2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)
IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V53.No.1.49

Cryptanalysis of A Self-Certified Threshold Proxy Signature Scheme Ased on Elliptic Curve Discrete Logarithm Problem

Qi Xie[+]

School of Information Science and Engineering
Hangzhou Normal University
Hangzhou 310036, P R China

Abstract. Digital signatures based on self-certified public key systems are more efficient because the authentication of the users' public keys can be implicitly accomplished with the signature verification. In 2009, Xue et al. proposed first self-certified threshold proxy signature scheme based on the elliptic curve discrete logarithm problem (ECDLP). In this paper, we show that Xue et al.'s scheme can not resist the proxy warrant revision attack. That is, their scheme fails to meet the security properties of strong undeniability and strong unforgeability. Further, we propose a possible improvement to overcome their weakness.

Keywords: Proxy signature; threshold proxy signature; self-certified public key; proxy warrant revision attack

1. Introduction

The concept of proxy signature was first proposed by Mambo et al. in 1996 [1][2]. A proxy signature scheme allows a signer to delegate the signing capability to a designated person, called the proxy signer, the proxy signer can generate proxy signature of a message on behalf of the original signer. A secure proxy signature scheme should meet the following properties [3][4]:

(1) Verifiability: From the proxy signature, the verifier can be convinced of the original signer's agreement on the signed message.

(2) Strong identifiability: Anyone can determine the identity of the corresponding proxy signers from the proxy signature.

(3) Strong non-deniability: Once a proxy signer creates a valid proxy signature on behalf of an original signer, both the original signer and the proxy signer cannot repudiate the proxy signature.

(4) Strong non-forgeability: A designated proxy signer can create a valid proxy signature for the original signer. But the original signer and other third parties who are not designated as a proxy signer cannot create a valid proxy signature.

(5) Prevention of misuse: The proxy signer cannot use the proxy key to sign messages that have not been authorized by the original signer.

In 1997, Kim et al.[5] and Zhang [6] proposed a new concept of proxy signature, called (*t, n*) threshold proxy signature, which the proxy signature key is shared among a group of *n* proxy signers delegated by the original signer. Any *t* or more proxy signers can cooperatively sign messages on behalf of the original signer, but *t*-1 or fewer proxy signers cannot. Later on, many threshold proxy signature schemes and their improvement schemes were proposed [7]-[14].

[+] Corresponding author.
E-mail address: qixie68@yahoo.com.cn.

In 1991, Girault proposed the self-certified public keys cryptosystem [15], which each user's public key is signed by the certification authority (CA) using the CA's private key, while the corresponding private key is only known by the user. The authentication of the public key can be implicitly accomplished with the signature verification. Therefore, self-certified public keys contribute to reducing the amount of storage and computation in public key schemes. In 2009, Xue et al. proposed first self-certified threshold proxy signature scheme based on the elliptic curve discrete logarithms [16], they claimed that their scheme can resist all existing attacks.

However, in this paper, we will show that Xue et al.'s scheme can not resist the proxy warrant revision attack. That is, their scheme fails to meet the security properties of strong undeniability and strong unforgeability. To overcome their weakness, we propose a possible improvement to against our attack.

The remainder of this article is organized as follows. In Section II, we briefly review Xue et al.'s scheme. Then, we present an attack on Xue et al.'s scheme in Section III. After that, we present the possible improvement and security analysis in Section IV. Finally, the paper is concluded in Section V.

2. Brief review of Xue et al.'s Scheme

In this section, we review Xue et al.'s threshold proxy signature scheme based on the elliptic curve discrete logarithms, which consists of four stages: system initialization, registration, proxy share generation, threshold proxy signature generation and verification.

2.1 System initialization stage

A trusted CA chooses a large prime q, an elliptic curve E over $GF(q)$, let G be a base point on E with order n, $h()$ denotes a one-way collision resistant cryptographic hash function. CA randomly chooses the private key $\delta \in Z_q^*$, and computes his public key $\beta = \delta G$. The parameters q, $h()$, G, n and β are made public, while the CA's private key δ is kept secret.

2.2 Registration stage

Each user U_i with the identifier ID_i performs the interactive steps with the CA:

Step 1: U_i chooses an integer $t_i \in Z_q^*$, computes $v_i = h(t_i \| ID_i)G$ and sends (v_i, ID_i) to the CA.

Step 2: After receiving (v_i, ID_i), the CA chooses $a_i \in Z_q^*$, computes $y_i = v_i + a_i G$, $w_i = a_i + \delta h(y_{ix} \| ID_i) \bmod n$, where y_{ix} denotes the *x*-coordinate of point y_i on *E*, and returns (y_i, w_i) to U_i.

Step 3: U_i first computes $x_i = w_i + h(t_i \| ID_i) \bmod n$ and verifies the equation $\beta h(y_{ix} \| ID_i) + y_i = x_i G$. If it holds, user U_i accepts (x_i, y_i) as his private and public key pair.

2.3 Proxy share generation stage

Let $G_p = \{U_{p1}, U_{p2}, \cdots, U_{pl}\}$ be the set of l ($> t$, where t is a threshold value) proxy signers and U_o the original signer who wants to delegate his signing power to the proxy group G_p. U_o and the proxy group G_p perform the following steps:

Step 1: U_o chooses an integer $k_o \in Z_n$ and computes $K_o = k_o G$, $\sigma_o = k_o K_{ox} + x_o h(m_w \| K_{ox}) \bmod n$, where K_{ox} is the *x*-coordinate of point K_o on E, m_w is the warrant consisting of the original and the proxy signers' identifiers, the delegation duration, threshold value t, and so on.

Step 2: U_o sends (σ_o, m_w, K_o) to $U_{pi} \in G_p$ via a secure channel.

Step 3: $U_{pi} \in G_p$ verifies the validity of (σ_o, m_w, K_o) by checking the equation

$$\sigma_o G = K_{ox} K_o + h(m_w \| K_{ox})(\beta h(y_{ox} \| ID_o) + y_o).$$

If it holds, $U_{pi} \in G_p$ accepts the proxy share.

2.4 Threshold proxy signature generation and verification stage

Without loss of generality, let $D = \{U_{p1}, U_{p2}, \cdots, U_{pt}\}$ be t proxy signers to represent the proxy group G_p to sign a message m on behalf of the original signer U_o:

Step 1: Each $U_{pi} \in D$ chooses an integer $k_i \in Z_q^*$, computes $K_i = k_i G$, and sends K_i to other proxy signers in D and the designated clerk C, who is responsible for collecting and verifying the individual proxy signatures generated by the proxy signers and constructing the final threshold proxy signature.

Step 2: Each $U_{pi} \in D$ computes $K = \sum_{j=1}^{t} K_i$,

$$s_i = k_i K_x + (\sigma_o t^{-1} + x_{pi}) h(m \| ASID) (\mathrm{mod}\, n),$$

where $ASID$ denotes the identities of the actual proxy signers. Then, sends s_i to C.

Step 3: C verifies the validity of each $U_{pi} \in D$'s individual proxy signature by checking

$$\begin{aligned} s_i G = {} & K_i K_x \\ & + ((K_{ox} K_o + h(m_w \| K_{ox})(\beta h(y_{ox} \| ID_o) + y_o)) t^{-1} \\ & + (\beta h(y_{pix} \| ID_{pi}) + y_{pi})) h(m \| ASID), \end{aligned}$$

where $i = 1, 2, \cdots, t$. If all equations hold, C computes $S = \sum_{i=1}^{t} s_i \bmod n$, then the threshold proxy signature of message m is $(m_w, K_o, m, K, S, ASID)$.

When threshold proxy signature $(m_w, K_o, m, K, S, ASID)$ of message m is available, any verifier can verify the validity of the threshold proxy signature by checking

$$\begin{aligned} SG = {} & K_x K \\ & + ((K_{ox} K_o + h(m_w \| K_{ox})(\beta h(y_{ox} \| ID_o) + y_o)) \\ & + \sum_{i=1}^{t} (\beta h(y_{pix} \| ID_{pi}) + y_{pi})) h(m \| ASID). \end{aligned}$$

If it holds, verifier accepts the proxy signature. Otherwise, he rejects it.

3. Cryptanalysis of Xue et al.'s Scheme

Xue et al. claimed that their threshold-proxy signature scheme can resist all existing attacks. However, we will show that their scheme cannot resist the warrant revision attack mounted by the proxy signers. The details are as follows.

When the proxy group G_p obtains the proxy share (σ_o, m_w, K_o) from the original signer U_o, they can change the warrant m_w to $\overline{m_w}$, which can change the kind of messages to be delegated, the valid delegation time, the threshold value, the proxy signers, etc. For example, the proxy group changes the threshold value to $t+1$, changes the group G_p to $G_p \cup \{U_{p0}\}$, where U_{p0} is a new member in proxy group.

Since the proxy group adds a new member U_{p0}, therefore, U_{p0} needs to register to the CA as follows:

Step 1: U_{p0} chooses an integer $\overline{k_o} \in Z_n$, computes

$$\overline{K_o} = \overline{k_o} G,$$

$$d = h(\overline{m_w} \| \overline{K_{ox}})(h(m_w \| K_{ox}))^{-1} \bmod n,$$

$$v_{p0} = h(ID_{p0}) G + d K_o K_{ox}.$$

Then he sends (v_{p0}, ID_{p0}) to the CA.

Step 2: After receiving (v_{p0}, ID_{p0}), the CA chooses $a_{p0} \in Z_q^*$, computes

$$y_{p0} = v_{p0} + a_{p0} G,$$

$$w_{p0} = a_{p0} + \delta h(y_{p0x} \| ID_{p0}) \bmod n.$$

Then returns (y_{p0}, w_{p0}) to U_{p0}.

Step 3: After receiving (y_{p0}, w_{p0}), U_{p0} computes

$$x_{p0} = w_{p0} + h(ID_{p0}) \bmod n$$

and verifies the following equation

$$\beta h(y_{p0x} \| ID_{p0}) + y_{p0} - d K_o K_{ox} = x_{p0} G.$$

The correctness of the verification equation is proved as follows:

$$
\begin{aligned}
x_{p0}G &= w_{p0}G + h(ID_{p0})G \\
&= a_{p0}G + \delta h(y_{p0x} \| ID_{p0})G + h(ID_{p0})G \\
&= y_{p0} - v_{p0} + \delta h(y_{p0x} \| ID_{p0})G + h(ID_{p0})G \\
&= y_{p0} - (h(ID_{p0})G + dK_oK_{ox}) \\
&\quad + h(y_{p0x} \| ID_{p0})\beta + h(ID_{p0})G \\
&= \beta h(y_{p0x} \| ID_{p0}) + y_{p0} - dK_oK_{ox}
\end{aligned}
$$

Then, let $D = \{U_{p1}, U_{p2}, \cdots, U_{pt}\}$ and U_{p0} be $t+1$ proxy signers to sign a message m on behalf of the original signer U_o:

Step 1: Each $U_{pi} \in D \cup \{U_{p0}\}$ chooses an integer $k_i \in Z_q^*$, computes $K_i = k_iG$, and sends K_i to other proxy signers.

Step 2: Each $U_{pi} \in D \cup \{U_{p0}\}$ computes $K = \sum_{i=0}^{t} K_i$,

$$s_i = k_iK_x + (d\sigma_o(t+1)^{-1} + x_{pi})h(m \| ASID) (\mathrm{mod}\, n),$$

where $ASID$ denotes the identities of the actual proxy signers in $D \cup \{U_{p0}\}$.

Then, U_{pi} sends s_i to other signers.

Step 3: Eeach $U_{pi} \in D \cup \{U_{p0}\}$ computes

$$S = \sum_{i=0}^{t} s_i \bmod n,$$

$$\overline{S} = \overline{k_o}\,\overline{K_{ox}} h(m \| ASID) + S (\mathrm{mod}\, n).$$

Therefore, $(\overline{m_w}, \overline{K_o}, m, K, \overline{S}, ASID)$ is a valid threshold proxy signature of message m. The verification equation is

$$
\begin{aligned}
\overline{S}G &= K_xK + (\overline{K_o}\,\overline{K_{ox}} + h(\overline{m_w} \| \overline{K_{ox}})(\beta h(y_{ox} \| ID_o) + y_o) \\
&\quad + \sum_{i=0}^{t} (\beta h(y_{pix} \| ID_{pi}) + y_{pi}))h(m \| ASID).
\end{aligned}
$$

The correctness of the verification equation is proved as follows:

Because $dh(m_w \| K_{ox}) = h(\overline{m_w} \| \overline{K_{ox}}) \bmod n$, so

$$
\begin{aligned}
SG &= K_xK + (d(K_{ox}K_o + h(m_w \| K_{ox})(\beta h(y_{ox} \| ID_o) + y_o)) \\
&\quad + \beta h(y_{p0x} \| ID_{p0}) + y_{p0} - dK_oK_{ox} \\
&\quad + \sum_{i=1}^{t} (\beta h(y_{pix} \| ID_{pi}) + y_{pi}))h(m \| ASID) \\
&= K_xK + (h(\overline{m_w} \| \overline{K_{ox}})(\beta h(y_{ox} \| ID_o) + y_o) \\
&\quad + \sum_{i=0}^{t} (\beta h(y_{pix} \| ID_{pi}) + y_{pi}))h(m \| ASID),
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\overline{S}G &= \overline{k_o}\,\overline{K_{ox}} h(m \| ASID)G + SG \\
&= \overline{K_o}\,\overline{K_{ox}} h(m \| ASID) + K_xK \\
&\quad + (h(\overline{m_w} \| \overline{K_{ox}})(\beta h(y_{ox} \| ID_o) + y_o) \\
&\quad + \sum_{i=0}^{t} (\beta h(y_{pix} \| ID_{pi}) + y_{pi}))h(m \| ASID) \\
&= K_xK + (\overline{K_o}\,\overline{K_{ox}} + h(\overline{m_w} \| \overline{K_{ox}})(\beta h(y_{ox} \| ID_o) + y_o) \\
&\quad + \sum_{i=0}^{t} (\beta h(y_{pix} \| ID_{pi}) + y_{pi}))h(m \| ASID).
\end{aligned}
$$

Therefore, the attack is success.

4. Possible Improvement

The reason of Xue et al.'s scheme suffers from proxy warrant revision attack is that the CA does not check if the user knows the elliptic curve discrete logarithm of v_i submitted by the user in user registration

stage. To overcome their weakness, we only revise the user registration stage, other stages are the same as that of Xue et al.'s scheme.

In user registration stage, each user U_i with the identifier ID_i performs the interactive steps with the CA as follows:

Step 1: U_i chooses an integer $t_i \in Z_q^*$, computes

$$v_i = h(t_i \| ID_i)G,$$
$$v_i' = h(t_i \| ID_i)\beta,$$

then he sends (ID_i, v_i, v_i') to the CA.

Step 2: After receiving (ID_i, v_i, v_i'), the CA first checks if $v_i' = \delta v_i$. If it does not hold, he rejects it. Otherwise, he then chooses $a_i \in Z_q^*$, computes

$$y_i = v_i + a_i G,$$
$$w_i = a_i + \delta h(y_{ix} \| ID_i) \bmod n,$$

where y_{ix} denotes the *x*-coordinate of point y_i on *E*, and returns (y_i, w_i) to U_i.

Step 3: U_i first computes $x_i = w_i + h(t_i \| ID_i) \bmod n$ and verifies the equation

$$\beta h(y_{ix} \| ID_i) + y_i = x_i G.$$

If it holds, user U_i accepts (x_i, y_i) as his private and public key pair.

Next, we analysis the improvement scheme can resist our attack.

If the proxy group adds a new member U_{p0}, and U_{p0}'s private and public key pair (x_{p0}, y_{p0}) should satisfy

$$\beta h(y_{p0x} \| ID_{p0}) + y_{p0} - dK_o K_{ox} = x_{p0} G,$$

where $\overline{m_w}$ is a revised proxy warrant by proxy group and $d = h(\overline{m_w} \| \overline{K_{ox}})(h(m_w \| K_{ox}))^{-1} \bmod n$ for randomly chosen $\overline{k_o} \in Z_n$, $\overline{K_o} = \overline{k_o} G$. Therefore, U_{p0} should register to the CA and submit $v_{p0} = h(ID_{p0})G + dK_o K_{ox}$ to CA. However, U_{p0} cannot submit a valid v_{p0}' such as $v_{p0}' = \delta v_{p0}$ because he does not know k_o chosen by the original signer U_o in proxy share generation stage, and cannot compute k_o from $K_o = k_o G$ because it is an elliptic curve discrete logarithm problem. Therefore, CA will reject U_{p0} to register the system since (v_{p0}, v_{p0}') submitted by U_{p0} does not satisfy $v_{p0}' = \delta v_{p0}$.

That is, the attack is fail, the improvement scheme is secure.

5. Conclusions

In this paper, we have shown that Xue et al.'s scheme can not resist the proxy warrant revision attack. The proxy group can revise the proxy warrant and generate the valid proxy signature for any message. To overcome their weakness, we propose a possible improvement to against our attack.

6. Acknowledgment

This work is supported by National Natural Science Foundation of China (No.61070153) and Natural Science Foundation of Zhejiang province (No.Y1080831).

7. References

[1] M. Mambo, K. Usuda, E. Okamoto, "Proxy Signature for Delegating Signing Operation," Proc. 3rd ACM Conf.on Computer and Communications Security, 1996, pp. 48-57, doi:10.1145/ 238168.238185.

[2] M. Mambo, K. Usuda, E. Okamoto, "Proxy Signatures: Delegation of the Power to Sign Messages," IEICE Trans. Fundam. Electron. Commun. Comput. Sci., E79-A(9), 1996, pp.1338-1354.

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

[4] B. Lee, H. Kim, K. Kim, "Strong Proxy Signature and Its Application," Australasian Conference on Information Security and Privacy (ACISP'01), LNCS 2119, Springer-Verlag, 2001, pp. 603-608.

[5] S. Kim, S. Park, D Won, "Proxy signatures, revisited," Proc of ICICS'97, Springer-Verlag, 1997, pp.223-232.

[6] K. Zhang, “Threshold proxy signature schemes,” In: Information Security Workshop, 1997, pp.191-197.

[7] L. H. Li, S. F. Tzeng, M. S. Hwang, “Generalization of Proxy Signature- Based on Discrete Logarithms,” Computers & security, vol.22, 2003, pp. 245-255, doi: 10.1016/S0167-4048(03)00312-2.

[8] S. F. Tzeng, C. Y. Yang, M. S. Hwang, “A Nonrepudiable Threshold Multi-proxy Multisignature Scheme with Shared Verification,” Future Generation Computer Systems, vol.20, 2004, pp.887-893, doi: 10.1016/j.future.2004.01.002.

[9] Q. Xie, “Improvement of Tzeng et al.'s Nonrepudiable Threshold Proxy Signature Scheme with Known Signers,” Applied Mathematics and Computation, vol.158, 2005, pp.776-782, doi:10.1016/ j.amc.2004.09.037.

[10] C.L.Hsu, T.S.Wu, “Efficient nonrepudiable threshold proxy signature scheme with known signers against the collusion attack,” Applied Mathematics and Computation, 2005, 168(1): 305-319, doi: 10.1016/j.amc.2004.08.040.

[11] C.Y.Yang, S.F.Tzeng, M.S.Hwang, “On the efficiency of nonrepudiable threshold proxy signature scheme with known signers,” The Journal of Systems and Software, 73:507-514(2004), doi: 10.1016/j.jss.2003.09.022.

[12] Q. Xie, X. Yu, “Cryptanalysis of Two Nonrepudiable Threshold Proxy Signature Schemes,” International Journal of Network Security, vol.3, 2006, pp. 21-25

[13] H. Bao, Z. Cao, S. Wang, “Improvement on Tzeng et al.’s nonrepudiable threshold multi-proxy multi-signature scheme with shared verification,” Applied Mathematics and Computation, 169(2005)2, 1419-1430, doi:10.1016/j.amc.2004.10.075.

[14] Q. Xie, J. Wang, X. Yu, “Improvement of Nonrepudiable Threshold Multi-proxy Threshold Multi-signature Scheme with Shared Verification,” Journal of Electronics (china), vol.24, 2007, pp.806-811, doi: 10.1007/s11767-006-0047-z.

[15] M. Girault, “Self-certified public keys,” LNCS 547, Springer-Verlag, 1991, pp.490-497, doi: 10.1007/3-540-46416-6_42.

[16] Q. Xue, F. Li, Y. Zhou, J. Zhang, Z. Cao, H. Qian, “An ECDLP-based threshold proxy signature scheme using self-certified public key system,” First International ICST Conference, MobiSec 2009, Turin, Italy, LNICST 17, Springer-Verlag, 2009, pp. 58-70, doi: 10.1007/978-3-642-04434-2_6.