Notes on Proxy Signcryption and Multi-proxy Signature Schemes

Chunhua Pan[1], Shunpeng Li[1], Qihui Zhu[1], Chunzhi Wang[2], and Mingwu Zhang[1,2]
(Corresponding author: Mingwu Zhang)

College of Information, South China Agricultural University[1]
School of Computer Sciences, Hubei University of Technology[2]
(Email: csmwzhang@gmail.com)
(Received Dec. 26 2012; revised and accepted Oct. 1, 2013)

Abstract

Proxy signcrytipn scheme allows an original signer to delegate his signing power to a proxy such that the latter can signcrypt a message on behalf of the former. Recently, Lin et al. proposed a proxy signcryption with CCA and CMA security. In this work, we indicate that the Lin et al.'s proxy signcryption scheme does not hold the security of indistinguishability against adaptive chosen-ciphertext attacks and existential unforgeability against adaptive chosen-message attacks. Also, we show that the Jin-Wen's certificateless multi-proxy signature scheme does not hold the security of existential unforgeability against adaptive chosen-message attacks.
Keywords: Cryptanalysis, multi-proxy signature, proxy signcryption, unforgeability

1 Introduction

Proxy signcryption, first proposed by Gamage et al. [5, 12, 13], is a cryptographic primitive, which combines the functionality of a proxy signature scheme with that of an encryption, to allow an original signer to delegate his signing power to a proxy one such that the proxy can signcrypt a message on behalf of the delegator. The signcrypted message can only be decrypted by a designated recipient who is also responsible for verifying the recovered proxy signature function [2, 4, 9]. Recently, Lin et al. [10, 11] proposed an efficient proxy signcryption scheme based on bilinear pairings. Jin and Wen et al. [8] proposed a multi-proxy signature scheme in certificateless setting. They also stated that their scheme achieves the *confidentiality* against indistinguishability under adaptive chosen-ciphertext attacks (IND-CCA2) and *unforgeability* against existential forgery under adaptive chosen-message attacks (UEF-CMA2) in the random oracle models.

Multi-proxy signature, which was first introduced by Hwang and Shi et al. [3, 6, 7], could be viewed as a variation of the proxy signature primitive. In such a scheme, an original signer delegates his signing power to a group of proxy signers, and then only the cooperation of all proxy signers can generate proxy signatures, referred to as multi- proxy signatures, on behalf of the original signer.

In this work, we show that the Lin et al.'s proxy signcryption scheme [10] is insecure since they cannot obtain the unforgeability and forward security. We also indicate that the Jin-Wen's certificateless multi-proxy signature scheme does not hold the security of existential unforgeability against adaptive chosen-message attacks of their declared.

2 Model of Proxy Signcryption and Multi-proxy Signature

2.1 Proxy Signcryption

Definition 1. ***Proxy Signcryption.*** *A proxy signcryption contains four probabilistic polynomial-time algorithms:*

1) *Setup: Taking as input 1^k where k is a security parameter, the algorithm generates the systems public parameters pp*

2) *Proxy-Credential-Generation (PCG): The PCG algorithm takes as input the private key of original signer and outputs a corresponding proxy credential for the proxy signer.*

3) *Signcrypted-Message-Generation (SMG): The SMG algorithm takes as input a plaintext m, a proxy credential, the public key of designated recipient and the private key of proxy signer, and outputs signcrypted message δ.*

4) *Signature-Recovery-and-Verification (SRV): The SRV algorithm takes as input a signcrypted message δ, the private key of designated recipient and the public keys of original and proxy signers, and outputs a plaintext m and its converted ordinary*

proxy signature if the signcrypted message is valid, and returns an error symbol $\perp$ otherwise.

2.2 Certificateless Multi-proxy Signature

Definition 2. ***Certificateless Multi-proxy Signature.*** *A certificateless multi-proxy signature scheme is defined by a collection of probabilistic polynomial-time algorithms as follows:*

1) *Setup: Given a security parameter k, the PKG generates a master key s and the system parameters pp.*

2) *Partial-Private-Key-Extract (PPKE): Given a user's identity ID_i, the PKG produces the corresponding partial private key D_i with the master key s after verifying the user's identity.*

3) *User-Key-Generate (UKG): After receiving the partial private key D_i from PKG, the user with identity ID_i randomly selects a secret value x_i to construct his full private key sk_i with D_i, and publishes his public key P_i w.r.t x_i.*

4) *Sign: Given a message m, the user, whose identity is ID_i and public key is P_i, generates a signature σ on m with his private key sk_i.*

5) *Verify: Given a signature σ on message m, the verifier accepts it if σ is a valid signature relative to m, the signer's identity ID_i and his public key P_i and rejects otherwise.*

6) *Proxy-Key-Generate (PKG): It is a protocol between the original signer and all proxy signers formed by a group of interactive randomized algorithms. All participants take their identities ID_{OS} and $\mathsf{ID}_{PS_1}, \mathsf{ID}_{PS_2}, \cdots, \mathsf{ID}_{PS_n}$ as inputs. Additionally, the original signer also takes his secret key sk_{OS} and the delegation warrant w as inputs, where w includes the restrictions on the class of messages delegated, the identities of the original signer and all proxy singers, the period of delegation and etc. Every proxy signer also takes his secret key sk_{PS_i} as input. As a result, each proxy signer gets a multi-proxy signature secret key PSK_i which could be used to cooperatively produce multi-proxy signatures with other proxy signers.*

7) *Multi-Proxy-Sign (MPS): Given a message m which satisfies the requirements stated in w, all proxy signers cooperatively produce a multi-proxy signature σ_{MPS} on behalf of the original signer with the multi-proxy signature secret keys PSK_i for $i \in [n]$.*

8) *Multi-Proxy-Verify (MPV): Given a multi-proxy signature σ_{MPS} on message m under the warrant w, the verifier accepts it if σ_{MPS} is a valid signature relative to m and w by proxy signers $PS_1, PS_2, \cdots, PS_n$ on behalf of the original signer OS.*

For certificateless cryptosystems, the widely accepted notion of security was defined by Al-Riyami and Paterson [1]. According to their definitions, two types of adversaries with different capabilities were considered, which could be described as follows:

1) **Type I Adversary $\mathcal{A}_I$**: This type of adversary acts as a dishonest user who does not have access to the master key but has the ability to replace the public key of any entity with a value of his choice.

2) **Type II Adversary $\mathcal{A}_{II}$**: This type of adversary acts as a malicious PKG who has access to the master key but cannot perform the public key replacements.

3 Review of Lin et al.'s Proxy Signcryption

Setup Taking as input 1^k, the system authority selects two groups $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, \times)$ of the same prime order q. Let P be a generator of order q over $\mathbb{G}_1$, $\hat{e} : \mathbb{G}_1^2 \to G_2$ a bilinear pairing and $h_1 : \{0,1\} \times \mathbb{G}_1 \to \mathcal{Z}_q$, $h_2 : \mathbb{G}_1 \to \mathbb{G}_1$, $h_3 : \mathbb{G}_2 \times \mathbb{G}_1 \to \{0,1\}^k$ be collision resistant hash functions. The system publishes $pp = (\mathbb{G}_1, \mathbb{G}_2, q, P, \hat{e}; h_1, h_2, h_3)$. Each user $\mathcal{U}_i$ chooses his private key $x_i \in \mathcal{Z}_q$ and computes the corresponding public key as $Y_i = x_i P$.

PCG Let $\mathcal{U}_o$ be an original signer delegating his signing power to a proxy signer $\mathcal{U}_p$. $\mathcal{U}_o$ first chooses an integer $d \in \mathcal{Z}_q$ and a warrant m_w to compute $N = dP$, $\sigma = x_o + d(m_w) \bmod q$.

SMG To signcrypt a message $m \in \{0,1\}^k$ on behalf of the original signer U_o, U_p chooses $r \in \mathcal{Z}_q$ to compute: $R = rP$, $S = r(h_1(m, R) + x_p + \sigma)^{-1} P$, $V = \hat{e}(h_2(\sigma Y_v), x_p Y_v)$, $X = E_V(S)$ and $Y = h_3(V, R) \oplus m$. It outputs the ciphertext $\delta = (R, X, Y, N)$ together with the warrant m_w.

SRV Upon receiving $\delta = (R, X, Y, N)$, U_v computes $V = \hat{e}(h_2(x_v(Y_o + m_w N)), x_v Y_p)$, $m = h_3(V, R) \oplus Y$, $S = D_V(X)$, and accepts the message if $\hat{e}(h_1(m, R) + Y_p + Y_o + m_w N, S) = \hat{e}(P, R)$ holds.

3.1 Cryptanalysis

In this section, we give a forgery attack and a confidentiality attack to show that the Lin et al.'s scheme does not hold the claimed properties such as unforgeability against UEF-CMA2 and confidentiality against IND-CCA2.

3.1.1 Unforgeability Attack

Definition 3. ***Unforgeability of Proxy Signcryption.*** *A proxy signcryption scheme is said to achieve unforgeability against existential forgery under adaptive chosen-message attacks (EF-CMA) if there exists*

no probabilistic polynomial-time) forger $\mathcal{F}$ with non-negligible advantage in the following game played with a challenger $\mathcal{B}$:

Setup *$\mathcal{B}$ runs the $\mathsf{Setup}(1^k)$ algorithm and sends the system's public parameters pp to the forger $\mathcal{F}$.*

Phase 1 *The forger $\mathcal{F}$ can issue several kinds of following queries adaptively.*

- *PCG queries: $\mathcal{F}$ issues a PCG query with respect to the target proxy signer. $\mathcal{B}$ returns the corresponding warrant and its proxy credential (σ, N, m_w).*
- *SMG queries: $\mathcal{F}$ chooses a message m and a warrant, and $\mathcal{B}$ outputs the corresponding signcrypted message δ to $\mathcal{F}$.*
- *SRV queries: On receiving a signcrypted ciphertext δ with its warrant sent by $\mathcal{F}$, $\mathcal{B}$ returns a message m and its converted proxy signature if the signcrypted message δ is valid. Otherwise, an error symbol $\perp$ is returned.*

Forgery *$\mathcal{F}$ arbitrarily chooses a message m and produces a ciphertext δ^* which is not outputted by the SMG query. The forger $\mathcal{F}$ wins if δ^* is valid.*

Forgery Attacks. We now show that the receiver U_v may forge a new valid ciphertext $\tilde{\delta}$ for any message $\tilde{m}$ on behalf of the proxy signcrypter U_p. Receiver U_v does

1) Random pick $\tilde{r} \in \mathcal{Z}_q$, compute $\tilde{R} = \tilde{r}(Y_o + Y_p + m_w N)$ and $\tilde{S} = \tilde{r}P - \tilde{r} \cdot h_1(\tilde{m}, \tilde{R})P$.
2) Compute $V = \hat{e}(h_2(x_v(Y_o + m_w N)), x_v Y_p)$ using U_v's secret key x_v.
3) Compute $\tilde{Y} = \tilde{m} \oplus h_3(V, \tilde{R})$.
4) Set $\tilde{X} = E_V(\tilde{S})$.
5) Output the forged ciphertext $\tilde{\delta} = (\tilde{R}, \tilde{X}, \tilde{Y}, N)$.

The forged ciphertext $\tilde{\delta} = (\tilde{R}, \tilde{X}, \tilde{Y}, N)$ is valid for the decryption algorithm SRV: $V = \hat{e}(h_2(x_v(Y_o + m_w N)), x_v Y_p)$, $\tilde{m} = h_3(V, \tilde{R}) \oplus \tilde{Y}$, $S = D_V(\tilde{X})$.

$$\begin{aligned}
&\hat{e}(h_1(\tilde{m}, \tilde{R})P + Y_p + Y_o + m_w N, \tilde{S}) \\
=&\hat{e}(h_1(\tilde{m}, \tilde{R})P + Y_p + Y_o + m_w N, \tilde{r}P - \tilde{r}h_1(\tilde{m}, \tilde{R})P) \\
=&\hat{e}(Y_p + Y_o + m_w N, P)^{\tilde{r}} \\
=&\hat{e}(P, \tilde{R}).
\end{aligned}$$

Remark 1. *Because the verification equation is only to verify the components R and S where $R = rP$ and $S = r(h_1(m, R) + x_p + \sigma)^{-1}P = (h_1(m, R) + x_p + \sigma)^{-1}R$. We can construct the new $\tilde{R}, \tilde{S}$ such that $\tilde{R} = \tilde{r}(Y_o + Y_p + m_w N)$ and $\tilde{S} = \tilde{r}P - \tilde{r}h_1(m, \tilde{R})P$. Then $\tilde{R}, \tilde{S}$ have the same relation with R, S in the verification. i.e.,*

$$\hat{e}(h_1(m, R)P + Y_o + Y_p + m_w N, S) = \hat{e}(P, R) \Leftrightarrow$$

$$\hat{e}(h_1(m, R)P + Y_o + Y_p + m_w N, \tilde{S}) = \hat{e}(P, \tilde{R}).$$

Remark 2. *Actually, any user can forge a signcrypted ciphertext on behalf of the proxy signcrypter successfully, since anyone may compute the proxy agreement key V with his secret key.*

3.1.2 Confidentiality Attack

Definition 4. ***Confidentiality.*** *A proxy signcryption scheme is said to achieve the security requirement of confidentiality against indistinguishability under adaptive chosen-ciphertext attacks (IND-CCA2) if there is no PPT distinguisher $\mathcal{D}$ with non-negligible advantage in the following game played with a challenger $\mathcal{B}$.*

Setup *$\mathcal{B}$ first runs the $\mathsf{Setup}(1^k)$ algorithm and sends the system's public parameters pp to $\mathcal{D}$.*

Phase 1 *The distinguisher $\mathcal{D}$ adaptively issues PCG, SMG and SRV queries as those in Phase 1 of unforgeability definition.*

Challenge *$\mathcal{D}$ produces two plaintexts m_0 and m_1 of the same length, then $\mathcal{B}$ flips a coin $\eta \in \{0, 1\}$ and generates a ciphertext δ^* for m_η. The ciphertext δ^* is then delivered to $\mathcal{D}$ as a target challenge.*

Phase 2 *The distinguisher $\mathcal{D}$ issues new queries as those in Phase 1, except the SRV query for the target challenge δ^*.*

Guess *$\mathcal{D}$ outputs a bit η' and wins the game if $\eta' = \eta$.*

Confidentiality Attacks. We show that the scheme is not forward secure as the confidentiality definition declared. In the forward security definition, only designated recipient can decrypt the message legally. That is, it is infeasible for a distinguisher $\mathcal{D}$ to extract the message even though the signcrypter leaks his secret key to $\mathcal{D}$. To guess the message m_η in the ciphertext $\delta^* = (R, X, Y, N)$, $\mathcal{D}$ gets the guess η as follows:

1) Computes $V = \hat{e}(h_2(\sigma Y_v), x_p Y_v)$;
2) Recovers $m = Y \oplus h_3(V, R)$.
3) If $m = m_0$, $\mathcal{D}$ outputs $\eta' = 0$ as the guess, otherwise outputs $\eta' = 1$.

Remark 3. *In Lin et al.'s proxy signcryption scheme, the agreement key V between the proxy signcrypter and the decrypter is fixed and constant that does not import a randomness. This means that any ciphertext generated by proxy signcrypter U_p to U_v may be decrypted using this decrypted key V. This violates the probabilistic encryption principle.*

4 Cryptanalysis of Jin-Wen Certificateless Multi-proxy Signature Scheme

4.1 Review of Jin-Wen's Scheme

Setup Given a security parameter k, the PKG does as follows: first choose groups $\mathbb{G}$ and G_T of prime order q such that an admissible bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ can be constructed and pick an arbitrary generator P of $\mathbb{G}$; Choose a random number $s \in \mathcal{Z}_q$ as the master key msk and set $Q = sP$ as the master public key; Choose six different cryptographic hash functions that $H_1, h_2, H_3 : \{0,1\}^* \to \mathbb{G}$ and $H_4, H_5, H_6 : \{0,1\}^* \to \mathcal{Z}_q$; Finally, output and publish system parameters $pp = (\mathbb{G}, \mathbb{G}_T, \hat{e}, P, Q, H_1, H_2, H_3, H_4, H_5, H_6)$ while keeping the master key $msk = s$.

PPKE Given a user's identity $\mathsf{ID} \in \{0,1\}^*$, the PKG generates the partial private key for the user by computing $D = sH_1(\mathsf{ID})$ and sends D to user ID.

UKG The user with identity ID selects a random number $x \in \mathcal{Z}_q$, sets his public key as $P_{\mathsf{ID}} = xP$ and makes it public while keeping the secret value x and the partial private key D as his secret key sk_{ID}.

Sign To sign a message $m \in \{0,1\}^*$ with $sk = (x, D)$, the signer (identity ID and $pk = P_{\mathsf{ID}}$) first chooses a random number $r \in \mathcal{Z}_q$ and computes $R = rP$; computes $W = H_2(pp)$, $T = H_3(Q)$, $h = H_4(pp, m, \mathsf{ID}, P, R)$ and $V = hD + xW + rT$. Finally, outputs $\sigma = (R, V)$ as the signature.

Verify. The verifier checks whether $\hat{e}(V, P) = \hat{e}(hH_1(\mathsf{ID}), Q)e(W, P_{\mathsf{ID}})e(T, R)$ holds, where $W = H_2(pp)$, $T = H_3(Q)$, $h = H_4(pp, m, \mathsf{ID}, P_{\mathsf{ID}}, R)$.

PKG The proxy key generation algorithm performs as follows:

1) Delegation generation: To delegate the signing capability, the original signer o, with identity ID_o and public key P_o, first makes the signed warrant w which specifies the necessary proxy details, such as the identities of the original signer and the proxy signers, the type of messages delegated, the period of delegation and etc. Then he produces the delegation as follows
   a. Choose a random number $r_0 \in \mathcal{Z}_q$ and compute $R_0 = r_0 P$;
   b. Compute $h_0 = H_5(pp, w, \mathsf{ID}_o, P_o, R_0)$, $W = H_2(pp)$, $T = H_3(Q)$ and $V_0 = h_0 D_o + x_o W + r_0 T$;
   c. Send (w, R_0, V_0) to each proxy signer ps_i, $i = 1, \ldots, n$.
2) Delegation verification: After receiving the delegation (w, R_0, V_0) from the original signer o, each proxy signer ps_i confirms its validity by checking $\hat{e}(V_0, P) = e(h_0 H_1(\mathsf{ID}_o), Q)\hat{e}(W, P_o)\hat{e}(T, R_0)$, where $h_0 = H_5(pp, w, \mathsf{ID}_o, P_o, R_0)$, $W = H_2(pp)$, $T = H_3(Q)$. ps_i accepts it if the equation holds; otherwise, he requests a valid one from o, or terminates the protocol.
3) Proxy secret key generation: If all proxy signers ps_i confirm the delegation, each of them sets $PSK_i = (sk_{ps_i}, R_0, V_0)$ as his multiproxy signature secret key respectively.

MPS Every proxy signer ps_i computes $R_i = r_i P$ with random picked $r_i \in \mathcal{Z}_q$, and $V_i = h_i D_{ps_i} + x_{ps_i} W + r_i T$, where $W = H_2(pp)$, $T = H_3(Q)$ and $h_i = H_6(pp, w, m, \mathsf{ID}_{ps_i}, P_{ps_i}, R_i)$. Sends (w, R_0, V_0, R_i, V_i) to a clerk.
The clerk verifies its validity by checking the equations $\hat{e}(V_0, P) = \hat{e}(h_0 H_1(\mathsf{ID}_o), Q)\hat{e}(W, P_o)\hat{e}(T, R_0)$ and $\hat{e}(V_i, P) = \hat{e}(h_i H_1(\mathsf{ID}_{ps_i}), Q)\hat{e}(W, P_{ps_i})\hat{e}(T, R_i)$. Then it generates the multi-proxy signature as $\sigma_{MPS} = (w, R_{MPS}, V_{MPS})$ where $R_{MPS} = (R_0, R_1, \ldots, R_n)$ and $V_{MPS} = \sum_i V_i$.

MPV To verify a multi-proxy signature $\sigma_{MPS} = (w, R_{MPS}, V_{MPS})$ of the message m, the verifier checks whether: $\hat{e}(V_{MPS}, P) = \hat{e}(h_0 H_1(\mathsf{ID}_o) + \sum_i h_i H_1(\mathsf{ID}_{ps_i})) \cdot \hat{e}(W, \sum_{i \in \Omega} P_i)$.
$\hat{e}(T, \sum_i R_i)$, where $h_0 = H_5(pp, w, \mathsf{ID}_o, P_o, R_0)$, $\Omega = \{o, ps_1, \ldots, ps_n\}$, $W = H_2(pp)$, $T = H_3(Q)$ and $h_i = H_6(pp, w, m, \mathsf{ID}_{ps_i}, P_{ps_i}, R_i)$.

4.2 Forgery Analysis

In this section, we give a forgery attack to show that the Jin-Wen scheme does not hold the claimed security.

4.2.1 Clerk's Forgery

A clerk $\mathcal{C}$ can forgery a multiproxy signature on behalf of the new original delegator $\tilde{o}$. First, $\mathcal{C}$ requests a Proxy-key-gen query between the original delegator $\tilde{o}$ and multi-proxy $ps_1, \ldots, ps_n$, then he gets a delegation $(\tilde{w}, \tilde{R}_0, \tilde{V}_0)$.

After clerk $\mathcal{C}$ obtains all multi-proxy signatures (w, R_0, V_0, R_i, V_i) on the message m from ps_i $(i = 1, \ldots, n)$, $\mathcal{C}$ replaces w, R_0, V_0 with $\tilde{w}, \tilde{R}_0, \tilde{V}_0$ respectively.

Adversary can forge a valid multi-proxy signature $(\tilde{w}, \tilde{R}_0, \tilde{V}_0, \tilde{R}_i, \tilde{V}_i)$ on message $\tilde{m}$ under warrant $\tilde{w} \neq w$ by ID_i where $i = 2, \ldots, n+1$ on behalf of ID_1 (or the challenger). To forge a valid multi-proxy signature, adversary $\mathcal{A}$ does

1) $\mathcal{A}$ makes a warrant $\tilde{w}$ that the original signer's identity is ID_1, the proxy signers's identities are $\mathsf{ID}_2, \ldots, \mathsf{ID}_{n+1}$.
2) $\mathcal{A}$ requests a signature query on $(\mathsf{ID}_1, \tilde{w})$, and obtains an answer $(\tilde{U}_1, \tilde{V}_1)$.

3) $\mathcal{A}$ performs extraction queries for $\mathsf{ID}_2, \ldots, \mathsf{ID}_{n+1}$. That is, $\mathcal{A}$ knows the secret keys of identities $\mathsf{ID}_2, \ldots, \mathsf{ID}_{n+1}$. $\mathcal{A}$ can generate ID_i's proxy key $(sk_{id_i}, \tilde{U}_1, \tilde{V}_1)$.

4) $\mathcal{A}$ generates a valid multi proxy signature $\sigma_{MPS} = (\tilde{w}, \tilde{R}_{MPS}, \tilde{V}_{MPS})$.

Remark 4. *The Jin-Wen certificateless multi-proxy signature scheme, which can be viewed as a two-level hierarchical IBE scheme, is not secure in the proposed security model. The main reason is the direct employment of the proposed scheme that is a simple aggregation of standard signatures produced by multiple original signer and multi-proxy, respectively.*

5 Conclusion

In this work, two attacks were proposed to show that the Lin etl al.'s proxy signcryption scheme does not hold the indistinguishability against CCA2 and existential unforgeability against CMA. Also, existential forgery attacks was presented to demonstrate that a certificateless multi-proxy signature proposed by Jin and Wen does not hold the existential unforgeability.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants 61370224, 61272404 and 61170135, Guangdong Natural Science Foundation under Grant S2012010010383, and Key Program of Natural Science Foundation of Hubei Province under Grant 2013CFA046.

References

[1] S. S Al-Riyami and K. G. Paterson. "Certificateless public key cryptography,". in *Advances in Cryptology - Asiacrypt '03*, pp. 452–473. 2003.

[2] F. Cao and Z. Cao, "A secure identity-based proxy signature scheme," *Information Sciences*, vol. 179, no. 3, pp. 292–302, 2009.

[3] M. L. Das, A. Saxena, and D. B. Phatak, "Algorithms and approaches of proxy signature: A survey," *International Journal of Network Security*, vol. 9, no. 3, pp. 264–284, 2009.

[4] H. Elkamchouchi, M. Nasr, and R. Ismail, "A new efficient strong proxy signcryption scheme based on a combination of hard problems," in *IEEE International Conference on Systems, Man and Cybernetics*, pp. 5123–5127, 2009.

[5] C. Gamage, J. Leiwo, and Y. Zheng, "An efficient scheme for secure message transmission using proxy-signcryption," in *Proceedings of the Twenty Second Australasian Computer Science Conference*, pp. 18–21, 1999.

[6] M. S. Hwang, C. C. Lee, and S. F. Tzeng, "A new proxy signature scheme for a specified group of verifiers," *Information Sciences*, vol. 227, pp. 102–115, 2013.

[7] S. J. Hwang and C. H. Shi, "A simple multi-proxy signature scheme," in *Proceedings of the Tenth National Conference on Information Security*, pp. 134–138, 2000.

[8] Z. Jin and Q. Wen, "Certificateless multi-proxy signature," *Computer Communications*, vol. 34, no. 3, pp. 344–352, 2011.

[9] F. Li, X. Xin, and Y. Hu, "Id-based threshold proxy signcryption scheme from bilinear pairings," *International Journal of Security and Networks*, vol. 3, no. 3, pp. 206–215, 2008.

[10] H. Y. Lin, T. S. Wu, S. K. Huang, and Y. S. Yeh, "Efficient proxy signcryption scheme with provable CCA and CMA security," *Computers & Mathematics with Applications*, vol. 60, no. 7, pp. 1850–1858, 2010.

[11] S. Mashhadi, "A novel non-repudiable threshold proxy signature scheme with known signers," *International Journal of Network Security*, vol. 15, no. 4, pp. 274–279, 2013.

[12] M. Zhang, B. Yang, Z. Chen, and T. Takagi, "Efficient and adaptively secure broadcast encryption systems," *Security and Communication Networks*, vol. 6, no. 8, pp. 1044–1052, 2013.

[13] M. Zhang, J. Yao, C. Wang, and T. Takagi, "Public key replacement and universal forgery of SCLS scheme," *International Journal of Network Security*, vol. 15, no. 1, pp. 115–120, 2013.

Chunhua Pan is a lecturer at College of Information, South China Agriculutral University. His research interests focus on Secure Computations and Network Protocols.

Shunpeng Li is a postgraduate student at College of Information, South China Agriculutral University. His research is in the field of Information Security and Cryptography.

Qihui Zhu is a postgraduate student at College of Information, South China Agriculutral University. His research is in the field of Information Processing and Security.

Chunzhi Wang is a professor at School of Computer Sciences, Hubei University of Technology. Her research interests focus on Network Protocol and System Security.

Mingwu Zhang is a professor at School of Computer Sciences, Hubei University of Technology. He is a senior member of Chinese Computer Federation, a senior member of Chinese Association for Cryptologic Research (CACR), and a member of IEEE Computer Society. His research interests include Secure Multi-party Computation and Information Security.