Threshold Signature Scheme with Strong Forward Security Based on Chinese Remainder Theorem

Ya-ge Cheng[1], Zhi-juan Jia[1(✉)], Bei Gong[2], Li-peng Wang[1], and Yan-fang Lei[1]

[1] College of Information Science and Technology, Zhengzhou Normal University, Zhengzhou 450044, China
897373693@qq.com, 13676951984@163.com
[2] College of Computer Sciences, Beijing University of Technology, Beijing 100124, China

Abstract. The traditional cryptosystem is based on the security of private key. While the private key is leaked, the signature information may be exposed. Based on this, a threshold signature scheme with strong forward security based on Chinese remainder theorem is proposed. The signature is generated through the cooperation of members, which solve the problem of authoritative fraud introduced by the dealer. The private key is updated periodically to handle the threat caused by the private key leakage. Security analysis shows that the existing signatures will not be affected by the compromise of the corresponding private keys, and do not allow for forgery of the future signatures, which shows that the new scheme has the forward security and the backward security. The efficiency analysis shows that our scheme is more efficient compared with the well-known existing schemes in the literature.

Keywords: Strong forward security · Threshold signature · Chinese remainder theorem · Secret sharing

1 Introduction

In the era of explosive development of the Internet today, while it brings convenience to people, it also faced the problems such as privacy leaked and information tampering. The rapid development of the network has promoted the widespread application of digital signature technology, however, the biggest challenge of digital signature technology is the leakage of the private key, which make the information seriously inaccurate, in this context, the idea of forward security came into being.

In 1997, Anderson [1] first proposed the concept of forward security at the cryptography conference in Europe. The core idea was the update of the key. Then Bellare and Miner [2] proposed forward theory based on One-Schnorr and Fiat-Shamir's schemes in 1999, in which implemented a forward-secure digital signature scheme for the first time. In 2000, Anderson [3] summarized the forward security scheme and

J. Li et al. (Eds.): SPNCE 2019, LNICST 284, pp. 15–28, 2019.
https://doi.org/10.1007/978-3-030-21373-2_2

proposed two security: forward safety and backward safety. In 2001, Mike Burmester et al. proposed a strong forward security definition [4], it means a signature system will not affect the previous and subsequent signatures when the current key is compromised. Its proposal greatly improves the efficiency of the signature.

The literature [5] based on the zero-knowledge proof proposed a forward-backward secure digital signature scheme. Literature [6] proposed a forward-backward security digital signature based on the strong RSA hypothesis. Literature [7] proposed a two-way secure signature scheme based on discrete logarithm problem. Literature [8] proposed a proxy signature scheme with strong forward security based on the ElGamal scheme. The literature [9] proposed forward and backward security group signature scheme based on Lagrangian difference polynomial. In literature [10] the dual key is introduced on the basis of Guillou-Quisquater signature system and Rabin cryptosystem, proposed a strong forward-secure digital signature scheme. The literature [11] based on the bilinear pairing algorithm proposed a verifiable strong forward secure ring signature scheme, both in signature and verification process requires bilinear pairing calculation, which makes the signature efficiency lower. All of the solutions above have strong forward security but are inefficient.

In [12], a signature scheme with forward security based on the Chinese remainder theorem was proposed. In [13], given a subgroup signature scheme. In [14], Tang proposed a group blind signature scheme. A group signature scheme was proposed in [15]. The signature schemes above were all based on the Chinese remainder theorem, all of which have forward security but no backward security.

Based on the above researches, a threshold signature scheme based on the Chinese remainder theorem with strong forward security is proposed. The scheme does not require a trusted center and solves the problem of authoritative fraud in the trusted center. Through cooperation, the partial signatures synthesized the final signature. It also supports the members' private keys updated periodically to ensure strong forward security of the signature system.

2 Prerequisite Knowledge

2.1 Forward Security Theory

The forward security theory [2] means the entire signature time is divided into cycles and the public key remains unchanged throughout the signature time, but the member private key is continuously updated as the signature cycle progresses. In each cycle, signatures are generated by using the member's current private keys. When a member's private key is leaked in a certain period, due to the update of the private key, the malicious attacker cannot forge the signature information before the period, so the signatures before the current period are secure.

The implementation of forward security theory is as follows:

1. Divide the validity period of the signature into T periods;
2. The public key remains unchanged throughout the signature time, and the private key is dynamically updated as time passes;

3. In j cycle, member P_i counts $SK_{ij} = h(SK_{i(j-1)})$, where h is a one-way function;
4. P_i deletes $SK_{i(j-1)}$ immediately after calculating $SK_{i(j-1)}$. Thus, even if an attacker obtains the j cycle's private key of P_i, it cannot obtain any information of the private keys before the period. The update of private key is shown as follows (Fig. 1).

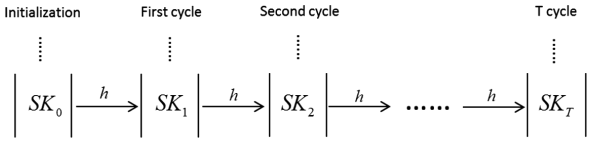


Fig. 1. Schematic diagram of private key update

2.2 Strong Forward Security

Strong forward security means that if the current key of a signature system is given away, it will not have any effect on the signature before and after the current period. It mainly includes two aspects of security:

1. Forward security: refers to that the key's leak of the current period will have no effect on the signature information before this;
2. Backward security: refers to that the key's leak of the current period will have no effect on the signature that will to be generated.

2.3 Asmuth-Bloom Secret Sharing Scheme

The Asmuth-Bloom [16] secret sharing scheme was proposed by Asmuth and Bloom in 1983. Its mainly includes three steps:

1. Initialize

Suppose DC is a secret distributor, $P = \{P_1, P_2, \cdots, P_n\}$ is a collection of n members, the threshold is t and the secret is S. The DC selects a large prime $q(q > S)$, A is an integer, $d = \{d_1, d_2, \cdots, d_n\}$ is a strictly increasing sequence of positive integers, and d satisfies the following conditions:

(1) $0 \le A \le M/q - 1$;
(2) $d_1 < d_2 < \cdots < d_n$;
(3) $gcd(d_i, d_j) = 1, (i \ne j)$;
(4) $gcd(d_i, q) = 1, (i = 1, 2, \cdots, n)$;
(5) $M = \prod_{i=1}^{t} d_i > q \prod_{i=1}^{t-1} d_{n-t+1}$.

2. Secret distribution

DC calculation $z = S + Aq$ and $z_i = zmodd_i, (i = 1, 2, \cdots, n)$, send (z_i, d_i) to $P_i(i = 1, 2, \cdots, n)$ as a secret share of P_i.

3. Secret recovery

Any t members can recover secrets. After exchanging secrets between members, any member can establish the following congruence equations:

$$z \equiv z_i (\text{mod } d_i)$$

According to the Chinese remainder theorem, the congruence equation has a unique solution:

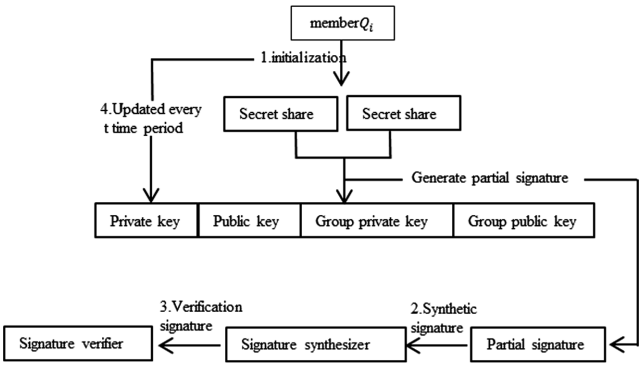
$$z = \sum_{i=1}^{t} \frac{D}{d_i} e_i X_i modD, (i = 1, 2, \cdots, t)$$

So, we can find $S = z - Aq$.

3 The Proposed Scheme

Based on the Chinese remainder theorem, this paper proposes a dynamic threshold signature scheme with strong forward security. This solution does not require a dealer and the member private keys' are updated regularly, which keeping the group public key unchanged, to ensure the scheme strong forward security. The architecture diagram of the scheme is shown as follows (Table 1):

Table 1. signature scheme architecture diagram



The solution consists of four steps: initialization, signature generation, signature verification and the updated of private key. The initialization phase generates a secret share, calculates verification information, and generates member keys and group keys. The signature compositor combines the partial signature into the final signature and it is verified by the signature verifier. The member private key is updated every t time periods.

3.1 System Initialization

$Q = \{Q_1, Q_2, \cdots, Q_n\}$ is a collection of n members, p and q is two large prime numbers that satisfy $q/p - 1$, $d = \{d_1, d_2, \cdots, d_n\}$ is a set of strictly monotonically increasing positive integer sequences which satisfies the Asmuth-Bloom secret sharing scheme, t is the threshold, g is the generator element on the finite field, M is the message to be signed, $N = \prod_{i=1}^{t} d_i$ is the product of the t smallest d_i.

1. Generate secret shares:

(1) Q_i selects α_i^0 and N_i^0 randomly to satisfy the following conditions:

$$0 < \alpha_i^0 < [q/n] \tag{1}$$

$$0 < N_i^0 < [N/q^2 - 1]/n \tag{2}$$

(2) Q_i calculates the verification information ω_i^0 and φ_{ij}^0:

$$\omega_i^0 = g^{(\alpha_i^0 + N_i^0 q)} modp \tag{3}$$

$$\tau_{ij}^0 = (\alpha_i^0 + N_i^0 q - L_{ij}^0)/d_j \tag{4}$$

$$\varphi_{ij}^0 = g^{\tau_{ij}^0} modp \tag{5}$$

broadcast ω_i^0, φ_{ij}^0.

(3) Q_i calculates secret shares for other members:

$$L_{ij}^0 = (\alpha_i^0 + N_i^0 q) modd_j \tag{6}$$

retains L_{ii}^0, broadcasts $g^{\alpha_i^0}$, $g^{N_i^0}$ and sends L_{ij}^0 to Q_j.

2. Generate members' private keys

Q_j verifies the correctness of the message from Q_i through (7, 8)

$$g^{\alpha_i^0} \cdot g^{N_i^0 q} modp = \omega_i^0 \tag{7}$$

$$((g^{L_{ij}^0} modp)\ ((\varphi_{ij}^0)^{d_j} modp)) modp = \omega_i^0 \tag{8}$$

If they are right, then Q_j calculates his private key:

$$H_j^0 = \sum_{i=1}^{n} L_{ij}^0 modd_j \tag{9}$$

So the member's personal public key is:

$$C = g^{H_j^0} \tag{10}$$

3. Generate a group key:

According to the sub-secrets α_i^0 selected by each member, the group public key is:

$$PK = \prod_{i=1}^{n} g^{\alpha_i^0} modp \tag{11}$$

Then the group private key is:

$$SK = \sum_{i=1}^{n} \alpha_i^0 \tag{12}$$

3.2 Generate Signature

1. Q_i selects a random number $x_i \in Z_p$ and calculates:

$$z_i = g^{x_i} modp \tag{13}$$

broadcasts g^{x_i}.

2. After Q_j receives z_i, it calculates:

$$z = g^{\sum_{i=1}^{t} x_i} modp = \prod_{i=1}^{t} g^{x_i} \mathrm{modp} = \prod_{i=1}^{t} z_i modp \tag{14}$$

3. Q_i calculates:

$$V_i^0 = \frac{D}{d_i} e_i H_i^0 modD \tag{15}$$

4. Q_i calculates part of the signature R_i^0:

$$R_i^0 = M \cdot z \cdot x_i + V_i^0 modD \tag{16}$$

then, sends the partial signatures (M, z, R_i^0) to the signature compositor.

5. After the signature compositor receives the partial signature of the t members, synthesize them

$$R = \left(\sum_{i=1}^{n} R_i^0 modD \right) modq \tag{17}$$

so the signature of the M is (M, z, R)

3.3 Verify Signature

When the certifier receives the signature of M, it verifies the signature:

$$g^R \equiv z^{M \cdot z} \cdot PKmodp \tag{18}$$

If the equation is true, the signature (M, z, R) of the M is valid.

3.4 Private Keys Update

The update of private keys can prevent attacks effectively. Assume that the update cycle is T, the detailed update algorithm is shown as follows

1. Q_i selects a random number N_i^T to satisfy the initial conditions.
2. Q_i calculates the update factors:

$$L_{ij}^T = L_{ij}^{(T-2)} + N_i^T q mod d_j \tag{19}$$

sends it to Q_j, broadcasts $g^{L_{ij}^{(T-2)}}$ and $g^{N_i^T}$;

3. Q_i calculates verification information ω_i^T and φ_{ij}^T.

$$\begin{aligned} \omega_i^T &= g^{L_{ij}^{(T-2)} + N_i^T q} modp; \\ \tau_{ij}^T &= (L_{ij}^{(T-2)} + N_i^T q - L_{ij}^T)/d_j; \\ \varphi_{ij}^T &= g^{\tau_{ij}^T} modp; \end{aligned}$$

and broadcasts them.

4. When Q_j received the messages L_{ij}^T, ω_i^T and φ_{ij}^T, verifies the correctness through the following two equations:

$$g^{L_{ij}^{(T-2)}} \cdot (g^{N_i^T})^q modp = \omega_i^T \tag{20}$$

$$((g^{L_{ij}^T} modp)\ ((\varphi_{ij}^T)^{d_j} modp)) modp = \omega_i^T \tag{21}$$

5. If Q_j has a private key during T-2 is $H_j^{(T-2)}$, then the private key for the T period after update is:

$$H_j^T = H_j^{(T-2)} + \sum_{i=1}^{n} L_{ij}^T modd_j \tag{22}$$

The new private key can still be used for signature and verification.

4 Analysis of the Proposed

4.1 Correctness Analysis

Theorem 1. The signature generated by the updated private key is valid. That is to prove that the (18) formula is established.

Prove:

$$\begin{aligned}
H_j^T &= H_j^{(T-2)} + \sum_{i=1}^{n} L_{ij}^T modd_j \\
&= H_j^{(T-3)} + \sum_{i=1}^{n} L_{ij}^{(T-2)} modd_j + \sum_{i=1}^{n} L_{ij}^T modd_j = \ldots \\
&= H_j^0 + \sum_{i=1}^{n} L_{ij}^0 modd_j + \ldots + \sum_{i=1}^{n} L_{ij}^T modd_j \\
&= H_j^0 + \sum_{i=1}^{n} \left(\sum_{r=1}^{T} L_{ij}^r \right) modd_j \\
&= \sum_{i=1}^{n} \left(\alpha_i^0 + N_i^0 q \right) + \sum_{i=1}^{n} \left[\sum_{r=1}^{T} L_{ij}^{(T-2)} + N_i^T \right] modd_j \\
&= \sum_{i=1}^{n} \left(\alpha_i^0 + \sum_{r=1}^{T} N_i^r q \right) + \sum_{i=1}^{n} \sum_{r=1}^{T-1} L_{ij}^{(T-2)} modd_j \\
&= \sum_{i=1}^{n} \left(\alpha_i^0 + \sum_{r=1}^{T} N_i^r q \right) + \sum_{i=1}^{n} \sum_{r=1}^{T-2} \left(\alpha_i^0 + N_i^r q \right) modd_j \\
&= 2 \sum_{i=1}^{n} \left(\alpha_i^0 + \sum_{r=1}^{T-2} N_i^r q \right) + N_i^T q modd_j; (j = 1, 2, \cdots, n)
\end{aligned}$$

make

$$G^T = \frac{1}{2} \sum_{i=1}^{n} \left(\alpha_i^0 + \sum_{r=1}^{T} N_i^r q \right) + \sum_{i=1}^{n} \sum_{r=1}^{T-2} \left(\alpha_i^0 + N_i^r q \right) modd_j \tag{23}$$

Then

$$H_j^T = 2G^T mod d_j,\ (j = 1, 2, \cdots, n).$$

Solving the congruence equations according to the Chinese remainder theorem:

$$\begin{cases} H_1^T \equiv 2G^T mod d_1 \\ H_2^T \equiv 2G^T mod d_2 \\ \qquad \vdots \\ H_t^T \equiv 2G^T mod d_t \end{cases}$$

Get a unique solution:

$$G^T = \frac{1}{2}\sum_{i=1}^{t} \frac{D}{d_i} e_i H_i^T mod D$$

Make

$$V_i^T = \frac{D}{d_i} e_i H_i^T mod D$$

Then

$$G^T = \frac{1}{2}\sum_{i=1}^{t} V_i^T mod D,$$

It can be known from the formulas (1), (2) and (19):

$$\begin{aligned} G^T &= \frac{1}{2}\left[\sum_{i=1}^{n}\left(\alpha_i^0 + \sum_{r=1}^{T} N_i^r q\right) + \sum_{i=1}^{n}\sum_{r=1}^{T-2}\left(\alpha_i^r + N_i^r q\right)\right] \\ &\le \frac{1}{2}\left\{\sum_{i=1}^{n}\left(\alpha_i^0 + q\cdot\left[\frac{N}{2q^2} - 1\right]/n\right) + \sum_{i=1}^{n}\sum_{r=1}^{T-2}\left(\alpha_i^0 + q\cdot\left[\frac{N}{2q^2} - 1\right]/n\right)\right\} \\ &\le \frac{1}{2}\left[n\cdot\left(\frac{q}{n} + q\cdot\left[\frac{N}{2q^2} - 1\right]/n\right)\right) + n\cdot\left(\frac{q}{n} + q\cdot\left[\frac{N}{2q^2} - 1\right]/n\right)\right] \\ &\le \frac{1}{2}\cdot 2\cdot n\cdot\left\{\frac{q}{n} + q\cdot\left[\frac{N}{q^2} - 1\right]/n\right\} \\ &\le \left\{q + q\cdot\left[\frac{N}{q^2} - 1\right]\right\} \\ &\le \frac{N}{q} \end{aligned}$$

According to the literature [17], when t > 2

$$M \cdot z \cdot \sum_{i}^{t} x_i + G^T \leq D$$

According to formula (16, 17)

$$\begin{aligned} R &= \left(\sum_{i=1}^{t} R_i^0 modD \right) modq \\ &= \left(\sum_{i=1}^{t} M \cdot z \cdot x_i + K_i^0 modD \right) modq \\ &= \left[\left(M \cdot z \cdot \sum_{i=1}^{t} x_i + G^T \right) modD \right] modq \\ &= \left[\left(M \cdot z \cdot \sum_{i=1}^{t} x_i + G^T \right) \right] modq \end{aligned}$$

According to Eq. (23)

$$\begin{aligned} G^T &= \frac{1}{2} \left[\sum_{i=1}^{n} \left(\alpha_i^0 + \sum_{r=1}^{T} N_i^r q \right) + \sum_{i=1}^{n} \sum_{r=1}^{T-2} (\alpha_i^0 + N_i^r q) modd_j \right] \\ &= \sum_{i=1}^{n} \alpha_i^0 modq \end{aligned}$$

So have

$$\begin{aligned} R &= [(M \cdot z \cdot \sum_{i=1}^{t} x_i + \sum_{i=1}^{t} \alpha_i^0)] modq \\ g^R &\equiv g^{[(M \cdot z \cdot \sum_{i=1}^{t} x_i + \sum_{i=1}^{n} \alpha_i^0)] modq} \\ &\equiv z^{M \cdot z} \cdot PKmodp \end{aligned}$$

Equation (18) is established, so the signature is valid.

4.2 Forward Security Analysis

If a member's private key is leaked in a certain period, no one else can falsify the signatures before it.

Suppose an attacker has stolen the personal private key H_j^T of the member Q_j of the T period, and the attacker wants to calculate $H_j^{(T-1)}$, then the attacker must calculate $\sum_{i=1}^{n} L_{ij}^T mod d_j$, and

$$\begin{aligned} L_{ij}^T &= L_{ij}^{(T-2)} + N_i^T qmod d_j \\ &= L_{ij}^{(T-3)} + (N_i^{(T-2)} + N_i^T) qmod d_j \\ &= L_{ij}^{(T-4)} + (N_i^0 + \ldots\ldots + N_i^{(T-2)} + N_i^T) qmod d_j \\ &= L_{ij}^0 + \sum_{r=1}^{T} (N_i^r - N_i^{(T-1)}) qmod d_j \end{aligned}$$

So

$$\sum_{i=1}^{n} L_{ij}^T mod d_j = \sum_{i=1}^{n} (L_{ij}^0 + \sum_{r=1}^{T} (N_i^r - N_i^{(T-1)}) qmod d_j)$$

The attacker needs to obtain the random numbers N_i^r of the first T cycles and the initial secret share L_{ij}^0 of all members in a limited time, however they are secretly selected by the members, so it is difficult. The initial secret share is $L_{ij}^0 = (\alpha_i^0 + N_i^0 q) mod d_j$, since α_i^0 and N_i^0 are secretly selected and saved by members, so it is not possible to get.

In the stage of generating secret shares, an attacker may intercept the broadcast messages $g^{\alpha_i^0}$ and $g^{N_i^0}$ to calculate the secret shares L_{ij}^0, but it is difficult for the attacker to calculate the discrete logarithm problem in the limited time, so it is impossible to get for the attacker.

During the private key update phase, the attacker may intercept the broadcast information $g^{L_{ij}^{(T-2)}}$ and attempt to obtain L_{ij}^{T-2} directly, but it is still a discrete logarithm problem, solving this problem is extremely difficult, so the attacker cannot obtain it within a limited time through calculation.

Therefore, the attacker cannot calculate the member's private key before the period based on the private key of the current period, the scheme has forward security.

4.3 Backward Security Analysis

If the attacker wants to falsify the members' private key after the current cycle, it is not possible.

The member's private key is $H_j^T = H_j^{(T-2)} + \sum_{i=1}^{n} L_{ij}^T mod d_j$, if an attacker wants to falsify the private key after the current, suppose the attacker wants to fake the private key of the T + 1 period, the attacker must calculate $H_j^{(T-1)}$ and $\sum_{i=1}^{n} L_{ij}^{T+1}$, from the analysis in the previous paragraph, it is impossible for an attacker to calculate the private keys before the period in the effective time, so the attacker cannot obtain

$H_j^{(T-1)}$. In addition $\sum_{i=1}^{n} L_{ij}^{T+1} = \sum_{i=1}^{n} (L_{ij}^{(T-1)} + N_i^{T+1} qmodd_j)$, if the attacker want to get $\sum_{i=1}^{n} L_{ij}^{T+1}$, he must calculate $L_{ij}^{(T-1)}$ and $N_i^{T+1} q$, while both of them are selected secretly by members, so it is impossible to get. Through analysis, the attacker cannot calculate the secret share of T + 1 cycle, so it is impossible to forge the member's private key after the cycle.

Therefore, the attacker cannot get the members' private keys after the current period in a limited time, so the scheme is backward security.

5 Performance Analysis

5.1 Efficiency Analysis

Since the modulo-addition operation and the modulo-subtraction operation are negligible compared with other operations, the scheme mainly analyzes the follow aspects of bilinear pair, hash, modular power, modular multiplication and Modular inverse. For ease of understanding, this article defines the following symbols:

This scheme analyzes the three stages of key update, signature generation and signature verification, and compares the calculation complexity between the literature [8, 10, 11], the comparison results are shown in Table 2 below.

Table 2. Time complexity representation symbol.

Operation	Symbol	Time complexity representation
Bilinear pair	e	$o(e(x))$
Hash	h	$o(h(x))$
Modular power	m	$o((lbn)^k)$
Modular multiplication	c	$o(\cdot lbn)$
Modular inverse	u	$o((lbn)^{-1})$

Table 3 is the comparison results of the calculation complexity between this article and other schemes. All of the solutions above have strong forward security. Through analysis, it can be found that the calculation complexity of this scheme is significantly lower than the others.

The computational complexity of the three stages in [8] is higher than that in this paper. In [10] and [11] the algorithm in the update phase is lower than this paper, but it is higher in the stage of generating signature and verification signature.

The order of algorithms complexity involved in the scheme is as follows $e > m > u > h > c$, that is, the bilinear pair calculation has the highest complexity, followed by the modulus power, the modular inverse, and the modular multiplication. This paper mainly includes modular power, modular multiplication and modular inverse, while other schemes all need hash operation. Literature [11] required bilinear

Table 3. Comparison of calculation complexity.

Schemes	Update phase	Signature generation phase	Signature verification phase
This article	$4o((lbn)^k)+2o(\cdot lbn)+o((lbn)^{-1})$	$o((lbn)^k)+3o(\cdot lbn)+o((lbn)^{-1})$	$o((lbn)^k)+2o(\cdot lbn)$
Literature [8]	$2o(h(x))+5o((lbn)^k)+5o(\cdot lbn)+o((lbn)^{-1})$	$o(h(x))+2to((lbn)^k)+2to(\cdot lbn)+o((lbn)^{-1})$	$2o(h(x))+4o((lbn)^k)+5o(\cdot lbn)$
Literature [10]	$3(t+2)o((lbn)^k)$	$o(h(x))+4o((lbn)^k)+3o(\cdot lbn)$	$o(h(x))+4o((lbn)^k)+3o(\cdot lbn)$
Literature [11]	$2to((lbn)^k)$	$2to(h(x))+2to((lbn)^k)+to(e(x))+to((lbn)^{-1})$	$t\left[2o(e(x))+o(h(x))+o((lbn)^k)\right]$

pair calculation which of the computational complexity is significantly higher than the others. So, it is obvious that the operation of this scheme is simpler and the computational complexity is lower than the others.

5.2 Simulation

The environment of the simulation experiment is: 64-bit Window 10 operating system, MyEclipse2015 system, CPU is Intel Core i5-8300H processor, clocked at 2.3 GHz, memory 8 GB. The simulation experiment was carried out on the time overhead between the scheme and the literature [11] in the signature generation and verification phase. The result is shown below:

It can be seen from Fig. 2 that both of the scheme and the literature [11] have an increasing trend with the increase of the number of members. From the experimental result, the scheme [11] takes more time than the scheme proposed. This is because scheme [11] requires bilinear pairing operations in both the signature generation and verification phases, which is computationally complex than the other operations.

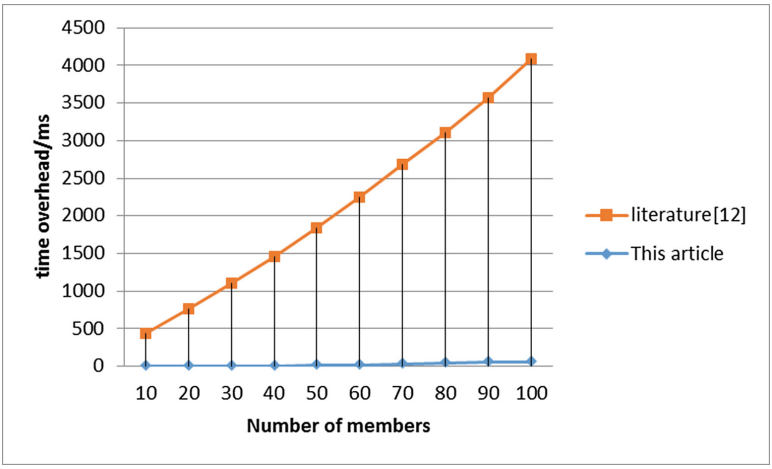


Fig. 2. Relationship between number of members and time overhead

6 Conclusion

In this manuscript, we proposed a threshold signature scheme with strong forward security. The scheme does not need a dealer. Through periodically update member private keys, it solved the problem of forgery or falsification of signatures due to private key leaks.

References

1. Anderson, R.: Invited lecture. In: Proceedings of Fourth Annual Conference on Computer and Communication Security, pp. 1–7. ACM Press, New York (1997)
2. Bellare, M., Miner, S.K.: A forward-secure digital signature scheme. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 431–448. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_28
3. Anderson, R.: Two remarks on public key cryptology. In: Proceedings of the 4th ACM Conference on Computer and Communication California, pp. 16–30. Springer, USA (1997)
4. Burmester, M., Chrissikopoulos, V., Kotzanikolaou, P., Magkos, E.: Strong forward security. In: Dupuy, M., Paradinas, P. (eds.) SEC 2001. IIFIP, vol. 65, pp. 109–121. Springer, Boston, MA (2002). https://doi.org/10.1007/0-306-46998-7_8
5. Wang, M.W., Hu, Y.X.: A forward-backward security digital signature scheme. J. Xidian Univ. **41**(2), 71–78 (2014)
6. Li, C., He, M.X.: A forward and backward security digital signature scheme. In: China Cryptography Society Annual Conference (2008)
7. Wang, Y.B.: Two-way secure signature scheme based on discrete logarithm. J. Qinghai Normal Univ. (Nat. Sci. Ed.) **2**, 6–10 (2016)
8. Yang, J., Qian, H.F., Li, Z.B.: A proxy signature scheme with strong forward security. Comput. Eng. **34**(17), 162–166 (2008)
9. Ye, J., Ding, Y., Liu, Y.N.: Forward and backward secure group signature scheme based on verifiable random number. J. Lanzhou Univ. Technol. **37**(1), 86–90 (2011)
10. Xu, G.B., Jiang, D.H., Liang, Q.X.: A strong forward secure digital signature scheme. Comput. Eng. **39**(9), 167–169 (2013)
11. Yang, X.D.: Research on improved verifiable strong forward security ring signature scheme. Comput. Appl. Softw. **30**(4), 319–322 (2013)
12. Wang, Y., Hou, Q.F., Zhang, X.Q., Huang, M.J.: Dynamic threshold signature scheme based on Chinese remainder theorem. J. Comput. Appl. **38**(4), 1041–1045 (2018)
13. Shi, R.H., Zhou, Y.: A forward-safe dynamic subgroup signature scheme. Comput. Eng. Appl. **42**(30), 130–133 (2006)
14. Tang, L.W., Du, W.Z.: Forward secure group blind signature scheme based on Chinese remainder theorem. J. Comput. Appl. **32**(s1), 53–55 (2012)
15. Ou, H.W., Zhang, S.W.: Forward security group signature based on Chinese remainder theorem. J. Comput. Appl. **31**(s1), 98–100 (2011)
16. Asmuth, C., Bloom, J.: A modular approach to key safeguarding. IEEE Trans. Inf. Theory **29**(2), 208–210 (1983)
17. Hou, Z.F., Tan, M.N.: A CRT-basted (tn) threshold signature scheme without a dealer. J. Electron. Inf. Technol. **11**(3), 975–986 (2015)