Some Basic Principles for Proxy Signature Schemes Based on ECDLP[*]

Fengying Li
Dept. of Education Information Technology
East China Normal University
200062, Shanghai, China
fyli@sjtu.edu.cn

Zhenfu Cao
Dept. of Computer Science and Engineering
Shanghai Jiaotong University
200240, Shanghai, China
zfcao@cs.sjtu.edu.cn

Qingshui Xue
Dept. of Computer Science and Engineering
Shanghai Jiaotong University
201101, Shanghai, China
xue-qsh@sjtu.edu.cn

Abstract **- Nowadays, most proxy signature schemes are based on the difficulty of DLP (Discrete Logarithm Problem) or ECDLP (Elliptical Curve Discrete Logarithm Problem). As though many proxy signature schemes based on DLP or ECDLP have been proposed, it makes us discouraged that some disadvantages can be found after a new or modified proxy signature scheme was designed after short time. How to solve the question? How to design secure and valid proxy signature scheme? How to prove them secure? Now, it is too difficult for us to prove one scheme secure, but if we can have some principles to conform to when designing some proxy signature schemes based on DLP or ECDLP, it will be helpful. It will be able to make the scheme designer to make few mistakes, that's to say, by these principles, they can judge their schemes meet basic secure conditions. If designers don't conform to these principles, it can easily be seen that their schemes are definitely insecure. It is all known by us that until now there are not these principles in the real life. By some hints from some attacks, especially forgery attacks, it seems to us that we have found three basic principles which should be conformed to when proxy signature schemes are proposed. The first principle is that the existent forms of public parameters in proxy signatures in the proxy signature verification congruence make a key role on the security property of unforgeability. The second principle is that any public parameter in the proxy signature can't lonely exist in the proxy signature verification congruence in the form of bases or exponents. The third principle is that any public parameter in the proxy signature should exist in the proxy signature verification equation in the form of not only exponents and bases, but also hashes. In addition, some examples are given.**

Index Terms – Cryptograph, digital signature, proxy signature, principle, ECDLP

I. INTRODUCTION

The proxy signature scheme [1], a variation of ordinary digital signature schemes, enables a proxy signer to sign messages on behalf of the original signer. Proxy signature schemes are very useful in many applications such as electronics transaction and mobile agent environment.

Mambo et al. [1] provided three levels of delegation in proxy signature: full delegation, partial delegation and delegation by warrant. In full delegation, the original signer gives its private key to the proxy signer. In partial delegation, the original signer produces a proxy signature key from its private key and gives it to the proxy signer. The proxy signer uses the proxy key to sign. As far as delegation by warrant is concerned, warrant is a certificate composed of a message part and a public signature key. The proxy signer gets the warrant from the original signer and uses the corresponding private key to sign. Since the conception of the proxy signature was brought forward, a lot of proxy signature schemes have been proposed [2-14, 16-18].

Recently, many threshold proxy signature schemes were proposed [2, 6-14]. In threshold proxy signature schemes, a group of n proxy signers share the secret proxy signature key. To produce a valid proxy signature on the message m, individual proxy signers produce their partial signatures on that message, and then combine them into a full proxy signature on m. In a (t,n) threshold proxy signature scheme, the original signer authorizes a proxy group with n proxy members. Only the cooperation of t or more proxy members is allowed to generate the proxy signature. Threshold signatures are motivated both by the demand which arises in some organizations to have a group of employees agree on a given message or document before signing, and by the need to protect signature keys from attacks of internal and external adversaries.

In 1999, Sun proposed a threshold proxy signature scheme with known signers [9]. Then Hwang et al. [7] pointed out that Sun's scheme was insecure against collusion attack. By the collusion, any $t-1$ proxy signers among t proxy signers can

[*] This work is supported by the PhD Program Scholarship Fund of ECNU under Grant No. 2010005, the National Natural Science Foundation of China under Grant 60873217 and National Basic Research Program of China (973 Program) under Grant No.2007CB311100.

COLLABORATECOM 2010, October 9-12, Chicago, USA

DOI 10.4108/icst.collaboratecom.2010.35

cooperatively obtain the secret key of the remainder one. They also proposed an improved scheme which can guard against the collusion attack. After that, [6] showed that Sun's scheme was also insecure against the conspiracy attack. In the conspiracy attack, t malicious proxy signers can impersonate some other proxy signers to generate valid proxy signatures. To resist the attack, they also proposed a scheme. Hwang et al pointed out [8] that the scheme in [7] was also insecure against the attack by the cooperation of one malicious proxy signer and the original signer. In 2002, Li et al. [2] proposed a threshold proxy signature scheme full of good properties and performance.

The multi-proxy signature scheme was first proposed in [14]. The multi-proxy signature scheme is a special case of the threshold proxy signature scheme. The multi-proxy signature scheme allows an original signer to authorize a group of proxy members can generate the multi-signature on behalf of the original signer.

In a designated-verifier proxy signature scheme, the proxy signature will be verified only by a designated verifier chosen by the proxy signer. In 1996, Jakobsson et al. designed a designated-verifier proxy signature scheme for the first time [1]. In [21], Dai et al. proposed a designated-verifier proxy signature scheme based on discrete logarithm problems. However, in 2003, Wang pointed out that the original signer alone can forge valid proxy signatures to frame the proxy signer [16]. In 2004, Li et al. proposed a designated-verifier proxy signature scheme from bilinear pairings [17].

In 1984, Shamir proposed identity (ID)-based cryptography to simplify key management and remove the necessity of public key certificates [18]. In 2001, a practical ID-based encryption scheme was found by Boneh and Franklin, who took advantage of the properties of suitable bilinear parings (the Weil or Tate pairing) over supersingular elliptic curves [19].

Designated-verifier proxy signature scheme provides both the security properties of designated verifier signatures and those of proxy signatures. As far as the property of verifiability is concerned, the designated-verifier proxy signature scheme should meet the property of restrictive verifiability which means that only the designated verifier can verifier the validity of proxy signatures.

In 2003, Cha and Cheon [20] designed an ID-based signature scheme using GDH groups. Under the random oracle model, their scheme is proved to be secure against existential forgery on adaptively chosen messages and ID attacks supposing CDHP (Computational Diffie-Hellman Problem) is intractable.

Nowadays, most proxy signature schemes are based on the difficulty of DLP (Discrete Logarithm Problem) or ECDLP (Elliptical Curve Discrete Logarithm Problem). As though many proxy signature schemes based on DLP or ECDLP have been proposed, it makes us discouraged that some disadvantages can be found after a new or modified proxy signature scheme was designed after short time. How to solve the question? How to design secure and valid proxy signature scheme? How to prove them secure? Now, it is too difficult for us to prove one scheme secure, but if we can have some principles to conform to when designing some proxy signature schemes based on DLP or ECDLP, it will be helpful. It will be able to make the scheme designer to make few mistakes, that's to say, by these principles, they can judge their schemes meet basic secure conditions. If designers don't conform to these principles, it can easily be seen that their schemes are definitely insecure. It is all known by us that until now there are no these principles in the real life.

In the paper, by some hints from some attacks, especially forgery attacks, it seems to us that we have found three basic principles which should be conformed to when proxy signature schemes are proposed.

We will organize the rest of the paper. In section 2, we will detail the Principle 1. The Principle 2 will be stated in the section 3. In section 4, the Principle 3 will be detailed. One proxy scheme conforming to the three principles will be described in Section 5. Some remarks will be stated in Section 6. Finally, the conclusion will be given in Section 7.

II. Principle 1: The existent forms of public parameters in proxy signatures in the proxy signature verification congruence make a key role on the security property of unforgeability

The principle tells us that public parameters in proxy signatures exist in the proxy signature verification equation in the form of bases, exponents or hashes. The three existent forms-bases, exponent and hash will have a pretty large impact on the security of proxy signature schemes. As far as the principle is concerned, it is evident that it is correct, as attackers or forgers produce valid proxy signatures by finding properly modified public parameters satisfying the proxy signature verification equations. Though most of us know it, we don't regard it as one of vital principles when some proxy signature schemes are proposed or modified. Of course, we can regard it as one axiom. Thus, equipped with the idea, we will be able to design more valid and secure proxy signature schemes or other types of schemes such as signature schemes and proxy decryption schemes.

III. Principle 2: Any public parameter in the proxy signature can't lonely exist in the proxy signature verification congruence in the form of bases or exponents

A. Explanation

Principle 2 tells us that on one hand, if some public parameters in the proxy signature lonely exist in the proxy signature verification equation in the form of bases, the proxy signature scheme will probably suffer from the forgery attack from the original signer, the proxy signer or any third party; on the other hand, if some public parameters in the proxy signature lonely exist in the proxy signature verification congruence in the form of exponents, similarly, the proxy signature scheme will probably suffer from the forgery attack from the original signer, the proxy signer or any third party.

B.Example 1

In 2005, we proposed a threshold proxy signature scheme from bilinear pairing [23]. Recently, Cao and Lin [22] pointed out that our scheme was insecure and an adversary could forge a valid threshold proxy signature for any message on behalf of the proxy signers and the original signer. Our scheme will be briefly stated as followed.

We will define the following notations. Let G_0 and G_1 denote cyclic groups of prime order q, let P be a generator of G_0 and the bilinear pairing is given as $e: G_0 \times G_0 \to G_1$. Choose two cryptographic hash functions $H_1: \{0,1\}^* \times G_0^* \to Z_q^*$ and $H_2: \{0,1\}^* \to G_0^*$. The original signer has a secret key $sk = x_o$, randomly chosen from Z_q^* and a public key $pk = Y_o = x_o P$ which is certified by CA (Certificate Authority). Let $\{P_1, P_2, .., P_n\}$ be the proxy group of n proxy signers in such a way that a proxy signature can be created by any subset of t or more proxy signers. Each proxy signer has a secret key $sk_i = x_i$ randomly chosen from Z_q^* and a public key $pk_i = Y_i = x_i P$ which is certified by CA as well.

Our scheme consists of three stages: the proxy sharing, the proxy signature generation and the proxy signature verification.

Secret share generation: Let m_w be the warrant that is composed on the identities of the original signer and the proxy signers, the threshold value t, and the valid delegation time. In the stage, the original signer computes the partial proxy signing keys from his secret key and delivers them to each proxy signer.

Proxy signature generation: Let m be a message to be signed, any t or more proxy signers cooperate and sign the message m on behalf of the proxy group. Without loss of generality, let $D = \{P_1, P_2, ..., P_t\}$ be the actual proxy signers and *ASID* (Actual Signers' ID) be the collection of identities of all the users in D. The proxy signature of m generated by the scheme is 6-tuple $(m, U, m_w, \sigma, K, ASID)$.

Proxy signature verification: To make sure that the proxy signature $(m, U, m_w, \sigma, K, ASID)$ is indeed signed by the signers in D, the recipient can verify the validity of the proxy signature by checking if the following equation holds or not.

$$e(P,\sigma) = e(U + (H_1(m_w,U))Y_o + K + \sum_{i=1}^{n} Y_i + \sum_{i=1}^{t} Y_i, H_2(m)) \qquad (1)$$

If it holds, the recipient accepts the signature, otherwise rejects.

Cao and Lin [22] forged a valid proxy signature as follows. An adversary chooses a set of actual signers' identities $\{P_1, P_2, ..., P_t\}$, a proxy warrant m_w, a message m, two random numbers $r \in Z_q^*$ and $U \in G_0^*$. He or she computes $K = rP - (U + (H_1(m_w,U))Y_0 + \sum_{i=1}^{n} Y_i + \sum_{i=1}^{t} Y_i)$ and $\sigma = rH_2(m)$. Then, the 6-tuple $(m, U, m_w, \sigma, K, ASID)$ satisfies the verification equation eq. (1) where *ASID* is the collection of identities $\{P_1, P_2, ..., P_t\}$.

From the proxy signature $(m, U, m_w, \sigma, K, ASID)$ and its verification equation eq. (1), we know that the public parameters (U, m_w, σ, K) in $(m, U, m_w, \sigma, K, ASID)$ exists in equ. (1) in the form of bases, but not exponents. Thus, the scheme is attacked successfully.

From the above example, it can be known that not conforming to the principle 2, designed proxy signature schemes are not secure, at least for the forgery attack.

IV. PRINCIPLE 3: ANY PUBLIC PARAMETER IN THE PROXY SIGNATURE SHOULD EXIST IN THE PROXY SIGNATURE VERIFICATION EQUATION IN THE FORM OF NOT ONLY EXPONENTS AND BASES, BUT ALSO HASHES.

A. Explanation

From the principle 3, we know that if any public parameter in the proxy signature doesn't exist in the proxy signature verification congruence in the form of bases, exponents or hash, the proxy signature scheme maybe is insecure and it will possibly be attacked from the forgery by the original signer, the proxy signer or any third party. The three conditions-bases, exponents and hash, any of them should be met.

Of course, it is a little difficult for us to design this kind of proxy signature scheme which satisfies the principle 3. Since the proxy signature scheme was proposed, lots of proxy signatures schemes have been proposed. But, few of them is secure, that's to say, after they were proposed for not a long time, some researchers can find their weaknesses such as suffering from forgery attacks. Until now, it is the truth. To some extent, it makes some researchers somewhat depressed.

B. One example

From the above Example 1, we can know: the public parameter U in the proxy signature $(m, U, m_w, \sigma, K, ASID)$ exists in the verification equation eq. (1) in the form of bases and hashes, but not exponents; the public parameter m_w in $(m, U, m_w, \sigma, K, ASID)$ exists in eq. (1) only in the form of hashes, but not bases and exponents; the public parameters σ and K exists in eq. (1) only in the form of bases, but not exponents and hashes. In addition, the message m in $(m, U, m_w, \sigma, K, ASID)$ exists in eq. (1) only in the form of hashes, but not bases and exponents.

Similarly, due to not conforming to Principle 3, the above proxy signature scheme is insecure.

V. ONE CONTROVERSY

In the following, we will detail the Zhang and Kim's scheme [25].

Setup: Takes as input a security parameter k, and returns a master key s and system parameters $\Omega = (G_1, G_2, q, \hat{e}, P, P_{pub}, H_1, H_2)$, where $(G_1,+)$ and $(G_2,\cdot)$ are two cyclic groups of order q, $\hat{e}: G_1 \times G_1 \to G_2$ is an admissible bilinear map, $P_{pub} = sP, H_1 : \{0,1\}^* \to G_1^*$ and $H_2 : \{0,1\}^* \times G_2 \to Z_q$ are hash functions.

Extract: For a given identity ID_U ,computes $Q_U = H_1(ID_U) \in G_1^*, d_U = sQ_U$. PKG (Public Key Generator) returns d_U as the user's secret key. In the following description, denote $Q_x = H_1(ID_x)$.

Delegate: For secret key d_A and a warrant m_w, the original signer A computes $r_A = \hat{e}(P,P)^k$, where $k \in Z_q^*, c_A = H_2(m_w, r_A), U_A = c_A d_A + kP$, and outputs the delegation $W_{A \to B} = (m_w, r_A, U_A)$.

DVerify: Once the proxy signer B receives $W_{A \to B}$, he computes $c = H_2(m_w, r_A)$. If $r_A = \hat{e}(U_A, P)(\hat{e}(Q_A, P_{pub}))^{-c}$, he accepts the delegation.

PKgen: If B accepts the delegation $W_{A \to B}$, he computes the proxy signing key d_p as $d_p = H_2(m_w, r_A) \cdot d_B + U_A$.

PSign: Let d_p be B's proxy signing key, for a message m, B chooses $k \in Z_q^*$ at random and computes $r_P = \hat{e}(P,P)^k, c_P = H_2(m, r_P), U_P = c_P d_p + kP$, and lets $(m, \tau) = (m, r_P, U_P, m_w, r_A)$ be the proxy signature for m.

PVerify: For a proxy signature (m, r_P, U_P, m_w, r_A), a recipient first checks if the proxy signer and the message conform to m_w. Then he computes $c_P = H_2(m, r_P)$ and verifies whether $r_P = \hat{e}(U_P, P)(r_A \cdot \hat{e}(Q_A + Q_B, P_{pub})^{H_2(m_w, r_A)})^{-c_P}$. If both steps succeed, the proxy signature on behalf of A is valid.

ID: The proxy signer's identity ID_B can be revealed by m_w.

In Zhang's scheme, we can get the verification congruence as follows,

$$r_P = \hat{e}(U_P, P)(r_A \cdot \hat{e}(Q_A + Q_B, P_{pub})^{H_2(m_w, r_A)})^{-H_2(m, r_P)} \quad (2)$$

From the proxy signature (m, r_P, U_P, m_w, r_A) and its verification congruence eq. (2), we know that the public parameters (r_P, r_A) in (m, r_P, U_P, m_w, r_A) exist in the verification equation eq. (2) in the form of bases, exponents and hashes; the parameters (m, m_w) in (m, r_P, U_P, m_w, r_A) exist in eq. (2) in the form of exponents and hashes, but not bases; the public parameter U_P in (m, r_P, U_P, m_w, r_A) exists in eq. (2) only in the form of bases, but not exponents and hashes. If based on Principle 3, the Zhang's scheme is maybe insecure.

In 2005, Gu and Zhu [24] proved Zhang and Kim's scheme was secure by their security model. If our idea is right, it can prove that Gu and Zhu's secure model is not reasonable. On the contrary, if Gu and Zhu's proof of Zhang and Kim's scheme holds, our principle is wrong. We are eager to get the answer.

VI. REMARKS

We abstract the principles from some attacks. As to its correctness, until now, we can't prove it. In the following related research, we hope the answer will be found.

.VII. CONCLUSIONS

In the paper, some basic principles by which proxy signature designers can design more valid and secure proxy signature schemes, have been proposed by us. Since proxy signature schemes were proposed, there have been no such principles to be used by the scheme designers. With these principles, it's easier for proxy signature scheme designer or modifier to design or modify reasonable proxy signature schemes. It seems to us that these principles are of importance. Especially, when we will design some models for all kinds of proxy signature schemes based on DLP or ECDLP, they are more useful, though they are not proved correct.

ACKNOWLEDGMENT

We thank the reviewers for their valuable comments that helped us improve the quality and presentation of our work.

REFERENCES

[1] M. Mambo, K. Usuda, and E. Okamoto, "Proxy Signature for Delegating Signing Operation," *Proceedings of the 3.th ACM Conference on Computer and Communications Security*, New Dehli, India, ACM Press, New York, 1996, pp. 48-57.

[2] J.G. Li and Z.F. Cao, "Improvement of a Threshold Proxy Signature Scheme," *Journal of Computer Research and Development*, vol. 39, no. 11, pp. 515-518, 2002 (in Chinese).

[3] J.G. Li, Z.F. Cao, and Y.C. Zhang, "Improvement of M-U-O and K-P-W Proxy Signature Schemes," *Journal of Harbin Institute of Technology (New Series)*, vol. 9, no. 2, pp. 145-148, 2002.

[4] J.G. Li, Z.F. Cao, and Y.C. Zhang, "Nonrepudiable Proxy Multi-signature Scheme," *Journal of Computer Science and Technology*, vol. 18, no. 3, pp. 399-402, 2003.

[5] J.G. Li, Z.F. Cao, Y.C. Zhang, and J.Z. Li, "Cryptographic Analysis and Modification of Proxy Multi-signature Scheme," *High Technology Letters*, vol. 13, no. 4, pp. 1-5, 2003 (in Chinese).

[6] C.L Hsu, T.S. Wu and T.C. Wu, "New Nonrepudiable Threshold Proxy Signature Scheme with Known Signers," *The Journal of Systems and Software*, 58, pp.119~124, 2001.

[7] M.S. Hwang, I.C. Lin and J.L. Lu Eric, "A Secure Nonrepudiable Threshold Proxy Signature Scheme with Known Signers," *International Journal of Informatica*, vol. 11, no. 2, pp.1-8, 2000.

[8] S.J Hwang and C.C. Chen, "Cryptanalysis of Nonrepudiable Threshold Proxy Signature Scheme with Known Signers," *INFORMATICA*, vol. 14, no. 2, pp.205-212, 2003.

[9] H.M. Sun, "An Efficient Nonrepudiable Threshold Proxy Signature Scheme with Known Signers," *Computer Communications*, vol. 22, no. 8, pp. 717-722, 1999.

[10] H.M. Sun, N.Y. Lee, and T. Hwang, "Threshold Proxy Signature," *IEEE Proceedings-computers & Digital Techniques,* 146(5), pp. 259-263, 1999.

[11] K. Zhang, "Threshold Proxy Signature Schemes," *Information Security*

Workshop, Japan, 1997, pp. 191-197.

[12] C.L. Hsu, T.S. Wu, and T.C. Wu, "Improvement of Threshold Proxy Signature Scheme," *Applied Mathematics and Computation*, 136, pp. 315-321, 2003.

[13] C.S. Tsai, S.F. Tzeng, and Hwang, M.S., "Improved Nonrepudiable Threshold Proxy Signature Scheme with Known Signers," *INFORMATICA*, vol. 14, no. 3, pp. 393-402, 2003.

[14] S.J. Hwang and C.H. Shi, "A Simple Multi-Proxy Signature Scheme," *Proceeding of the Tenth National Conference on Information Security*, Taiwan, 2000.

[15] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their application," *Lect. Notes Comput. Sci.*, 1070, pp. 143-154, 1996.

[16] G. Wang, "Designated-verifier proxy signatures for e-commerce," *Proc. IEEE 2004 Int. Conf. on Multimedia and Expo (ICME 2004)*, 3, pp. 1731-1734 , 2004.

[17] X. Li, K. Chen, and S. Li, "Designated-verifier proxy signatures for e-commerce from bilinear pairings," *Proc. of Int. Conf. on Computer Communication*, pp. 1249-1252, 2004.

[18] A. Shamir, "Identity-based cryptosystems and signature schemes," *Lect. Notes Comput. Sci.*, 196, pp. 47-53, 1984.

[19] D. Boneh, and M. Franklin, "Identity-based encryption from the Weil pairing," *Lect. Notes Comput. Sci.*, 2139, pp. 213-229, 2001.

[20] J.C. Cha and J.H. Cheon, "An identity-based signature from gap diffie-hellman groups," *Lect. Notes Computer Sci.*, 2567, pp. 18-30, 2003.

[21] J. Dai, X. Yang, and J. Dong, "Designated-receiverproxy signature scheme for electronic commerce," *Proc. of IEEE International Confernece on System,Man and Cybernetics*,. Oct. 5-8, vol. 1, pp. 384-389, 2003.

[22] T.J. Cao and D.D. Lin, "Security analysis of some threshold signature schemes and multi-signature scheme," *LNCS 3822*, pp. 233-241, 2005.

[23] H. Qian, Z. Cao, and Q., Xue, "Efficient Pairing-Based Threshold Proxy Signature Scheme with Known Signers," *INFORMATICA*, vol.16, no. 2, pp. 261-274, 2005.

[24] C. Gu and Y. Zhu, "Provable security of ID-based proxy signature schemes," *LNCS 3619*, pp. 1277-1286, 2005.

[25] F. Zhang and K. Kim, "Efficient ID-based blind signature and proxy signature from bilinear pairings," *LNCS 2727*, pp. 312-323, 2003.