ECDLP Based Proxy Multi-signature Scheme

Ramanuj Chouksey[1], R. Sivashankari[1], and Piyush Singhai[2]

[1] Name, VIT University, Vellore
ramanuj@vit.ac.in, sivashankari.r@vit.ac.in
[2] Name, Knowlarity Communications Private Limited
singhai.piyush@gmail.com

Abstract. A Proxy signature scheme enables a proxy signer to sign a message on behalf of the original signer. In this paper, we propose efficient and secure Proxy multi-signature scheme based on elliptic curve cryptosystem. Our scheme satisfy all the proxy requirements and require only elliptic curve multiplication and elliptic curve addition which needs less computation overhead compared to modular exponentiation also our scheme is withstand against original signer forgery and public key substitution attack.

1 Introduction

Signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature. A digital signature allows an entity, called the designator or original signer, to generate a signature on any message using his private key such that the receiver can verify the validity of the signature and authentication of the signer using signer's certified public key. But in the case of absence of original signer, digital signature schemes are not applicable. Proxy signature schemes were proposed to address the problem in traditional signature schemes.

Proxy signature allows the original signer to delegate his signing power to a person called proxy signer who can replace the original signer, in case of say, temporal absence, lack of time or computational power, etc. Then the verifier can check the validity of signature, identity of the proxy signer and the original signer's agreement using original signers, proxy signer's certified public keys. Proxy signatures have been used in numerous practical applications, like e-commerce, electronic agreement, mobile agents, mobile communications, distributed computing, electronic voting, etc.

The concept of proxy signature was first proposed in 1996 by Mambo et. al [8]. Based on the delegations types, they classified proxy signature into *full delegation, partial delegation* and, *delegation by warrant* schemes. In a full delegation, As name implies, the complete delegation (the private key of original signer) is transferred to proxy signer. So a signature by proxy signer is indistinguishable from that created by an original signer. In a Partial delegation, A new secret key (proxy signature key) is computed by the the original signer using his private key. Using this secret key, the proxy signer can generate a proxy signature on any message. For security requirements, it is computationally infeasible for the proxy signer to derive the original signer's private key from the proxy signature key. However, in such schemes the range of messages a proxy signer can sign is not limited. This weakness eliminated by warrant schemes so using delegation by

N. Meghanathan et al. (Eds.): Advances in Computing & Inform. Technology, AISC 176, pp. 71–79.
springerlink.com

warrant specifies the types of messages, delegation period and identity of signer (proxy or original), etc. In paper [1] they mention two kinds of proxy signature schemes depending on whether the original signer can generate the same proxy signature as the proxy signers do. The one is proxy-unprotected and other is proxy-protected in which anyone else, including the original signer, cannot generate the same proxy signatures. This difference is important in practical applications and this thing also avoid potential disputes between the original signer and proxy signer.

After the Mambo's scheme [8] that is one to one i.e. one original signer and one proxy signer there are so many proxy signature scheme have been proposed [9]. To meet the requirements of various rapidly growing applications, different types of proxy signature schemes have been evolved. Those are threshold proxy signatures, nominative proxy signatures, one-time proxy signatures, multi-proxy signatures, proxy multi-signatures, proxy blind signatures, etc. Unlike one to one scheme the proxy multi-signature scheme proposed by Yi et al's [4] allows two or more original signers to delegate his signing power to single proxy signer to sign the messages for all original signers. Yi et al's [4] proposed two types of proxy multi-signature scheme one is Mambo like proxy multi-signature scheme and another is Kim like proxy multi-signature scheme. Sun's [5] showed that both scheme are insecure. The Mambo-like proxy multi-signature scheme in [4] suffers from the public key substitution attack easily. The Kim-like proxy multi-signature scheme in [4] suffers from a kind of direct forgery. After that Sun [5] proposed two proxy multi-signature schemes one is Proxy protected proxy multi-signature scheme (Mambo like) and another is Proxy unprotected proxy multi-signature scheme (Kim like). Between these two schemes, one scheme provides the protection for proxy signers while another scheme does not. In these schemes, the secure channel is not necessary. However, Sun's [5] and Yi et al's [4] schemes have the common disadvantage that is size of the proxy signature depend on the number of original signers and both schemes involve exponential operation to verify proxy signature. Accordingly, an improvement is proposed to change the exponential operations into elliptic curve multiplicative ones. The elliptic curve cryptosystem can achieve a level of security equal to that of RSA or DSA but has a lower computational overhead and a smaller key size than both of these. Therefore, it is used in Sun's schemes [5] to improve their efficiency.

In light of the high computational overhead of Suns schemes [5] and Yi et. al's [4] scheme a new Efficient Multi signature scheme has been proposed by Tzer-shyong chen and Gwo-shiuan et. al's [10]. After that Tzer-shyong chen and Kuo-Hsuan et. al's [11] proposed A tracable proxy multi signature scheme. These scheme are based on ECC that can perform more efficiently then those based on DLP. These schemes are based on Elliptic curve discrete logarithm problem(ECDLP). These schemes makes size of the proxy signature independent of the number of original signers, so the computation overhead required for the verification is reduced. For improving Sun's [5] and Yi et al's [4] schemes so many DLP based schemes also proposed like Chien-Lung Hsu et. al's [2] and Guilin Wang et. al's [7]. But these schemes involve exponential operation to verify proxy signature. But Hsu et. al's [2] scheme and Tzer-shyong chen et. al's [11] schemes are insecure against malicious original signer. For the Hsu et. al's scheme in [2], which is suffer from cheat attack that is shown by Feng Cao and Zhenfu Cao [3] that means

a malicious original signer can cheat the Certificate Authority into extracting a proxy signing key of a proxy signer. Furthermore, this attack can be used by proxy signer to cheat CA into extracting proxy signing key without the knowledge of the original signer. Yumin Yuan [12] also give improvement of this scheme. In addition to this Tzer-shyong chen and Gwo-shiuan et. al's. [10] and Tzer-shyong chen and Kuo-Hsuan et. al's [11] are also vulnerable to one original signer and all original signer proxy signing forgery attack respectively that is shown by Je Hong Park and Bo Gyeong Kang and Sangwoo Park in [6]. Original signer forgery attack means malicious original signer can generate valid proxy signature which looks like that it is generated by proxy signer. For generating valid proxy signature original signer forges proxy signing key and uses it to make a signature forgery.

In this paper we propose an efficient and secure proxy multi-signature scheme and analyze the security of the scheme. We show that our scheme are secure against the original signersforgery and public key substitution attack.

The rest of the paper is organized as follows. Sect. 2 we show the proxy requirements and security assumptions. Sect. 3 introduce the Tzer-shyong chen and Kuo-Hsuan et. al's proxy multi signature scheme and security analysis and possible attack in the scheme. In section Sect. 4 we show our proxy multi signature propose scheme and analyze its security and efficiency. Finally, Sect. 5 discusses some application.

2 Preliminaries

In 1996, Mambo, Usuda and Okamoto [8] first addressed the basic properties that a proxy signature scheme for partial delegation should satisfy, and defined them as follows:

- Verifiability
- Identifiably
- Unforgeability
- Undeniability
- Prevention of misuse

2.1 Security Assumption

The proxy multi signature scheme in this paper is based on some security assumption.

- **Elliptic curve Discrete logarithm Problem (ECDLP):** Consider the equation $Q = kP$ where $Q, P, E_p(a;b)$ and $k < p$. It is relatively easy to calculate Q given k and P, but it is relatively hard to determine k given Q and P. This is called the discrete logarithm problem for elliptic curves.
- **EC Diffie-Hellman Key Exchange:**
  - A's Private key and public key n_A and $P_A = n_A \times G$, This is point in $E_q(a;b)$.
  - B similarly selects a private key n_B and computes a public key P_B.
  - A generates the secret key $K = n_A \times P_B$ and B generates the secret key $K = n_B \times P_A$.
  - Both having same secret key, $n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A$

3 Tzer-Shyong Chen and Kuo-Hsuan et al.'s Proxy Multi-signature Scheme

This scheme is improved version of Tzer-shiuan et. al scheme [10]. In Tzer-shiuan et. al scheme each original signer sends information in individual manner but in this scheme [11] each original signer calculate some group commitment value that is common for all and then generate information using it and then send to proxy signer. This scheme has four phases.

1. proxy public key generation phase: All original and proxy signer generate their public and private key in this phase.
2. proxy signing key generation phase: For delegating signing power to proxy signer each original signer A_i performs following steps
   - A_i securely selects a random number $k_i \in \{1,2,\ldots,t-1\} \setminus d_i$ and computes $R_i = k_i B = (x_{R_i}, y_{R_i})$.
   - Broadcast R_i to the other original signer.
   - upon receiving R_j computes $R = \sum_{i=1}^{n} R_i = (x_R, y_R)$.
   - Then computes $s_i = d_i h(m_w, x_{Q_i}, x_{Q_P}, x_R)$ - $k_i \bmod t$.
   - Sends sub delegation parameter (m_w, s_i) to proxy signer.
3. Sub delegation parameter verification and secret key generation:
   - using Sub delegation parameter (m_w, s_i) proxy signer first calculate $R_i'=(x_{R_i'}, y_{R_i'})$ as follows

$$R_i' = Q_i \times h(m_w, x_{Q_i}, x_{Q_P}, x_R) - s_i \times B.$$

and checks

$$x_{R_i'} = x_{R_i} \bmod t.$$

   - if all parameter is valid then proxy signer compute proxy signing key as follows

$$d = d_P + \sum_{i=1}^{n} s_i \bmod t.$$

4. Proxy signature generation and verification: Proxy multi-signature is attached to the message m in the form of $(m, m_w, R, Sig_d(m))$, where $Sig_d(m)$ means the signature generated by designated scheme using the proxy signature key d. For verifying signature, verifier computes proxy public key Q corresponding to the proxy signing key d as

$$Q = Q_P + \sum_{i=1}^{n} h(m_w, x_{Q_i}, x_{Q_P}, x_R) Q_i - R.$$

   with this proxy public key the verifier confirms the validity of signature by validating the verification equation.

Now we discuss the security of this scheme. This scheme is suffer from one attack that is original signer forgery attack, that is described by Je Hong park, Bo Gyeong Kang et. al [6] Original signers forgery attack in which conspiracy of all original signers to

generate valid proxy multi-signature without the agreement of proxy signer. They show how attack is possible as follows.

The original signer A_i select random number k_i and then compute

$$R_i = k_i B \text{ for } 1 \leq i \leq n.$$

Furthermore A_i adds Q_P to R_1 and then computes

$$R = \sum_{i=1}^{n} R_i = Q_P + \left(\sum_{i=1}^{n} k_i \right) B$$

The forged proxy signing key that is generated by all original signers is as follows

$$d = \sum_{i=1}^{n} d_i h(m_w, x_{Q_i}, x_{Q_P}, x_R) - k_i$$

from the verification equation, proxy public key Q computed by verifier as follows

$$Q = Q_P + \sum_{i=1}^{n} Q_i h(m_w, x_{Q_i}, x_{Q_P}, x_R) - R$$

$$= Q_P + \left(\sum_{i=1}^{n} d_i h(m_w, x_{Q_i}, x_{Q_P}, x_R) \right) B - \left(\sum_{i=1}^{n} k_i \right) B - Q_P = dB$$

This means verifier will convinced that any proxy multi signature signed by using the forged signing key d are generated by agreement of all original signer and P. so this scheme is not proxy protected.

4 Our Proposed Proxy Multi-signature Scheme

A new ECDLP based proxy multi signature scheme is presented in this paper. The proposed scheme is independent to the number of original signer and we are also using the merits of ECC so the overhead of computation and communication cost due to modular exponential operation also reduced. Our scheme also secure against original signer forgery attack and public key substitution attack without using any encryption and decryption. This scheme involves three parties: original signers A_i $1 \leq i \leq n$, proxy signer p and verifier v and scheme is divided into four phases those are as follows:

1. System initialization phase: The following parameters over the elliptic curve domain must be known
   F_p**:** A field size p (a large prime number (512 bits)).
   E**:** An elliptic curve of the form $(y^2 = x^3 + ax + b \bmod p)$ over F_p, where $a, b \in F_p$ such that $4a^3 + 27b^2 \neq 0 (\bmod\ p)$. $E(F_p)$ represents a set of points $(x, y) \in F_p \times F_p$ which satisfy E and with an additional point called point at infinity O. The cardinality of E should be divisible by a large Prime number because of the issue of security raised by Pohlig and Hellman.
   B**:** $(B \neq O)$ A finite point on E with an order t (a large prime number).

Every participant has a private/public key pair $(d_Z, Q_Z = (x_Z, y_Z) = d \times B)$, where $d_i \in [1, t-1]$ such that $x_Z \neq 0$. Subscript Z indicates the identification of participant Z. $Q_{ZX} = Q_Z \times d_X = Q_X \times d_Z$ is the Diffie-Hellman shared secret key between the persons Z and X.

2. Key generation phase: In this phase original signer delegates his signing power to proxy signer by generating a key (proxy signing key) using his private key d_o and message warrant m_w. The value (proxy signing key) is equivalent to the signing of warrant by original signer using his private key. The Steps to generate the proxy key are shown below:

Part 1: Private/Public key generation phase: All original signers and the designated proxy signer are authorized to select their own individual secret keys. All original signer and proxy signer randomly selects a number $d_i \in [1, t-1]$ this number is his private key and then using this they calculate public key that is $Q_i = d_i \times B = (x_{Q_i}, y_{Q_i})$. If $x_{Q_i} \neq 0$, d_i is the secret key and Q_i is the public one.

Part 2: Proxy signing key generation phase

Step 1: Each original signer selects random number (secret key) $t_{o_i} \in \{1, 2, 3, \ldots, t-1\} \backslash d_i$, and then computes $k_i = t_{0_i} \times B = (x_{k_i}, y_{k_i})$ and also computes

$$v_i = d_i h(m_w, x_{Q_i}) + t_{0_i} \times x_{Q_i} \bmod t.$$

Now each original signer broadcast (v_i, k_i) to other original signers. Each original use v_i for authenticate himself to the other original signers so that anyone except valid original signer cannot send these parameter and k_i use for generating group commitment value.

Step 2: Each original signer after receiving (v_i, k_i) first verify the parameter and checks all parameters are correct i.e.

$$R_i' = h(m_w, x_{Q_i}) \times Q_i + k_i \times x_{Q_i} \bmod t.$$

if $v_i \times B = (x_{v_i}, y_{v_i})$ and $x_{v_i} = x_{R_i'} \bmod t$ then original signer accepts (v_i, k_i) as a valid parameter.

Step 3: Then all original signers calculate group commitment value $K = \sum_{i=1}^{n} k_i = (x_K, y_K)$.

Step 4: Now each original signer calculate sub delegation parameter using his secret key, private key and group commitment value that is

$$x_o = \sum_{i=1}^{n} x_{Q_i}$$

$$\sigma_i = d_i x_{Q_i} h(m_w, x_{Q_i}, x_K, x_p, x_o) + t_{O_i} x_K \bmod t.$$

Step 5: Now each original signer sends sub delegation parameter to proxy signer with the help of Diffie-Hellman key exchange so that their is mutual authentication between original signer and proxy signer. First original signer makes the parameter that has to be send as follows:

Step 6: Original signer choses a random number $\beta \in [1, t-1]$ and calculates $\lambda_1 = \beta \times Q_{O_i}$, $\lambda_2 = \beta \times Q_{O_i P} = (x_2, y_2)$, and $\lambda_3 = \sigma_i \times Q_{O_i P} = (x_c, y_c)$

such that $x_2 \neq 0$, otherwise he has to repeat this step with another random number.

Step 7: Sends the proxy share as $(\lambda_1, \lambda_3, (x_2 \times \sigma_i) \bmod t, x_2 \times k_i, K, m_w)$ to p.

Step 8: Upon receiving the value $(\lambda_1, \lambda_3, (x_2 \times \sigma_i) \bmod t, x_2 \times k_i, K, m_w)$ from the original signer, p gets the partial proxy share σ_i back as below: Calculates $\lambda_2 = \lambda_1 \times d_p = (x_2, y_2)$, using this λ_2 he will calculate $\sigma_i = \sigma_i \times x_2 \times x_2^{-1} \bmod t$ and $k_i = x_2 \times k_i \times x_2^{-1}$ and then verifies the validity of σ_i by checking the equation $\lambda_3 =^? \sigma_i \times Q_{O_i P}$ and $R_i' = h(m_w, x_{Q_i}, x_K, x_p, x_o) \times Q_i \times x_{Q_i} + k_i \times x_K \bmod t$. if $\sigma_i \times B = (x_{\sigma_i}, y_{\sigma_i})$ and $x_{\sigma_i} = x_{R_i'} \bmod t$ then proxy signer accepts sub delegation parameter. If the equality gets hold, both validity of the share and authentication of original signer are proved.

Step 9: Proxy multi signature secret key generation: After validating all sub delegation parameter proxy signer computes proxy signing secret key on behalf of all original signer as follows:
first proxy signer calculate

$$\sigma_0 = \sum_{i=1}^{n} \sigma_i$$

and then proxy signer generate random number l and calculate $L = l \times B = (x_L, y_L)$. finally calculate proxy signing secret key $d_{p'}$

$$d_{p'} = \sigma_0 \times x_{Q_p} + h(m_w, x_K, x_p, x_o, x_L) \times l + d_p \times x_L \bmod t.$$

3. Proxy multi signature generation phase: After generation of proxy signing key proxy signer sign the message m on a behalf of original signer using secret key $d_{p'}$ and resultant signature is $sign_{d_{p'}}(m)$ and later proxy signer p choses a random number $\beta' \in [1, t - 1]$ and calculates $\lambda_4 = \beta' \times Q_p$, $\lambda_5 = \beta' \times Q_{vp} = (x_3, y_3)$, such that $x_3 \neq 0$, otherwise he has to repeat this step with another random number. And proxy signer send the proxy multi signature $(\lambda_4, x_3 \times L, x_3 \times K, m_w, m, sign_{d_{p'}}(m))$ to v.
4. Proxy multi signature verification phase: Upon receiving proxy multi signature from proxy signer verifier first calculates $\lambda_5 = \lambda_4 \times d_v = (x_3, y_3)$, using this x_3 he will calculate $L = x_3 \times L \times x_3^{-1} \bmod t$ and $K = x_3 \times K \times x_3^{-1}$. Now verifier computes proxy public key using these value and public key of original signer and proxy signer.

$$Q_{p'} = \sum_{i=1}^{n} Q_i h(m_w, x_{Q_i}, x_K, x_p, x_o) x_{Q_p} + K \times x_K + L \times h(m_w, x_K, x_p, x_o, x_L) + Q_p \times x_L.$$

using this proxy multisignature public key verifier will validates $sign_{d_{p'}}(m)$ and verify the correctness of the verification equation.

5 Security Analysis

Security analysis and some discussion are given below. First of all we show the correctness of verification equation that means derivation of proxy multi-signature public key

$$Q_{p'} = d_{p'} \times B = \sigma_0 \times x_{Q_p} \times B + h(m_w, x_K, x_p, x_o, x_L) \times l \times B + d_p \times B \times x_L \bmod t$$

$$Q_{p'} = d_{p'} \times B = \sum_{i=1}^{n} \sigma_i \times x_{Q_p} \times B + h(m_w, x_K, x_p, x_o, x_L) L + Q_p \times x_L \bmod t.$$

After putting σ_i value we get Q_p that is

$$Q_{p'} = \sum_{i=1}^{n} Q_i h(m_w, x_{Q_i}, x_K, x_p, x_o) x_{Q_p} + K \times x_K + L \times h(m_w, x_K, x_p, x_o, x_L) + Q_p \times x_L.$$

There are some security concern as follows:

1. ECDLP: The proposed scheme is based on ECDLP, therefore attacker has to face the difficulty of solving the ECDLP so that he will be unable to derive the secret key from public key so forging the signature is difficult for attacker.
2. Parameter passing using Diffie-Hellman: The original signer and proxy signer sends the parameter to proxy signer and verifier respectively in Diffie-Hellman fashion. In which they calculate $\lambda_1, \lambda_2, \lambda_3, \ldots$ etc, these values are calculated using the public key and the private key of the receiver and sender so that it is guaranteed that the parameter is coming directly from the particular sender and also the sender is assured that only the receiver can see the values. Similarly the Receiver has surety that the parameter is coming from the actual sender. Hence there is mutual authentication between the sender and the receiver. Forging the parameter is as difficult as solving ECDLP. With this method we can also stop "original signer forgery"- sometimes original signer generates proxy signature and sends to the verifier, in that case it is not verified whether the parameter is coming from the original signer. We can now trace who is the sender and who is the receiver and the original signer can not bypass the proxy signer.
3. Public key substitution attack: The proxy signature verification equation at the verifier side is combine with the public keys of the original signer, the proxy signer and one way hash function. Forging a public key in the verification equation the attacker has to face the problem of ECDLP and one way hash function that is more difficult. If we look at verification eq. that is

$$Q_{p'} = K \times x_K + L \times h(m_w, x_K, x_p, x_o, x_L) + Q_p \times x_L + Q_1 h(m_w, x_{Q_1}, x_K, x_p, x_o) x_{Q_p} + Q_2 h(m_w, x_{Q_2}, x_K, x_p, x_o) x_{Q_p} + \ldots + Q_n h(m_w, x_{Q_n}, x_K, x_p, x_o) x_{Q_p}$$

In this equation x_o is summation of all x-coordinate of all original signers public key. In one case original signer $Q1$ may forge his public key and randomly selects

a pair $Q'_1 = (x_{Q'_1}, y_{Q'_1})$ as his public key, now for satisfying the verification equation original signer can change only value of K by changing his share in K but changing the value K that means in each term in the equation there will be a change and changing the public key also affects x_o in one way hash function hence the difficulty of so doing is harder then the ECDLP.

6 Conclusion

We have proposed a proxy multi signature scheme based on elliptic curve cryptosystem (ECC). The proposed scheme is secure then Tzer-Shyong Chen [11], Kuo-Hsuan et. al [2] and Sun et. al scheme [5] and computation cost is independent of the number of original signers. Our scheme uses Diffie-Hellman for sending the sub delegation parameter and we are not using any encryption and decryption method for sending parameter. So we also reduce the cost of encryption and decryption. Besides this Our scheme is able to withstand the public key substitution attack and original signer forgery attack.

References

1. Nonmember, B.L., Member, K.K.: Strong proxy signatures, December 13 (1999)
2. Hsu, C.-L., Wu, T.-S., He, W.-H.: New proxy multi-signature scheme. Applied Mathematics and Computation 162(3), 1201–1206 (2005)
3. Cao, F., Cao, Z.: Cryptanalysis on a proxy multi-signature scheme
4. Yi, G.X.L., Bai, G.: Proxy multi-signature scheme: a new type of proxy signature scheme. Electronics Letters 36, 134–138 (2000)
5. Sun, H.: On proxy multi-signature schemes. In: Proceedings of the International Computer Symposium, pp. 65–72 (2000)
6. Park, J.H., Kang, B.G., Park, S.: Cryptanalysis of Some Group-Oriented Proxy Signature Schemes. In: Song, J.-S., Kwon, T., Yung, M. (eds.) WISA 2005. LNCS, vol. 3786, pp. 10–24. Springer, Heidelberg (2006)
7. Guilin Wang, J., Bao, F., Deng, R.H.: Proxy signature scheme with multiple original signers for wireless e-commerce applications. Infocomm Security Department, Institute for Infocomm Research (I2R). 21 Heng Mui Keng Terrace, Singapore 119-613. IEEE (2004)
8. Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures for delegating signing operation. In: Neuman, C. (ed.) Proceedings of the 3rd ACM Conference on Computer and Communications Security, pp. 48–57. ACM Press, New Delhi (March 1996)
9. Kim, S., Park, S., Won, D.: Proxy signatures. In: Proc. 1st International Information and Communications Security Conference, pp. 223–232 (1997)
10. Chen, T.-S., Chung, Y.-F., Huang, G.-S.: Efficient proxy multi-signature schemes based on the elliptic curve cryptosystem. Computers & Security 22(6), 527–534 (2003)
11. Chen, T.-S., Chung, Y.-F., Huang, K.-H.: A traceable proxy multi-signature scheme based on the elliptic curve cryptosystem. Applied Mathematics and Computation 159(1), 137–145 (2004)
12. Yuan, Y.: Improvement of Hsu-Wu-He's Proxy Multi-signature Schemes. In: Jonker, W., Petković, M. (eds.) SDM 2005. LNCS, vol. 3674, pp. 234–240. Springer, Heidelberg (2005)