Research Article

Cryptanalytic Performance Appraisal of Improved CCH2 Proxy Multisignature Scheme

Raman Kumar and Nonika Singla

Department of Computer Science and Engineering, DAV Institute of Engineering and Technology, Jalandhar, Punjab 144004, India

Correspondence should be addressed to Raman Kumar; er.ramankumar@aol.in

Received 28 August 2013; Revised 7 November 2013; Accepted 10 February 2014; Published 27 April 2014

Academic Editor: Wang Xing-yuan

Copyright © 2014 R. Kumar and N. Singla. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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 multisignature scheme based on elliptic curve cryptosystem. We have represented time complexity, space complexity, and computational overhead of improved CCH1 and CCH2 proxy multisignature schemes. We have presented cryptanalysis of improved CCH2 proxy multisignature scheme and showed that improved CCH2 scheme suffered from various attacks, that is, forgery attack and framing attack.

1. Introduction

During the last decade there has been an exponential growth in the number of handheld devices being used all over the world. Devices with limited processing capability such as PDA and smart cards also exchange information over the networks. To provide the confidentiality and authenticity of information is a challenging task in a network environment which consists of constrained devices. The security of public key systems is based on the relative complexity of the underlying mathematical problem. For example, the security of RSA depends on integer factorizing systems and that of DSA depends on discrete logarithm systems [1]. Proxy signatures are very useful tools when one needs to delegate his/her signing capability to another party. Relatively longer key lengths are required to maintain the security of a cryptosystem, because the computational power for cryptanalysis increases. This increases the need for higher computational power in devices to achieve reasonable security. But handheld devices like PDAs, smart cards, and so forth have limited processing capability and therefore the overheads associated with communication must be minimal.

1.1. Various Terms

- (i) Proxy signature: proxy signature, as a variant of ordinary digital signature, allows one party (original signer) to delegate his/her signing capability to another party (proxy signer) such that the proxy signer can sign messages on behalf of the original signer [2].
- (ii) Proxy multisignature: in proxy multisignature, one proxy signer can create signature on behalf of group of original signers.
- (iii) Multiproxy signature: in multiproxy signature, multiple proxy signers can create signature on behalf of one original signer.
- (iv) Multiproxy multisignature: in multiproxy multisignature, multiple proxy signers can create signature on behalf of multiple original signers.
- (v) Proxy unprotected proxy signature: in proxy unprotected, the proxy signer generates proxy signatures only with the proxy signing key given by the original

signer. So the original signer can also generate the same proxy signatures.

- (vi) Proxy protected proxy signature: in proxy protected, the proxy signer generates proxy signature not only with the proxy signing key given by the original signer but also with his own private key. Therefore, anyone else, including the original signer, cannot generate the same proxy signatures.

According to authenticated degree, Mambo et al. [4] give a classification, that is, full delegation, partial delegation, and delegation by warrant.

In full delegation, the original signer gives the same secret key(s) to proxy signer that he has, so that proxy signer can create the 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. Proxy signer can sign range of messages. 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, and so forth.

1.2. Elliptic Curve over Finite Field $GF(p)$. In 1985, elliptic curve cryptography was introduced by Miller [5] and Koblitz [6]. ECC is an attractive public key cryptosystem due to small key size and low computational overhead.

Equation of the elliptic curve on a prime field $GF(p)$ is

$$y^2 \pmod{p} = x^3 + ax + b \pmod{p}, \quad (1)$$

where $4a^3 + 27b^2 \pmod{p} \neq 0$. Here the elements of the finite field are integers between 0 and $p-1$, where p is a large prime number and greater than 3. The operations are performed using modular arithmetic. Modular arithmetic works like ordinary arithmetic except that the answers of the calculation are reduced to its remainder on division by p . The variables and coefficients all take values in set of integers from 0 through $p-1$. The prime number p is chosen such that there is finitely large number of points on the elliptic curve to make the cryptosystem secure. Point multiplication is calculated by two elliptic curve operations, that is, point addition and point doubling (Figure 2). The rules for point addition and point doubling over $GF(p)$ are explained below and see Figure 1.

1.3. Point Addition. The elliptic curve addition is different from simple addition. J and K are two distinct points on the elliptic curve; that is, $J = (x_1, y_1)$ and $K = (x_2, y_2)$ [3].

Let $L = J + K = (x_3, y_3)$ be obtained through the following rules:

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \pmod{p}, \\ y_3 &= \lambda(x_1 - x_3) - y_1 \pmod{p}, \end{aligned} \quad (2)$$

where $\lambda = (y_2 - y_1)/(x_2 - x_1) \pmod{p}$ is the slope of the line through J and K .

- (i) If $K = -J$, that is, $K = (x_1, -y_1 \pmod{p})$, then $J + K = O$, where O is the point at infinity.
- (ii) If $K = J$, then $J + K = 2J$; then point doubling equations are used.

1.4. Point Doubling. Consider a point J such that $J = (x_1, y_1)$, where $y_1 \neq 0$ [3]. Let $L = 2J$ where $L = (x_2, y_2)$. Then

$$\begin{aligned} x_2 &= \lambda^2 - 2x_1 \pmod{p}, \\ y_2 &= \lambda(x_1 - x_2) - y_1 \pmod{p}, \end{aligned} \quad (3)$$

where $\lambda = ((3 \times 12 + a)/(2y_1)) \pmod{p}$ is the tangent at point J and a is one of the parameters chosen with the elliptic curve.

- (i) If $y_1 = 0$, then $2J = O$, where O is the point at infinity.

1.5. Domain Parameters for Elliptic Curve over $GF(p)$ Are (p, a, b, B, t) . p is the prime number defined for finite field $GF(p)$. $a, b \in GF(p)$ are the two coefficients defining the curve $y^2 \pmod{p} = (x^3 + ax + b) \pmod{p}$. B is the generator point (XB, YB) and t is the order of B .

1.6. Various Security Parameters. A proxy signature should have security properties [4] and they are as follows.

- (1) Strong unforgeability: proxy signatures can be created only by designated proxy signer. Original signer or any other party cannot generate 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 undeniability: once valid proxy signature is created by proxy signer, he/she cannot repudiate signature creation.
- (5) Distinguishability: proxy signatures are distinguishable from ordinary signature created by original signer.

1.7. Various Attacks

- (i) Public key substitution attack: in this attack, original signer can generate proxy multisignature by updating his own public key [7].
- (ii) Original signer's forgery attack: in this attack, original signers can generate valid proxy multisignature without agreement of proxy signer and verifier will be convinced that any proxy multisignature generated by using forged signing key is generated by agreement of all, original signer and proxy signer [8]. Under the name of proxy signer, original signer can forge valid proxy multisignature.
- (iii) Framing attack: in this attack, any user P is framed by malicious users $A_1, A_2, \dots, A_n$. User P does not receive any delegation from the users $A_1, A_2, \dots, A_n$, but the malicious users $A_1, A_2, \dots, A_n$ can forge a proxy multisignature for message m by user P on behalf of users $A_1, A_2, \dots, A_n$ [9].

FIGURE 1: Point addition [3].

FIGURE 2: Point doubling [3].

2. Review of Existing Schemes

Mambo et al. [4] define the different types of delegations like full delegation, partial delegation, and delegation by warrant. They have proposed a proxy signature scheme that is based on discrete logarithm problem. They have compared different proxy signature schemes like Schnorr scheme, Elgamal scheme, and Okamoto scheme on the basis of message length and amount of computational work. In all the above schemes, the amount of computational work in partial delegation is smaller than that with delegation by warrant. In partial delegation, a proxy signer can create proxy signature forever because valid period is not specified. In this case the original signer can revoke the signing capability of the proxy signer by two ways, that is, (1) to make revocation list publicly seen and (2) to change the public key of original signer and all proxies of honest proxy signers are updated accordingly.

Mambo et al. [10] proposed a new type of proxy signature scheme based on discrete logarithm problem. Scheme that is proposed by Mambo et al.'s [4] holds sufficient properties if original signer is trustworthy and never cheats. If the original signer is not trustworthy then proxy protected proxy signature scheme is important and they have introduced proxy protected proxy signature scheme based on discrete logarithm problem.

Kim et al. [11] have introduced two new types of proxy signature schemes based on discrete logarithm problem, that is, partial delegation with warrant and threshold delegation. Partial delegation with warrant combines the benefits of partial delegation and delegation with warrant. Valid period can be specified in proxy signature for partial delegation with warrant, so their new scheme does not require an additional proxy revocation protocol. In threshold delegation, the original signer delegates the power to sign message in such a way

that from the designated group of n proxy signers, t or more proxy signers can create signature and $t - 1$ or less proxy signers cannot create signature on behalf of original signer.

Yi et al. [12] proposed a new type of proxy signature scheme, that is, proxy multisignature scheme in which a proxy signer can create signature on behalf of two or more original signers. They give the overview of proxy monosignature schemes, that is, Mambo et al.'s [4] and Kim et al.'s [11]. They have introduced Mambo-like proxy multisignature scheme and Kim-like proxy multisignature scheme. Their schemes are proxy unprotected schemes; that is, original signer can also create proxy signature.

Sun [7] analyzes the proxy signature and proxy multisignature schemes and their analysis indicates that these schemes suffer from the public key substitution attack and direct forgery attack. They analyze Yi et al.'s [12] proxy multisignature schemes and shows that these schemes suffer from public key substitution attack (an original signer can forge proxy multisignature by updating his own public key) and direct forgery attack (one original signer can generate forged proxy multisignature on arbitrary message for multiple original signers). They proposed a new proxy protected and proxy unprotected proxy multisignature schemes which do not suffer from these attacks.

Lee et al. [13] provide new classifications of proxy signature scheme, that is, strong and weak proxy signature, designated and nondesignated proxy signature, and self-proxy signature. They proposed a strong nondesignated proxy signature scheme. The proposed scheme does not specify proxy signer so it can be applied to multiproxy signature in which multiple original signers can delegate their signing capabilities to proxy signers.

Chen et al. [14] proposed a new proxy protected proxy signature scheme which is based on elliptic curve discrete logarithm problem. They analyze the performance of Sun [7] and the proposed scheme on the basis of time complexity.

Chen et al. [1] proposed an improved scheme in which the exponential operations are changed into elliptic curve multiplicative ones. Sun [7] improvement increases security but requires complex operations to derive the proxy public key that is required to verify the proxy multisignature. ECC has lower computational overhead and a smaller key size than that of RSA or DSA and ECC can achieve a level of security equal to that of the RSA or DSA. This proposed scheme is called CCH1 scheme. They compared the Sun [7] and proposed proxy multisignature schemes. The time complexity of proposed scheme is reduced and performance is enhanced without loss of security.

Chen et al. [15] introduced a traceable proxy multisignature scheme. This scheme makes size of proxy signature independent of number of original signers, so computation overhead means none of operations required for verification is greatly reduced. This proposed scheme is called CCH2 scheme. They compare the Sun [7] and proposed proxy multisignature schemes on the basis of time complexity.

Hwang et al. [16] proposed a generalized version of the $(t_1/n_1 - t_2/n_2)$ proxy signature scheme based on elliptic curve discrete logarithm problem. In a generalized proxy signature scheme with known signers, any t_1 or more original signers

out of n_1 original signers ($1 \leq t_1 \leq n_1$) can represent the original group to delegate the signing capability, and t_2 or more proxy signers out of n_2 proxy signers ($1 \leq t_2 \leq n_2$) can represent the proxy group to sign message on behalf of the group of original signers. They have discussed special cases, namely, the $(t_1/n_1 - 1)$ proxy signature (proxy multisignature) scheme, $(1 - t_2/n_2)$ proxy signature scheme (multiproxy signature), and $(1 - 1)$ proxy signature scheme.

Wang et al. [17] present security analysis of some proxy signature schemes, that is, Mambo et al.'s [10] and Lee et al.'s [13]. By identifying several attacks, they show that all these schemes are insecure.

Wang et al. [2] review Chen et al.'s [14] proxy protected proxy signature scheme based on elliptic curve cryptosystem and they show that it is vulnerable to an original signer forgery attack. They present an improved scheme which is secure against the proposed attack.

Park et al. [8] show that proxy multisignature schemes proposed by Chen et al. [1, 15] are insecure against the malicious original signer(s). They review the CCH1 and CCH2 schemes and analyze their security. These schemes are vulnerable to proxy signing forgery attack by one or all original signers.

Chang et al. [18] proposed a proxy protected signature scheme based on ECDSA which satisfies security properties. They show that the time complexity of proxy signature is similar in both proxy signature based on ECDSA and ECDSA.

Li and Xue [19] have reviewed CCH1 and CCH2 proxy multisignature schemes based on elliptic curve cryptography. Park et al. [8] show that these schemes suffer from forgery attack by one or all original signers. They have proposed improved CCH1 and improved CCH2 schemes that do not suffer from forgery attack.

Tutanesu et al. [3] examine that ECC is more attractive cryptosystem than conventional cryptosystem (RSA/DSA) for mobile devices, which are limited in terms of their CPU, power, and network connectivity. ECC is fast and can be implemented with less hardware, because of shorter key length. They have presented the application of ECC, that is, internet, smart cards, PDAs, and PCs. Their opinion is that ECC could become the next generation of PKC.

Wang and Yu [20] have discussed two fatal flaws of the cryptosystem which are based on the logistic map and proposed by Wang and Xiang are pointed out. According to this, cryptanalysts could recover the plaintext by the chosen plaintext attacked in a short time. Authors proposed a remedial improvement which can avoid the flaws and enhance the security of the cryptosystem.

In this paper [21], have analyzed the security of a parallel keyed hash function based on chaotic neural network proposed by Wang and Zhao recently. Weak keys and forgery attacks against Wang and Zhao's scheme are demonstrated. Both theoretical analysis and experimental results show that the parallel keyed hash function is not security. Besides, some improvement measures are presented to enhance the security of the parallel keyed hash function.

Wang and He [22] have introduced a novel image encryption method based on a skew tent map that is proposed

recently. In this paper, some flaws of this algorithm are pointed out and then a chosen plaintext attack against it is presented. Both theoretical analysis and experimental simulation indicate that the plain image can be recovered exactly from the cipher image without the secret key. So it can be seen that this algorithm is not secure enough to be applied in network communication.

Wang and Liu [23] have cryptanalysis of a parallel subimage encryption method with high-dimensional chaos.

3. Estimation of Time, Space, and Computational Overhead of Improved CCH1 and CCH2 Schemes

3.1. Implementation of Improved CCH1 Proxy Multisignature Scheme. There are four phases—the system initialization phase, the key generation phase, the proxy multisignature generation phase, and the proxy multisignature verification phase [19].

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

- (i) a field size p , which is odd prime;
- (ii) two parameters $a, b \in F_p$ to define the equation of elliptic curve E over F_p (i.e., $y^2 = x^3 + ax + b \pmod{p}$), where $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$;
- (iii) a finite point $B = (x_B, y_B)$ whose order is a large prime number in $E(F_p)$, where B is a point in $E(F_p)$, where $B \neq O$, because O denotes an infinity point;
- (iv) 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.

- (i) 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 “ $\times$ ” indicates the multiplication of a number by an elliptic curve point.
- (ii) The proxy signer is provided with a private key $1 \leq d_p \leq t - 1$ and a corresponding public key $Q_p = d_p \times B = (x_{Q_p}, y_{Q_p})$. 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 $k_i \in \{1, 2, \dots, t - 1\}/d_i$ as secret key.

Step 2 (group commitment value generation). They then computes

$$R_i = k_i \times B = (x_{R_i}, y_{R_i}). \quad (4)$$

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

Step 3 (subdelegation 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:

$$s_i = d_i * x_{Q_i} * h(M_w, R_i) - k_i * x_{R_i} \pmod{t}, \quad (5)$$

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 subdelegation parameter for A_i is (M_w, R_i, s_i) .

Step 4 (subdelegation parameter verification). After the proxy signer has received the subdelegation parameters, then the proxy signer P computes

$$S_i \times B = (x_{Q_i} * h(M_w, R_i)) * Q_i - x_{R_i} R_i \quad (6)$$

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

Step 5 (proxy multisignature secret key generation). He then computes the proxy multisignature secret key as follows:

$$d = d_p * x_{Q_p} + \sum_{i=1}^n s_i \pmod{t}. \quad (7)$$

Phase 3. Proxy multisignature generation phase: the proxy multisignature affixed to the m is in the form of $(m, m_w, R, \text{Sig}_d(m))$, where $\text{Sig}_d(m)$ 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 chooses random number j where $1 \leq j \leq t - 1$ and calculates $J = j \times B = (J_x, J_y)$.

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

$$y = j - d * e \pmod{t}, \quad (8)$$

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

Phase 4. Proxy multisignature 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

$$Q = Q_P * x_{Q_P} + \sum_{i=1}^n (h(M_w, R_i) * Q_i - x_{R_i} \times R_i). \quad (9)$$

With the value, the verifier can confirm the validity of $\text{Sig}_d(m)$ 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 this equation satisfies then valid signature generated otherwise not.

3.2. Implementation of Improved CCH2 Proxy Multisignature Scheme. There are four phases—the system initialization phase, the key generation phase, the proxy multisignature generation phase, and the proxy multisignature verification phase [19].

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

- (i) a field size p , which is an odd prime;
- (ii) two parameters $a, b \in F_p$ to define the equation of elliptic curve E over F_p (i.e., $y^2 = x^3 + ax + b \pmod{p}$), where $4a^3 + 27b^2 \neq 0 \pmod{p}$;
- (iii) a finite point $B = (x_B, y_B)$ whose order is a large prime number in (F_p) , where B is a point in $E(F_p)$, where $B \neq O$, because O denotes an infinity point;
- (iv) 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.

- (i) 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 “ $\times$ ” indicates the multiplication of a number by an elliptic curve point.
- (ii) The proxy signer is provided with a private key $1 \leq d_p \leq t - 1$ and a corresponding public key $Q_p = d_p \times B = (x_{Q_p}, y_{Q_p})$. 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 $k_i \in \{1, 2, \dots, t - 1\}/d_i$ as secret key.

Step 2 (group commitment value generation). Then computes $R_i = k_i \times B = (x_{R_i}, y_{R_i})$. If $x_{R_i} = 0$, 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

$$i = (x_R, y_R). \quad (10)$$

Step 3 (subdelegation 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:

$$s_i = d_i * h(M_w, x_{Q_p}, x_{Q_i}, x_R) - k_i * x_R \pmod{t}, \quad (11)$$

where $h(\cdot)$ is a hash function and the warrant M_w contains information such as the IDs of all original signers and proxy signer. Then, the subdelegation parameter for A_i is (M_w, s_i) .

Step 4 (subdelegation parameter verification). After the proxy signer has received the subdelegation parameters then the proxy signer P computes

$$s_i * B = h(M_w, x_{Q_p}, x_{Q_i}, x_R) * Q_i - x_R \times R_i \quad (12)$$

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

Step 5 (proxy multisignature secret key generation). They then compute the proxy multisignature secret key as follows:

$$d = d_p + \sum_{i=1}^n s_i \pmod{t}. \quad (13)$$

Phase 3. Proxy multisignature generation phase: the proxy multisignature affixed to the m is in the form of $(m, m_w, R, \text{Sig}_d(m))$, where $\text{Sig}_d(m)$ 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 chooses random number j where $1 \leq j \leq t - 1$ and calculates $J = j \times B = (J_x, J_y)$.

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

$$y = j - d \times e \pmod{t}, \quad (14)$$

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

Phase 4. Proxy multisignature 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

$$Q = Q_P + \sum_{i=1}^n h(M_w, x_{Q_p}, x_{Q_i}, x_R) \times Q_i - R \times x_R. \quad (15)$$

FIGURE 3: Entropy for the proposed scheme.

TABLE 1: Compression ratio (in %) in each scheme.

Schemes	Compression ratio (in %)
CCH1	75
CCH2	89
The proposed scheme	75

With the value, the verifier can confirm the validity of $\text{Sig}_d(m)$ 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 this equation satisfies then valid signature generated otherwise not.

4. Performance Analysis of the Proposed Scheme

The analysis reports of the proposed scheme are given below.

4.1. Entropy. In this case, the value of entropy is the measure of the tendency of a process, to be entropically *favoured*, or to proceed in a particular direction. Moreover, entropy provides an indication for a specific encryption method. We have analyzed our hypothesis on the basis of entropy generated [24].

Figure 3 shows the entropy for the proposed scheme. The Figure 4 shows that compression ratio required in each scheme. Table 1 lists the name and compression ratio required in each scheme.

4.2. Floating Frequencies/Intuitive Synthesis. Floating frequencies/intuitive synthesis in its completed three part entirety which takes full advantage of the time complexity, space complexity, and communication overhead provided by the digital medium. We have calculated floating frequency of threshold proxy signature scheme [24]. Figure 5 shows floating frequencies/intuitive synthesis for the proposed scheme.

4.3. ASCII Histogram. The ASCII histogram proved to be very useful since it helped enormously in debugging code involving probability calculations with simple print statements. Probabilistic simulations are extremely hard to test

FIGURE 4: Radar chart showing compression ratio required in each scheme.

FIGURE 5: Floating frequencies/intuitive synthesis for the proposed scheme.

because the results of a given operation are never strictly the same. However, they should have the same probability distribution, so by looking at the rough shape of the histogram, you tell if your calculations are going in the right direction. In this context, we have calculated ASCII histogram for our threshold proxy signature scheme [24]. Figure 6 shows ASCII histogram for the proposed scheme.

4.4. Autocorrelation. A mathematical representation of the degree of similarity between a given time series and a lagged version of itself over successive time intervals. It is the same as calculating the correlation between two different time series, except that the same time series is used twice—once in its original form and once lagged one or more time periods. The term can also be referred to as “lagged correlation” or “serial correlation.” In this, we have calculated autocorrelation for threshold proxy signature scheme [24]. Figure 7 shows autocorrelation for the proposed scheme.

TABLE 2

Number	Substring	CCH1 Frequency (in %)	Frequency1 Frequency	CCH2 Frequency (in %)	Frequency2 Frequency	Proposed technology Frequency (in %)	Frequency3 Frequency
1	N	10.9477	201	10.9477	201	10.9477	201
2	I	10.512	193	10.512	193	10.512	193
3	T	9.3137	171	9.3137	171	9.3137	171
4	E	8.5512	157	8.5512	157	8.5512	157
5	S	6.9172	127	6.9172	127	6.9172	127
6	R	5.3377	98	5.3377	98	5.3377	98
7	A	4.3028	79	4.3028	79	4.3028	79
8	O	4.085	75	4.085	75	4.085	75
9	C	3.8126	70	3.8126	70	3.8126	70
10	D	3.8126	70	3.8126	70	3.8126	70
11	U	3.5403	65	3.5403	65	3.5403	65
12	F	3.268	60	3.268	60	3.268	60
13	P	3.268	60	3.268	60	3.268	60
14	G	3.2135	59	3.2135	59	3.2135	59
15	L	3.2135	59	3.2135	59	3.2135	59
16	H	3.159	58	3.159	58	3.159	58
17	Y	2.9412	54	2.9412	54	2.9412	54
18	M	2.0697	38	2.0697	38	2.0697	38
19	X	1.9063	35	1.9063	35	1.9063	35
20	V	1.6885	31	1.6885	31	1.6885	31
21	W	1.3617	25	1.3617	25	1.3617	25
22	J	0.817	15	0.817	15	0.817	15
23	B	0.817	15	0.817	15	0.817	15
24	K	0.5447	10	0.5447	10	0.5447	10
25	Q	0.4902	9	0.4902	9	0.4902	9
26	Z	0.1089	2	0.1089	2	0.1089	2

FIGURE 6: ASCII histogram for the proposed scheme.

FIGURE 7: Autocorrelation for the proposed scheme.

4.5. Histogram Analysis. A histogram is a graphical representation showing a visual impression of the distribution of data. We have analyzed histogram for all schemes. Table 2 lists the histogram analysis for overall threshold proxy signature schemes [24]. Figure 8 shows radar chart showing overall analysis for all schemes.

5. Graphical Representation of Time, Space, and Computational Overhead of Improved CCH1 Scheme

5.1. Time Complexity, Space Complexity, and Computational Overhead. When determining the time complexity of an algorithm, we measure how fast the computing requirements grow as the size of the input grows. We generate graphs to analyze the time complexity of the schemes. The space complexity of a program is the number of elementary objects that this program needs to store during its execution. We generate graphs to analyze the space complexity of the schemes. The computational overhead includes two types of communication in the schemes: number of transmissions and number of broadcasts. We generate graphs to analyze the computational overhead of the schemes [24]. Figures 9, 10, 11, 12, 13, 14, 15 and 16 show time complexity, space

FIGURE 8: Radar chart showing overall analysis for all schemes.

FIGURE 9: Time complexity of improved CCH1 scheme with varying value of field size (p). FIGURE 10: Time complexity of improved CCH2 scheme with varying value of field size (p).

complexity, and computational overhead for the CCH1 and CCH2 schemes.

6. Cryptanalysis of Improved CCH2 Proxy Multisignature Scheme

See Figures 15, 16, 17, 18, 19, 20, 21, 22, and 23.

In improved CCH2 proxy multisignature scheme, elliptic curve based Schnorr signature scheme is used to generate and verify the signature.

6.1. Forge a Proxy Multisignature. Suppose proxy signer P signed a message with his private key d_p ; the signature is (m, m_w, R, e, y'') where $e = h(m, j_x)$ and $y'' = j - d_p *$

FIGURE 11: Space complexity of improved CCH1 scheme with varying value of field size (p).

FIGURE 12: Space complexity of improved CCH2 scheme with varying value of field size (p).

FIGURE 14: Computational overhead of improved CCH2 scheme with varying value of field size (p).

FIGURE 13: Computational overhead of improved CCH1 scheme with varying value of field size (p).

FIGURE 15: Time complexity of improved CCH1 scheme with varying value of field size (p).

$e \pmod{t}$. Upon receiving the signature (m, m_w, R, e, y'') the malicious original signers $A_1, \dots, A_n$ can forge a valid proxy signature as follows.

(i) Compute $y' = \sum_{i=1}^n s_i * e \pmod{t}$.

(ii) Compute $y = y'' - y'$.

Finally, the malicious original signers $A_1, \dots, A_n$ can forge a valid proxy signature (m, m_w, R, e, y) . The following shows why the proxy signature (m, m_w, R, e, y) is valid.

Proof. Consider the following:

$$\begin{aligned}
 y &= y'' - y' \\
 &= j - d_p * e \pmod{t} - \sum_{i=1}^n s_i * e \pmod{t} \\
 &= j - \left(d_p + \sum_{i=1}^n s_i \right) e \pmod{t}, \\
 y &= j - d * e \pmod{t}.
 \end{aligned} \tag{16}$$

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

$$Q = Q_P + \sum_{i=1}^n h(M_w, x_{Q_p}, x_{Q_i}, x_R) \times Q_i - R \times x_R. \tag{17}$$

With the value, the verifier can confirm the validity of $\text{Sig}_d(m)$ 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 this equation satisfies then valid signature generated otherwise not. $\square$

6.2. Forge the Proxy Signer's Signature. After getting signature (m, m_w, R, e, y) where $e = h(m, j_x)$ and $y = j - d * e \pmod{t}$, the original signer $A_1, \dots, A_n$ can forge proxy signer P 's signature on message m as follows.

(i) Each A_i computes $s_i * e$ where $i = 1, 2, \dots, n$.

(ii) Compute $y' = \sum_{i=1}^n s_i * e \pmod{t}$.

(iii) Compute $y'' = y + y'$.

(iv) (m, m_w, R, e, y'') is valid signature on message m .

FIGURE 16: Time complexity of improved CCH2 scheme with varying value of field size (p).

FIGURE 18: Space complexity of improved CCH1 scheme with varying value of field size (p).

FIGURE 17: Time complexity of enhanced scheme with varying value of field size (p).

FIGURE 19: Space complexity of improved CCH2 scheme with varying value of field size (p).

The malicious original signers can forge a valid signature (m, m_w, R, e, y'') on message m with respect to proxy signer P 's private key d_p .

The following shows why the signature (m, m_w, R, e, y'') is valid.

Proof. Consider the following:

$$\begin{aligned}
 y'' &= y + y' \\
 &= j - d * e + \sum_{i=1}^n s_i * e \pmod{t} \\
 &= j - \left(d_p + \sum_{i=1}^n s_i \right) e + \sum_{i=1}^n s_i * e \pmod{t}, \\
 y'' &= j - d_p * e \pmod{t}.
 \end{aligned} \tag{18}$$

Proxy multisignature verification: when the verifier verifies the signature, he or she uses proxy public value Q_p corresponding to the proxy signature key d_p . With the value,

the verifier can confirm the validity of $\text{Sig}_d(m)$ by validating the verification equality of the designated signature scheme.

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

Step 2. And compute $e' = h(\bar{J}_x, m)$. Then check that $e' = e$ and if this equation satisfies then valid signature generated otherwise not. $\square$

6.3. Framing Attack. In this attack, malicious users $A_1, A_2, \dots, A_n$ also can forge a proxy multisignature for message m by some user P on behalf of users $A_1, A_2, \dots, A_n$, such that user P was never designated by users $A_1, A_2, \dots, A_n$. Suppose proxy signer P signed a message with his private key d_p ; the signature is (m, m_w, R, e, y'') where $e = h(m, j_x)$ and $y'' = j - d_p * e \pmod{t}$.

Upon receiving the signature (m, m_w, R, e, y'') , then the malicious original signers $A_1, \dots, A_n$ can forge a valid proxy signature as follows.

- (i) The malicious users $A_1, A_2, \dots, A_n$ pretend to produce a forged warrant m_w , which records the delegation information such as identities of the malicious users $A_1, A_2, \dots, A_n$ and user P .

FIGURE 20: Space complexity of enhanced scheme with varying value of field size (p).

FIGURE 21: Computational overhead of improved CCH1 scheme with varying value of field size (p).

(ii) 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 $R_i = k_i \times B = (x_{R_i}, y_{R_i})$ and broadcasts R_i to other users.

(iii) On receiving R_j ($1 \leq j \leq n, j \neq i$), A_i calculates

$$R = \sum_{i=1}^n R_i = (x_R, y_R), \quad (19)$$

$$s_i = d_i * h(M_w, x_{Q_p}, x_{Q_i}, x_R) - k_i * x_R - (\text{mod } t).$$

Note that user P does not receive any information from the malicious users $A_1, A_2, \dots, A_n$.

(i) Compute $y' = \sum_{i=1}^n s_i * e(\text{mod } t)$;

(ii) Compute $y = y'' - y'$.

Finally the malicious users can forge a valid signature (m, m_w, R, e, y) on message m by some user P on behalf of users $A_1, \dots, A_n$, such that user P was never designated by users $A_1, \dots, A_n$. The following shows why the signature (m, m_w, R, e, y) is valid.

FIGURE 22: Computational overhead of improved CCH2 scheme with varying value of field size (p).

FIGURE 23: Computational overhead of enhanced scheme with varying value of field size (p).

Proof. Consider the following:

$$\begin{aligned} y &= y'' - y' \\ &= j - d_p * e(\text{mod } t) - \sum_{i=1}^n s_i * e(\text{mod } t) \\ &= j - \left(d_p + \sum_{i=1}^n s_i \right) e(\text{mod } t), \\ y &= j - d * e(\text{mod } t). \end{aligned} \quad (20)$$

From above we can see that an innocent user P is framed by the malicious users $A_1, \dots, A_n$.

Proxy multisignature 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

$$Q = Q_p + \sum_{i=1}^n h(M_w, x_{Q_p}, x_{Q_i}, x_R) \times Q_i - R \times x_R. \quad (21)$$

With the value, the verifier can confirm the validity of $\text{Sig}_d(m)$ 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 this equation satisfies then valid signature generated otherwise not. $\square$

7. Conclusion

In this paper, we have discussed implementation of improved CCH1 and improved CCH2 proxy multisignature scheme based on elliptic curve cryptosystem. We have represented time complexity, space complexity, and computational overhead of improved CCH1 and CCH2 proxy multisignature schemes. We have presented cryptanalysis of improved CCH2 proxy multisignature scheme and showed that improved CCH2 scheme suffers from various attacks, that is, forgery attack and framing attack.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors also wish to thank many anonymous referees for their suggestions to improve this paper.

References

- [1] T. S. Chen, Y. F. Chung, and G. S. Huang, "Efficient proxy multisignature schemes based on the elliptic curve cryptosystem," *Computers and Security*, vol. 22, no. 6, pp. 527–534, 2003.
- [2] S. Wang, G. Wangt, F. Bao, and J. Wang, "Cryptanalysis of a proxy-protected proxy signature scheme based on elliptic curve cryptosystem," in *Proceedings of the IEEE 60th Vehicular Technology Conference, VTC2004-Fall: Wireless Technologies for Global Security*, pp. 3240–3243, September 2004.
- [3] I. Tutanescu, C. Anton, L. Ionescu, and D. Caragata, "Elliptic curves cryptosystems approaches," in *Proceedings of the International Conference on Information Society (i-Society '12)*, pp. 357–362, 2012.
- [4] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, pp. 48–57, March 1996.
- [5] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology-Crypto '85*, vol. 218 of *Lecture Notes in Computer Science*, pp. 417–426, 1986.
- [6] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.
- [7] H. M. Sun, "On proxy multisignature schemes," in *Proceedings of the International Computer Symposium*, pp. 65–72, 2000.
- [8] J. H. Park, B. G. Kang, and S. Park, *Cryptanalysis of Some Group-Oriented Proxy Signature Schemes*, vol. 3786 of *Lecture Notes in Computer Science*, 2006.
- [9] C. Feng and C. Zhenfu, "Cryptanalysis on a proxy multi-signature scheme," in *Proceedings of the 1st International Multi-Symposiums on Computer and Computational Sciences (IMSCCS '06)*, vol. 2, pp. 117–120, IEEE, April 2006.
- [10] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: delegation of the power to sign messages," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E79-A, no. 9, pp. 1338–1353, 1996.
- [11] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited," in *Proceedings of the Information and Communications Security (ICICS '97)*, vol. 1334 of *Lecture Notes in Computer Science*, pp. 223–232, 1997.
- [12] L. Yi, G. Bai, and G. Xiao, "Proxy multi-signature scheme: a new type of proxy signature scheme," *Electronics Letters*, vol. 36, no. 6, pp. 527–528, 2000.
- [13] B. Lee, H. Kim, and K. Kim, "Strong proxy signature and its application," in *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics (SCIS '01)*, pp. 603–608, 2001.
- [14] T. S. Chen, T. P. Liu, and Y. F. Chung, "A proxy-protected proxy signature scheme based on elliptic curve cryptosystem," in *Proceedings of the IEEE TENCOM Region 10 Conference on Computers, Communications, Control and Power Engineering*, pp. 184–187, October 2002.
- [15] T. S. Chen, Y. F. Chung, and G. S. Huang, "A traceable proxy multisignature scheme based on the elliptic curve cryptosystem," *Applied Mathematics and Computation*, vol. 159, no. 1, pp. 137–145, 2004.
- [16] M. S. Hwang, S. F. Tzeng, and C. S. Tsai, "Generalization of proxy signature based on elliptic curves," *Computer Standards and Interfaces*, vol. 26, no. 2, pp. 73–84, 2004.
- [17] G. L. Wang, F. Bao, J. Y. Zhou, and R. H. Deng, "Security analysis of some proxy signatures," in *Information Security and Cryptology*, vol. 2971 of *Lecture Notes in Computer Science*, pp. 305–319, 2004.
- [18] M. H. Chang, I. T. Chen, and M. T. Chen, "Design of proxy signature in ECDSA," in *Proceedings of the 8th International Conference on Intelligent Systems Design and Applications (ISDA '08)*, pp. 17–22, November 2008.
- [19] F. Li and Q. Xue, "Two improved proxy multi-signature schemes based on the elliptic curve cryptosystem," *Communications in Computer and Information Science*, vol. 234, no. 4, pp. 101–109, 2011.
- [20] X. Wang and C. Yu, "Cryptanalysis and improvement on a cryptosystem based on a chaotic map," *Computers and Mathematics with Applications*, vol. 57, no. 3, pp. 476–482, 2009.
- [21] X. Wang and J. Zhao, "Cryptanalysis on a parallel keyed hash function based on chaotic neural network," *Neurocomputing*, vol. 73, no. 16–18, pp. 3224–3228, 2010.
- [22] X. Wang and G. He, "Cryptanalysis on a novel image encryption method based on total shuffling scheme," *Optics Communications*, vol. 284, no. 24, pp. 5804–5807, 2011.
- [23] X. Wang and L. Liu, "Cryptanalysis of a parallel sub-image encryption method with high-dimensional chaos," *Nonlinear Dynamics*, vol. 73, no. 1–2, pp. 795–800, 2013.
- [24] R. Kumar, H. K. Verma, and R. Dhir, "Cryptanalysis and performance evaluation of enhanced threshold proxy signature scheme based on RSA for known signers," *Mathematical Problems in Engineering*, vol. 2013, Article ID 790257, 24 pages, 2013.