# A Secure Multi-party Signature Scheme Based on Trust Mechanism

Yage Cheng[1], Mingsheng Hu[1](✉), Lipeng Wang[1], Yanfang Lei[1], Junjun Fu[1], Bei Gong[2], and Wei Ma[1]

[1] Zhengzhou Normal University, Zhengzhou 450044, China
15652698433@163.com
[2] Beijing University of Technology, Beijing 100124, China

**Abstract.** Aiming at the problem of trust, we propose a secure multi-party signature scheme based on trust mechanism. In this scheme, we introduce a trust vector with time-stamped and form a trust matrix composed of multi-dimensional vectors to record the behavior of the participants periodically. Finally, a trusted evaluation mechanism is established for the participants. Under the premise of participant trustworthiness, a secure multi-party dynamic threshold signature scheme is constructed by secret sharing technology. The security analysis shows that the scheme can effectively suppress the vandalism of malicious participants. it is forward security and can resist mobile attacks. Performance analysis shows that the scheme has lower computational complexity and higher execution efficiency.

**Keywords:** Secure multi-party computation · Trusted mechanism · Trust matrix · Secret sharing · Threshold signature

## 1 Introduction

Secure multi-party computing (SMC) protocol is an academic field that is very active in cryptography. It has strong theoretical and practical significance. Broadly speaking, all cryptographic protocols are a special case of secure multiparty computing. It plays an important role in data mining, statistical analysis, privacy protection and confidential electronic voting etc. It was first proposed by Yao in the 1980s, which was an extension of the millionaire problem [1]. After extensive research by Goldreich et al. [2], secure multi-party computing has become a research hotspot in the international cryptography.

In [3], a secure routing decision scheme based on trust mechanism is proposed. This scheme introduces a trust vector to realize the collection of the evidence chain. The literature [4] sets behavioral trust and energy trust of the nodes to the same weight, and comprehensively considers historical behavior and existing behavioral to record the trusted behavior of the node. In [5], a rational secure multi-party computing protocol based on reputation mechanism is proposed. The scheme is based on Lagrange difference polynomial, which requires many polynomial calculations and is less efficient. The literature [6] proposes a weight-based way to calculate the trust value of the nodes.

W. Han et al. (Eds.): CTCIS 2019, CCIS 1149, pp. 119–132, 2020.
https://doi.org/10.1007/978-981-15-3418-8_9

There are many signature schemes based on secret sharing, such as the literature [7–9]. They can verify the credibility of the participant in the secret share generation stage but cannot verify the credibility of the participant behavior in the signature phase. The literature [10] is based on the Lagrange interpolation polynomial, which has a large amount of computation. The literature [11] is based on the bilinear pairing algorithm. The scheme requires bilinear pairing calculation in the signature and verification process, which makes the signature efficiency low. The literature [12] is based on a secure multi-party fair secret sharing scheme. In recovery phase, the scheme implements the privacy protection function through secure multi-party computing.

In the signature scheme based on trust mechanism, Literature [13] first proposed a conceptual model of "Virtual identity authentication based on trust delivery". The model proposes the establishment, authorization, storage and maintenance rules of trust to ensure the security of the virtual identity authentication process. Literature [14] designed a dynamic credibility evaluation model in a distributed environment. In this scheme the Shapley entropy is introduced into the process of credibility evaluation, so that the credibility evaluation result of the new scheme can more accurately reflect the dynamic behavior of the node. Literature [15] based on the basic automatic trust negotiation model combine with the idea of secure multi-party computing theory, proposes an automatic trust negotiation protocol based on secure multi-party computing to achieve privacy protection.

Based on the above research, this paper designs a secure multi-party signature scheme, introduces a trust matrix to record the participant's trusted behavior, and dynamically binds it to the signature process as evidence. Its overall structure is shown in Fig. 1:

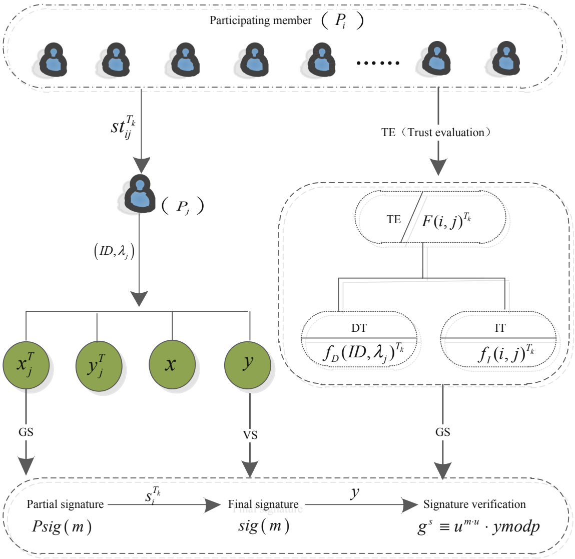

**Fig. 1.** Secure multi-party signature scheme based on trust mechanism

# 2 Prerequisite

## 2.1 Secure Multi-party Computation

Secure Multi-party Computation [16] is used to solve the problem of privacy protection between a group of untrusted participants. It must ensure the independence of the input, the correctness of the calculation, and the confidentiality of the participants' privacy. It is an effective way to solve the privacy calculation problem between two or more participants. It can ensure that the participants complete the computing task without revealing the privacy data. The model is as follows:

Suppose that in a distributed network, there are a group of $n$ participants $P = \{P_1, P_2, \cdots, P_n\}$ who do not trust each other. Each member has the secret data $x = \{x_1, x_2, \cdots, x_n\}$. The participants secretly input the secrets $x_i$, and cooperate with each other to execute functions $f : (x_1, x_2, \cdots, x_n) \to (y_1, y_2, \cdots, y_n)$. Finally, each participant gets their own output $y_i$. In this process, each participant cannot obtain any information from other participants except for their own output information.

In a secure multiparty computing protocol, the person who attempts to destroy the agreement is called an attacker. An attacker can corrupt a subset of participants, and depending on the type of attacker's corrupted participants, the attacker can be divided into two categories [17]:

**Passive Attacker:** If the corrupted are half-honest participants, that is, the attacker can only get the input, output and intermediate results of the corrupted participant, but it cannot change the input and intermediate results, nor can it stop the operation of the agreement, then the attacker called a passive attacker.

**Active Attacker:** The corrupted person has malicious participants, that is, the attacker can not only get the input, output and intermediate results of the corrupted participant, but also can cause the corrupted participant to change the input, the intermediate result information, and even can stop the operation of the agreement, then call this attacker an active attacker.

## 2.2 Asmuth-Bloom Secret Sharing Scheme

In 1983, Asmuth and Bloom [18] proposed the Asmuth-Bloom secret sharing scheme. It mainly includes three steps:

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

**Secret Distribution.** DC calculation $z = S + Aq$, here, $z_i = z \bmod d_i$, DC send $(z_i, d_i)$ to $P_i$ as a secret share of $P_i$.

**Secret Recovery.** Any $t$ members can recover secrets. After exchanging secrets between members, any member can establish the following congruence equations: $z = z_i \bmod d_i$. According to the Chinese remainder theorem, the congruence equation has a unique solution: $z = \sum_{i=1}^{t} \frac{D}{d_i} e_i z_i \bmod D$. Here, $\frac{D}{d_i} e_i \equiv 1 \bmod d_i$ So, we can find $S = z \bmod q$.

# 3 Proposed Scheme

This paper evaluates the credibility of participants by introducing a time-stamped trust matrix, and dynamically binds them to the signature process as the evidence to record the participants' behavior. Based on the trustworthiness of participants, a secure multi-party trusted threshold signature scheme was designed by using secret sharing technology and threshold signature scheme.

## 3.1 A Secure Multi-party Signature Scheme Based on Trust Mechanism

**Key Generation**

1. System initialization: $P = \{P_1, P_2, \cdots, P_n\}$ denotes a set of $n$ members. $p, q$ are large primes satisfy $\frac{q}{p-1}$. There is a strictly monotonically increasing positive integer sequence $d = \{d_1, d_2, \cdots, d_n\}$ satisfies the Asmuth-Bloom secret sharing scheme, and $D = \prod_{i=1}^{t} d_i$ is the product of the $t$ smallest $d_i$. The threshold is $t$, and $g \in GF(p)$ is the generated element. $m$ is the message to be signed. Published $p, q, g, t, d, D$.
2. Select the secret tag information: $P_i (i = 1, 2, \cdots, n)$ secretly selects the secret tag $st_{ij}^0 = \left(c_{i1}^0, c_{i2}^0, \cdots, c_{in}^0\right)$ and sends them to $P_j$ , $(j = 1, 2, \cdots, n, j \neq i)$. At the same time, $P_i$ keeps $c_{ii}^0$. Then calculate $st_i^0 = c_{i1}^0 + c_{i2}^0 + \cdots + c_{in}^0 = \sum_{j=1}^{n} c_{ij}^0$ and broadcasts $g^{c_{ii}^0}$ and $g^{st_i^0}$
3. Generate identity tag information: The participant selects the random number $r_i$ and calculates $r = \sum_{i=1}^{n} r_i$, then broadcasts $g^{r_i}$ and $g^r$. Let $ID_i = \lambda_i (\bmod\, d_i)$, and its solution is $ID = \sum_{i=1}^{n} \frac{D}{d_i} e_i \lambda_i \bmod D$, then the identity tag information of the participant $P_i$ is $(ID, \lambda_i)$. Here $\lambda_i = r + st_i^0$, and $e_i$ satisfies $\frac{D}{d_i} e_i \equiv 1 \bmod d_i$.
4. Calculate verification information: Let $\mu_i^0 = st_i^0 + c_{ii}^0$, $\alpha_i^0 = \lambda_i + c_{ii}^0$, $\beta_i^0 = st_i^0 q + r$. According to the broadcast information $g^{c_{ii}^0}$, $g^{st_i^0}$ and $g^r$, if

$$\left(g^{\left(\alpha_i^0 - \mu_i^0\right)} \bmod p\right) \cdot \left(\left(g^{st_i^0}\right)^q \bmod p\right) \bmod p = \left(g^{\left(\beta_i^0\right)}\right) \bmod p,$$

then $P_j$ receives the information $c_{ij}^0 (i \neq j)$ sent by $P_i (i \neq j)$.

5. Generate secret share: Then $P_j(j = 1, 2, \cdots, n)$ calculates its own secret share $ss_j^0{=}c_{1j}^0 + c_{2j}^0 + \ldots\ldots + c_{nj}^0 = \sum\limits_{i=1}^{n} c_{ij}^0$.
6. Generate keys: $P_i(i = 1, 2, \cdots, n)$ calculate $x_i^0 = (ss_i^0 q + r_i) \bmod d_i$ together to generate the personal private key $x_i^0$. Then the public key of the participant is $y_i^0 = g^{x_i^0} \bmod p$. The system private key is $x = \sum\limits_{i=1}^{n} r_i \bmod p$, and the system public key is $y = g^x \bmod p$.

**Trust Evaluation (TE)**
A time-stamped trust vector is generated by dynamically updating interaction information between participants. Then, as the basis for the trustworthiness of the participants a trust matrix (TM) is constituted by the trust vectors. The specific process is as follows:
The trusted evaluation function is:

$$F(i,j)^{T_k} = \frac{1}{2}\left[f_D(i,j)^{T_k} + f_I(i,j)^{T_k}\right]$$

As shown above, $F(i,j)^{T_k}$ represents a trusted evaluation of participant $i$ by $j(i = 1, 2, \cdots, n)$ during the $k$ update cycle, which consists of direct trust and indirect trust. Direct trust (DT) $f_D(i,j)^{T_k}$ is the identity tag metric trust evaluation value of $P_i$, here $f_D(ID, \lambda_j)^{T_k} {\subseteq} \{0, 1\}$. The indirect trust (IT) $f_I(i,j)^{T_k}$ is the trust metric evaluation of $P_j$ on $P_i$, here $f_I(i,j)^{T_k} {\subseteq} \{0, 1\}$. Let $f(x) = [x]$ be the rounding function, so $F(i,j)^{T_k} = \left\{0, \frac{1}{2}, 1\right\}$. Here $T$ denotes the update cycle.

The direct trust calculation is based on the function $g^{\lambda_i}{=}g^r \cdot g^{st_i^0}$. If the equation $g^{\lambda_i}{=}g^r \cdot g^{st_i^0}$ is true, the participant identity information is credible, that is $P_i$ is trusted then $f_D(i,j)^{T_k} = 1$, otherwise 0.

The indirect trusted calculation is determined by the verification equation

$$\left(g^{\left(\alpha_i^{T_k} - \mu_i^{T_k}\right)} \bmod p\right) \cdot \left(\left(g^{st_i^{T_k}}\right)^q \bmod p\right) \bmod p = \left(g^{\left(\beta_i^{T_k}\right)}\right) \bmod p$$

if the equation is true, then $f_I(i,j)^{T_k}{=}1$ ,otherwise equal to 0.

The participant $P_j(j = 1, 2, \ldots\ldots, n)$ generates the trust vector $TV$ by the trust value of $P_i$ in cycle $k$.

$$TV_i^{T_k} = \begin{bmatrix} F_{i1}^{T_k} & F_{i2}^{T_k} & \cdots & F_{in}^{T_k} \end{bmatrix}$$

here $F_{ii}^{T_k}{=}1$, and the trust vector of $P_i(i = 1, 2, \cdots\cdots, n)$ is formed into the trust matrix $(TM)$ is a reliable judgment basis for the participant behavior.

$$TM_{ij}^{T_k} = \begin{bmatrix} F_{11}^{T_k} & F_{12}^{T_k} & \cdots & F_{1n}^{T_k} \\ F_{21}^{T_k} & F_{21}^{T_k} & \cdots & F_{2n}^{T_k} \\ \vdots & \vdots & \vdots & \vdots \\ F_{n1}^{T_k} & F_{n2}^{T_k} & \cdots & F_{nn}^{T_k} \end{bmatrix}$$

So, every participant $P_i$ retains the credible estimates of the first $k$ cycles of all participants.

$$TM_{i}^{T_k} = \begin{bmatrix} F_{i1}^{1} & F_{i2}^{1} & \cdots & F_{in}^{1} \\ F_{i1}^{2} & F_{i2}^{2} & \cdots & F_{in}^{2} \\ \vdots & \vdots & \vdots & \vdots \\ F_{i1}^{T_k} & F_{i2}^{T_k} & \cdots & F_{in}^{T_k} \end{bmatrix}$$

In cycle $k+1$, $P_i$ update $TM_i^k$ and save it.

$$TM_{i}^{T_k} = \begin{bmatrix} F_{i1}^{1} & F_{i2}^{1} & \cdots & F_{in}^{1} \\ F_{i1}^{2} & F_{i2}^{2} & \cdots & F_{in}^{3} \\ \vdots & \vdots & \vdots & \vdots \\ F_{i1}^{T_k} & F_{i2}^{T_k} & \cdots & F_{in}^{T_k} \end{bmatrix} \rightleftharpoons \begin{bmatrix} F_{i1}^{T_{k+1}} & F_{i2}^{T_{k+1}} & \cdots & F_{in}^{T_{k+1}} \end{bmatrix} = TV_{ij}^{T_{k+1}}$$

$$\downarrow$$

$$TM_{i}^{T_{k+1}} = \begin{bmatrix} F_{i1}^{1} & F_{i2}^{1} & \cdots & F_{in}^{1} \\ F_{i1}^{2} & F_{i2}^{2} & \cdots & F_{in}^{2} \\ \vdots & \vdots & \vdots & \vdots \\ F_{i1}^{T_{k+1}} & F_{i2}^{T_{k+1}} & \cdots & F_{in}^{T_{k+1}} \end{bmatrix}$$

When the participant is not trusted, it will be excluded.

### Generating a Signature (GS)

1. $P_i(i = 1, 2, \cdots, t)$ select random numbers $P_i(i = 1, 2, \cdots, t)$, and calculate $u_i = g^{l_i \cdot \frac{\sum_{j=1}^{t} F(i,j)^{T_k}}{n}} \bmod p$. Then $P_i(i = 1, 2, \cdots, t)$ send it to $P_j$ and broadcast it.
2. After $P_j$ receive $u_i$, the intermediate variable $u = g^{\frac{1}{t}\sum_{i=1}^{t}\left(l_i \cdot \sum_{j=1}^{t} F(i,j)^{T_k}\right)} \bmod p = \prod_{i=1}^{t} g^{l_i \cdot \frac{\sum_{j=1}^{t} F(i,j)^{T_k}}{t}} \mathrm{modp} = \prod_{i=1}^{t} u_i \bmod p$ is calculated together by $P_j(j = 1, 2, \cdots, t)$.
3. $P_i(i = 1, 2, \cdots, t)$ calculate the partial signature $s_i^0 = m \cdot u \cdot l_i + w_i^0 \bmod D$ together, so the partial signature of each participant is $(m, u, s_i)$, where $w_i^0 = \sum_{i=1}^{t} \frac{D}{d_i} e_i x_i^0 \bmod D$ .

4. $P_i$ co-calculates $s = \left(\sum_{i=1}^{t} s_i^0 \text{mod} D\right) \text{mod} q$ to generate the final signature $sig(m)$, then the signature of message $m$ is $sig(m) = (m, u, s)$.

**Verification Signature (VS)**
$P_i$ Verifies the equation $g^s \equiv u^{m \cdot u} \cdot ymodp$ by the system public key $y$. If the equation is true, the signature of the message $m$ is valid.

**Dynamic Update**
Due to the existence of mobile attacks, an attacker may obtain somebody's private key of the participant through a long-term stable attack. However dynamically updating the keys of the participants' can effectively prevent mobile attacks and increase the security. The solution keeps the system public key unchanged during the whole update process, retains the function of using the system public key to access historical signature information. Set the update period to be $T$.

Every $T$ cycle, the participants update secret tag $C^{T_k} = \left(st_{ij}^{T_k}\right)^{t \times t}$, and co-calculate the private key generation function $x_i^{T_k} = (ss_i^{T_k} q + r_i) \bmod d_i$, and then update the private keys $x_i^{T_k}$.

After the update is complete, the participants can also generate the signature according to the signature process 1–4.

### 3.2 A Secure Multi-party Signature Protocol Based on Trust Mechanism

Protocol: Trust-based secure multi-party signing protocol

Input: $P_1, P_2, \cdots, P_n$ enter the secret tag information $st_{ij}^{T_k}$ and the random positive integers $r_i$ and $l_i$.

Output: signature of message $m$.

1. Initialize to select system parameters. The participants select secret tag information $st_{ij}^{T_k}$ and random positive integers $r_i$ and $l_i$.
2. $P_i (i = 1, 2, \cdots, n)$ co-calculates private keys $x_i^0 = (ss_i^0 q + r_i) \bmod d_i$, then the public key of $P_i$ is $y_i^0 = g^{x_i^0} \bmod p$. The system private key is $x = \sum_{i=1}^{n} r_i \bmod p$, and the system public key is $y = g^x \bmod p$. Public $y_i^0$ and $y$.
3. The credibility of the participants are evaluated according to the trusted evaluation function $F(i,j)^{T_k}$, and the untrusted are eliminated.
4. $P_i$ calculates $u_i = g^{l_i \cdot \frac{\sum_{j=1}^{t} F(i,j)^{T_k}}{t}} \text{mod} p$ and broadcasts it. Then $P_i$ calculates the intermediate variable $u = g^{\frac{1}{t}\sum_{i=1}^{t}\left(l_i \cdot \sum_{j=1}^{t} F(i,j)^{T_k}\right)} \text{mod} p = \prod_{i=1}^{t} g^{l_i \cdot \frac{\sum_{j=1}^{t} F(i,j)^{T_k}}{t}} \text{modp} = \prod_{i=1}^{t} u_i \text{mod} p$ .

5. $P_i$ calculates the partial signature $s_i^0 = m \cdot u \cdot l_i \cdot \frac{\sum_{j=1}^{t} F(i,j)^0}{t} + w_i^0 \text{mod} D$, so the partial signature is $(m, u, s_i)$, where $w_i^0 = \sum_{i=1}^{n} \frac{D}{d_i} e_i x_i^0 modD$.
6. $P_i$ co-calculates $s = \left( \sum_{i=1}^{n} s_i^0 modD \right) modq$ and generates the final signature $sig(m)$. Then, the signature of the message $m$ is $sig(m) = (m, u, s)$.
7. $P_i$ Verify the equation $g^s \equiv u^{m \cdot u} \cdot ymodp$. If it is true, the signature of the message $m$ is valid.
8. Every $T$ cycle, the participants jointly calculate $x_i^0 = (ss_i^0 q + r_i) \bmod d_i$ and get their own new private keys, and with the new private keys they execute the signature process from 1 to 7.

# 4 Correctness and Safety Analysis

## 4.1 Correctness Analysis

**Theorem 1:** The signatures generated by the participants jointly calculated are valid. According to the construction of the protocol, the private key of $P_i$ is

$$x_i^{T_k} = (ss_i^{T_k} q + r_i) \bmod d_i.$$

Let

$$\alpha = ss_i^{T_k} q + r_i \tag{1}$$

so

$$x_i^{T_k} = \alpha \text{mod} d_j.$$

According to the Chinese remainder theorem, we can solve the congruence equations:

$$\begin{cases} x_1^{T_k} \equiv \alpha \text{mod}\, d_1 \\ x_2^{T_k} \equiv \alpha \text{mod}\, d_2 \\ \vdots \\ x_t^{T_k} \equiv \alpha \text{mod}\, d_t \end{cases}$$

get a unique solution: $\alpha = \sum_{i=1}^{t} \frac{D}{d_i} e_i x_i^{T_k} \bmod D$

Let $w_i^{T_k} = \sum_{i=1}^{t} \frac{D}{d_i} e_i x_i^{T_k} \bmod D$

Then $\alpha = \sum_{i=1}^{t} w_i^{T_k} \bmod D$

When $t > 2$, according to [19], $m \cdot u \cdot \frac{1}{t}\sum_{i=1}^{t}\left(l_i \cdot \sum_{j=1}^{t} F(i,j)^{T_k}\right) + \alpha < D$, so there is

$$\begin{aligned} s &= \left(\sum_{i=1}^{t} s_i^{T_k} modD\right) modq \\ &= \left[\sum_{i=1}^{t}\left(m \cdot u \cdot l_i \cdot \frac{\sum_{j=1}^{t} F(i,j)^{T_k}}{t} + w_i^{T_k}\right) \bmod D\right] \bmod q \\ &= \left(m \cdot u \cdot \frac{1}{t}\sum_{i=1}^{t}\left(l_i \cdot \sum_{j=1}^{t} F(i,j)^{T_k}\right) + \alpha\right) \bmod q \end{aligned}$$

From (1) $\alpha = ss_{ij}^{T_k} q + r_i$, we can get

$$\begin{aligned} s &= \left[m \cdot u \cdot \frac{1}{t}\sum_{i=1}^{t}\left(l_i \cdot \sum_{j=1}^{t} F(i,j)^{T_k}\right) + \sum_{j=1}^{t} ss_{ij}^{T_k} q + r_i\right] \bmod q \\ &= \left(m \cdot u \cdot \frac{1}{t}\sum_{i=1}^{n}\left(l_i \cdot \sum_{j=1}^{t} F(i,j)^{T_k}\right) + \sum_{i=1}^{t} r_i\right) \bmod q \end{aligned}$$

So

$$\begin{aligned} g^s &\equiv g^{m \cdot u \cdot \frac{1}{t}\sum_{i=1}^{t}\left(l_i \cdot \sum_{j=1}^{t} F(i,j)^{T_k}\right) + \sum_{i=1}^{t} r_i} \bmod p \\ &\equiv g^{m \cdot u \cdot \frac{1}{t}\sum_{i=1}^{t}\left(l_i \cdot \sum_{j=1}^{t} F(i,j)^{T_k}\right)} \cdot g^{\sum_{i=1}^{t} r_i} \bmod p \\ &\equiv u^{m \cdot u} \cdot y \bmod p \end{aligned}$$

If the equation is established by verification, we can say that the signature is valid. So, the correctness of the agreement is proved.

### 4.2 Security Analysis

**Theorem 2:** It has the forward security.

The private keys of the participants were updated regularly to ensure the forward security.

Suppose that if an attacker who wants to get the private key $x_i^{T_k} = (ss_i^{T_k} q + r_i) \bmod d_i$ of participant $P_i$ in cycle $k$, the attacker needs to obtain both the secret tag information $ss_i^{T_k}$ and the personal privacy $r_i$. He needs to calculate $ss_i^{T_k} q + r_i$, but $ss_i^{T_k} = c_{i1}^{T_k} + c_{i2}^{T_k} + , \cdots ,$ $+ c_{in}^{T_k}$ ,which is obtained from the other participants $P_i (i = 1, 2, \cdots, n-1)$. It is difficult for an attacker to attack all participants simultaneously in a finite time to get $ss_i^{T_k}$.

The attacker may want to calculate $ss_i^{T_k}$ by $g^{ss_i^{T_k}}$, however the problem is based on the discrete logarithm problem that he cannot obtain it by calculation. Similarly, $r_i$ is secretly selected by the participant, and the attacker cannot directly obtain it. The attacker may want to calculate $ss_i^{T_k}$ through the broadcast information $g^{ss_i^{T_k}}$, which also belongs to the discrete logarithm calculation problem, so it is difficult for the attacker to obtain it.

Therefore, it is difficult for an attacker to obtain the participants' private keys by calculating the discrete logarithm problem. It has the forward security.

**Theorem 3:** It is resistant to mobile attacks.

The mobile attack means that when an attacker successfully invades and controls a participant, he can transfer the attack target to other participants of the system. A mobile attacker may not be able to completely invade and control all participants in a short period of time, but if there is enough time, he can obtain almost all secret shares to damage the system security. So, it is necessary to update the private keys regularly to prevent the mobile attacks.

This paper is based on the $(t, n)$ threshold secret sharing scheme, which requires at least $t$ thresholds for solving congruence equations. Less than $t$ cannot be solved. So, only the attacker successfully invades $t$ or more participants in the same period, which may affect the security of the system.

Suppose an attacker invades in $k$ cycle, and the private key of $P_i$ is $x_i^{T_k} = (ss_i^{T_k} q + r_i) \bmod d_i$. If the attacker wants to obtain the private key, it is necessary to obtain the secret tag $ss_i^{T_k}$ and the random number $r_i$ of $t$ members simultaneously within the finite time. The attacker may intercept the broadcast information $g^{ss_i^{T_k}}$ and $g^{r_i}$, and try to calculate $ss_i^{T_k}$ and $r_i$ to obtain the private keys. But through calculate $g^{ss_i^{T_k}}$ and $g^{r_i}$ to get $ss_i^{T_k}$ and $r_i$ are discrete logarithm problems, it is impossible for the attacker to calculate it within the effective time. So, it can effectively resist mobile attacks.

**Theorem 4:** Trusted evaluation mechanism can effectively identify malicious participants.

The trusted evaluation mechanism dynamically monitors the behavior of participants in times to ensure the participants are dynamically trusted.

The trusted evaluation function is

$$F(i,j)^{T_k} = \frac{1}{2}\left[f_D(i,j)^{T_k} + f_I(i,j)^{T_k}\right]$$

here $f_D(ID, \lambda_j)^{T_k}$ is the direct trust metric function, and $ID_i = \lambda_i (\bmod\, d_i)$ is the participant's identity tag. It has a unique solution $ID = \sum_{i=1}^{n} \frac{D}{d_i} e_i \lambda_i \bmod D$. If an attacker pretends to be a trusted participant to forge a pseudo-private key $x_i^{T'_k} = (ss_i^{T'_k} q + r_i) \bmod d_i$ , according to the participant identity information, there must be $ss_i^{T'_k} \neq ss_i^{T_k}$, $\lambda_i^{T'_k} \neq \lambda_i^{T_k}$, so $ID' \neq ID$. Then, it can be judged that the participant behave is abnormally and it is not trusted. In addition, based on the Chinese remainder theorem to solve the congruence equations is a large modulus decomposition problem. So it is impossible for an attacker to solve $e_i$ by $D$ and $d_i$, then it is also impossible to obtain the private keys through the identity information $ID$ of the participant.

In addition, $f_I(i,j)^{T_k}$ is the indirect trusted metric function, which is generated by interaction verification:

$$\left(g^{\left(\alpha_i^0-\mu_i^{T_k}\right)} \bmod p\right)\cdot\left(\left(g^{st_i^0}\right)^q \bmod p\right) \bmod p=\left(g^{\left(\beta_i^0\right)}\right) \bmod p$$

due to,

$$\begin{aligned}\mu_i^0&=st_i^0+c_{ii}^0, \alpha_i^0=\lambda_i+c_{ii}^0,\\ \beta_i^0&=st_i^0q+r, \lambda_i=st_i^0+r_i,\end{aligned}$$

So,

$$\begin{aligned}&=\left(g^{\left(st_i^0+r_i+c_{ii}^0-st_i^0-c_{ii}^0\right)} \bmod p\right)\cdot\left(\left(g^{st_i^0}\right)^q \bmod p\right) \bmod p\\ &=\left(g^{r_i} \bmod p\right)\cdot\left(\left(g^{st_i^0}\right)^q \bmod p\right) \bmod p\\ &=\left(g^{r_i}\cdot g^{st_i^0\cdot q}\right) \bmod p\\ &=\left(g^{r_i+st_i^0\cdot q}\right) \bmod p\\ &=g^{\beta} \bmod p\end{aligned}$$

If it passes the verification, $P_i$ is trusted. Then $f_D(i,j)^{T_k}=1$ and $f_I(i,j)^{T_k}=1$. So

$$F(i,j)^{T_k}=\frac{1}{2}\left[f_D(i,j)^{T_k}+f_I(i,j)^{T_k}\right]=1,$$

otherwise it is 0.

# 5 Performance Analysis

## 5.1 Efficiency Analysis

The scheme is based on the Asmuth-Bloom secret sharing scheme, which mainly involves calculations such as modular multiplication, modular addition, and modular subtraction, and it requires only one inverse calculation. So, it reduces the computational complexity, decreases time consumption, and improves computational efficiency. Compared with the Shamir secret sharing scheme based on the Lagrange difference polynomial and the bilinear pairing operation, the scheme has obvious advantages.

For convenience of description, this paper defines the symbolic representation method of Table 1 below.

**Table 1.** Symbol description

| Description | Symbol | Complexity representation |
|---|---|---|
| Logarithm operation | $e$ | $o(e(x))$ |
| Modular power operation | $m$ | $o((lbn)^k)$ |
| Modular inverse | $u$ | $o((lbn)^{-1})$ |
| Hash function calculation | $h$ | $o(h(x))$ |

Table 2 is the comparison of the computational complexity of this paper and the literature [10] and [11]. The literature [10] is based on the Lagrange difference polynomial. It has higher polynomial order and complicated calculation, which leads to lower execution efficiency. In [11], a forward-and-secure signature algorithm is constructed by using bilinear pairwise properties. The scheme introduces bilinear pairwise operations and hash calculations when generating the signature. In the verification process, two bilinear operations are required to verify. It greatly increases the system calculation complexity and the execution efficiency.

From Table 2, we can find that the calculation efficiency of this paper is significantly higher than the others.

**Table 2.** Computational complexity comparison

| Scheme | Signature generation | Signature verification |
|---|---|---|
| Ours | $t\left[o((lbn)^k)+3o(\cdot lbn)\right]$ | $o((lbn)^k)+o(\cdot lbn)$ |
| Lit. [10] | $(4t+1)o(lbn)^k+3o(h(x))$ | $3to(lbn)^k+2to\left((lbn)^{-1}\right)+o(h(x))$ |
| Lit. [11] | $2to(h(x))+2to((lbn)^k)+to(e(x))+to((lbn)^{-1})$ | $t\left[2o(e(x))+o(h(x))+o((lbn)^k)\right]$ |

## 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 of the scheme and the literature [10] in the signature and verification phase. The result is shown below:

It can be seen from Fig. 2 that ours' scheme and the literature [10] both have an increasing trend with the increase of the number of members, this is because the signature process is positively related to the number of the members. From the experimental data, the literature [10] takes more time than ours' scheme. This is because the literature [10] requires bilinear pairing in the signature generation and verification stages, and the computational complexity is relatively high.

It can be seen from Fig. 3 that the efficiency of this paper is improved by about 90% compared with the literature [10], which greatly reduces the time overhead and improves the execution efficiency.

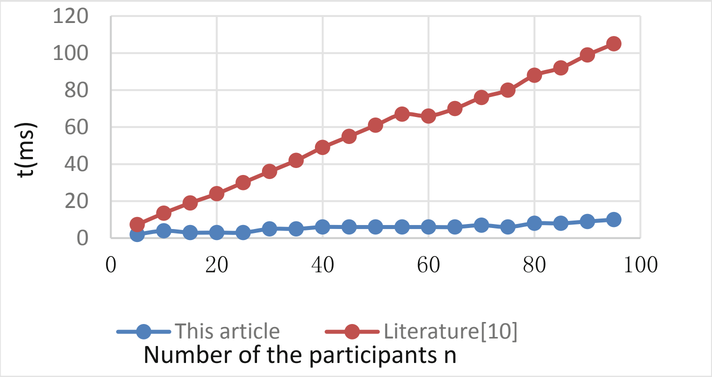

**Fig. 2.** Relationship between the number of members n and time overhead

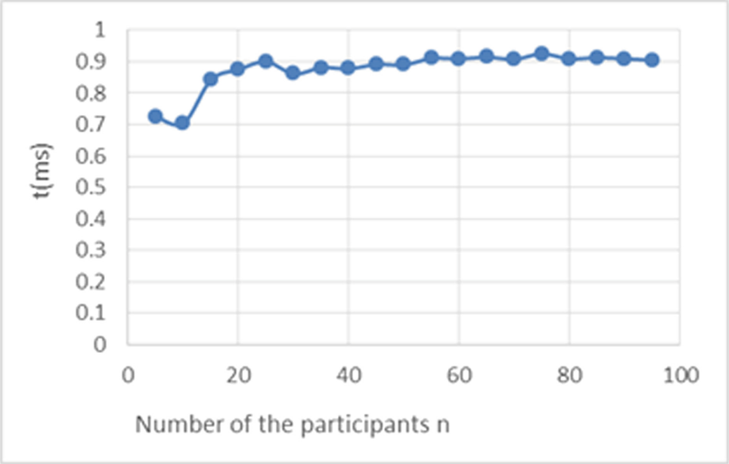

**Fig. 3.** Relationship between efficiency and number of members

## 6 Summary

Based on secure multi-party computing, this paper establishes a trust evaluation mechanism to detect the credibility of the participants. A time-stamped trust matrix is introduced to record the behavior of the participants, which is traceable. It has no trusted center and the secret shares are generated by the interaction among the participants, which has verifiable function. Regularly update the private keys to make it forward-secure and resistant to mobile attacks. And it is based on the Chinese remainder theorem, which reduces the computational complexity, has a small amount of computation, and improves the efficiency of execution.

## References

1. Yao, A.C.: Protocols for secure computations. In: Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science, Piscataway, NJ, pp. 160–164. IEEE Press (1982)
2. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: Proceedings of the 19th Annual ACM Conference on Theory of Computing, pp. 218–229. ACM Press, New York (1987)
3. Li, F., Si, Y.L., Chen, Z., Lu, N., Shen, L.M.: Trust-based security routing decision method for opportunity network. J. Softw. **29**(09), 2829–2843 (2018)
4. Qin, D.Y., Jia, S., Yang, S.X., Ma, J.Y., Zhang, Y., Ding, Q.: Research on secure routing mechanism of wireless sensor networks based on trust perception. J. Commun. **38**(10), 60–70 (2017)
5. Zheng, W.: Research on rational and secure multi-party computing protocol based on secret sharing under multiple mechanisms. Beijing University of Technology, Beijing (2018)
6. Jiang, J., Han, G., Wang, F., et al.: An efficient distributed trust model for wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. **26**(5), 1228–1237 (2015)
7. Cheng, Y.G., Hu, M.S., Gong, B., Wang, L.P., Xu, E.F.: Dynamic threshold signature scheme with strong forward security. Comput. Eng. Appl., 1–13 (2019). Accessed 04 June 2019. http://kns.cnki.net/kcms/detail/11.2127.tp.20190417.1134.006.html
8. Cheng, Y.G., Jia, Z.J., Hu, M.S., Gong, B., Wang, L.P.: A threshold signature scheme applying for blockchain electronic voting scene, pp. 1–9 (2019). Accessed 04 June 2019. http://kns.cnki.net/kcms/detail/51.1307.TP.20190507.1540.002.html
9. Wang, L.P., Hu, M.S., Jia, Z.J., Gong, B, Zhang J.L.: Blockchain voting scene signature scheme based on Chinese remainder theorem. Comput. Appl. Res., 1–8. Accessed 16 June 2019. https://doi.org/10.19734/j.issn.1001-3695.2018.08.0566
10. Yang, X.D.: Research on improved verifiable strong forward security ring signature scheme. Comput. Appl. Softw. **30**(4), 319–322 (2013)
11. Xu, P.: Proactive threshold RSA signature scheme based on polynomial secret sharing. J. Electron. Inf. **38**(09), 2280–2286 (2016)
12. Fu, Z.Y., Zhang, Y.H., Xu, J.G.: A Fair secret sharing scheme based on secure multi-party. Math. Model. Appl. **7**(02), 30–35 (2018)
13. Wang, L.: Research on Mobile Business Virtual Identity Authentication Mechanism based on Trust Transfer. Beijing Jiaotong University, Beijing (2015)
14. Zhu, Y.W.: Research on Privacy Protection Technology and its Application in Distributed Environment. University of Science and Technology of China, Hefei (2012)
15. Wang, W.: Research on Automatic Trust Negotiation Protocol based on Secure Multi-party Computing. Hunan University, Changsha (2012)
16. Dou, J.W., Li, S.D.: Study on secure multi-party computing scheme for data equality problem. Acta Electronica Sinica **46**(05), 1107–1112 (2018)
17. Li, Q.: Research and Application of Secure Multi-party Computing Protocol. Shanghai Jiaotong University, Minhang (2003)
18. Asmuth, C., Bloom, J.: A modular approach to key safeguarding. IEEE Trans. Inf. Theor. **29** (2), 208–210 (1983)
19. Hou, Z.F., Tan, M.N.A.: CRT-basted (t, n) threshold signature scheme without a deeler. J. Electron. Inf. Technol. **11**(3), 975–986 (2015)