Original Signer's Forgery Attacks on Discrete Logarithm Based Proxy Signature Schemes

Tianjie Cao and Xianping Mao
(Corresponding author: Tianjie Cao)
School of Computer Science and Technology, China University of Mining and Technology
Xuzhou Jiangsu 221008, China
(Email: tjcao@cumt.edu.cn)

(Received Oct. 17, 2005; revised and accepted Nov. 23, 2005 & Feb. 3, 2006)

Abstract

A proxy signature scheme enables a proxy signer to sign messages on behalf of the original signer. In this paper, we demonstrate that a number of discrete logarithm based proxy signature schemes are vulnerable to an original signer's forgery attack. In this attack, a malicious original signer can impersonate a proxy signer and produce a forged proxy signature on a message. A third party will incorrectly believe that the proxy signer was responsible for generating the proxy signature. This contradicts the strong unforgeability property that is required of proxy signatures schemes. We show six proxy signature schemes vulnerable to this attack including Lu et al.'s proxy blind multi-signature scheme, Xue and Cao's proxy blind signature scheme, Fu et al. and Gu et al.'s anonymous proxy signature schemes, Dai et al. and Huang et al.'s nominative proxy signature schemes are all insecure against the original signer's forgery.

Keywords: Cryptanalysis, discrete logarithm, proxy signatures

1 Introduction

The concept of a proxy signature was first introduced by Mambo, Usuda and Okamoto [8, 9]. In a proxy signature scheme, the original signer delegates his signing capability to a proxy signer, thereby enabling the proxy signer to sign messages on behalf of the original signer. Upon receiving a proxy signature on a given message, a verifier can not only validate its correctness by a given verification procedure, but also be convinced of the original signer's agreement on the signed message.

A secure proxy signature scheme should satisfy the following unforgeability property.

Unforgeability: Only the designated proxy signer could 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 a proxy signer cannot create a valid proxy signature purporting to have been generated by the proxy signer. Due to unforgeability property, a verifier can be convinced of the original signer's agreement on the signed message from the proxy signature and can determine the identity of the corresponding proxy signer.

Various extensions have been proposed to the basic proxy signature primitives which can be used in different application situations. A large selection of the schemes are based upon the discrete logarithm problem.

In 1982, Chaum introduced the concept of blind signature scheme [1]. Using the blind scheme, a user can obtain the signature on any given message, without revealing any information about the message or its signature. Combining proxy signatures and blind signatures Lin and Jan introduced the first proxy blind signature scheme [6]. A proxy blind signature should satisfy the blindness property.

Blindness: it allows a user to acquire a proxy signature on a message without revealing anything about the message or its signature to the proxy signer. When a proxy blind signature is verified and opened, any one can verify it, but no one except the signature requester can link this signature to its previous signing process instance.

The proxy blind multi-signature requires that the message is signed by a proxy signer whose signing power is delegated from all the original signers. In 2005, Lu et al. proposed a proxy blind multi-signature schemes [7], in which two or more original signers can jointly delegate their signing power to a single proxy signer.

Recently, with the indepth research on electronic cash and anonymous electronic voting, it is necessary to protect the privacy of the participants. An anonymous proxy signature scheme should provide proxy- anonymous property and anonymity revocation property.

Proxy-anonymous: Proxy signers may sign messages on the behalf of the original signer while protecting their identities against other third parties.

Anonymity Revocation: In the case of a dispute, the

actual proxy signer of the proxy signature can be revoked by the original signer.

A nominative signature scheme is a signature scheme in which signatures can only be verified by a designated verifier chosen by the signer. Nominative proxy signatures should provide restrictive verifiability property.

Restrictive verifiability: Only the nominative verifier can verify the validity of proxy signatures.

To guarantee undeniability of a signer, all the variants of proxy signature schemes should provide unforgeability property. It implies that only the proxy signer could create a valid proxy signature on behalf of the original signer, any third parties even the original signer cannot create a valid proxy signature.

In this paper, we demonstrate that a number of discrete logarithm based proxy signature schemes are vulnerable to an original signer's forgery attack. In this attack, a malicious original signer can impersonate a proxy signer and produce a forged proxy signature on a message. A third party will incorrectly believe that the proxy signer was responsible for generating the proxy signature. This contradicts the strong unforgeability property that is required of proxy signatures schemes. We show six proxy signature schemes vulnerable to this attack including Lu et al.'s proxy blind multi-signature scheme (the Lu-Cao-Zhou scheme) [7], Xue and Cao's proxy blind signature scheme (the Xue-Cao scheme) [11], Fu et al. and Gu et al.'s anonymous proxy signature schemes (the Fu-Kou-Xiao scheme and the Gu-Li-Yang scheme) [3, 4], Dai et al. and Huang et al.'s nominative proxy signature schemes (the Dai-Yang-Dong scheme and the Huang-Hao-Wang scheme) [2, 5] are all insecure against the original signer's forgery. In other words, these schemes do not possess the unforgeability property which is a desired security requirement for a proxy signature scheme.

2 Cryptanalysis of Lu-Cao-Zhou's Scheme

2.1 Review of Lu-Cao-Zhou's Scheme

Based upon discrete logarithm problem, Lu-Cao-Zhou proposed a proxy blind multi-signature scheme [7]. In this subsection, we briefly describe Lu-Cao-Zhou's proxy blind multi-signature scheme.

p and q are two large prime integers such that $q|p-1$ and g is a generator with order q in Z_p^*. Let $A_1, A_2, \cdots, A_n$ be the n original signers and B be the designated proxy signer. Every original signer $A_i(1 \le i \le n)$ has a private key x_i and the corresponding public key y_i, where $x_i \in_R Z_q^*$ and $y_i = g^{x_i} (\mathrm{mod} p)$. Proxy signer B also holds his own key pair (x_B, y_B), where $x_B \in_R Z_q^*$ is the private one, and $y_B = g^{x_B} (\mathrm{mod} p)$ the public one. Furthermore, $H(\cdot)$ is a universal secure hash functions.

The original signer $A_i(1 \le i \le n)$ selects $k_i \in_R Z_q^*$ at random, and computes (r_i, s_i).

$$\begin{aligned} r_i &= g^{k_i} (\mathrm{mod} p) \\ s_i &= x_i H(m_w, r_i) + k_i (\mathrm{mod} q), \end{aligned}$$

where m_w is the designated proxy warrant negotiated by all original signers, which records the delegation policy including limits of authority, valid periods of delegation and proxy signature, and all identities and the public keys of the original signers. The original signer $A_i(1 \le i \le n)$ publishes (r_i, m_w).

After the original signer $A_i(1 \le i \le n)$ produces sub proxy secret s_i, he encrypts and signs it before sending it to proxy signer B. After proxy signer B received the message, he decrypts and verifies the proxy sub secret key s_i. Proxy signer B generates the proxy secret key sk:

$$sk = \sum_{i=1}^{n} s_i + x_B (\mathrm{mod} q).$$

The proxy public key α, which is used in the proxy signature verification stage, is generated according with proxy signer and all original signers's public key and all $r_i(1 \le i \le n)$ published by original signers.

$$\alpha = y_B \prod_{i=1}^{n} (r_i y_i^{H(m_w, r_i)}) (\mathrm{mod} p).$$

After the proxy secret key sk has been generated, proxy signer B can calculate blind signatures on behalf of all the original signers. Assume requester C asks proxy signer B to make a blind signature on message m. The proxy signer B and the requester C can use the blind Schnorr signature scheme [10] to generate a proxy blind signature. After proxy blind signature was generated, anyone can verify it.

2.2 Cryptanalysis

We show that Lu-Cao-Zhou's proxy blind multi-signature scheme is insecure against the original signer's forgery. The original signer can forge the proxy secret key.

We assume the original signer A_n is an attacker. A_n selects proxy warrant m_w and $r_i \in_R Z_p^*(1 \le i \le n-1)$ at random. A_n computes

$$r_n = (y_B \prod_{i=1}^{n-1} (r_i y_i^{H(m_w, r_i)}))^{-1} (\mathrm{mod} p).$$

A_n publishes m_w and $r_i \in_R Z_p^*(1 \le i \le n-1)$. A_n can forge a valid proxy secret key sk.

$$sk = x_n H(m_w, r_n) (\mathrm{mod} q).$$

The proxy public key α can be generated by any verifier as follows.

$$\alpha = y_B \prod_{i=1}^{n} (r_i y_i^{H(m_w, r_i)} (\mathrm{mod} p).$$

Now we show that $\alpha \equiv g^{sk}(\mathrm{mod}p)$ as follows:

$$\begin{aligned}\alpha &\equiv y_B \prod_{i=1}^{n}(r_i y_i^{H(m_w,r_i)}) \\ &\equiv (y_B \prod_{i=1}^{n-1}(r_i y_i^{H(m_w,r_i)}))r_n y_n^{H(m_w,r_n)} \\ &\equiv y_n^{H(m_w,r_n)} \\ &\equiv g^{x_n H(m_w,r_n)} \\ &\equiv g^{sk}(\mathrm{mod}p).\end{aligned}$$

After proxy secret key sk has been forged, A_n can impersonate proxy signer B and calculate blind signatures on behalf of all original signers $A_i(1 \le i \le n)$.

3 Cryptanalysis of Xue-Cao's Scheme

3.1 Review of Xue-Cao's Scheme

In Xue-Cao's scheme [11], the parameters are defined as follows. p and q are two large primes such that $q|p-1$. g is a generator with order q in Z_p^*. The original signer A's secret key and public key are $x_A \in Z_q^*$ and $y_A = g^{x_A}(\mathrm{mod}p)$ respectively. Likewise, the proxy signer B's secret key and public key are $x_B \in Z_q^*$ and $y_B = g^{x_B}(\mathrm{\ mod\ } p)$ respectively. h is a public secure hash function.

In the proxy phase, A selects $\overline{k} \in Z_q^*$ and computes $\overline{r} = g^{\overline{k}}(\mathrm{mod}p)$ and $\overline{s} = \overline{k} + x_A h(m_w, \overline{r})$ where m_w is a warrant. A sends the 3-tuple $(m_w, \overline{r}, \overline{s})$ to B. B checks whether the following equation holds or not

$$g^{\overline{s}} \equiv \overline{r} y_A^{h(m_w,\overline{r})}(\mathrm{mod}p).$$

If it holds, B continues to compute

$$\begin{aligned}s' &= \overline{s} + x_B y_B(\mathrm{mod}q) \\ y_P &= g^{s'} = g^{\overline{s}} y_B^{y_B} = \overline{r} y_A^{h(m_w,\overline{r})} y_B^{y_B}(\mathrm{mod}p),\end{aligned}$$

as his/her secret and public proxy signature key, respectively. After proxy secret key s' has been generated, proxy signer B can calculate blind signatures on behalf of the original signer A. Anyone can verify the proxy blind signature using proxy public key y_P.

3.2 Cryptanalysis

We show that Xue-Cao's proxy blind signature scheme is insecure against the original signer's forgery. In Xue-Cao's proxy blind signature scheme, if the original signer A can construct 3-tuple $(s', m_w, \overline{r})$ such that $g^{s'} \equiv \overline{r} y_A^{h(m_w,\overline{r})} y_B^{y_B}(\mathrm{mod}p)$, the original signer A can forge a valid proxy secret key s'. The detail attack is described as follows. A selects proxy warrant m_w and $r \in Z_q^*$ at random. A computes $\overline{r} = (y_B^{y_B})^{-1} g^r(\mathrm{mod}p)$. A computes

$$\begin{aligned}s' &= r + x_A h(m_w, \overline{r})(\mathrm{mod}q) \\ y_P &= \overline{r} y_A^{h(m_w,\overline{r})} y_B^{y_B}(\mathrm{mod}p),\end{aligned}$$

as his/her secret and public proxy signature key, respectively. Now we show that $y_P \equiv g^{s'}(\mathrm{mod}p)$ as follows:

$$\begin{aligned}y_P &\equiv \overline{r} y_A^{h(m_w,\overline{r})} y_B^{y_B} \\ &\equiv (y_B^{y_B})^{-1} g^r y_A^{h(m_w,\overline{r})} y_B^{y_B} \\ &\equiv g^r y_A^{h(m_w,\overline{r})} \\ &\equiv g^{r+x_A h(m_w,\overline{r})} \\ &\equiv g^{s'}(\mathrm{mod}p).\end{aligned}$$

After proxy secret key s' has been forged, A can impersonate proxy signer B and calculate blind signatures on any message.

4 Cryptanalysis of Fu-Kou-Xiao's Scheme

4.1 Review of Fu-Kou-Xiao's Scheme

In [3], Fu et al. proposed a proxy signature scheme with proxy signer's privacy anonymity. The parameters are same as Xue-Cao's proxy blind signature scheme.

In SETUP step, the original signer A blinds all of his designated proxy signer's actual identities by giving every proxy signer a new identity, called proxy identity. To blind the identity of a proxy signer B, the original signer A randomly chooses a number $k_B \in Z_q^*$ and computes $ID_P = h(k_B, ID_B)$. A secretly sends ID_P to B and records the tuple (ID_B, ID_P, k_B), for later use of anonymity revocation.

In DELEGATE step, A computes $r = g^{k_B}(\mathrm{\ mod\ } p)$ and $s_A = x_A h(w_B, r) + k_B \mathrm{\ mod\ } p$ where m_B is a warrant. A sends the signature and the warrant together (r, s_A, m_B) to B. B verifies s_A by $g^{s_A} \equiv y_A^{h(w_B,r)} r^r \mathrm{\ mod\ } p$.

In SIGN step, B generates a new secret and public key pair (x_{BP}, y_{BP}) where $y_{BP} = g^{x_{BP}} \mathrm{\ mod\ } p$. B makes y_{BP} public under the name of ID_P. B computes $x_P = (s_A + x_{BP}) \mathrm{\ mod\ } p$ and $T = ID_P ID_B y_A^{x_{BP}}$. Here x_P is the proxy signing key of B, y_B is the corresponding public key. B uses a conventional signature scheme to produce a proxy signature of message m by computing $s = Sign(m, x_P)$. The tuple $(m, s, r, w_B, T, y_{BP})$ is a proxy signature of m on behalf of A.

To verify a proxy signature, a verifier computes $y_P = y_A^{h(w_B,r)} r^r y_{BP} \mathrm{\ mod\ } p$ and checks $Ver(m, s, y_P)$ true or not.

In OPEN step, the original signer A firstly verifies the proxy signature's validity and then computes $a = T/y_{BP}^{x_A} \mathrm{\ mod\ } p$ and $b = T/(ID_P y_{BP}^{x_A}) \mathrm{\ mod\ } p$. A checks all of the recorded tuple (ID_P, ID_B, k_B) to find the corresponding one that satisfies $ID_B = b$ and $ID_B ID_P = a$. Therefore, the actual identity of the proxy signer is ID_B in (ID_P, ID_B, k_B).

4.2 Cryptanalysis

Original signer's forgery also exists in Fu et al.'s anonymous proxy signature scheme. In Fu et al.'s scheme, the original signer A randomly chooses a number $k_B \in Z_q^*$, computes $ID_P = h(k_B, ID_B)$ and records the tuple (ID_B, ID_P, k_B). A selects proxy warrant m_B and $r, t \in Z_q^*$ at random. A computes $y_{BP} = (r^r)^{-1} g^t (\mathrm{mod} p)$.

A computes

$$\begin{aligned} x_P &= t + x_A h(m_B, r) (\mathrm{mod} q) \\ y_P &= y_A^{h(w_B, r)} r^r y_{BP} \bmod p, \end{aligned}$$

as his/her secret and public proxy signature key, respectively. A computes $T = ID_P ID_B y_{BP}^{x_A}$.

Now we show that $y_P \equiv g^{x_P} (\mathrm{mod} p)$ as follows:

$$\begin{aligned} x_P &\equiv y_A^{h(w_B, r)} r^r y_{BP} \\ &\equiv y_A^{h(w_B, r)} r^r (r^r)^{-1} g^t \\ &\equiv y_A^{h(w_B, r)} g^t \\ &\equiv g^{t + x_A h(w_B, r)} \\ &\equiv g^{x_P} (\mathrm{mod} p). \end{aligned}$$

After the proxy secret key x_P has been forged, A can impersonate proxy signer B and calculate a signature on any message. In the OPEN procedure, we also have $a = T / y_{BP}^{x_A} \bmod p = ID_A ID_P$ and $b = T/(ID_P y_{BP}^{x_A}) \bmod p = ID_B$.

5 Cryptanalysis of Gu-Li-Yang's Scheme

5.1 Review of Gu-Li-Yang's Scheme

In [4], Gu et al. proposed a anonymous proxy signature scheme without a trusted party. The parameters are defined as follows. p and q are two large primes such that $q | p - 1$. g is a generator with order q in Z_p^*. The original signer M's secret key and public key are $x_M \in Z_q^*$ and $y_M = g^{x_M} (\mathrm{mod} p)$ respectively. Likewise, the proxy signer P's secret key and public key are $x_P \in Z_q^*$ and $y_P = g^{x_P} (\mathrm{mod} p)$ respectively. h is a public secure hash function.

The original signer M sends his identity ID_M and the warrant m_w to proxy signer P through a secure channel. The proxy signer P randomly chooses two number k_P, $k_1 \in Z_q^*$ and computes $K_P = g^{k_P} (\mathrm{mod} p)$, $s_P = x_P + k_P K_P (\mathrm{mod} q)$, $K_1 = g^{k_1} (\mathrm{mod} p)$, $s_1 = x_P h(K_P, ID_P, K_1) + k_1 (\mathrm{mod} q)$. P sends K_P, ID_P, K_1 and s_1 to M. The original signer M accepts (K_P, ID_P, K_1, s_1) by checking whether the following equation holds: $g^{s_1} \equiv y_P^{h(K_P, ID_P, K_1)} (\mathrm{mod} p) g^{s_1} \equiv y_P^{h(K_P, ID_P, K_1)} K_1 (\mathrm{mod} p)$. If (K_P, ID_P, K_1, s_1) passes this checking, M records the tuple (K_P, ID_P) for later use for anonymity revocation and writes $Y_P = y_P K_P^{K_P}$ into m_w.

The original signer M randomly chooses a number $k_M \in Z_q^*$ and computes $K_M = g^{k_M}$, $s_M = x_M h(m_w, K_M) + k_M (\mathrm{mod} q)$ where m_w is a modified warrant. M sends the signature and the warrant together (K_M, s_M, m_w) to P in a secure manner. P verifies s_M by $g^{s_M} \equiv y_M^{h(m_w, K_M)} K_M (\mathrm{mod} p)$.

P computes $s = s_M + s_P (\mathrm{mod} p)$ and signature $\sigma = Sign(m, s)$. Here s is the proxy signing key of B. The tuple $(m, \sigma, K_M, m_w, ID_M)$ is a proxy signature of m on behalf of M.

To verify a proxy signature, a verifier computes $v = y_M^{h(m_w, K_M)} K_M Y_P \bmod p$ and checks $Ver(m, \sigma, v)$ true or not.

In OPEN step, the original signer M firstly verifies the proxy signature's validity, and then checks all of the recorded tuple (K_P, ID_P) to find the corresponding one that satisfies $Y_P = y_P K_P^{K_P} (\mathrm{mod} p)$ where Y_P can be obtained from m_w. Therefore, the actual identity of the proxy signer is ID_P in (K_P, ID_P).

5.2 Cryptanalysis

Original signer's forgery also exists in Gu et al.'s anonymous proxy signature scheme. In Gu et al.'s scheme, the original signer M randomly chooses a number $k_P \in Z_q^*$, computes $K_P = g^{k_P} (\mathrm{mod} p)$, $Y_P = y_P K_P^{K_P} (\mathrm{mod} p)$ and records the tuple (K_P, ID_P).

M selects a proxy warrant m_w and writes Y_P into m_w. M selects $r \in_R Z_p^*$ at random and computes $K_M = (Y_P)^{-1} g^r (\mathrm{mod} p)$.

M computes

$$\begin{aligned} s &= r + x_M h(m_w, K_M) (\mathrm{mod} q) \\ v &= K_M y_M^{h(m_w, K_M)} Y_P \bmod p, \end{aligned}$$

as his/her secret and public proxy signature key, respectively. Now we show that $v \equiv g^s (\mathrm{mod} p)$ as follows:

$$\begin{aligned} v &\equiv K_M y_M^{h(m_w, K_M)} Y_P \\ &\equiv (Y_P)^{-1} g^r y_M^{h(m_w, K_M)} Y_P \\ &\equiv g^r y_M^{h(m_w, K_M)} \\ &\equiv g^{r + x_M h(m_w, K_M)} \\ &\equiv g^s (\mathrm{mod} p). \end{aligned}$$

After proxy secret key s has been forged, M can impersonate proxy signer P and calculate a signature on any message m. In the OPEN procedure, we also have $Y_P = y_P K_P^{K_P} (\mathrm{mod} p)$.

6 Cryptanalysis of Dai-Yang-Dong and Huang-Hao-Wang Schemes

6.1 Review of Dai-Yang-Dong and Huang-Hao-Wang Schemes

Two similar nominative proxy signature schemes were independently proposed by Dai et al. [2] and Huang et al. [5]. Here we only describe Dai-Yang-Dong's designated-receiver proxy signature scheme. The parameters are defined as follows. p and q are two large primes such that $q|p-1$. g is a generator with order q in Z_p^*. The original signer A's secret key and public key are $x_A \in Z_q^*$ and $y_A = g^{x_A}(\mathrm{mod}p)$ respectively. Likewise, the proxy signer B's secret key and public key are $x_B \in Z_q^*$ and $y_B = g^{x_B}(\mathrm{mod}p)$ respectively, and the verifier C's secret key and public key are $x_C \in Z_q^*$ and $y_C = g^{x_C}(\mathrm{mod}p)$ respectively. h is a public secure hash function.

In the proxy phase, A selects $k_A \in Z_q^*$ and computes $r_A = g^{k_A}(\mathrm{mod}p)$, $e_A = h(M, y_C, r_A)$ and $s_A = x_A e_A + k_A(\mathrm{mod}q)$. The receiver C is designated by the original signer A through the form of the receiver's public key y_C in the signature. A sends the 5-tuple (M, s_A, r_A, y_C, y_A) to B. B checks whether the following equation holds or not

$$g^{s_A} = y_A^{h(M,y_C,r_A)} r_A(\mathrm{mod}p)$$

If it holds, B continue to compute

$$\begin{aligned} x_P &= s_A + x_B(\mathrm{mod}q) \\ y_P &= g^{x_P} = y_A^{h(M,y_C,r_A)} r_A y_B(\mathrm{mod}p), \end{aligned}$$

as his/her secret and public proxy signature key, respectively. After proxy secret key x_P has been generated, proxy signer B can calculate the designated verifier signature on behalf of the original signer. The designated verifier C can verify the proxy signature using his/her secret key.

6.2 Cryptanalysis

We show that Dai et al. and Huang et al.'s nominative proxy signature schemes are insecure against the original signer's forgery.

In Dai-Yang-Dong's scheme, if the original signer A can construct 3-tuple (x_P, M, r_A) such that $g^{x_P} \equiv y_A^{h(M,y_C,r_A)} r_A y_B(\mathrm{mod}p)$, the original signer A can forge a valid proxy secret key x_P. The detailed attack is described as follows.

A selects M and $r \in Z_q^*$ at random. A computes $r_A = (y_B)^{-1} g^r(\mathrm{mod}p)$. A computes

$$\begin{aligned} x_P &= r + x_A h(M, y_C, r_A)(\mathrm{mod}q) \\ y_P &= y_A^{h(M,y_C,r_A)} r_A y_B(\mathrm{mod}p), \end{aligned}$$

as his/her secret and public proxy signature key, respectively. Now we show that $y_P \equiv g^{x_P}(\mathrm{mod}p)$ as follows:

$$\begin{aligned} y_P &\equiv y_A^{h(M,y_C,r_A)} r_A y_B \\ &\equiv y_A^{h(M,y_C,r_A)} (y_B)^{-1} g^r y_B \\ &\equiv y_A^{h(M,y_C,r_A)} g^r \\ &\equiv g^{r + x_A h(M,y_C,r_A)} \\ &\equiv g^{x_P}(\mathrm{mod}p). \end{aligned}$$

After proxy secret key x_P has been forged, A can impersonate proxy signer B and calculate the designated verifier signature on message M. The same weakness also exists in Huang-Hao-Wang's scheme.

7 Conclusion

In this paper, we presented the cryptanalysis of many proxy signature scheme recently published in [2, 3, 4, 5, 7, 11]. Our results show that these schemes are all insecure, i.e., forgeable. In our attacks, an original signer could generate a valid proxy secret key without the knowledge of the proxy signer's secret key.

Acknowledgments

This work was supported by the Science and Technology Foundation of CUMT. We thank the anonymous reviewers for their valuable suggestions in improving the quality of this paper.

References

[1] D. Chaum, "Blind signatures for untraceable payments", in *Crypto'82*, pp. 199-203, 1982.

[2] J. Dai, X. Yang, and J. Dong, "Designated-receiver proxy signature scheme", *Journal of Zhejiang University (Engineer Science)*, vol. 38, no. 11, pp. 1422-1425, 2004.

[3] X. Fu, W. Kou, and G. Xiao, "A proxy signature scheme with proxy signer's privacy anonymity," in *Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business*, pp. 257-260, 2004.

[4] L. Gu, Z. Li, and Y. Yang, "A Anonymous Proxy Signature Scheme without a Trusted Party", *Journal of Beijing University of Posts and Telecommunication*, vol. 28, no. 1, pp.48-50, 2005.

[5] Z. Huang, Y. Hao, and Y. Wang, "Nominative Signature and Nominative Proxy Signature", *Journal of Electronics & Information Technology*, vol. 26, no. 12, pp.1996-2001, 2004.

[6] W. D. Lin, and J. K. Jan, "A security personal learning tools using a proxy blind signature scheme", in *Proceedings of International Conference on Chinese Language Computing*, pp. 273-277, 2000.

[7] R. Lu, Z. Cao, and Y. Zhou, "Proxy blind multi-signature scheme without a secure channel", *Applied Mathematics and Computation*, vol. 1649, no. 1, pp.179-187, 2005.
[8] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature: Delegation of the power to sign messages", *IEICE Transactions on Fundamentals*, vol. E79-A, no. 9, pp. 1338-1353, 1996.
[9] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation", in *Proceedings of 3rd ACM Conference on Computer and Communications Security*, pp. 48-57, 1996.
[10] T. Okamoto, "Provable secure and practical identification schemes and corresponding signature schemes", in *Crypto'92*, LNCS 740, pp. 31-53, 1992.
[11] Q. Xue and Z. Cao, "A new proxy blind signature scheme with warrant", in *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems*, pp. 1385-1390, 2004.

Tianjie Cao received his B.S. and M.S. degrees in Mathematics from Nankai University in 1990 and 1993 respectively. Currently he is Associate Professor of Computer Science at China University of Mining and Technology. He is also ph.D candidate at the State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences. His research interests include cryptographic protocols and network security. E-mail address: tjcao@is.iscas.ac.cn

Xianping Mao received his B.S. degree in computer science from China University of Mining and Technology in 2004. He is now working toward the M.S. degree in the same University. His research interests lie in information security. E-mail address: xpmoore@126.com