ACM Press, pp. 48-57, 1996.

[11] A. Nicolosi, M. Krohn, Y. Dodis, and D. Mazieres, "Proactive Two-Party Signatures for User Authentication," *Proc. 10th Annual Network and Distributed System Security Symposium* (*NDSS'03*), The Internet Society, 2003. `http://www.isoc.org/isoc/conferences/ndss/`

[12] H.-U. Park and I.-Y. Lee, "A Digital Nominative Proxy Signature Scheme for Mobile Communications," *Proc. Information and Communications Security* (*ICICS'01*), LNCS 2229, pp. 451-455, Springer-Verlag, 2001.

[13] D. Pointcheval and J. Stern, "Security Arguments for Digital Signatures and Blind Signatures," *Journal of Cryptology*, Vol 13, No. 3, pp. 361-369, 2000.

[14] R.L. Rivest, A. Shamir, and L.M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, No. 2, pp. 120-126, Feb. 1978.

[15] C. Schnorr, "Efficient Signature Generation by Smart Cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161-174, 1991.

[16] H.-M. Sun, N.-Y. Lee, and T. Hwang, "Threshold Proxy Signatures," *IEE Proc.-Computers & Digital Techniques*, vol. 146, no. 5, pp. 259-263, Sept. 1999.

[17] G. Wang, F. Bao, J. Zhou, and R. H. Deng, "Security Analysis of Some Proxy Signatures," *Proc. Information Security and Cryptology - ICISC 2003*, LNCS 2971, Springer-Verlag, pp. 305-319, 2004. Preliminary version is available at `http://eprint.iacr.org/2003/196.`

[18] G. Wang, F. Bao, J. Zhou, and R. H. Deng, "Comments on a Threshold Proxy Signature Scheme Based on the RSA Cryptosystem," *IEEE Transactions on Knowledge and Data Engineering*, IEEE Computer Society, 2004 (to appear).

[19] G. Wang, "Designated-Verifier Proxy Signature Schemes," *Proc. the 8th IFIP Conference on Communication and Multimedia Security* (*CMS'04*), Kluwer, 2004.

[20] K. Zhang, "Threshold Proxy Signature Schemes," *Proc. Information Security Workshop* (*ISW'97*), LNCS 1396, Springer-Verlag, pp. 282-290, 1997.