Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

ROMSS: a rational optional multi-secret sharing scheme based on reputation mechanism

Ruonan Shi[1], Yuling Chen[1*], Chaoyue Tan[1], Yun Luo[1] and Tao Li[1]

Abstract

The traditional threshold secret sharing scheme only allows the participants' sub-secret shares to be used once in the reconstruction process. Several multi-secret sharing schemes have been proposed that are related to cloud computing, aiming to improve reconstruction efficiency. Rational secret sharing is a technique that combines secret sharing with game theory. In traditional rational multi-secret sharing, participants must reconstruct all secrets, resulting in unnecessary overhead. Rational participants will act dishonestly to maximize their own interests, leading to a prisoner's dilemma and incomplete secret reconstruction. Additionally, when sharing multiple secrets, the Dealer must distribute the sub-secret shares of all secrets to the participants, increasing overhead. In this paper, we propose a rational optional multi-secret sharing scheme based on a reputation mechanism that selectively reconstructs secrets according to participants' needs in the context of cloud computing. Our scheme introduces a reputation mechanism to evaluate participants' reputation values to avoid their dishonest behaviors. Furthermore, we adopt a broadcast encryption matrix so that participants only need to receive a single sub-secret share to participate in multi-secret reconstruction. Our security analysis shows that the proposed scheme can effectively constrain the self-interested behavior of rational participants and reduce the overhead in the process, thus multi-secret sharing scheme can provide more efficient and secure solutions for secret sharing in key management and distributive storage for the cloud scenarios.

Keywords Multi-secret sharing, Cloud computing, Rationality, Reputation mechanism

Introudction

In recent years, cloud computing has become the primary technology for users to perform computing tasks. However, with the increasing amount of data being stored due to the era of big data, there are growing concerns about data leakage and security issues [1–3]. The growing quantity of data and security requirements are all relying on stable and efficient computation to ensure the secure of data [4]. Secret sharing technology is widely used in cloud storage to ensure data security. The confidentiality of data also makes secret sharing technology useful in various fields, such as finance, medical care, and government [5–7]. In 1979, Shamir [8] proposed the famous (t, n) threshold secret sharing scheme, which is a basic component of information security and a security protocol that provides confidentiality and robustness services for various ciphers [9]. The initial secret sharing scheme involves a Dealer and n participants. The Dealer splits the secret into n sub-secret shares and sends them to the n participants. If there are t or more participants cooperating in the reconstruction using Lagrange interpolation, the original secret can be reconstructed. Cooperation of less than t participants cannot provide any information about the secret. The advantages of the secret sharing technique are that it can distribute data processing in a

*Correspondence:
Yuling Chen
[1] State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang, China

decentralized manner, thus enhancing the security of data [10, 11]. Even if some secret information is leaked, the attacker cannot reconstruct the complete secret information. Additionally, it can improve the reliability of the data [12], as even if a participant fails, the integrity of the data can still be restored by other participants. Therefore, secret sharing can be used to address some practical issues, such as API selection in mobile application development, resource optimization based on CNN [13–16].

Multi-secret sharing technology is a complex research field that involves key issues such as trust, data security, and reliability of the parties [17]. To better address these challenges [18], researchers have proposed various multi-secret sharing schemes. These schemes can be used without revealing sensitive data, allowing multiple participants to process data together and achieve effective information sharing and collaboration. Despite the successful application of secret sharing schemes, there are still some problems, such as limitations in security, recovery, and efficiency. These issues need to be addressed to improve the performance of multi-secret sharing schemes that are specifically designed for cloud computing environments..

Halpern [19] applied the rational assumption in game theory to secret sharing, introducing the concept of rational secret sharing and proving that many traditional secret sharing schemes cannot be realized under this assumption. Designing new rational secret sharing schemes has become a new research direction. In the case of rational participants, each participant is assumed to use various strategies to maximize benefits [20]. This makes the reconstruction of the secret difficult since participants may choose strategies unfavorable to others in order to maximize their own interests. To address the problems of security recovery and efficiency in multi-secret sharing, we drew on the idea of dynamic multi-key encryption [21]. This ensures that the underlying plaintext is not exposed during security calculation on encrypted data, and it addresses the difficulty of reconstruction caused by rational participants. In this paper, we propose a rational optional multi-secret sharing scheme based on a reputation mechanism to analyze the behavior of rational participants. We call this scheme ROMSS, which enables participants to ensure the security and recovery of secret information while protecting their own interests. Our analysis results show that the proposed scheme is efficient, practical, and ensures safety and reliability.

In order to implement the secret sharing scheme in cloud environment, the distributed nature and security requirements of cloud computing are utilized to apply the secret sharing technology, as shown in Fig. 1, the user uploads their secret to the master server of the cloud system, which is then distributed through the secret sharing distribution phase. The secret is divided into n shares and sent to n storage nodes, ensuring that each node has a small portion of the secret, and that a minimum of t shares is required to reconstruct the secret. When other users request access to the secret, the master server reconstructs the secret using the sub-secret shares sent by each node, and returns the result to the user to complete the query. Based on this, users can achieve privacy data security protection under cloud computing, optimizing data encryption and decryption while ensuring the security and privacy of data.

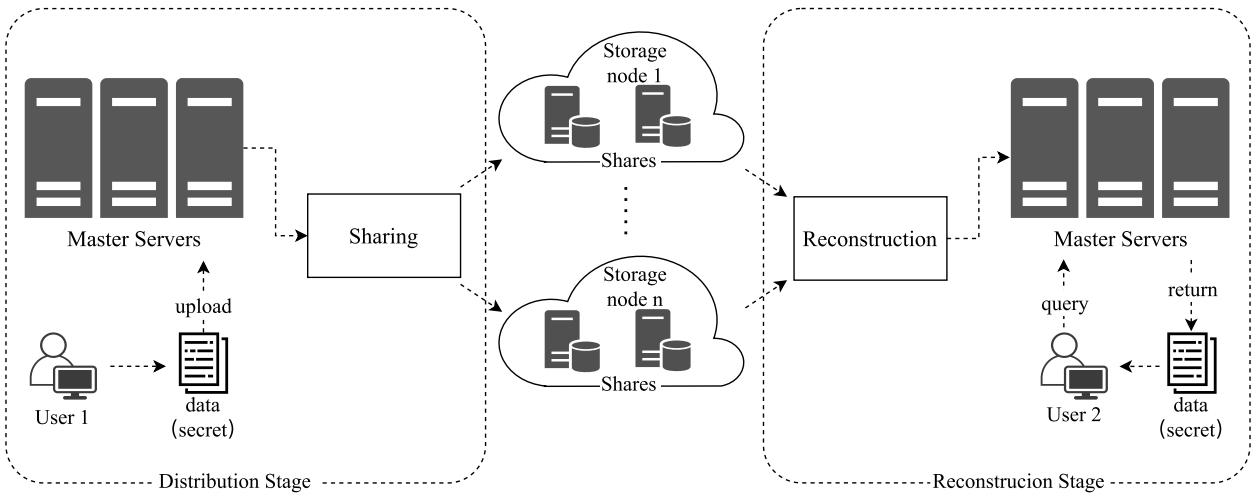

Fig. 1 Secret sharing in cloud computing

Related work

In game theory, John Nash introduced the concept of Nash equilibrium [22], which represents a state of balance and is a solution concept. Halpern and Teague [19] studied the concept of Nash equilibrium and applied it to secret sharing. Rational participants choose the best strategy and eliminate inferior strategies to keep the secret reconstruction process in a balanced state. Maleka [23] studied the repeated game model of sharing a secret in a rational secret sharing scheme. However, the repeated game requires multiple process games, and once the players know that they are in the last sub-game of the secret reconstruction, the players have no incentive to be honest. Ong [24] studied the perfect equilibrium of the sub-game, but the model assumes that there are only a small number of honest participants.

Basar [25] introduced the concept of game theory, in general rational secret sharing, participants will pursue the maximization of interests, leading to the prisoner's dilemma in the whole process. Therefore, an incentive mechanism is necessary to ensure the honest behavior of the participants. Zhang and Liu [26] designed a credible penalty mechanism in rational secret sharing based on extensive games. Jin [27] proposed a rational secret sharing scheme based on a reputation mechanism, which can complete the reconstruction in one round.

To ensure the efficiency of secret sharing, In 1992 Simmons [28] proposed the application scenario of multi-secret sharing. However, this scheme requires participants to store a large number of sub-secret shares, which can be inconvenient to manage and lead to inefficiency in secret reconstruction. In multi-secret sharing, there is a scheme that utilizes the security of RSA digital signature [29] based on the difficulty of large number decomposition. Chen [30] proposed a solution to the prisoner's dilemma based on the blockchain to complete the secret reconstruction and completed a simulation experiment. Yurek [31] proposed a resilient asynchronous complete secret sharing protocol. While these schemes have achieved certain results, there are still some challenges in multi-secret sharing, such as participants not being able to reconstruct the required secrets according to their own needs and the need to ensure trust between participants.

Introducing game theory in secret sharing assumes participants are rational and have a utility function evaluated by themselves. In the reconstruction process, participants act rationally to maximize their own interests. This can lead to dishonest behavior, making it difficult to complete the reconstruction. In traditional multi-secret sharing, the Dealer calculates the sub-secret share of all secrets for the participants, and participants also interact on secrets that are not needed, increasing the overhead. And most existing work does not reduce costs from the perspective of the Dealer [32].

Therefore, in the current rational secret sharing scheme, the problem to be solved is that the participants choose dishonest behaviors in order to maximize their own interests during the reconstruction process, and at the same time reduce the cost of the process while ensuring security. Therefore, based on the reputation mechanism, this paper designs an optional rational multi-secret sharing scheme. Our contribution is as follows:

1 We propose an optional multi-secret sharing scheme in the context of cloud computing. Participants can selectively reconstruct secrets without reconstructing all secrets. Therefore, the Dealer does not need to calculate the sub-secret shares of all secrets for participants, reducing the overhead in the process of secret distribution.
2 We propose a method that only needs one sub-secret share to reconstruct multiple secrets. The Dealer calculates the participants' sub-secret shares during the secret distribution process, participants receive unique shares sent by the Dealer and generate their required sub-secret shares using the content of the broadcasted matrix by the Dealer, which reduces the interaction between the Dealer and the participants, and improves the efficiency of multi-secret sharing.
3 We introduce a reputation mechanism into the scheme and add deposit payments. Participants with high reputation values are preferentially selected for reconstruction. Participants' behavior directly affects the increase or decrease of reputation values, and the payment of deposits can promote participants to complete the reconstruction honestly, avoiding dishonest behavior during the reconstruction process. By leveraging the inherent mathematical properties and computational complexity of elliptic curve cryptography, the sub-secret shares are encrypted, ensuring the overall security of the scheme.

The rest of this paper is organized as follows. The third part introduces the preliminary knowledge that this article needs to understand. In the fourth part, a rational optional multi-secret sharing scheme based on reputation mechanism is described in detail. In the fifth part, this scheme is compared from various aspects, and analyzed for the correctness and security. Finally, the article is concluded in the sixth section.

Preliminary

This part will briefly introduce the concepts of incomplete information game, threshold secret sharing, elliptic curve encryption algorithm, Nash equilibrium in secret sharing.

Incomplete information game

Our scheme combines game theory with secret sharing, and the rational participants in the scheme want to maximize their own benefits, try to keep the secret out of the hands of other participants while getting it for themselves, and the participants in the reconstruction process are aware of the set of strategies F_i in the process, but do not know which strategy $f_{i,j}$ in the set of strategies F_i will be chosen by other participants, which is the extensive incomplete information game [33]. The process of the game is that the participants try to make the most favorable behavioral choice without knowing what strategies the other party will use.

Definition 1

In the game process [34], strategy combination $F_i = f_{i,j}(1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m)$ is $P_i's$ the policy selection set, i is the participant number, j is the secret number:

For example: $F_i = (f_{i,1}, f_{i,3}, f_{i,9} \ldots)$ represents the strategic choice made by this participant on the secret numbered 1, secret numbered 3 and secret numbered 9 and so on.

Definition 2

In the process of game [30], the strategy combination of a secret reconstruction is f_j:

For example: $f_1 = (f_{1,1} \ldots f_{i-1,1}, f_{i,1}, f_{i+1,1} \ldots f_{n,1})$ represents the strategy choice of all participants who need this secret in the reconstruction process of the first secret.

Cloud computing

Cloud computing is an Internet-based computing paradigm that serves as the foundation for the next generation of computing. It provides computational services, including data, storage, software, computing, and applications, to local devices through the Internet. As the popularity of cloud computing increases, it becomes essential to address the security and data protection concerns associated with it. To provide cloud services, sensitive data from all clients needs to be stored in the cloud host, and ensuring data security and personal privacy becomes paramount. Cloud service providers must safeguard this data and personal information from unauthorized access by both internal and malicious external parties. Consequently, several secure cloud computing schemes based on secret sharing approaches have been proposed [35].

Threshold secret sharing

Shamir (t, n) threshold secret sharing scheme constructs $t-1$ degree polynomial according to the threshold value, takes the secret S as a constant term, and calculates the function value at n points as the sub-secret share of n participants. Split the secret into n copies and send them to n participants, so that any participant greater than or equal to t can reconstruct the secret S, and less than t participants can not obtain any information of the secret S [8]. The specific secret distribution and reconstruction process is as follows.

Secret distribution stage:

(1) Dealer selects a polynomial of degree $t-1$:

$$f(x) = S + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \tag{1}$$

Where $a_0 = S$ is a secret value, $a_1, a_2, \ldots, a_{t-1}$ is a finite field Randomly selected elements in F_q.

(2) The Dealer calculates $f(i)$ as the sub-secret of participant $P_i (i = 1, 2 \ldots n)$ and send them to the corresponding participants.

Secret reconstruction stage:

(3) The participants who perform the secret reconstruction send the sub-secret $f(i)$ to other participants respectively.

(4) Participants calculate the secret according to the Lagrange interpolation formula:

$$S = \sum_{i-1}^{t} f(i) \prod_{j=1, j \neq i}^{t} \frac{-j}{i-j} \tag{2}$$

Shamir (t, n) threshold secret sharing scheme meets the following requirements:

(1) Any number of participants greater than or equal to t can reconstruct the polynomial $f(x)$ according to the Lagrange interpolation formula, so that the secret S can be recovered correctly and meet the correctness.

(2) Any less than t participants can not reconstruct the polynomial $f(x)$, so they can not reconstruct any information about secret S and meet the security conditions.

Therefore, Shamir (t, n) threshold scheme is a complete secret sharing scheme [36].

Elliptic curve cryptography

Elliptic curve cryptography (ECC), an algorithm for establishing public key encryption, is an asymmetric encryption algorithm based on elliptic curve mathematical theory. The use of elliptic curve [37] in cryptography was independently proposed by Neal Koblitz and Victor Miller in 1985. The main advantage of ECC is that in some cases it uses a smaller key than other methods, providing equivalent or higher level of security.

The principle of elliptic curve encryption and decryption algorithm: set the private key and public key as d and Q respectively, and calculate $Q = dG$, where G is a base point on the ellipse. If G and dG are known on the elliptic

curve, it is very difficult to calculate d, that is, if the public key and base point are known, it is very difficult to calculate the private key.

In the elliptic curve encryption algorithm [38], ECC is based on the properties of a specific type of equation generated by an array, which is composed of points where lines and axes intersect. Multiplying a point on the curve by a number can produce another point on the curve, but even if you know the original number and result, it is difficult to find the original number. The equation based on elliptic curve has a very valuable property for encryption technology, relatively speaking, the equation is relatively easy to execute, but the reverse is very difficult.

Nash equilibrium

Nash equilibrium [22], also known as non-cooperative game equilibrium, is an important term in game theory. In the process of game, regardless of the other party's strategy choice, one party will choose a certain strategy, which is called the dominant strategy. If any participant chooses the best strategy when the strategies of all other participants are determined, then this combination is defined as Nash equilibrium.

Definition 3

In the process of game, Nash equilibrium $f^* = (f_i^*, f_{-i}^*)$ [39], shows that it is impossible to get higher benefits when participants get out of Nash equilibrium:

$$u_i(f_i, f_{-i}^*) \leq u_i(f_i^*, f_{-i}^*) \tag{3}$$

In other words, when the whole game reaches Nash equilibrium, the strategy chosen by the participants is the best response to the strategy chosen by other participants, and no participant can get higher profits beyond Nash equilibrium.

Prisoner's dilemma [40] is a representative example in game theory, reflecting that the best choice for individuals is not the best choice for teams. In the natural state, participants will fall into the prisoner's dilemma in the process of secret reconstruction, and rational participants will choose dishonest behavior to maximize their own benefits. Participants' behaviors [41] includes sending the sub-secret share honestly, refusing to send the sub-secret share and sending the wrong sub-secret share (We call refusing to send sub-secret shares and sending wrong sub-secret shares as dishonesty), The honest participants will only follow the rules of the scheme and will not perform malicious operations [42].

The following is what happens when each participant chooses different strategies and the different benefit results (benefit: $a > b > c > d$) that can be obtained by choosing different strategies:

1. $u(b, b)$: Participants choose honest behavior, and other participants choose honest behavior;
2. $u(d, a)$: Participants choose honest behavior, and other participants choose dishonest behavior;
3. $u(a, d)$: Participants choose dishonest behavior, and other participants choose honest behavior;
4. $u(c, c)$: Participants choose dishonest behavior, while other participants choose dishonest behavior.

From the above benefits, it can be seen that the participants will get the highest benefits if they choose dishonest behavior. Therefore, in the process of game, if there are no other measures, the participants will choose dishonest behavior to maximize their own interests, and the reconstruction will eventually fall into the prisoner's dilemma.

ROMSS

In this section, we propose a rational optional multi-secret sharing scheme based on reputation mechanism in the context of cloud computing. Due to the efficiency problem of multi-secret sharing, we introduce the concept of optional into multi-secret sharing, reduce the data transmission between the Dealer and participants, and describe in detail how this scheme can reconstruct multiple secrets through a single sub-secret, and introduce reputation mechanism, Ensure the safety and reliability of the reconstruction process, quantify the credibility value of participants, so that participants can only take honest actions to participate in the reconstruction, and realize the maximization of benefits. Before the detailed description of the scheme in this paper, we define the specific symbols required in Table 1.

$A = \{honest, cheat, silent\}$ represents that participants can choose to be honest, lying and silent.

$U = \{a, b, c, d\}$ where

a: The secret is obtained by itself but not by other participants.

Table 1 Symbol description

	description
n	Total number of participants
P_i	The i'th participant
M	Set of all secrets
m	Total number of secrets
S_i	The i'th secret
R_{S_i}	The evaluation value of the i'th secret
B_{P_i}	Security deposit of the i'th participant
A	Set of behaviors
U	Set of utility

b: They and other participants get secrets.

c: Neither he nor other participants can get secrets.

d: They can't get secrets while other participants get secrets.

Obviously, under the condition that all participants are rational, the order of benefits is $a > b > c > d$.

Reputation mechanism

In this scheme, the evaluation function of reputation mechanism is added. Participants choose partners according to the reputation value. We control the reputation value from multiple aspects. The reputation value of participants is updated after each reconstruction, so that participants can choose partners according to the reputation value before the next reconstruction.

Before each round of reconstruction, there will be a stage of reputation value update. Participants will choose partners according to the level of reputation value, according to the reputation value formula $X_{i,k} = A_{i,k} + B_{i,k} + C_{i,k}$ [43], we can consider the historical reputation value of participant i, the participation degree of historical reconstruction, and the factors of evaluating the increase and decrease of participant $i's$ reputation value. Because in [43] considers the degree of time decay. However, in this scheme, the reputation value of participants will be reduced as long as they take dishonest actions, and the impact is not affected by the distance of time, we evaluate the reputation value from three aspect [44]. Therefore, the new credit mechanism evaluation function is designed as follows.

$$V_{i,k} = V_{i,k-1} + Y_{i,k} + C_{i,k} \tag{4}$$

Where $V_{i,k-1}$ is the reputation value of participant i in the previous round, $Y_{i,k}$ is the influence degree of historical behavior, $C_{i,k}$ is the historical reconstruction participation of participants.

The influence degree of historical behavior:

$$Y_{i,k} = \begin{cases} \sum_{k=1}^{r} e^{-V_{i,k-1}} & T_{in} = 1 \\ 0 & T_{in} = 0 \\ \sum_{k=1}^{r} e^{-\frac{V_{i,k-1}^2}{2}} - 1 & T_{in} = -1 \end{cases} \tag{5}$$

Where r is the number of rounds of reconstruction. T_{in} is the refactoring behavior T_n that identifies participant i, The value of n is an evaluation method for participants to participate in reconstruction, which can be expressed as:

$$T_n = \begin{cases} 1 & honest \\ 0 & not\ participant \\ -1 & dishonest \end{cases} \tag{6}$$

where $e^{-V_{i,k-1}}$ is when the participants participate in the reconstruction honestly to increase the reputation value, but the increase speed will gradually slow down to prevent the participants' reputation value from expanding indefinitely. $e^{-\frac{V_{i,k-1}^2}{2}}$ is a normal distribution function, which is used to reduce the reputation value of participants when they make dishonest behaviors in the reconstruction process. Due to the curve of normal distribution, when the reputation value of participants is high, the greater the punishment they will receive for dishonest behaviors, and the more the credit value will be deducted.

The historical reconstruction participation:

$$C_i = \frac{T_i}{T} \tag{7}$$

Where T is the sum of reconstruction times in the past period, T_i is the total number of times that participant i participated in the reconstruction process in the past period. C_i can reflect the degree of participation of participant i. The higher the degree of participation of participant i, the greater the positive impact on reputation evaluation.

Scheme process

In the following, we will describe the specific process of multi-secret sharing, the work of Dealer and participants in each stage, and the general diagram of this program is shown in Fig. 2: In the figure, the secret sharing is divided into the secret distribution phase and the secret reconstruction phase, where the Dealer sends the secret number for statistics, calculates the secret value and the sub-secret share and forms the encryption matrix and broadcasts it to the participants, participants select the participants with high reputation value for cooperative interaction, and finally sends the reconstructed secret for verification to complete the whole process of secret sharing.

Secret distribution phase

Step 1 $Dealer \rightarrow M = \{S_1, S_2 \ldots S_m\}$ and broadcasts set$\{S_1, S_2 \ldots S_m\}$:

Dealers will share m secrets among n participants, and the Dealers will share m secrets $\{S_1, S_2 \ldots S_m\}$ according to the secret information S_{inf} numbers all secrets, and broadcasts the number set$\{S_1, S_2 \ldots S_m\}$through the broadcast channel.

Step 2 Define $\{S_i\}_{i \in need_{P_i}}$ and $PK_{Dealer} \rightarrow \{S_i\}_{i \in need_{P_i}}$:

Participants select secret S according to their own needs, and define a secret selection set $S = \{S_1, S_2 \ldots S_m\}$, where $S_i \in \{0, 1\}$ indicates whether the i'th secret is selected and uses the public key of the Dealers $PK_{Dealer} \rightarrow \{S_i\}_{i \in need_{P_i}}$ is encrypted, packaged and sent to the Dealers.

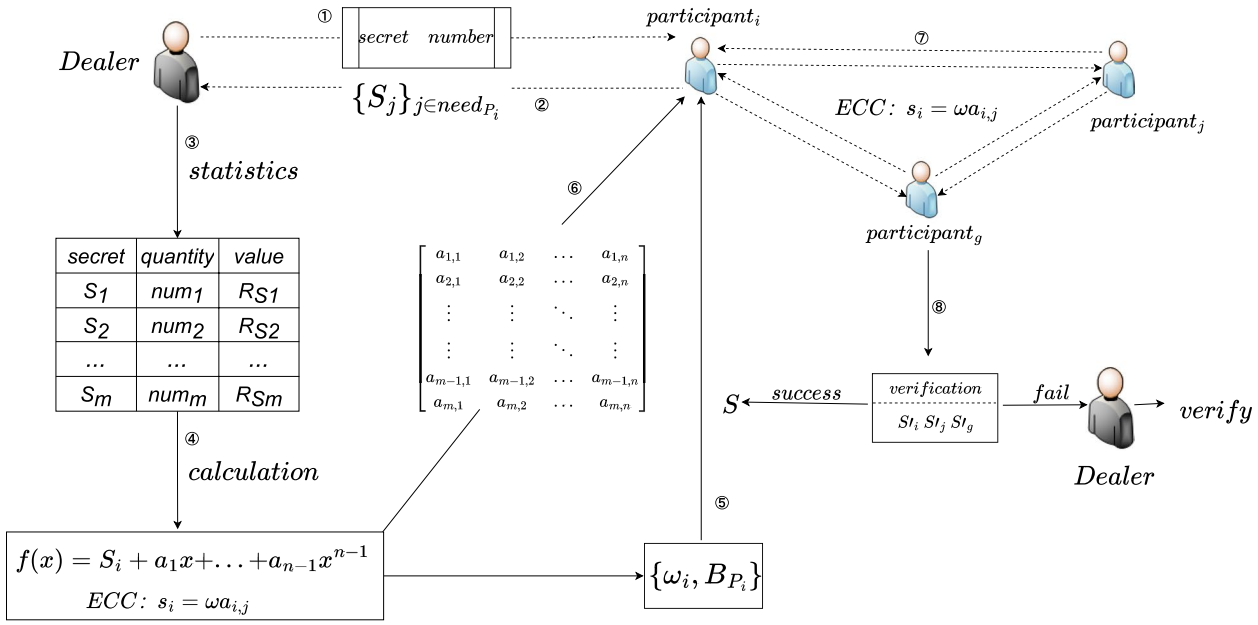

Fig. 2 View of the scheme

Step 3 Calculate value of secret R_{S_i}:

The Dealers will receive the encrypted secret set $PK_D\{S_i\}$ uses the private key to match its $SK_{Dealer} \rightarrow S_{i\{i\in need_{P_i}\}}$ to decrypt and count the required number of each secret to form a list of required numbers, $num_S = \{num_{S_1}, num_{S_2} \ldots num_{S_m}\}$, num_{S_i} represents the required number of the i'th secret. In this scheme, the value R_{S_i} of each secret is determined by its demand. The more participants want to reconstruct the secret, we think its value is greater, and the less participants need the secret, we think its value is smaller. So in this scheme, we designed a secret value evaluation formula:

$$R_{S_i} = a - e^{-num_{S_i}} \tag{8}$$

Where a denotes the highest value of the secret, num_{S_i} denotes the quantity of the i-th secret required by participants. the basic idea of the formula is that the value of the secret is positively correlated with the number of its needed, the higher the value of the secret with the higher number of needed, and the lower the value of the secret with the lower number of needed. We simulate the value assessment formula, the value of the secret is positively correlated with the number of needed, as shown in Fig. 3. We calculate the values of different secrets based on the number of times they are selected. The secret value (vertical axis representing the value of secret S_i) increases with the number of desired shares (horizontal axis representing the quantity of shares required for secret S_i by participants). This indicates that the value of a secret increases as more participants require it. However, to prevent the value from growing infinitely, the curve eventually converges to a certain value, ensuring a limited range for the secret's value, and e^{-x} in the evaluation formula is a kind of monotonically decreasing function, and as the number of desired increases, e^{-x}decreases, then $a - e^{-x}$ increases, and the whole value range is kept between [0, a], so that it does not make the value of the secret too large and lead to the participants' inability to submit the margin of the secret value.

Step 4 Calculate value of secret B_{P_i}:

The Dealer calculates the total value of the secret required by each participant according to the secret number sent by the participant, which corresponds to the deposit required by the participant:

$$B_{P_i} = \sum_{i\in need_{p_i}} R_{S_i} \tag{9}$$

where $need_{p_i}$ denotes the secret required by participant i. In this scheme, when calculating the evaluation value of the secret, we assume that the value of the secret itself is less than the value we set for it. Therefore, when reconstructing the secret, once it is found that the participant has chosen dishonest behavior, it will lose the deposit submitted, which is unfavorable for rational participants.

Step 5 Calculate $f(i)$ and thought $ECC \rightarrow \omega_i a_{i,j} = s_i$:

The Dealer selects a univariate polynomial of order t according to shamir's threshold secret sharing scheme, and the secret S is a constant term: $f(x) = S + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}$, $a_1, a_2 \ldots a_{t-1}$are elements for the randomly selected

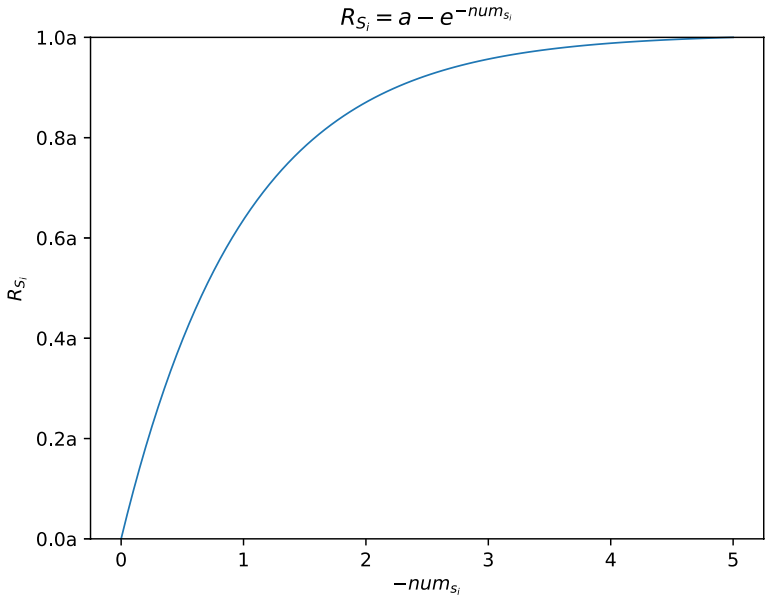

Fig. 3 $R_{S_i} = a - e^{-num_{S_i}}$

in F_q, calculate the sub-secret share $f(i)$ of each participant,i.e. s_i. i is the participant number; For each participant in the finite field F_q select a large number randomly in $\omega_i \in F_q$. Calculate the corresponding number $a_{i,j}$ of each participant in the matrix according to the elliptic curve encryption ECC algorithm $a_{i,j}$: $\omega_i a_{i,j} = s_i$. Send a set to each participant $PK_{P_j}\{\omega_i, B_{P_i}\}$, the set uses the public key of the corresponding participant PK_{P_j}, where the set contains a large number ω_i and the amount of security deposit B_{P_i}.

Step 6 Broadcasts a matrix:

After the participant submits the deposit within the time limit for submitting the deposit, the Dealer broadcasts a matrix to all participants:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ a_{m-1,1} & a_{m-1,2} & \cdots & a_{m-1,n} \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

the row number represents the participant's number, and the column number represents the secret number, where each position corresponds to the number of operations of each sub-secret share of the participant, and the participant sends a large number in the set based on the information given by the matrix and combined with the Dealer ω_i. The real sub-secret share can be obtained by calculation. For example, participant P_1 calculates the sub-secret shares of secret $s_1 = a_{1,1}\omega_1$.

The secret distribution phase of secret sharing is shown in Algorithm 1, where we represent the distribution of m secrets among n participants by the Dealer, and counts the required number of secrets based on the required secret values returned by the participants, the value of the secret is evaluated for each required number of secrets. Then the sub-secret shares are calculated and the encrypted matrix is broadcasted, completing the entire secret distribution phase.

Input: secret S,participants P,Total number of secret m,Total number of participants n,Public key of Dealer PK_{Dealer}.
Output: matrix, $\{\omega_i, B_{P_i}\}$.

for *Dealer* **do**
  $M\{S_1, S_2 \ldots S_m\} \leftarrow$ *number the secret* S
  broadcasts $M\{S_1, S_2 \ldots S_m\}$
end
for *each* P_i *in* P **do**
  Select required secret need$_i$ *from* M
  $Dealer \leftarrow Enc_{PK_{Dealer}}(need_i)$
end
for *Dealer* **do**
  $SK_{Dealer} \rightarrow S_{i(i \in need_{P_i})}$
  Statistics $S \rightarrow num_S = \{num_{S_1}, num_{S_2} \ldots num_{S_m}\}$
  value assessment $\rightarrow R_{S_i} = a - e^{-num_{S_i}}$
  bond $B_{P_i} = \sum_{i \in need_{P_i}} R_{S_i}$
  choose a polynomial to calculate subsecret $f(x) = S + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$
  randomly select a number ω_i *and* $\omega_i a_{i,j} = s_i$
  broadcast/*send a matrix include* $a_{i,j}$
end

Algorithm 1 The Dealer distributes m secrets among n participants

Secret reconstruction stage

Step 1 Send $PK_j \rightarrow s_i$ to P_i and receive $PK_i(s_j)$:

In the reconstruction process, participants combine according to their own strategies F_i Select participants with high reputation value for cooperative reconstruction. The strategy set f adopted by each participant after the reconstruction starts will reach a Nash equilibrium state f^*, and the participants will calculate the sub-secret share s_i. Use the public key PK of the receiver participant and send it to the receiver $PK_j \rightarrow s_i$. At the same time, receive the sub-secret share sent by other participants $PK_i(s_j)$.

Step 2 $SK_i \rightarrow s_j$ and calculate S:

Participants decrypt the received sub-secret share with private key SK: $SK_i \rightarrow s_j$. According to the sub-secret share of itself and other participants, Lagrange algorithm $S = \sum_{i-1}^{t} f(i) \prod_{j=1, j\neq i}^{t} \frac{-j}{i-j}$ calculates the final secret S.

Step 3 Verify the correctness of the secret S:

After the reconstruction of the secret is completed, the participants will publish and verify the reconstructed secret on the synchronous broadcast channel. When the secret value reconstructed by all participants is the same ($S_{P_1} = S_{P_2} = \cdots = S_{P_t}$), the Dealer sends: the secret is the same. However, once different secret values appear in the reconstruction secrets sent by all reconstruction participants ($S_{P_1} = S_{P_2} = \cdots \neq S_{P_t}$), the Dealer sends: the secrets are not the same. it means that some participants have taken dishonest behavior during the reconstruction process, and the Dealer immediately verify the participants in the reconstruction phase. Verify according to the sub-secret share s_i sent by each participant during the reconstruction process, and the last secret S' sent, and find out dishonest participants to confiscate the deposit and reduce the reputation value penalty, then send the real secret value S to participants who honestly participate in the reconstruction.

Step 4 Honest P_i update reputation values:

After the reconstruction is completed, the reputation value is updated. Then the last round of refactoring is over, the reputation value of each participant is updated. When the refactoring is safely executed, it means that all participants are honestly refactoring, and the reputation value will increase accordingly. Through $X_{i,k} = V_{i,k-1} + Y_{i,k} + C_{i,k}$ to calculate a new round of reputation value, where $Y_{i,k} = \begin{cases} \sum_{k=1}^{r} e^{-V_{i,k-1}} & T_{in} = 1 \\ 0 & T_{in} = 0 \\ \sum_{k=1}^{r} e^{-\frac{V_{i,k-1}^2}{2}} - 1 & T_{in} = -1 \end{cases}$. Because the participants participate honestly, they only need to increase the reputation value of the previous round through the logarithmic function of $e^{-V_{i,k-1}}$ to calculate the reputation value of the new round.

Step 5 Dishonest P_i update reputation values:

When the reconstruction is completed and enters the verification phase, if some participants are detected to have dishonest behavior in the reconstruction phase, the change of the reputation value will be greatly reduced, and the dishonest participants will pass the $e^{\frac{-V_{i,k-1}^2}{2}} - 1$ normal distribution function on the reputation value of the previous round to reduce, due to the nature of the normal distribution function, which is high in the middle and low in both ends, it is calculated on this basis and subtracted by one, so that the participants with reputation value will be punished for dishonest behavior, and the reputation value will be reduced more. Through this, the reputation value of the new round is calculated.

The secret reconstruction phase in secret sharing is shown in Algorithm 2, where we indicate that participants interact to complete the secret reconstruction. Participants select other participants with high reputation value for cooperative reconstruction, perform the calculation of sub-secret shares based on the encryption matrix broadcasted by the Dealer and interact with the cooperators. The reconstruction of the secret is completed by the Lagrange calculation with the received sub-secret shares, and the reconstructed secret is broadcasted for verification, followed by the verify of the Dealer, if the participants broadcast the same secret, the reconstruction is considered to be executed correctly and the secret is reconstructed successfully. Once the process has different secrets, the Dealer verifies the participants, punishes the dishonest participants, and sends the correct secret value to the honest participants, completing the whole secret reconstruction phase.

Input: sub-secret s_i
Output: The secret after reconstruction S
Initialize $R_S \leftarrow \{0, 0, ..., 0\}$
for *each* P_i *in* P **do**
  Choose cooperative partners with high reputation values to reconstruct secret
  $s_i = a_{i,j} \times \omega_i$
  send $Pk_{P_j}\{s_i\} \rightarrow P_j$
  $Enc_{Sk_{P_i}} \leftarrow P_j^{'}$ *sub − secret*
  $S = \sum_{i-1}^{t} f(i) \prod_{j=1, j\neq i}^{t} \frac{-j}{i-j}$
  broadcastsend S
end
for *Dealer* **do**

  check if all Secret are the same
  if *Secret is same* **then**
    send to all P_i("*Secret is same*")
  else if *Secret is not same* **then**
    send to all P_i("*Secret is not same*")
    check all sub − secret form all participants to find malicious P_i
    update malicious P_i' *Reputation value*
    deduct confiscation of the malicious P_i' *deposit*
    correct $S \rightarrow honest(P_i)$
  end
end

Algorithm 2 P_i calculate the sub-secrets and participate in reconstruction

Scheme analysis

This scheme uses shamir's (t, n) threshold secret sharing scheme for secret reconstruction according to the nature of Lagrange algorithm. A reputation mechanism is added to the scheme, and participants can choose participants with high reputation value for cooperative reconstruction to ensure the reliability of the reconstruction environment. The scheme is full of rational participants, and the self-interested participants are the ones who will choose honest behavior to maximize their own interests. The proof of this scheme is given in detail below.

Correctness

This scheme is based on Shamir's (t, n) threshold secret sharing. In the sub-secret distribution stage, the Dealer uses an unary polynomial $f(x) = S + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}$ with the constant item as the secret value S calculates the sub-secret share $f(i)$, where the sub-secret share $f(i) = \omega_i a_{i,j}$, and the participants receive The received share is ω_i, but the secret share for interacting with other participants is $f(i)$ calculated by the participant based on the ω_i sent by the Dealer and $a_{i,j}$ in the broadcast matrix. In the reconstruction phase, the participant use Lagrange interpolation method $S = \sum_{i-1}^{t} f(i) \prod_{j=1,j\neq i}^{t} \frac{-j}{i-j}$ for calculation:

$$\begin{aligned} f(0) =& f(1)\left(\frac{0-x_2}{x_1-x_2}\frac{0-x_3}{x_1-x_3}\cdots\frac{0-x_t}{x_1-x_t}\right) \\ &+\cdots+f(t)\left(\frac{0-x_1}{x_t-x_1}\cdots\frac{0-x_{t-1}}{x_t-x_{t-1}}\right) \end{aligned} \tag{10}$$

where $f(0)$ is the constant term S, the secret value, in the $t-1$st order equation. The participants are calculated by the participant's number and the sub-secret shares received from other participants, and the secret S is recovered according to the Lagrange interpolation method $f(i)\prod_{j=1,j\neq i}^{t}\frac{-j}{i-j}$, and since the equation is a polynomial of order $t-1$, a minimum of t sub-secret shares is required in the calculation to solve for the secret S. Otherwise, no secret S can be obtained for any information.

$$\begin{aligned} S =& s_1\left(\frac{0-2}{1-2}\frac{0-3}{1-3}\cdots\frac{0-t}{1-t}\right)+\cdots+s_t\left(\frac{0-1}{t-1}\frac{0-2}{t-2}\cdots\frac{0-(t-1)}{t-(t-1)}\right) \\ =& s_1\prod_{j=1,j\neq 1}^{t}\frac{-j}{1-j}+s_2\prod_{j=1,j\neq 2}^{t}\frac{-j}{2-j}+\cdots+s_t\prod_{j=1,j\neq t}^{t}\frac{-j}{t-j} \\ =& \sum_{i=1}^{t} s_i\prod_{j=1,j\neq i}^{t}\frac{-j}{i-j} \end{aligned}$$

During the reconstruction phase, participants calculate their sub-secret shares based on the matrix sent by the Dealer, $s_i = \omega_i a_{i,j}$, which is the inverse process of secret distribution. By doing so, participants are able to reconstruct the sub-secret shares accurately. Afterwards, participants reconstruct the secret using Lagrange interpolation, as shown in Eq. 10. Thus, the entire secret sharing process ensures computational correctness.

Security analysis

In the multi-secret sharing process, we want to prevent participants from maliciously calculating the large number ω_i owned by other participants in the reconstruction process. Using ECC encryption method to ensure the security of the reconstruction process, as well as the dishonest behavior of rational participants in the reconstruction process, in order to maximize their own interests, leading to the prisoner's dilemma in the reconstruction process. Conduct behavioral analysis of participants, and finally ensure that participants will all choose honest behavior to secretly reconstruct and achieve behavioral security among participants.

Theorem 1 During the reconstruction process, ECC encryption ensures that participants cannot launch the received information of other participants through the interacting sub-secrets, achieving the security of the reconstruction.

proof: During the reconstruction process, the participant calculates the sub-secret s_i of the S_i secret according to the ω_i sent by the Dealer and $a_{i,j}$ in the broadcast matrix, and sends s_i to other For the participants, the Dealer uses the ECC encryption algorithm to encrypt the sub-secret s_i, and the participants cannot deduce the large number ω_i owned by other participants through the received sub-secret share, the proof is as follows:

In the elliptic curve encryption algorithm, the principle of the encryption and decryption algorithm is as follows:

$$Q = dG \tag{11}$$

where G is a base point on the ellipse, and G and dG are known on the elliptic curve. d is very difficult, that is to say, knowing the public key and the base point, it is very difficult to calculate the private key: This scheme finds a number d for two points Q and G on the elliptic curve, so that $Q = dG$. Where the point on the elliptic curve satisfies a specific equation, namely

$$y^2 = x^3 + ax + b \tag{12}$$

a and b are curve parameters, and Q and G are points on the curve. Now suppose that the participants only know the curve parameters a and b, and the coordinates of points Q and G, namely

$$Q = (x_Q, y_Q), G = (x_G, y_G) \tag{13}$$

The goal of the participant is to find the number d so that$Q = dG$. To solve this problem, participants need to

find the possible point dG on the curve to verify which is Q. Because the number of points on the curve is very large, it is unrealistic to enumerate all possible points.

In the reconstruction process, participants calculate the sub-secret share through large numbers and the number of operations in the matrix:

$$s_i = \omega_i a_{i,j} \tag{14}$$

Therefore, when a participant receives the sub-secret share s_i sent by other participants, where Q is the received sub-secret share s_i, d is the large number ω_i of other participants, and G is given in the broadcast matrix The operand $a_{i,j}$, the equation is easy to execute in the forward direction, that is, it is easy to find s_i, after knowing ω_i and $a_{i,j}$, but it is very difficult to calculate in reverse. Knowing s_i, and $a_{i,j}$ but not finding ω_i.Therefore, in the reconstruction phase, participants cannot deduce the large number ω_i owned by other participants, which ensures the security of reconstruction in calculation.

Theorem 2 In terms of behavioral choice, the increase or decrease of reputation value makes rational participants behave honestly, in order to maximize their own interests and ensure the behavioral safety of rational participants in the reconfiguration process.

Proof: In our scheme, when participants make behavior choices, they will eventually choose honest behaviors to maximize their own interests, and the whole process will not fall into the prisoner's dilemma, so the reconstruction can also be performed correctly, as shown below:

In Fig. 4, we can see that participants will eventually choose honest behaviors to carry out the reconstruction process, and the path is the path marked in red. Assuming that there are three participants in the process of refactoring. For example:

- $for\ P_3 : u_3(honest) = b + d + d + a > u_3(dishonest)$
  $= a + d - B + d - B + d - B$

P_3 will choose honest behavior to maximize its own interests.

- $for\ P_2 : u_2(honest) = b + a + d + d - B > u_2(dishonest)$
  $= d + d - B + a + d - B$

P_2 will choose honest behavior to maximize its own interests.

- $for\ P_1 : u_1(honest) = b + d + a + d - B > u_1(dishonest)$
  $= d + a + d - B + d - B$

P_1 will choose honest behavior to maximize its own interests.

Because in the reconstruction process, once a participant has committed dishonest behavior and is discovered, the Dealer will verify the entire reconstruction process. Once the dishonest participant is found, the deposit B_{P_i} will be deducted and the reputation value will be deducted $V_{i,k}$ greatly reduces the penalty.

In Table 2, we can see the utility results of participants under different behaviors, and we show the results of each participant performing honest behaviors as well as dishonest behaviors. The final benefit of dishonest behaviors is lower than that of honest behaviors, and under the evaluation of reputation mechanism, participants choose honest behaviors in order to maximize their own benefits, proving that the scheme can motivate participants to behave honestly and finally achieve the security of the reconstruction process.

After the reconstruction phase, there is a verification stage where participants broadcast the reconstructed secret. If different reconstructed secrets are observed during the broadcast, the secret distributor verifies the participants involved in the reconstruction process using the received sub-secret shares. If any participant is found to have distributed incorrect sub-secret shares maliciously, the secret distributor deducts the submitted deposit from that participant and significantly lowers their reputation score. The correct secret is then sent to other honest participants. This effectively prevents malicious behavior by participants and ensures the security of the scheme to a certain extent.

Performance analysis

In this scheme, we propose the concept of optionality into the secret sharing, so that participants do not need to reconstruct all the secrets in the multi-secret sharing, but only need to select and reconstruct the secrets according to their own needs, which gives participants more choices than the previous multi-secret sharing schemes. It also improves the efficiency of reconstruction. The Dealer does not need to calculate sub-shares of all secrets for every participant, reducing the computational overhead for the Dealer. In this paper, we propose a sharing scheme that only requires a single sub-secret share to reconstruct multiple secrets, compared to the scheme of, which requires a number of sub-secret shares to reconstruct multiple secrets, and this scheme reduces the transmission data in the generation of sub-secret shares. In this paper, a single sub-secret share can reconstruct multiple secrets due to an ECC-based encryption scheme that allows the participants themselves to calculate the secret shares by means of a broadcast matrix, while the Dealer only needs to send a single share. A single sub-secret share can reconstruct multiple secrets, reducing the transmission cost and the interaction

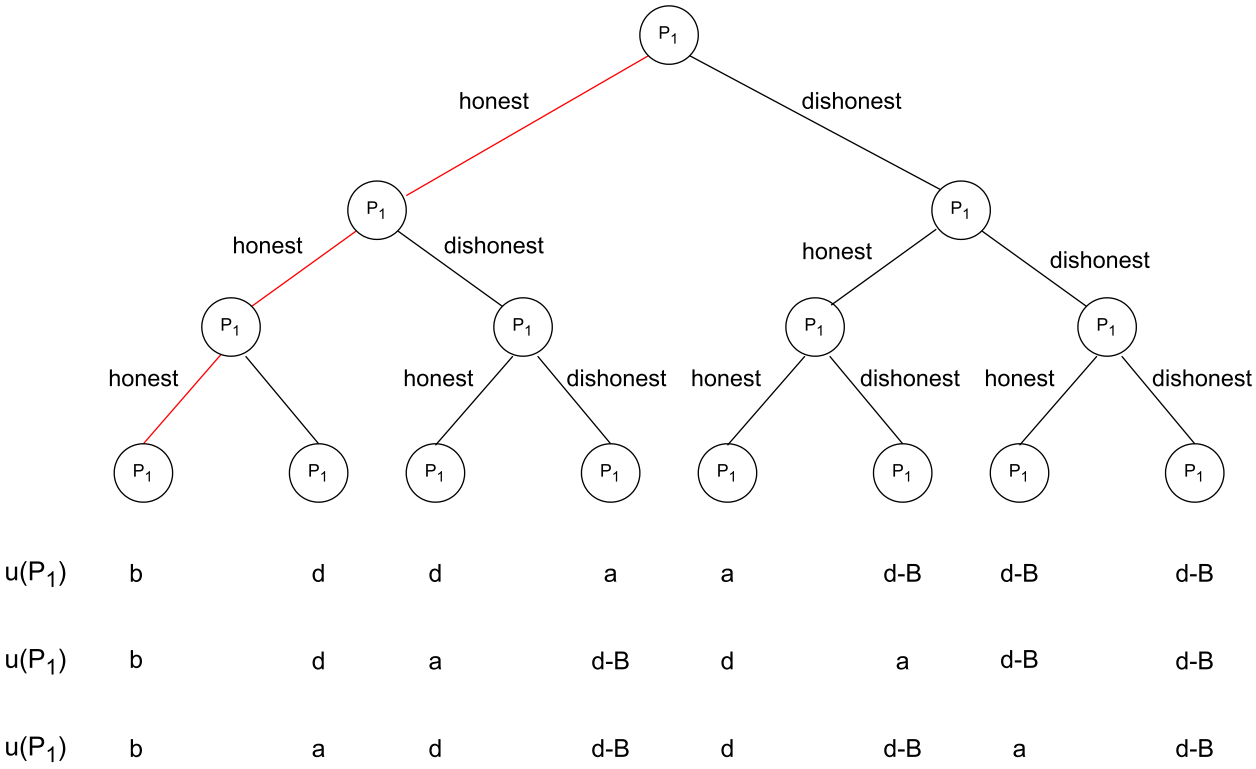

Fig. 4 Game

Table 2 Utility of participants

	P_1	P_2	P_3	$u(P_1)$	$u(P_2)$	$u(P_3)$
1	honest	honest	honest	b	b	b
2	honest	dishonest	honest	d	a	d
3	honest	honest	dishonest	d	d	a
4	honest	dishonest	honest	a	$d-B$	$d-B$
5	dishonest	honest	honest	a	d	d
6	dishonest	dishonest	honest	$d-B$	$d-B$	a
7	dishonest	honest	dishonest	$d-B$	a	$d-B$
8	dishonest	dishonest	dishonest	$d-B$	$d-B$	$d-B$

frequency between the Dealer and the participants. As shown in Table 3 , the functional advantages of this scheme are compared with other schemes.

Since an optional multi-secret sharing scheme is proposed, participants also do not need to reconstruct all secrets in the secret reconstruction stage, but only need to reconstruct the secrets they need, so the number of reconstruction rounds is reduced, and the number of reconstruction rounds of participants in this scheme is only related to the number of secrets they need θ, and other schemes need to reconstruct all secrets, so the number of reconstruction rounds is the total number of secrets m. In this scheme, the Dealer needs to generate m polynomials and calculate the sub-secret shares of participants and encrypt them with ECC, then the computational complexity of the secret distribution phase is $O(N^2)$, and in the secret reconstruction phase, participants need to calculate the sub-secret shares of their own required secrets and interact to calculate the secret S. The computational complexity also reaches $O(N^2)$. In the secret distribution phase, m secret numbers need to be broadcasted, and participants also need to encrypt the selected secrets and transmit them to the Dealer, after which the Dealer sends the calculated sub-secret

Table 3 Comparison with existing MSS schemes

	Optional	Single secret share	Hardness assumption
TangZhang [45]	×	×	polynomial
Fulin Li [46]	×	×	DH
Zhangjian [36]	×	✓	DLP
Our scheme	✓	✓	Bilinear Maps

shares to n participants and then sends the matrix as well as the sub-secret shares, and the total data transmission is $O(N^2)$. Compared with other multi-secret sharing schemes, our scheme is reduced in terms of computational complexity as well as the amount of data transmission. As shown in Table 4, Comparison of overhead between our scheme and other schemes.

Conclusion

Secret sharing has various practical applications in cloud computing scenarios, such as key transfer protocols, attribute-based encryption, and secure multi-party computation. Secret sharing has always been applicable to cloud computing, where multiple users distribute their data to servers. However, when applying secret sharing to cloud systems, the number of shares can be significant. In this paper, we propose a secret sharing scheme that reduces the number of shares while ensuring computational security when used in cloud systems. We approach rational secret sharing as a game process where participants make behavioral choices based on their interests and eventually reach a Nash equilibrium for the whole process. The development from secret sharing to multi-secret sharing has improved the reconstruction efficiency but increased the communication overhead during the interaction process. Therefore, we propose an optional multi-secret sharing scheme that enables users to select secrets on-demand for reconstruction and reduces interaction. The scheme evaluates the value of secrets based on demand and incorporates a reputation mechanism to calculate participants' reputation value to prevent dishonest behavior and ensure the security of the entire process. In this scheme, the secret distributor utilizes elliptic curve encryption for the broadcast matrix during secret reconstruction, allowing participants to reconstruct multiple secrets with only one share sent by the secret distributor. For future research, we aim to minimize communication overhead while improving efficiency and addressing other issues in multi-secret sharing. We also aim to introduce the reputation mechanism into more secret sharing schemes, which can serve as a constraint for participants and a safety evaluation criterion to ensure the sharing process is carried out safely and correctly in cloud computing scenarios. By closely integrating with the practical application requirements of cloud computing, we can further promote the development and application of secret sharing in the field of cloud computing.

Table 4 Comparison with existing MSS schemes

	Reconstruction rounds	data transfer volume	algorithmic complexity
TangZhang [45]	m	$O(N^3)$	$O(N^2)$
Fulin Li [46]	m	$O(N^3)$	$O(N^2)$
Zhangjian [36]	m	$O(N^3)$	$O(N^3)$
Our scheme	θ	$O(N^2)$	$O(N^2)$

Acknowledgements
We are thankful to State Key Laboratory of Public Big Data of Guizhou University for providing an environment for editing manuscripts.

Authors' contributions
Ruonan Shi wrote the main manuscript text, Chaoyue Tan and Yun Luo prepared tables and figures, Yuling Chen and Tao Li provided helpful suggestions and revised the manuscript. All authors reviewed the manuscript.

Funding
This research is supported in part by the National Natural Science Foundation (61962009, 62202118), in part by Top Technology Talent Project from Guizhou Education Department (Qianjiao ji [2022]073).

Availability of data and materials
The materials generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 18 April 2023 Accepted: 28 July 2023
Published online: 05 August 2023

References

1. Zhou X, Yang X, Ma J, Kevin I, Wang K (2021) Energy-efficient smart routing based on link correlation mining for wireless edge computing in IoT. IEEE Internet Things J 9(16):14988–14997
2. Qi L, Yang Y, Zhou X, Rafique W, Ma J (2021) Fast anomaly identification based on multiaspect data streams for intelligent intrusion detection toward secure industry 4.0. IEEE Trans Ind Inf 18(9):6503–6511
3. Zhou X, Xu X, Liang W, Zeng Z, Yan Z (2021) Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart IoT. IEEE Internet Things J 8(16):12588–12596
4. Zhou X, Liang W, Yan K, Li W, Kevin I, Wang K, Ma J, Jin Q (2022) Edge-enabled two-stage scheduling based on deep reinforcement learning for internet of everything. IEEE Internet Things J 10(4):3295–3304
5. Li Z, Xu X, Hang T, Xiang H, Cui Y, Qi L, Zhou X (2022) A knowledge-driven anomaly detection framework for social production system. IEEE Trans Comput Soc Syst pp 1–14. https://doi.org/10.1109/TCSS.2022.3217790

6. Kong L, Li G, Rafique W, Shen S, He Q, Khosravi MR, Wang R, Qi L (2022) Time-aware missing healthcare data prediction based on arima model. IEEE/ACM Trans Comput Biol Bioinforma 1–10. https://doi.org/10.1109/TCBB.2022.3205064
7. Kong L, Wang L, Gong W, Yan C, Duan Y, Qi L (2021) Lsh-aware multitype health data prediction with privacy preservation in edge environment. World Wide Web 25:1–16
8. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613
9. Harn L, Xia Z, Hsu C, Liu Y (2020) Secret sharing with secure secret reconstruction. Inf Sci 519:1–8
10. Yang Y, Yang X, Heidari M, Khan MA, Srivastava G, Khosravi M, Qi L (2022) Astream: Data-stream-driven scalable anomaly detection with accuracy guarantee in IIoT environment. IEEE Trans Netw Sci Eng 1. https://doi.org/10.1109/TNSE.2022.3157730
11. Dai H, Yu J, Li M, Wang W, Liu AX, Ma J, Qi L, Chen G (2022) Bloom filter with noisy coding framework for multi-set membership testing. IEEE Trans Knowl Data Eng 1–14. https://doi.org/10.1109/TKDE.2022.3199646
12. Xu X, Gu J, Yan H, Liu W, Qi L, Zhou X (2023) Reputation-aware supplier assessment for blockchain-enabled supply chain in industry 4.0. IEEE Trans Ind Inf 19(4):5485–5494
13. Qi L, Lin W, Zhang X, Dou W, Xu X, Chen J (2022) A correlation graph based approach for personalized and compatible web apis recommendation in mobile app development. IEEE Trans Knowl Data Eng 1. https://doi.org/10.1109/TKDE.2022.3168611
14. Wu S, Shen S, Xu X, Chen Y, Zhou X, Liu D, Xue X, Qi L (2023) Popularity-aware and diverse web apis recommendation based on correlation graph. IEEE Trans Comput Soc Syst 10(2):771–782
15. Jia Y, Liu B, Dou W, Xu X, Zhou X, Qi L, Yan Z (2022) Croapp: a CNN-based resource optimization approach in edge computing environment. IEEE Trans Ind Inf 18(9):6300–6307
16. Wang F, Wang L, Li G, Wang Y, Lv C, Qi L (2021) Edge-cloud-enabled matrix factorization for diversified apis recommendation in mashup creation. World Wide Web 25:1–21
17. Luo Y, Chen Y, Li T, Wang Y, Yang Y, Yu X (2022) An entropy-view secure multiparty computation protocol based on semi-honest model. J Organ End User Comput 34(10):1–17
18. Zhou X, Hu Y, Wu J, Liang W, Ma J, Jin Q (2022) Distribution bias aware collaborative generative adversarial network for imbalanced deep learning in industrial IoT. IEEE Trans Ind Inform 19(1):570–580
19. Halpern J, Teague V (2004) Rational secret sharing and multiparty computation: Extended abstract. In: Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, New York, NY, USA, STOC '04, p 623–632. https://doi.org/10.1145/1007352.1007447
20. Li T, Wang Z, Yang G, Cui Y, Chen Y, Yu X (2021) Semi-selfish mining based on hidden markov decision process. Int J Intell Syst 36(7):3596–3612
21. Chen Y, Dong S, Li T, Wang Y, Zhou H (2021) Dynamic multi-key fhe in asymmetric key setting from lwe. IEEE Trans Inf Forensic Secur 16:5239–5249
22. Kreps DM (1989) Nash equilibrium. Game theory 167–177. https://doi.org/10.1007/978-1-349-20181-5_19
23. Maleka S, Shareef A, Rangan CP (2008) Rational secret sharing with repeated games. Lect Notes Comput Sci 4991:334–346
24. Ong SJ, Parkes DC, Rosen A, Vadhan S (2009) Fairness with an honest minority and a rational majority. In: Reingold O (ed) Theory of Cryptography, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 36–53. https://doi.org/10.1007/978-3-642-00457-5_3
25. Başar T (2021) Game Theory: A General Introduction and a Historical Overview, Springer International Publishing, Cham, pp 881–886. https://doi.org/10.1007/978-3-030-44184-5_26
26. Zhang Z, Liu M (2011) Unconditionally secure rational secret sharing in standard communication networks. In: Information Security and Cryptology-ICISC 2010: 13th International Conference, Seoul, Korea, December 1-3, 2010, Revised Selected Papers 13, pp 355–369. https://doi.org/10.1007/978-3-642-24209-0_24
27. Jin J, Zhou X, Ma C, Wang X (2016) A rational secret sharing relying on reputation. In: 2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS), pp 384–387. https://doi.org/10.1109/INCoS.2016.40
28. Simmons GJ (1992) An Introduction to Shared Secret and/or Shared Control Schemes and Their ApplicationThis work was performed at Sandia National Laboratories and supported by the U.S. Department of Energy under contract number DEAC0476DPOO789, pp 441–497. https://doi.org/10.1109/9780470544327.ch9
29. Wu J, Tao W (2004) Threshold multi-secret sharing scheme. Acta Electron Sin 32(Supp):1688–1689
30. Chen Z, Tian Y, Peng C (2021) An incentive-compatible rational secret sharing scheme using blockchain and smart contract. Sci China Inf Sci 64:1–21
31. Yurek T, Luo L, Fairoze J, Kate A, Miller A (2021) hbacss: How to robustly share many secrets. Cryptol ePrint Arch. https://doi.org/10.14722/ndss.2022.23120
32. Wang Y, Li T, Liu M, Li C, Wang H (2022) Stsiiml: Study on token shuffling under incomplete information based on machine learning. Int J Intell Syst 37:11078 – 11100
33. Harsanyi JC, Harsanyi JC (1982) Games with incomplete information played by "bayesian" players, i–iii part i. the basic model. Pap Game Theory 115–138. https://doi.org/10.1287/mnsc.1040.0270
34. Liu H, Li X, Tian Y, Luo B, Ma J, Peng C (2020) A rational and fair secret sharing scheme. J Comput Res Dev 43(8):17
35. Yang CN, Lai JB (2013) Protecting data privacy and security for cloud computing based on secret sharing. In: 2013 International Symposium on Biometrics and Security Technologies, pp 259–266. https://doi.org/10.1109/ISBAST.2013.46
36. Zhang J, Lin C, Ding J, Lin X, Li C (2021) A secure multi-use threshold multi-secret sharing scheme. J Comput Syst Appl 30(5):276–281
37. Koblitz N (1987) Elliptic curve cryptosystems. Math Comput 48(177):203–209
38. Liu A, Ning P (2008) Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. In: 2008 International Conference on Information Processing in Sensor Networks (ipsn 2008), pp 245–256. https://doi.org/10.1109/IPSN.2008.47
39. Holt CA, Roth AE (2004) The nash equilibrium: A perspective. Proc Natl Acad Sci 101(12):3999–4002
40. Rapoport A (1989) Prisoner's dilemma. Game Theory pp 199–204. https://doi.org/10.1007/978-1-349-20181-5_23
41. Abraham I, Dolev D, Gonen R, Halpern J (2006) Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing, pp 53–62. https://doi.org/10.1145/1146381.1146393
42. Luo Y, Chen Y, Li T, Wang Y, Yang Y (2021) Using information entropy to analyze secure multi-party computation protocol. In: 2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pp 312–318. https://doi.org/10.1109/DASC-PICom-CBDCom-CyberSciTech52372.2021.00061
43. Xie Z, Zhang Z, Li L, Feng Y, Chen J (2022) Improved practical byzantine fault tolerance algorithm based on consortium blockchain. J Comput Sci 49(11):360–367
44. Chen Y, Sun J, Yang Y, Li T, Niu X, Zhou H (2022) Psspr: a source location privacy protection scheme based on sector phantom routing in wsns. Int J Intell Syst 37(2):1204–1221
45. Zhang T, Ke X, Liu Y (2018) (t, n) multi-secret sharing scheme extended from harn-hsu's scheme. EURASIP J Wirel Commun Netw 2018:1–4
46. Li F, Hu H, Zhu S, Yan J (2022) A fully dynamic multi-secret sharing scheme with redundant authorization. Cryptogr Commun 1–18. https://doi.org/10.1007/s12095-022-00613-3

Publisher's Note