Hindawi
Security and Communication Networks
Volume 2022, Article ID 1595520, 7 pages
https://doi.org/10.1155/2022/1595520

Research Article

Multiparty Strict Coin-Tossing Protocols Based on the Eigenvalue

Zhang YanShuo,[1] **Wang ZeHao**,[2] **Zhang Le**,[1] **and Xia Chao**[1]

[1]*Beijing Electronic Science & Technology Institute, Beijing 100070, China*
[2]*Data Communication Science and Technology Research Institute, Beijing 100191, China*

Correspondence should be addressed to Zhang YanShuo; zhang_yanshuo@163.com

Received 18 November 2021; Revised 26 February 2022; Accepted 19 May 2022; Published 29 May 2022

Academic Editor: Hu Xiong

The coin-tossing protocol is an important research area in cryptography. It generates a random bit with uniform distribution even if some participants might fraud. However, traditional coin-tossing protocol could not solve the situation of multiparty. It only divides participants into two parts. In this paper, a new kind of multiparty strict coin-tossing protocol based on the eigenvalue of matrix was proposed. First, matrix tampering attacks can be resisted. On the other hand, collusion attack which was caused by the addition of the Lagrange interpolation formula could be overcome. The analysis shows that the correctness and security of both protocols was guaranteed. Based on the above statements, comparing with the classic coin-tossing protocols, the proposed scheme has the advantage of resisting parties aborting, low complexity, and practicability.

1. Introduction

In network communication that the communicating party is not in the same geographical position, once the judgment needs to be made, both parties should compare the guessing result and ensure the information is not disclosed at the same time. The coin-tossing protocol can be seen as an application case for secure multiparty computation.

In cryptography, suppose Alice and Bob throw coins, and before the results are revealed, neither side wants to let the other one knows their own result, which is one of the important models for multiparty confidential computing [1]. Obviously, as there is no third-party arbitration, the fairness based on fraud prevention has become the most important consideration for the coin-tossing protocol.

Many scholars have conducted research on the coin-tossing protocol. In 1982, Blum introduced the problem of tossing a fair coin through a modem [2]. In 1990, Ben et al. proposed a coin throw problem in Reference [3]. In 2003, Lindell et al. raised the fair coin-tossing protocol of two-party [4]. Kun et al. raised the coin-tossing protocol based on knapsack problem [5].

Apparently, these protocols are limited to two parties and have not solved the problem of multiparty participation in coin-tossing. On the other hand, they did not solve the problem that all the participants have to decide their order in a fair way rather than be divided into two parts.

In this paper, based on the matrix eigenvalues and eigenvectors, we have first proposed a new kind of strict multiparty coin-tossing protocol. Furthermore, we applied the Lagrange interpolation formula to design an improved strict multiparty tossing protocol which can resist collusion attacks. At last, analysis of both protocols and specific examples are proposed.

2. Basic Knowledge

2.1. Coin-Tossing Protocol. The definition of coin-tossing protocol is as follows:

Definition 1. [6] Coin-tossing protocols are protocols that generate a random bit with uniform distribution, although some corrupted parties might try to bias the output. The coin-tossing protocol is used as a building block in many cryptographic protocols.

Secure multiparty computation allows distrustful parties to compute it correctly and privately [4, 7, 8].The coin-tossing protocol raises questions of fairness and how

corrupted parties can influence the scheme [9, 10].This is the problem we are going to discuss in the following section.

2.2. Eigenvalue and Eigenvector. The eigenvalue and eigenvector are defined as follows:

Definition 2. [11] Let $\mathbf{A}$ be a n-order matrix, if the number λ and n-dimensional nonzero column vectors $\mathbf{p}$ make the equation be established.

$$\mathbf{Ap} = \lambda\mathbf{p}. \tag{1}$$

Then, the number λ is called the eigenvalue of the matrix $\mathbf{A}$, and the nonzero vector $\mathbf{p}$ is called the eigenvector of $\mathbf{A}$ corresponding to the eigenvalue λ. Equation (1) can also be written as follows:

$$(\mathbf{A} - \lambda\mathbf{E})\mathbf{p} = 0. \tag{2}$$

Equation (2) is a homogeneous linear system of n equations with n unknowns.

2.3. Lagrange Interpolation Formula. Let $n+1$ distinct interpolation points (nodes) x_j, $j = 0, 1 \ldots n$, be given, together with corresponding numbers f_j, which may or may not be samples of a function f. Unless stated otherwise, we assume that the nodes are real, although most of the results and comments generalize to the complex plane. Let $\prod_n$ denote the vector space of all polynomials of degree at most n. The classical problem addressed here is that of finding the polynomial $p \in \prod_n$ that interpolates f at the points x_j, i.e.,

$$p(x_j) = f_j, \quad j = 0, \ldots, n. \tag{3}$$

The problem is well-posed, i.e., it has a unique solution that depends continuously on the data. Moreover, as explained in virtually every introductory numerical analysis text, the solution can be written in the Lagrange form [12]:

$$\begin{aligned} p(x) &= \sum_{j=0}^{n} f_j l_j(x), \\ l_j(x) &= \frac{\prod_{k=0,k\neq j}^{n} (x - x_k)}{\prod_{k=0,k\neq j}^{n} (x_j - x_k)}. \end{aligned} \tag{4}$$

The Lagrange polynomial l_j corresponding to the node x_j has the following property:

$$l_j(x_k) = \begin{cases} 1, & j = k, \\ 0, & \text{otherwise}, \end{cases} \quad j, k = 0, \ldots, n. \tag{5}$$

2.4. Meaning of Strict Multiparty. We could compare the protocols described in Section 3 to the grouping process of a soccer game. A group of players are fairly and randomly divided into team A and team B. This process only divides the participants into two parts, but does not draw the strict order.

Therefore, considering the order of all participants, we could define the meaning of the word "strict." Its work process is more like drawing lots. All players need to decide their order in a fair way. We associate this idea with the matrix and propose a kind of the strict multiparty coin-tossing protocol.

3. Classic Coin-Tossing Protocols

3.1. Blum's Coin-Tossing Protocol. Suppose two sides of the communication are Alice and Bob. They execute the following protocol [13]:

Step1: Alice chooses a random bit a and sends a commitment c = commit(a) to Bob.

Step2: Bob chooses a random bit b and sends it to Alice.

Step3: Alice sends the bit a to Bob together with decommit(c).

Step4: If Bob does not abort during the protocol, Alice outputs $a \oplus b$, otherwise she outputs a random bit.

Step5: If Alice does not abort during the protocol and c is a commitment to a, and then Bob outputs $a \oplus b$, otherwise he outputs a random bit.

3.2. Coin-Tossing Protocol Based on Quadratic Residue. Suppose two sides of the communication are Alice and Bob. The protocol is as follows [14]:

Step1: Bob chooses large prime numbers p, q and calculate $n = pq$, then chooses random number a that satisfied with Jacobi symbol [15] $(a/n) = 1$ and sends n, a to Alice.

Step2: Alice guesses if a is the quadratic residue of n. Telling the result to Bob.

Step3: Bob tells Alice she is right or not and sends p, q to Alice.

Step4: Alice checks p, q's parity and calculates $n = pq$.

3.3. Coin-Tossing Protocol Based on One Way Function. Suppose two sides of the communication are Alice and Bob. They both hold a one way function $f(x)$ and do not know $f^{-1}(x)$. The protocol is as follows [16]:

Step1: Bob chooses a random number x and sends Alice $y = f(x)$.

Step2: Alice guesses the parity of x and tells the result to Bob.

Step3: Bob tells Alice she is right or not and sends x to Alice.

4. Multiparty Coin-Tossing Protocol Based on the Eigenvalue

Suppose there are n participants who are marked as P_i $(i = 1, 2 \ldots n)$. The protocol is based on finite field Z_q where $q > n$ and a secret matrix A which is held by P_1. It is

worth mentioning that matrix **A** has the following two properties:

(1). **A** is a n-order matrix.

(2). The eigen equation of **A** has no multiple roots which means **A** has n different eigenvalues.

Suppose **A**'s eigenvalues are $\lambda_i (i = 1, 2 \ldots n)$ and corresponding eigenvectors are $\mathbf{p}_i$ $(i = 1, 2 \ldots n)$. The content of the protocol is as follows:

Step 1: Participant P_1 chooses a secret n-order matrix **A**. P_1 announces the main diagonal of **A** and all eigenvectors $\mathbf{p}_i$ $(i = 1, 2 \ldots n)$.

Step 2: Participants $P_i (i = 2 \ldots n)$ randomly select an eigenvector from $\mathbf{p}$ i $(i = 1, 2 \ldots n)$ and the last one belongs to participant P_1. None of the eigenvectors could be chosen twice.

Step 3: Participant P_1 announces the secret matrix **A**. All participants calculate $\lambda_i (i = 1, 2 \ldots n)$ of their own according to equation (1).

Step 4: Sort $\lambda_i (i = 1, 2 \ldots n)$ in ascending sequence, then each participant could get the corresponding order.

As can be seen from the above protocol, the final order of each participant depends only on the size of the eigenvalues. It could not prevent multiple participants in the conspiracy from exchanging eigenvectors to adjust the order. This means that this protocol cannot resist collusion attack. We use the Lagrange interpolation formula to make up for this security hole.

5. Improved Multiparty Coin-Tossing Protocol Based on the Eigenvalue

Suppose there are n participants who are marked as $P_i (i = 1, 2 \ldots n)$. The protocol is based on finite field Z_q where $q > n$ and a secret matrix **A** which is held by P_1. It is worth mentioning that matrix **A** has the following two properties:

(1) **A** is a n-order matrix.

(2) The eigen equation of **A** has no multiple roots which means **A** has n different eigenvalues.

Suppose **A**'s eigenvalues are $\lambda_i (i = 1, 2 \ldots n)$ and corresponding eigenvectors are $\mathbf{p}_i$ $(i = 1, 2 \ldots n)$. The content of the protocol is as follows:

Step 1: Participant P_1 chooses a secret n-order matrix **A**. P_1 announces the main diagonal of **A** and all eigenvectors $\boldsymbol{p}_i$ $(i = 1, 2 \ldots n)$.

Step 2: Participants $P_i (i = 2, \cdots n)$ randomly select an eigenvector from $\mathbf{p}_i$ $(i = 1, 2 \ldots n)$ and the last one belongs to participant P_1. None of the eigenvectors could be chosen twice.

Step 3: Participant P_1 announces the secret matrix **A**. All participants calculate $\lambda_i (i = 1, 2 \ldots n)$ of their own according to equation (1).

Step 4: All participants $P_i (i = 1, 2 \ldots n)$ randomly select constant $n_i \in Z_q$ to form $(i, n_i)(i = 1, 2 \ldots n)$ and calculate polynomial according to equation (3):

$$p(x) = \sum_{i=1}^{n} n_i l_i(x),$$
$$l_i(x) = \frac{\prod_{k=1, k\neq i}^{n} (x - k)}{\prod_{k=1, k\neq j}^{n} (i - k).} \tag{6}$$

As there are n points in total, so $p(x)$ is a $(n-1)$-th degree polynomial at most:

$$p(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}, \quad \left(a_0, a_1 \ldots a_{n-1} \in Z_q\right). \tag{7}$$

We choose the coefficient of the nonzero minimum degree term in $p(x)$, suppose it is a_j.

Step 5: All participants $P_i (i = 1, 2 \ldots n)$ calculate:

$$s_i = \lambda_i a_j (i = 1, 2 \ldots n). \tag{8}$$

Sort $s_i (i = 1, 2 \ldots n)$ in the ascending sequence, then each participant could get the corresponding order of themselves.

6. Instance of the Protocol

The protocol is based on finite field Z_{23}. Suppose there are 6 participants who is marked as P_i $(i = 1, 2, 3)$ and a secret matrix **A** is held by $P1$. **A** is a 6-order matrix which is designed as follows:

$$A = \begin{pmatrix} 5 & 13 & 18 & 0 & 4 & 6 \\ 2 & 18 & 17 & 20 & 4 & 7 \\ 0 & 17 & 0 & 21 & 0 & 8 \\ 2 & 12 & 17 & 22 & 4 & 11 \\ 2 & 20 & 20 & 0 & 5 & 1 \\ 2 & 12 & 17 & 20 & 5 & 3 \end{pmatrix}. \tag{9}$$

All eigenvalues and related eigenvectors pairs $(\lambda_i, \mathbf{p}_i)$ $(i = 1, 2, \ldots, 6)$ are as follows:

$$(\lambda_1, \mathbf{p}_1) = \left(1, (1\,1\,0\,1\,0\,1)^{\mathrm{T}}\right); (\lambda_2, \mathbf{p}_2) = \left(2, (1\,1\,1\,0\,1\,1)^{\mathrm{T}}\right),$$
$$(\lambda_3, \mathbf{p}_3) = \left(3, (0\,1\,0\,1\,1\,1)^{\mathrm{T}}\right); (\lambda_4, \mathbf{p}_4) = \left(4, (1\,0\,1\,0\,1\,0)^{\mathrm{T}}\right),$$
$$(\lambda_5, \mathbf{p}_5) = \left(5, (1\,1\,0\,1\,1\,1)^{\mathrm{T}}\right); (\lambda_6, \mathbf{p}_6) = \left(6, (1\,0\,1\,1\,0\,1)^{\mathrm{T}}\right). \tag{10}$$

Step1: According to the protocol, participant $P1$ holds the secret matrix **A** and announces the main diagonal: $(5, 18, 0, 22, 5, 3)$ and $\mathbf{p}_i (\mathbf{i} = 1, 2 \ldots 6)$ to all members.

Step2: Assume that $P2$ chooses eigenvector $\mathbf{p}2$, $P3$ chooses eigenvector $\mathbf{p}5$, $P4$ chooses eigenvector $\mathbf{p}1$, $P5$ chooses eigenvector $\mathbf{p}6$, and $P6$ chooses eigenvector $\mathbf{p}3$. The last eigenvector $\mathbf{p}4$ is left to $P1$.

Step3: Participant P_1 announces the secret matrix $\mathbf{A}$. All participants calculate $\lambda_i (i = 1, 2 \ldots n)$ of their own according to equation (1). So, the eigenvalues held by each participant are — P_1: 4; P_2: 2; P_3: 5; P_4: 1; P_5: 6; P_6: 3.

Step4: All participants $P_i (i = 1, 2 \ldots n)$ randomly select constant $n_i \in Z_q$ to form $(i, n_i)(i = 1, 2 \ldots n)$, assume that $P1$ chooses $n_1 = 5$, $P2$ chooses $n_2 = 8$, $P3$ chooses $n_3 = 1$, P4 chooses $n_4 = 12$, P5 chooses $n_5 = 6$, and $P6$ chooses $n_6 = 18$, we can obtain $p(x)$ according to equation (3):

$$\begin{aligned} p(x) = {} & 5 \times \frac{(x-2)(x-3)(x-4)(x-5)(x-6)}{(1-2)(1-3)(1-4)(1-5)(1-6)} \\ & + 8 \times \frac{(x-1)(x-3)(x-4)(x-5)(x-6)}{(2-1)(2-3)(2-4)(2-5)(2-6)} \\ & + 1 \times \frac{(x-1)(x-2)(x-4)(x-5)(x-6)}{(3-1)(3-2)(3-4)(3-5)(3-6)} \\ & + 12 \times \frac{(x-1)(x-2)(x-3)(x-5)(x-6)}{(4-1)(4-2)(4-3)(4-5)(4-6)} \\ & + 6 \times \frac{(x-1)(x-2)(x-3)(x-4)(x-6)}{(5-1)(5-2)(5-3)(5-4)(5-6)} \\ & + 18 \times \frac{(x-1)(x-2)(x-3)(x-4)(x-5)}{(6-1)(6-2)(6-3)(6-4)(6-5)} \\ = {} & 22x^5 - 2x^4 + 13x^3 - 12x^2 + 17x + 21. \end{aligned} \tag{11}$$

We choose the coefficient of the nonzero minimum degree term in $p(x)$, which is $a_0 = 21$

Step5: All participants $P_i (i = 1, 2 \ldots n)$ calculate:

$$\begin{aligned} & s_1 = \lambda_4 a_0 = 15; s_2 = \lambda_2 a_0 = 19; s_3 = \lambda_5 a_0 = 13; s_4 = \lambda_1 a_0 \\ & = 21; s_5 = \lambda_6 a_0 = 11; s_6 = \lambda_3 a_0 = 17. \end{aligned} \tag{12}$$

Sort $s_i (i = 1, 2 \ldots n)$ in the ascending sequence $-s_5, s_3, s_1, s_6, s_2, s_4$, the order of all members is $-P_5, P_3, P_1, P_6, P_2, P_4$.

7. Analysis of the Protocol

7.1. Analysis of Correctness. Because the protocol of multiparty is a kind of promotion of two-party, both have the same properties. We only need to analyze the situation of multiparty.

When it comes to classic two-party coin-tossing protocol (suppose two sides of the communication are Alice and Bob), a correct and effective process should meet the following three principles [17]:

(1) Alice must throw a coin before Bob guess.

(2) After Bob guessing, Alice can no longer throw coins.

(3) Bob does not know how the coins land before guessing.

Multiparty coin-tossing protocol also needs to meet these above principles. Under the premise of correct implementation of the protocol proposed in Sections 4 and 5, once participant P_1 announces the main diagonal of $\mathbf{A}$ and all eigenvectors $\mathbf{p}_i$ $(i = 1, 2 \ldots n)$, the "coin" has landed. Then, the step that every participant randomly chooses their own eigenvector can be seen as the "guess the front and back." Apparently, this satisfies the principle one.

The principle two is also satisfied. Since all eigenvectors have been selected in Step 2, so the coin throwing party P_1 cannot toss the coin again. On the other hand, because the main diagonal of $\mathbf{A}$ is made public, P_1 has no way to change the eigenvalue of $\mathbf{p}_i$ $(i = 1, 2 \ldots n)$.The proof is detailed in Section 6.

Obviously, the principle three is satisfied. Every participant has no need to know how the concrete structure of matrix $\mathbf{A}$. Participant P_1 cannot unilaterally deceive other participants for example tampering with the secret matrix as long as the protocol is executed correctly.

To summarize, both protocols are based on the basic coin-tossing protocol's principle of design. The correctness is proved.

7.2. Analysis of Security. There are three points worth discussing in terms of security. The first is the disclosure of the main diagonal and the eigenvectors of the secret matrix. This design prevents the matrix holder P_1 from tampering with the secret matrix. The second is the resistance of the collusion attack by the Lagrange interpolation formula. The third is verification of legal participants.

7.2.1. Protection against Matrix Tampering. What if P_1 is a fraud? Obviously if P_1 only announces all eigenvectors $\mathbf{p}_i$ $(i = 1, 2 \ldots n)$ of $\mathbf{A}$, he can manipulate the result of a coin toss by alter the secret matrix $\mathbf{A}$ to make $\lambda_i (i = 1, 2 \ldots n)$ being different. The design of making the main diagonal of $\mathbf{A}$ public can prevent this kind of fraud. The proof is as follows:

Proposition 1. *Only one exactly matrix can be determined by the main diagonal's elements and all eigenvectors.*

Prove: Suppose the n-order secret matrix is $\boldsymbol{A} = \begin{pmatrix} c_1 & x_{12} & \cdots & x_{1n} \\ x_{21} & c_2 & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & c_n \end{pmatrix}$ *whose elements is unknown except the main diagonal elements* $c_1, c_2 \ldots c_n$. *Besides, eigenvalues* $\lambda_i (i = 1, 2 \ldots n)$ *of* $\boldsymbol{A}$ *are unknown and eigenvectors* $\boldsymbol{p}_i$ $(i = 1, 2 \ldots n)$ *are all known. We suppose the column vector* $p_i = (p_{1i}, p_{2i} \ldots p_{ni})^{\mathrm{T}}$, $(i = 1, 2 \ldots n)$ *and vector composed of unknowns is* $\boldsymbol{x} = (x_{12} \cdots x_{1n}, x_{21}, x_{23} \cdots x_{2n} \cdots$

$x_{n1} \cdots x_{n(n-1)}, \lambda_1, \lambda_2 \cdots \lambda_n)^{\mathrm{T}}$, *According to formula (1), we can get* $\mathbf{A}\mathbf{p}_i = \lambda_i \mathbf{p}_i$, $(i = 1, 2 \cdots n)$, *which is equation set:*

$$\begin{cases} c_1 p_{1i} + x_{12} p_{2i} + \cdots + x_{1n} p_{ni} = \lambda_i p_{1i}, \\ x_{21} p_{1i} + c_2 p_{2i} + \cdots + x_{2n} p_{ni} = \lambda_i p_{2i}, \\ \vdots, \\ x_{n1} p_{1i} + x_{n2} p_{2i} + \cdots + c_n p_{ni} = \lambda_i p_{ni}, \end{cases} \quad (i = 1, 2 \ldots n). \tag{13}$$

We put the term with the unknowns on the left and the constant term on the right:

$$\begin{cases} x_{12} p_{2i} + \cdots + x_{1n} p_{ni} - \lambda_i p_{1i} = -c_1 p_{1i}, \\ x_{21} p_{1i} + \cdots + x_{2n} p_{ni} - \lambda_i p_{2i} = -c_2 p_{2i}, \\ \vdots, \\ x_{n1} p_{1i} + \cdots + x_{n(n-1)} p_{(n-1)i} - \lambda_i p_{ni} = -c_n p_{ni}, \end{cases} \quad (i = 1, 2 \ldots n). \tag{14}$$

The above system of nonhomogeneous linear equations can be considered as the form of $\mathbf{D}\mathbf{x} = \mathbf{b}$, $\mathbf{D}$ is the coefficient matrix and $\mathbf{b}$ is the vector consist of constant terms. The coefficient matrix $\mathbf{D}$ is as follows:

$$\mathbf{D} = \begin{pmatrix} \overbrace{p_{2i}, p_{3i} \cdots p_{ni}}^{n-1} & \overbrace{0, 0 \ldots 0}^{(n-1)(n-1)} & \overbrace{0, 0 \ldots 0}^{i-1} & -p_{1i} & \overbrace{0, 0 \ldots 0}^{n-i} \\ \overbrace{0, 0 \ldots 0}^{n-1} & \overbrace{p_{1i}, p_{3i} \cdots p_{ni}}^{n-1} & \overbrace{0, 0 \ldots 0}^{(n-1)(n-2)+(i-1)} & -p_{2i} & \overbrace{0, 0 \ldots 0}^{n-i} \\ & \ddots & & \vdots & \vdots \\ \overbrace{0, 0 \ldots 0}^{(n-1)(n-1)} & \overbrace{p_{1i}, p_{2i} \cdots p_{(n-1)i}}^{n-1} & \overbrace{0, 0 \ldots 0}^{i-1} & -p_{ni} & \overbrace{0, 0 \ldots 0}^{n-i} \end{pmatrix}, \quad (i = 1, 2 \ldots n). \tag{15}$$

When i is arranged from 1 to n row by row, the size of the coefficient matrix $\mathbf{D}$ is $n^2 \times n^2$.Because the eigenvector $\mathbf{p}_i$ $(i = 1, 2 \cdots n)$ are linearly independent, the elementary row operation cannot make any row of the matrix get all 0s which means the rank of $\mathbf{D}$ is full. We obtain the following conclusion:

$$r(\mathbf{D}) = r(\mathbf{D}|\mathbf{b}) = n^2. \tag{16}$$

The system of nonhomogeneous linear equations $\mathbf{D}\mathbf{x} = \mathbf{b}$ has a unique solution, only one exactly matrix $\mathbf{A}$ can be determined.

The proposition we just proposed directly limits P_1 to tamper with matrix elements or matrix eigenvalues. Once the main diagonal and all eigenvectors are published, the secret matrix $\mathbf{A}$ is locked. But there is still a security issue, what if two or more participants collude to deceive? For example, Alice and Bob exchange the eigenvector of themselves. At this time, the role of the Lagrange interpolation formula is reflected.

7.2.2. Protection against Collusion Attack. The main purpose of the introduction of the Lagrange interpolation formula is to prevent members from collusion attacks. This idea mainly comes from Shamir's Lagrange interpolation secret sharing threshold system [18–20].

The scheme in Section 4 directly determines the final strict order based on the sort of the eigenvalues. However, in the improved protocol proposed in Section 5, we do not directly sort the eigenvalues, all the participants negotiate a polynomial together and take a nonzero coefficient as a factor. This makes the final order completely random and is decided by all participants, and any collusion attack will not work.

7.2.3. Verification. Suppose participant p_1 wants to manipulate the result of a coin toss. The only way he can take is to alter the secret matrix $\mathbf{A}$. However, participants can identify the fraud in the following ways:

(1) The main diagonal and eigenvectors of $\mathbf{A}$ are not the same as what P_1 published.

(2) Cannot calculate the correct λ like λ is not in finite field Z_q.

(3) There are one or more repeated eigenvalues of matrix $\mathbf{A}$.

As long as any of the above three cases occur, it should be taken seriously because of fraud. At this time, participants who have an abnormal situation will report an error.

From this perspective, participant P_1 is under the supervision of all people. The protocol is reliable.

	Resist parties aborting	Low complexity	Practicability	Strict
Blum's coin-tossing protocol		√		
Coin-tossing protocol based on quadratic residue			√	
Coin-tossing protocol based on one-way function		√		
Almost-optimally fair multiparty coin-tossing	√		√	
Multiparty coin tossing in four rounds	√		√	
Coin-tossing protocol we proposed	√	√	√	√

Figure 1: Comparison with classic coin-tossing protocols.

8. Protocol Comparison

In Blum's coin-tossing protocol, there are problems caused by parties aborting the protocol. It is proved that the best case is 1/4 of the bias of the protocol [6]. In this paper, apparently eigenvalue secrete matrix can not only solve this problem but also sort multi participants strictly. We only need to focus on the final sequence rather than pay attention to the specific value. Legitimate users are not affected.

According to the coin-tossing protocol based on quadratic residue [7], the large prime numbers p, q are used to calculate composite number n. The execute of the protocol based on the quadratic residue calculation involving large prime numbers and congruence equations. Therefore, the computational complexity of the scheme is high. When it comes to the coin-tossing protocol based on the eigenvalue, the computational complexity is mainly based on the construction of secret matrix $\mathbf{A}$ which can be easily constructed. The reason lies in there is no need to care about the particular numbers of the eigenvalues.

Some classic coin-tossing protocols lack of practicality. For example, the coin-tossing protocols based on one-way function [21] have this drawback because there are no real one-way functions, the almost-optimally fair multiparty coin-tossing [22] and multiparty coin tossing in four rounds [23] have no low complexity and strict property. In this respect, the proposal in this paper has advantage of practicability. The program can easily construct a matrix that has the properties to meet the requirements in protocols we proposed. The protocols in this paper are convenient and reliable.

To summarize, a comparison of several coin-throwing protocols is shown in Figure 1, which shows that our proposed solution is more advantageous.

9. Conclusion

This paper first proposes a new kind of strict multiparty coin-tossing protocol based on the eigenvalue, then takes a step further to propose an improved version which is based on the Lagrange interpolation formula. The analysis shows the protocol is correct and can resist matrix tampering attack as long as collusion attack. Furthermore, we make sure the protocols based on the eigenvalue can resist parties aborting, have low complexity, and practicability which means they could easily be constructed.

The coin-tossing protocol can resist the attack proposed in literature [24, 25], which has been studied in literature [26].

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] W. Du and M. J. Atallah, "Secure multi-party computation problems and their applications: a review and open problems," in *Proceedings of New Security Paradigms Workshop 2001*, V. Raskin, S. J. Greenwald, and B. Timmerman, Eds., ACM Press, Cloudcroft, New Mexico, 2001.

[2] M. Blum, "Coin Flipping by Telephone: A Protocol for Solving Impossible problems," *SIGACT News*, vol. 15, pp. 133–137, 1982.

[3] M. Ben-Or and N. Linial, "Collective coin flipping," *Randomness and computation*, Academic Press, Cambridge, MA, USA, 1990.

[4] Y. Lindell, "Parallel coin-tossing and constant-round secure two-party computation," *Journal of Cryptology*, vol. 16, no. 3, pp. 143–184, 2003.

[5] S. Kun, Q. Shen, and M. Zhou, "Research of knapback on unbiased coin cast protocol," *Journal of University of*

Electronic Science and Technology of China, vol. 32, no. 4, pp. 417–419, 2003, in Chinese.

[6] A. Beimel, E. Omri, and I. Orlov, "Protocols for multiparty coin toss with a dishonest majority," *Journal of Cryptology*, vol. 28, no. 3, pp. 551–600, 2015.

[7] J. Katz, "On achieving the "best of both worlds" in secure multi-party computation," in *Proceedings of the 39th ACM Symp. on the Theory of Computing*, pp. 11–20, San Diego, California, USA, June 2007.

[8] T. Moran, M. Naor, and G. Segev, "An optimally fair coin toss," in *Theory of Cryptography*, pp. 1–18, Springer, Berlin, Heidelberg, 2009.

[9] R. Pass, "Bounded-concurrent secure multi-party computation with a dishonest majority," in *Proceedings of the 36th ACM Symp. on the Theory of Computing*, pp. 232–241, Chicago, IL, USA, June 2004.

[10] S. Goldwasser and Y. Lindell, "Secure multi-party computation without agreement," *Journal of Cryptology*, vol. 18, no. 3, pp. 247–287, 2005.

[11] J. Lenz and D. Mubayi, "Eigenvalues of non-regular linear quasirandom hypergraphs," *Discrete Mathematics*, vol. 340, no. 2, pp. 145–153, 2017.

[12] R. Baltensperger, "Improving the accuracy of the matrix differentiation method for arbitrary collocation points," *Applied Numerical Mathematics*, vol. 33, pp. 143–149, 2000.

[13] M. Blum, "Coin flipping by telephone a protocol for solving impossible problems," *ACM SIGACT News*, vol. 15, no. 1, pp. 23–27, 1983.

[14] A. Beimel, Y. Lindell, E. Omri, and I. Orlov, "1/p-secure multiparty computation without honest majority and the best of both worlds," *Journal of Cryptology*, vol. 33, 2011.

[15] S. D. Gordon and J. Katz, "Partial fairness in secure two-party computation," *Journal of Cryptology*, vol. 25, no. 1, pp. 14–40, 2012.

[16] S. D. Gordon, C. Hazay, J. Katz, and Y. Lindell, "Complete fairness in secure two-party computation," *Journal of the ACM*, vol. 58, no. 6, p. 24, 2011.

[17] X. L. Yang and X. J. Zuo, "A scheme of throwing coins based on quadratic residuals," *Computer Technology and Development*, vol. 26, no. 9, 2016, in Chinese.

[18] S. Zhu, L. Zhan, H. Qiang, D. Fu, W. Sun, and Y. Tang, "A Fuzzy Attribute-Based Authentication Scheme on the Basis of Lagrange Polynomial Interpolation," *Human Centered Computing. HCC 2014*, Springer, Cham, Switzerland, UK, 2014.

[19] X. Y. Gan and B. Liu, "Shamir threshold based encryption," *Applied Mechanics and Materials*, vol. 52-54, pp. 709–712, 2011.

[20] W. Li-Ping, "Lagrange Interpolation Polynomials and Generalized Reed-Solomon Codes over Rings of matrices," in *Proceedings of the 2012 IEEE International Symposium on Information Theory Proceedings*, pp. 3098–3100, IEEE, Cambridge, MA, USA, July 2012.

[21] M. Ahmad, S. Singh, and S. Khurana, "Cryptographic one-way hash function generation using twelve-terms 4D non-linear system," *International Journal of Information Technology*, pp. 1–9, 2018.

[22] B. Alon and E. Omri, "Almost-Optimally Fair Multiparty Coin-Tossing with Nearly Three-Quarters Malicious," *Proceedings, Part I, of the 14th International Conference on Theory of Cryptography*, Springer, Berlin, Heidelberg, pp. 307–335, 2016.

[23] M. Ciampi, R. Ostrovsky, L. Siniscalchi, and I. Visconti, "Delayed-Input Non-malleable Zero Knowledge and Multi-Party Coin Tossing in Four Rounds," *Theory of Cryptography Conference*, ACM, Baltimore, USA, pp. 711–742, 2017.

[24] S. Qiu and D. Wang, "Revisiting three anonymous two-factor authentication schemes for roaming service in global mobility networks," *J Surveill Secur Saf*, vol. 2, pp. 66–82, 2021.

[25] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 507–523, 2022.

[26] Y. Zhang, Z. Wang, Z. Wang, and H. Chen, "Verifiable three-party secure key exchange protocol based on eigenvalue," *Journal on Communications*, vol. 40, no. 12, pp. 149–154, 2019.