I.J. Computer Network and Information Security, **2014, 4, 46-54**
Published Online March 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2014.04.06

Security Analysis and Implementation of an Improved Cch2 Proxy Multi-Signature Scheme

Raman Kumar
Department of Computer Science and Engineering, D A V Institute of Engineering and Technology, Jalandhar, Punjab, India.
er.ramankumar@aol.in

Abstract **—** Many of the signature schemes are proposed in which the t out of n threshold schemes are deployed; but they still lack the property of security. In this paper, we have discussed implementation of improved CCH1 and improved CCH2 proxy multi-signature scheme based on elliptic curve cryptosystem. We have represented time complexity, space complexity and computational overhead of improved CCH1 and CCH2 proxy multi-signature schemes. We have presented cryptanalysis of improved CCH2 proxy multi-signature scheme and showed that improved CCH2 scheme is suffered from various attacks i.e. forgery attack and framing attack.

Index Terms **—** Proxy Signature; Enforceability; Secret Sharing.

I. INTRODUCTION

Elliptic curve cryptosystem was introduced by Koblitz [1] and Miller [2] in 1985. The attractive public key cryptosystem is ECC because ECC has shorter key size and faster computing speed. The public key cryptosystem's security is based on the relative complexity of the mathematical problem. For example, the security of RSA depends on integer factorizing problem and the security of DSA depends on discrete logarithm problem [10]. ECC is developed by integer points over elliptic curves in finite fields. The security of ECC is based on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP).When one person needs to delegate his/her signing capability to other party or person, the proxy signature is very useful tool. In 1996, Mambo et al.'s [3] introduced first proxy signature scheme. Proxy signature is a signature scheme which allows one party called original signer to delegate his/her signing capability to another party called Proxy signer and then on behalf of the original signer, proxy signer can create signature on messages and after signature creation, proxy signer sends these signatures to the verifier and then verifier verify these signatures. Yi et al.'s [6] presented proxy multi-signature scheme which enables one proxy signer to create signature on behalf of group of original signers.

Various security definitions:

A proxy signature should have security properties [4] and they are:

1. Strong Unforgeability: Only designated proxy signer can create proxy signatures. Original signer or any other party cannot create proxy signatures.
2. Verifiability: A verifier can be convinced of the original signer's agreement on the signed message from the proxy signature.
3. Strong Identifiability: From the proxy signature, anyone can determine the identity of corresponding proxy signer.
4. Strong Undenability: once valid proxy signature is created by proxy signer for an original singer, he/she cannot repudiate signature creation.
5. Distinguishability: Proxy signatures that are created by proxy signer are distinguishable from ordinary signatures that are created by original signer.

Various attacks:

- Public key substitution attack: By updating his/her own public key, an attacker can forge a valid proxy multi-signature [7].
- Original signer's forgery attack: Without agreement of proxy signer, original signers can create proxy multi-signature. Verifier will be sure that any proxy multi-signature created by using forged signing key are created by agreement of all original signers and proxy signer [15]. Under the name of proxy signer, original signer can forge valid proxy multi-signature.
- Transferring attack: In this attack, proxy signer's standard schnorr signature can be converted into proxy signature in which signer is regarded as proxy signer by the verifier and vice versa [13].
- Framing attack: In this attack, any user P can be framed by malicious users $A_{1,} A_2...A_n$. User P does not receive any delegation from the users $A_{1,} A_2...A_n$, but the malicious users $A_{1,} A_2...A_n$ can forge a proxy multi-signature for message by user P on behalf of users $A_{1,} A_2...A_n$ [16].

Mambo et al.'s [3] introduced different types of delegations i.e. full delegation, partial delegation and delegation by warrant. In full delegation, the original signer gives his/her own secret key to proxy signer, so that proxy signer can create same signature as original signer creates. In partial delegation, a proxy signer has proxy private key, which is different from original signer's private key and proxy signer can sign range of messages in partial delegation because delegation period is not specified. In delegation by warrant, warrant is added that specifies what kinds of messages are delegated, the delegation period, IDs of original signers and proxy signer etc. According to protection of proxy signer, partial delegation is further divided as proxy-unprotected and proxy-protected. In proxy unprotected, the original signer gives proxy signing key to the proxy signer and proxy signer create signature by using that key. So, in proxy unprotected proxy signature the original signer knows the proxy signing key and he/she can also create the same proxy signatures. In proxy protected, the proxy signer compute proxy signing key from his/her owns private key and the key given by original signer and then generates proxy signature. So in proxy protected proxy signature any third party including the original signer cannot create the same proxy signatures. In partial delegation, the signing capability of the proxy signer can be revoked by the original signer by two ways i.e. (1) Prepare revocation list and make it publicly seen. (2) The public key of original signer can be changed and accordingly all proxies of sincere proxy signers are updated. M. Mambo, K. Usuda, and E. Okamoto [4] have introduced proxy protected proxy signature scheme based on discrete logarithm problem. S. Kim, S. Park, and D. Won [5] have introduced two new types of proxy signature schemes i.e. proxy signature for partial delegation with warrant and proxy signature for threshold delegation. Partial delegation with warrant combines the benefits of partial delegation and delegation with warrant. H. M. Sun [7] analyzes Yi et al.'s [6] proxy multi-signature schemes and show that these schemes are suffered from public key substitution attack and direct forgery attack. They introduced a new proxy protected and proxy unprotected proxy multi-signature schemes which do not suffered from these attacks. B. Lee, H. Kim, and K. Kim [8] develop a strong non-designated proxy signature scheme and apply it to multi-proxy signature in which multiple original signers can delegate his signing capabilities to undetermined proxy signers. T. S. Chen, T. P Liu, and Y. F Chung [9] introduced a proxy protected proxy signature scheme based on elliptic curve discrete logarithm problem. On the basis of time complexity, they compare H. M. Sun [7] and the proposed scheme. T.S. Chen, Y.F. Chung and G.S. Huang [10] introduced an improved scheme in which the exponential operations are replaced by elliptic curve multiplicative ones. ECC with lower computational overhead and a smaller key size can achieve a level of security equal to that of the RSA or DSA. This proposed scheme is called CCH1 scheme. Without loss of security, the time complexity of proposed scheme is reduced and performance is enhanced. T.S. Chen, Y.F. Chung, G.S. Huang [11] introduced a traceable proxy multi-signature scheme. This proxy signature scheme is independent of number of original signers so, number of operations required for verification decreases. This proposed scheme is called CCH2 scheme. On the basis of time complexity they compare the sun's [7] and proposed proxy multi-signature scheme. M.S. Hwang, S.F. Tzeng and C.S. Tsai [12] introduced a generalized version of the (t1/n1-t2/n2) proxy signature scheme based on elliptic curve discrete logarithm problem. G. L. Wang, F. Bao, J. Y. Zhou, and R. H. Deng [13] analyze the security of some proxy signature schemes i.e. M. Mambo et al.'s [4] and B. Lee et al.'s [8] and show that all these schemes are suffered from various attacks. S. Wang, G. Wang, F. Bao and J. Wang [14] shows that chen et al.'s [9] is vulnerable to an original signer forgery attack and introduced an improved scheme which is secure against the proposed attack. J.H. Park, B.G. Kang, S. Park [15] shows that proxy multi-signature schemes proposed by Chen et al.'s [10, 11] are vulnerable to forgery attack by one or all original signers. M.H. Chang, I.T. Chen and M.T. Chen [17] presented a proxy protected signature scheme based on ECDSA which satisfies security properties. Fengying Li, Qingshui Xue [18] proposed improved CCH1 and improved CCH2 schemes that are not suffered from forgery attack. Tutanescu, C. Anton, L. Ionescu and D. Caragata [19] examines that ECC is an attractive public key cryptosystem than RSA or DSA for mobile devices, which have limited memory, processing capability and network connectivity. ECC can be implemented with less hardware, because of shorter key length. They have defined the application of ECC i.e. internet, smart cards, PDAs and PCs.

II. TIME, SPACE AND COMPUTATIONAL OVERHEAD OF IMPROVED CCH1 AND IMPROVED CCH2 PROXY MULTI-SIGNATURE SCHEME

A. Review of improved CCH1 proxy multi-signature scheme

There are four phases—Initialization phase, key generation phase, proxy multi-signature generation phase and proxy multi-signature verification phase [18].

Phase 1: Initialization phase over the elliptic curve domain following parameters must be known.

- ➢ A field size p, which is odd prime.
- ➢ Two parameters a, b Fp to define the equation of elliptic curve E over Fp (i.e., y2 = x3 + ax + b (mod p)), where 4a3 + 27 b2 ≠ 0 (mod p)
- ➢ A finite point B = (xB,yB) whose order is a large prime number in E(Fp), where B is a point in E(Fp). Where B ≠ O, because O denotes an infinity point.
- ➢ The order of B = t.

Phase 2: Key generation phase: This phase is further divided into two parts.

Part 1: Personal key generation phase: All original signers and the designated proxy signer select their own individual secret keys.

- For each $1\leq i \leq n$, the original signer A_i secretly selects a random number $1\leq d_i \leq t - 1$ as his private key, and computes the corresponding public key $Q_i = d_i \times B = (x_{Q_i}, y_{Q_i})$, where "×" indicates the multiplication of a number and elliptic curve point.
- The proxy signer is provided with a private key $1\leq d_p \leq t - 1$ and a corresponding public key $Q_p = d_p \times B = (x_{Qp}, y_{Qp})$.

Part 2: Proxy-signature secret key generation phase:

Step 1: (Secret key generation): For each $1\leq i \leq n$, the original signer A_i selects a random number k_i $\{1, 2\ldots t-1\}/ d_i$ as secret key.

Step 2: (Group commitment value generation): Then computes

$$R_i = k_i \times B = (x_{Ri}, y_{Ri}) \qquad (1)$$

If $x_{Ri} = 0$ then return to step 1; otherwise A_i gives R_i to other original signers.

Step 3: (Sub-delegation parameter generation): For each $1\leq i \leq n$, the original signer A_i uses his own secret keys d_i, k_i and the group commitment value x_R to compute the following:

$$s_i = d_i * x_{Q_i} * h(M_w, R_i) - k_i * x_{R_i} \pmod t \qquad (2)$$

Where h ( ) is a hash function and the warrant M_w contains information such as the IDs of all original signers and proxy signer. Then, the sub-delegation parameter for A_i is (M_w, R_i, s_i).

Step 4: (Sub-delegation parameter verification): After the proxy signer has received the sub-delegation parameters then the proxy signer P computes

$$S_i \times B = (x_{Q_i} * h(M_w, R_i))*Q_i - x_{R_i} R_i \qquad (3)$$

and check whether it holds. If it holds then the proxy signer accepts (M_w, R_i, s_i) as a valid sub-delegation parameter; otherwise, he can reject it and requests a valid one A_i , or terminate this protocol.

Step 5 :(Proxy multi-signature secret key generation): Then computes the proxy multi-signature secret key as follows:

$$d = dp* xQp + \sum ni=1si \bmod t \qquad (4)$$

Phase 3: Proxy multi-signature generation phase: The proxy multi-signature is in the form of (m, mw, R, Sigd(m)), where Sigd(m) is the signature generated by a designated signature scheme (EC-schnorr signature scheme) using the proxy signing key d and m is message.

Step 1: Proxy signer P choose random number j where $1\leq j \leq t-1$ and calculate $J = j \times B = (J_x, J_y)$.

Step 2: Compute $e = h(m, J_x)$ where $h(J_x, m)$ is hash function. If e = 0 then go to step 1.

Step 3: Compute

$$y = j - d * e \pmod t \qquad (5)$$

and the output $sign_d(m) = (e,y)$.

Phase 4: Proxy multi-signature verification phase: When the verifier verifies the signature, he or she calculates the proxy public value Q corresponding to the proxy signature key d as

$$Q = Q_P * x_{Q_p} + \sum_{i=1}^{n} (h(M_w, R_i) *Q_i - x_{R_i} \times R_i) \qquad (6)$$

With the value, the verifier can confirm the validity of Sig_d (m) by validating the verification equality of the designated signature scheme.

Step 1: Compute $J = y \times B + e \times Q = (J_x, J_y)$

Step 2: And compute $e' = h (J_x, m)$. Then check that $e' = e$ and if these are equal then valid signature otherwise not.

B. Review of improved CCH2 proxy multi-signature scheme

There are four phases—Initialization phase, key generation phase, proxy multi-signature generation phase and proxy multi-signature verification phase [18].

Phase 1: Initialization phase: Over the elliptic curve domain following parameters must be known.

- ➢ A field size p, which is a odd prime.
- ➢ Two parameters a, b Fp to define the equation of elliptic curve E over Fp (i.e., y2 = x3 + ax + b (mod p)), where 4a3 + 27 b2 ≠ 0 (mod p)
- ➢ A finite point B = (xB, yB) whose order is a large prime number in E(Fp) , where B is a point in E(Fp). Where B ≠ O, because O denotes an infinity point.
- ➢ The order of B = t.

Phase 2: Key generation phase: This phase is further divided into two parts:

Part 1: Personal key generation phase: All original signers and the designated proxy signer are authorized to select their own individual secret keys.

- For each $1\leq i \leq n$, the original signer A_i secretly selects a random number $1\leq d_i \leq t - 1$ as his private key, and computes the corresponding public key $Q_i = d_i \times B = (x_{Q_i}, y_{Q_i})$, where "×" indicates the multiplication of a number by an elliptic curve point.

- The proxy signer is provided with a private key $1 \leq d_p \leq t - 1$ and a corresponding public key $Q_p = d_p \times B = (x_{Qp}, y_{Qp})$. All public keys Q_i and Q_p must be certified by the CA.

Part 2: Proxy-signature secret key generation phase:

Step 1: (Secret key generation): For each $1 \leq i \leq n$, the original signer A_i selects a random number k_i $\{1, 2\ldots t-1\}/ d_i$ as secret key.

Step 2: (Group commitment value generation): Then computes $R_i = k_i \times B = (x_{Ri}, y_{Ri})$. If $x_{Ri} = 0$ then return to step 1; otherwise A_i broadcasts R_i to other original signers. On receiving R_j $(1 \leq j \leq n, j \neq i)$, A_i calculates

$$R_i = (x_R, y_R) \tag{7}$$

Step 3: (Sub-delegation parameter generation): For each $1 \leq i \leq n$, the original signer A_i uses his own secret keys d_i, k_i and the group commitment value x_R to compute the following.

$$s_i = d_i * h(M_w, x_{Qp}, x_{Qi}, x_R) - k_i * x_R (\text{mod } t) \tag{8}$$

Where h ( ) is a hash function and the warrant M_w contains information such as the IDs of all original signers and proxy signer. Then, the sub-delegation parameter for A_i is (M_w, s_i).

Step 4: (Sub-delegation parameter verification): After the proxy signer has received the sub-delegation parameters then the proxy signer P computes

$$si * B = h(Mw, xQp, xQi, xR) * Qi - xR \times Ri \tag{9}$$

checks whether it holds. If it holds then the proxy signer accepts (M_w, s_i) as a valid sub-delegation parameter; otherwise, he can reject it and requests a valid one A_i , or terminate this protocol.

Step 5 :(Proxy multi-signature secret key generation): Then computes the proxy multi-signature secret key as follows:

$$d = dp + \qquad i \quad \text{mod } t \tag{10}$$

Phase 3: Proxy multi-signature generation phase: The proxy multi-signature is in the form of (m,mw,R,Sigd(m)) , where Sigd(m) is the signature generated by a designated signature scheme (EC-schnorr signature scheme) using the proxy signing key d and m is message.

Step 1: Proxy signer P choose random number j where $1 \leq j \leq t-1$ and calculate $J = j \times B = (J_x, J_y)$.

Step 2: Compute $e = h(m, J_x)$ where $h(J_x, m)$ is hash function. If e = 0 then go to step 1.

Step 3: Compute

$$y = j - d \times e \ (\text{mod } t) \tag{11}$$

and the output $\text{sign}_d(m) = (e,y)$.

Phase 4: Proxy multi-signature verification phase: When the verifier verifies the signature, he or she calculates the proxy public value Q corresponding to the proxy signature key d as

$$Q = Q_P + (M_w, x_{Qp}, x_{Qi}, x_R) \times Q_i - R \times x_R \tag{12}$$

With the value, the verifier can confirm the validity of Sig_d (m) by validating the verification equality of the designated signature scheme.

Step 1: Compute $\bar{J} = y \times B + e \times Q = (\bar{J}_x, \bar{J}_y)$

Step 2: And compute $e' = h(\bar{J}_x, m)$. Then check that $e' = e$ and if these are equal then valid signature otherwise not.

C. Time, Space and Computational overhead of improved CCH1 and CCH2 scheme

This table shows the time, space and computational overhead of improved CCH1 and CCH2 proxy multi-signature scheme with varying value of field size (p).

TABLE NO. 1 Time, space and computational overhead of improved CCH1 and CCH2 scheme

Sr. No	p	Time complexity		Space complexity		Computational overhead	
		CCH1	CCH2	CCH1	CCH2	CCH1	CCH2
1.	5	1724	1549	1994	1810	211	175
2.	7	1619	1358	1894	1379	208	177
3.	11	1786	1369	1641	1515	210	180
4.	13	1503	1494	1565	1459	211	168
5.	17	1583	1426	1813	1524	210	171
6.	19	1777	1461	1774	1308	211	169
7.	23	1728	1572	1905	1505	218	177
8.	29	1761	1456	1979	1629	212	187
9.	31	1748	1448	1914	1602	213	170
10.	37	1590	1419	1918	1700	203	184
11.	41	1510	1407	1576	1524	213	187
12.	43	1535	1516	1624	1405	215	175
13.	47	1745	1359	1517	1387	204	168
14.	53	1539	1350	1963	1506	203	182
15.	59	1613	1431	1934	1508	215	172
16.	61	1624	1421	1987	1644	208	181
17.	67	1581	1350	1695	1661	217	170
18.	71	1636	1340	1964	1391	216	183
19.	73	1667	1391	1744	1648	217	175
20.	79	1779	1418	1813	1405	214	185
21.	83	1554	1363	1943	1464	208	173
22.	89	1583	1575	1904	1515	209	176
23.	97	1553	1438	1927	1517	219	184

D. Graphical representation of Time, Space and Computational overhead of improved CCH1 and CCH2 scheme

Graphs shows the time, space and computational overhead of improved CCH1 and CCH2 proxy multi-signature scheme with varying value of field size (p).

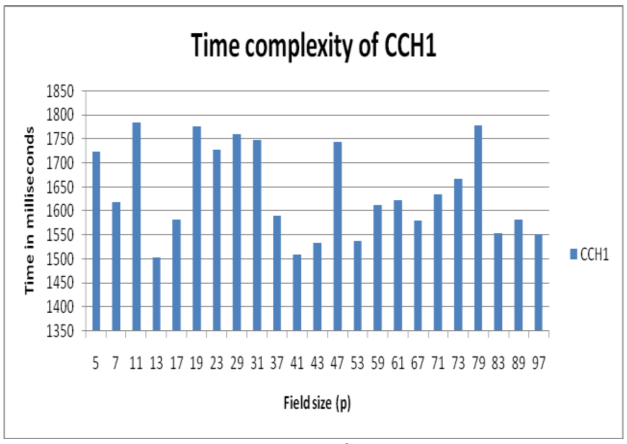

Figure. No. 1 Time complexity of improved CCH1 scheme with varying value of field size (p)

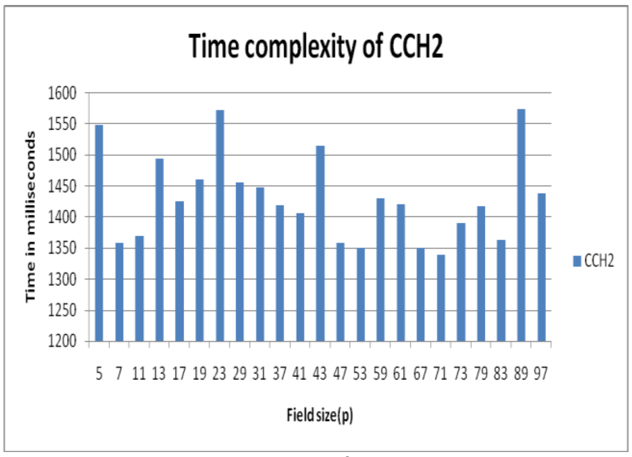

Figure. No. 2 Time complexity of improved CCH2 scheme with varying value of field size (p)

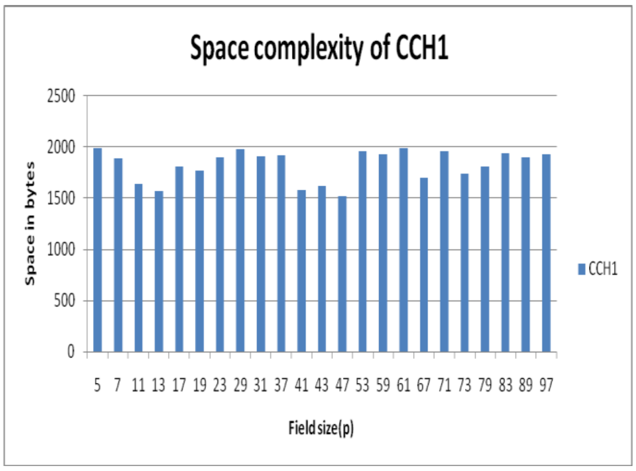

Figure. No. 3 Space complexity of improved CCH1 scheme with varying value of field size (p)

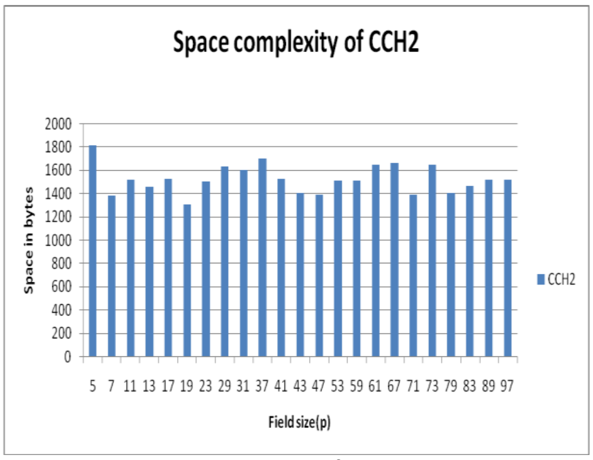

Figure. No. 4 Space complexity of improved CCH2 scheme with varying value of field size (p)

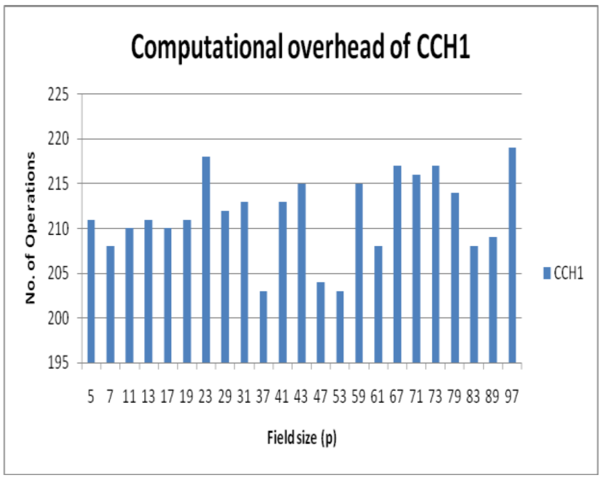

Figure. No. 5 Computational overhead of improved CCH1 scheme with varying value of field size (p)

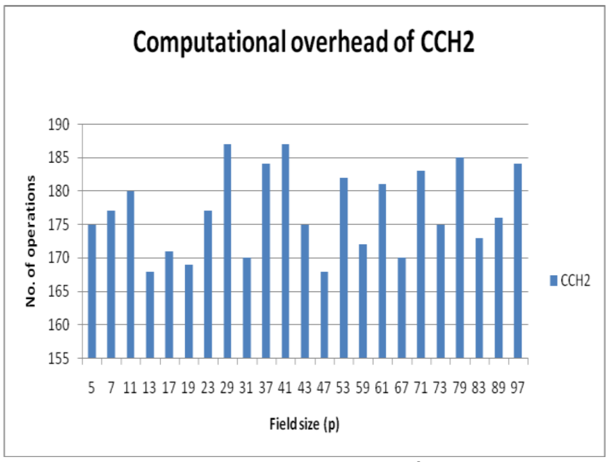

Figure. No. 6 Computational overhead of improved CCH2 scheme with varying value of field size (p)

III. CRYPTANALYSIS OF IMPROVED CCH2 PROXY MULTI-SIGNATURE SCHEME

Elliptic curve based schnorr signature scheme is used in improved CCH2 proxy multi-signature scheme to generate and verify the signature.

A. Forge a proxy multi-signature

Suppose proxy signer P signed a message with his private key dp and the signature that are generated is (m, mw, R, e, y ´) where e= h(m, jx) and y ´´= j-dp *e (mod t). Upon received the signature (m, mw, R, e, y ´) the malicious original signers A1…An can forge proxy signature by computing following operations:

- Compute y ´= $\sum_{i=1}^{n} s_i$ * e mod t.
- Compute y= y ´´- y ´

Finally, the malicious original signers A1…An can forge proxy signature (m, mw, R, e, y). The following proof shows that why the proxy signature (m, mw, R, e, y) is valid.

PROFF:

$$y = y' - y'$$
$$= j - d_p * e \bmod t - \quad {}_i * e \bmod t$$
$$= j - (d_p + \quad {}_i) e \bmod t$$
$$y = j - d*e \bmod t$$

Proxy multi-signature verification: When the verifier verifies the signature, he or she calculates the proxy public value Q corresponding to the proxy signature key d as

$$Q = Q_P + \qquad (M_w, x_{Qp}, x_{Qi}, x_R) \times Q_i - R \times x_R \qquad (13)$$

Then, the verifier can confirm the validity of Sigd (m) by validating the verification equality of the designated signature scheme.

Step 1: Compute $J = y \times B + e \times Q = (J_x, J_y)$

Step 2: And compute e ´= h (J_x,m) . Then check that e ´= e and if these are equal then valid signature otherwise not.

B. Forge the proxy signer's signature

After getting signature (m,mw,R,e,y) that are generated by proxy signer on behalf of original signers Where e= h(m, jx) and y = j-d*e (mod t), the original signer A1…An can forge proxy signer P's signature on message m by computing following opeartions:

- Each A_i compute s_i * e where i=1,2,..n.
- Compute y ´= $\sum_{i=1}^{n} s_i$ *e mod t.
- Compute y ´= y + y ´.
- (m, m_w, R, e, y ´) is valid signature on message m .

The malicious original signers can forge signature (m, mw, R, e, y ´) on message m with respect to proxy signer P's private key dp.

The following proof shows that why the proxy signature (m, mw, R, e, y ´) is valid

PROFF:

$$y' = y + y'$$
$$= j - d*e + \quad {}_i *e \bmod t$$
$$= j - (d_p + \quad {}_i) e + \quad {}_i *e \bmod t$$
$$y' = j - d_p * e \bmod t$$

Proxy multi-signature verification: When the verifier verifies the signature, he or she use proxy public value Qp corresponding to the proxy signature key dp. Then the verifier can confirm the validity of Sigd (m) by validating the verification equality of the designated signature scheme.

Step 1: Compute $J = y \times B + e \times Q_P = (J_x, J_y)$

Step 2: And compute e ´= h (J_x,m) . Then check that e ´= e and if these are equal then valid signature otherwise not.

C. Framing Attack

On behalf of users A1, A2… An, malicious users A1, A2… An can forge a proxy multi-signature for message m by some user P, such that user P was never designated by users A1,A2…An. Suppose proxy signer P signed a message with his private key dp, the signature is (m, mw, R, e, y ´) where e= h(m, jx) and y ´= j-dp *e (mod t).

Upon received the signature (m, mw, R, e, y ´), the malicious original signers A1…An can forge proxy multi-signature by performing following steps:

- The malicious users A_1,A_2…A_n pretend to produce a forge warrant m_w, which recording the delegation information such as identities of the malicious users A_1,A_2…A_n and user P .
- For each $1 \leq i \leq n$, the malicious user A_i selects a random number $1 \leq k_i \leq t - 1$, and then computes R_i = k_i X B = (x_{Ri}, y_{Ri}) and broadcast R_i to other users.
- On receiving R_j ($1 \leq j \leq n$, $j \neq i$), A_i calculates

$$R = \sum_{i=1}^{n} R_i = (x_R, y_R) \qquad (14)$$

$$s_i = d_i * h(M_w, x_{Qp}, x_{Qi}, x_R) - k_i * x_R \pmod t \qquad (15)$$

Note that user P doesn't receive any information from the malicious users A1, A2…An.

- Compute y ´= $\sum_{i=1}^{n} s_i$ *e mod t;
- Compute y= y ´´- y ´

Finally the malicious users can forge a valid signature (m, mw, R, e, y) on message m by some user P on behalf of users A1…An, such that user P was never designated by users A1…An. The following shows why the signature (m, mw, R, e, y) is valid.

PROFF:

$$y = y'' - y'$$
$$= j - d_p * e \bmod t - \sum_{i=1}^{n} s_i *e \bmod t$$
$$= j - (d_p + \sum_{i=1}^{n} s_i) e \bmod t$$
$$y = j - d*e \bmod t$$

From above we can see that the malicious users A1…An framed innocent user P.

Proxy multi-signature verification phase: When the verifier verifies the signature, he or she calculates the proxy public value Q corresponding to the proxy signature key d as

$$Q= Q_P+ \quad (M_w, x_{Qp}, x_{Qi}, x_R) \times Q_i - R \times x_R \qquad (16)$$

With the value, the verifier can confirm the validity of Sigd (m) by validating the verification equality of the designated signature scheme.

Step 1: Compute $\bar{J} = y \times B + e \times Q = (\bar{J}_x,\bar{J}_y)$

Step 2: And compute e ´= h ($\bar{J}_x$,m). Then check that e ´= e and if these are equal then valid signature otherwise not.

IV. ENHANCED PROXY MULTI-SIGNATURE SCHEME BASED ON ECC

There are four phases—Initialization phase, key generation phase, proxy multi-signature generation phase and proxy multi-signature verification phase.

Phase 1: System initialization phase: Before the whole scheme can be initialized, the following parameters over the elliptic curve domain must be known.

- ➢ A field size p, which is a large odd prime.
- ➢ Two parameters a, b Fp to define the equation of elliptic curve E over Fp (i.e., y2 = x3 + ax + b (mod p)), where 4a3 + 27 b2 ≠ 0 (mod p)
- ➢ A finite point B = (xB, yB) whose order is a large prime number in E(Fp) , where B is a point in E(Fp). Where B ≠ O, because O denotes an infinity point.
- ➢ The order of B = t.

Phase 2: Key generation phase: This phase can be further divided into two parts.

Part 1: Personal public key generation phase: All original signers and the designated proxy signer are authorized to select their own individual secret keys.

- For each 1≤ i ≤n, the original signer A_i secretly selects a random number $1\leq d_i \leq t-1$ as his private key, and computes the corresponding public key $Q_i = d_i \times B = (x_{Q_i}, y_{Q_i})$, where "×" indicates the multiplication of a number by an elliptic curve point.
- The proxy signer is provided with a private key $1\leq d_p \leq t-1$ and a corresponding public key $Q_p = d_p \times B = (x_{Qp}, y_{Qp})$. All public keys Q_i and Q_p must be certified by the CA.

Part 2: Proxy-signature secret key generation phase:

Step 1: (Secret key generation): For each 1≤ i ≤n, the original signer A_i selects a random number k_i {1, 2… t– 1}/ d_i as secret key.

Step 2: (Group commitment value generation): Then computes $R_i = k_i \times B = (x_{Ri}, y_{Ri})$. If $x_{Ri}=0$ then return to step 1; otherwise A_i broadcasts R_i to other original signers. On receiving R_j $(1 \leq j \leq n, j \neq i)$, A_i calculates

$$_i = (x_R,y_R) \qquad (17)$$

Step 3: (Sub-delegation parameter generation): For each 1≤ i ≤n, the original signer A_i uses his own secret keys d_i, k_i and the group commitment value x_R to compute the following.

$$s_i = d_i * h(M_w, x_{Qp}, x_{Qi}, x_R) - k_i * x_R \pmod t \qquad (18)$$

Where h ( ) is a public collision-resistant hash function and the warrant M_w contains information such as the IDs of all original signers and proxy signer, and the delegation period. Then, the sub-delegation parameter for A_i is (M_w,s_i).

Step 4: (Sub-delegation parameter delivery): For each 1≤ i ≤n, the original signer A_i sends (M_w, s_i) to the proxy signer via a public channel.

Step 5: (Sub-delegation parameter verification): After the proxy signer has received the sub-delegation parameters then the proxy signer P computes

$$s_i * B= h(M_w, x_{Qp}, x_{Qi}, x_R) * Q_i - x_R * R_i \qquad (19)$$

Checks whether it holds. If it holds then theproxy signer accepts (M_w,s_i) as a valid sub-delegation parameter; otherwise, he can reject it and requests a valid one A_i , or terminate this protocol.

Step 6 : (Proxy multi-signature secret key generation): Then computes the proxy multi-signature secret key as follows:

$$d = x_{Q_p} * d_p + \quad _i \mod t \qquad (20)$$

Phase 3: Proxy multi-signature generation phase: The proxy multi-signature affixed to the m is in the form of (m,mw,R,Sigd(m)), where Sigd(m) is the signature generated by a designated signature scheme (EC-schnorr signature scheme) using the proxy signing key d and m is message.

Step 1: Proxy signer P choose random number j where $1 \leq j \leq t-1$ and calculate $J = j \times B = (J_x,J_y)$.

Step 2: Compute e= h(m, J_x) where h(J_x, m) is hash function. If e = 0 then go to step 1.

Step 3: Compute

$$y = j - d \times e \pmod t \tag{21}$$

and the output $sign_d(m) = (e,y)$.

Phase 4: Proxy multi-signature verification phase: When the verifier verifies the signature, he or she calculates the proxy public value Q corresponding to the proxy signature key d as

$$Q = x_{Q_p} * Q_P + \quad (M_w, x_{Qp}, x_{Qi}, x_R) \times Q_i - R * x_R \tag{22}$$

With the value, the verifier can confirm the validity of Sig_d (m) by validating the verification equality of the designated signature scheme.

Step 1: Compute $\bar{J} = y \times B + e \times Q = (\bar{J}_x, \bar{J}_y)$
Step 2: And compute $e' = h(\bar{J}_x, m)$. Then check that $e' = e$ and if these are equal then valid signature otherwise not.

V. CONCLUSIONS

In this paper, we have reviewed improved CCH1 and improved CCH2 proxy multi-signature scheme based on elliptic curve cryptosystem. We have shown time complexity, space complexity and computational overhead of improved CCH1 and CCH2 proxy multi-signature schemes. We have presented cryptanalysis of improved CCH2 proxy multi-signature scheme and showed that improved CCH2 scheme is suffered from various attacks i.e. the malicious original signers can forge the proxy signer's signature, forge a proxy signature and frame anyone who was never designated as proxy signer. In addition, we have proposed an enhanced proxy multi-signature scheme based on ECC. Enhanced scheme does not suffer from these given attacks.

REFERENCES

[1] V.S. Miller, "Uses of Elliptic Curves in Cryptography", Advances in Cryptology - Crypo'85, LNCS 218, pp. 417-426, 1985.

[2] N. Koblitz, "Elliptic Curve Cryptosystems", Mathematics of Computation (AMC), Vo1.48, No.177, pp. 203-209, 1987.

[3] M. Mambo, K. Usuda, and E. Okamoto, "Proxy Signatures for Delegating Signing Operation", Proc. 3rd ACM Conference on Computer and Communications Security, ACM press, pp.48-57, 1996.

[4] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature: Delegation of the power to sign messages", IEICE Trans. Fundamentals, Vol. E79-A, No. 9, pp. 1338-1353, Sep. 1996.

[5] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited", Information and Communications Security -ICICS'97, LNCS 1334, pp. 223-232, 1997.

[6] L. Yi, G. Bai and G. Xiao, "Proxy Multi-signature Scheme: A New Type of Proxy Signature", Electronics Letters, Vol. 36, Issue.6, pp. 527-528, 2000.

[7] H. M. Sun, "On Proxy Multisignature Schemes", Proceedings of the International Computer Symposium, pp.65-72, 2000.

[8] B. Lee, H. Kim, and K. Kim, "Strong proxy signature and its application", In Proceedings of SCIS, pp. 603- 608, 2001.

[9] T. S. Chen, T. P Liu, and Y. F Chung, "A Proxy-Protected Proxy Signature Scheme Based on Elliptic Curve Cryptosystem", Proc. Of IEEE TENCOM, pp. 184-187, 2002.

[10] T.S. Chen, Y.F. Chung and G.S. Huang, "Efficient proxy multisignature schemes based on the elliptic curve cryptosystem", Computers & Security, Elsevier, Vol. 22, No. 6, pp 527-534, 2003.

[11] T.S. Chen, Y.F. Chung, G.S. Huang, "A traceable proxy multi-signature scheme based on the elliptic curve cryptosystem", Applied Mathematics Computation, Elsevier, Vol. 1591 No. 1, pp 137–145, 2004.

[12] M.S. Hwang, S.F. Tzeng and C.S. Tsai, "Generalization of proxy signature based on elliptic curves", Computers Standard & Interfaces, Elsevier, Vol. 26, Issue 2, pp. 73-84, 2004.

[13] G. L. Wang, F. Bao, J. Y. Zhou, and R. H. Deng, "Security Analysis of Some Proxy Signatures", Information Security and Cryptology, LNCS 2971, pp.305-319, 2004.

[14] S. Wang, G. Wang, F. Bao and J. Wang, "Cryptanalysis of A Proxy-Protected Proxy Signature scheme based on Elliptic Curve Cryptosystem", IEEE, pp 3240-3243, 2004.

[15] J.H. Park, B.G. Kang, S. Park, "Cryptanalysis of Some Group-Oriented Proxy Signature Schemes", LNCS, vol. 3786, pp. 10–24, 2006.

[16] Feng Cao, Zhenfu Cao, "Cryptanalysis on a Proxy Multi-Signature Scheme", Proceeding of the first international Multi-symposiums on computer and computational Sciences(IMSCCS), IEEE, Vol. 2, pp. 117-120, 2006.

[17] M.H. Chang, I.T. Chen and M.T. Chen, "Design of Proxy Signature in ECDSA", IEEE, 2008, pp 17-22, 2006.

[18] Fengying Li, Qingshui Xue, "Two Improved Proxy Multi-signature Schemes Based on the Elliptic Curve Cryptosystem", Communications in Computer and Information Science, Vol. 234, pp. 101-109, 2011.

[19] Sheetal Kalra, Sandeep K. Sood, "Elliptic Curve Cryptography: Survey and its Security

[20] Applications", ACAI Proceedings of the International Conference on Advances in Computing and Artificial Intelligence, ACM, pp 102-106,2011.

[21] I. Tutanescu, C. Anton, L. Ionescu and D. Caragata, "Elliptic Curves Cryptosystems Approaches", IEEE, pp 357-362, 2012.

[22] Elliptic curve cryptography tutorial: http://www.tataelxsi.com/whitepapers/ECC_Tut_v1_0.pdf

[23] Harsh Kumar Verma, Kamalpreet Kaur and Raman Kumar, "A Comparison of Threshold Proxy Signature Scheme", The 2008 International Conference on Security and Management, USA, 14-17 July, 2008.

[24] Raman Kumar and Harsh Kumar Verma, "An Advanced Secure (t, n) Threshold Proxy Signature Scheme Based on RSA Cryptosystem for Known Signers", IEEE 2nd International Advance Computing Conference, Thapar University, India, 19-20 February, 2010.

[25] Raman Kumar and Harsh Kumar Verma, "Secure Threshold Proxy Signature Scheme Based on RSA for Known Signers", Journal of Information Assurance and Security, JIAS, Vol. 5, Issue 1, pp. 319-326, 2010.

Author(s) Biographies

Mr. Raman Kumar *(er.ramankumar@aol.in)* is working as an Assistant Professor with the Department of Computer Science and Engineering, DAV Institute of Engineering and Technology, Jalandhar.

Before joining DAV Institute of Engineering and Technology, Jalandhar, He did his Bachelor of Technology *with honours* in Computer Science and Engineering from Guru Nanak Dev University; Amritsar *(A 5 Star NAAC University).* He did his Master of Technology *with honours* Computer Science and Engineering from Guru Nanak Dev University; Amritsar *(A 5 Star NAAC University)*. His major area of research is Cryptography, Security Engineering and Information security. He has published many papers in refereed journals and conference proceedings on his research areas.

How to cite this paper: Raman Kumar,"Security Analysis and Implementation of an Improved Cch2 Proxy Multi-Signature Scheme", IJCNIS, vol.6, no.4, pp.46-54, 2014. DOI: 10.5815/ijcnis.2014.04.06