Confidentially Compare Rational Numbers under the Malicious Model

Xin Liu[1,2], Xiaomeng Liu[1], Dan Luo[3], Gang Xu[4], Xiu-Bo Chen[2]*

[1] School of Information Engineering, Inner Mongolia University of Science and Technology, China
[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China
[3] Department of Computer Science and Technology, Tianjin Ren' ai College, China
[4] School of Information Science and Technology, North China University of Technology, China
lx2001.lx@163.com, 15552086354@163.com, rdjeans@gmail.com, gx@ncut.edu.cn, xb_chen@bupt.edu.cn

Abstract

Secure multi-party computation is a hotspot in the cryptography field, and it is also a significant means to realize privacy computation. The Millionaires' problem is the most fundamental problem among them, which is the basic module of secure multi-party computation protocols. Although there are many solutions to this problem, there are few anti-malicious adversarial protocols besides protocols based on Yao's garbled circuit. Only a few solutions have low efficiency, and there is no protocol for rational numbers comparison under the malicious model, which restricts the solution of many secure multi-party computation problems. In this paper, the possible malicious behaviors are analyzed in the existing Millionaires' problem protocols. These behaviors are discovered and taken precautions against through the triangle area formula, zero-knowledge proof, and cut-and-choose method, so the protocol of comparing confidentially rational numbers is proposed under the malicious model. And this paper adopts the real/ideal model paradigm to prove the security of the malicious model protocol. Efficiency analysis indicates that the proposed protocol is more effective than existing protocols. The protocol of rational numbers comparison under the malicious model is more suitable for the practical applications of secure multi-party computation, which has important theoretical and practical significance.

Keywords: Secure multi-party computation, Malicious model, Rational numbers comparison, Real/ideal model paradigm

1 Introduction

Data can be collected, used, and calculated more conveniently through the popularity of the Internet, big data, and cloud computing. But in the network of joint computing, data privacy is very easy to leak, resulting in huge losses. Privacy-based collaborative computing, which combines cryptography, distributed computing, and other technologies, is a hot research topic in privacy protection. It can complete joint computing tasks under the premise of protecting the data security of all parties. Secure multi-party computation (MPC) is the core technology to achieve multi-source data privacy collaborative computing [1-3].

In 1982, Professor Qizhi Yao proposed the famous Millionaires' problem [4] and introduced the concept of MPC, which is the beginning of private data secure comparison. The research field of MPC has also been extended to various application fields, including blockchain privacy protection [5-6], confidential data mining [7], and confidential computing geometry [8], which makes a large number of confidential collaborative computing problems to be solved effectively.

MPC is divided into the semi-honest model and the malicious model. Under the semi-honest model, scholars have designed many semi-honest MPC protocols, but in many real-world application scenarios, the participants are not semi-honest, and the protocols under the malicious model are more universal [9]. However, the design of MPC protocol is difficult under the malicious model, so many problems remain unresolved under the malicious model, and need to be studied.

The cut-and-choose method is often used to detect whether protocol participants have malicious behavior, which can reduce the probability of being deceived. In the method, Alice first selects m random numbers and uses her private data to calculate m sets of results, all of which are sent to Bob. Then Bob randomly selected m / 2 groups for validation, mainly verifying two aspects: (1) Whether the data provided by Alice was calculated from the same private data. (2) Is the calculation result of the m / 2 group that Alice sent to Bob and was selected correctly? If the verification is successful, Bob randomly selects one group from the remaining m / 2 groups and completes the subsequent protocol steps. If Alice sends an incorrect calculation result to deceive Bob, it will be discovered during the verification process. Therefore, the cut-and-choose method used in MPC protocols can reduce the probability of semi-honest participants being deceived.

Data secure comparison is one of the most classical problems in MPC, which is the basic module of other problems. It is of great theoretical and practical significance to study this problem. Nevertheless, most of the existing solutions are complex and inefficient, most of them can only be applied to the semi-honest model and cannot compare rational numbers. The main contributions of this paper are as follows:

(1) Firstly, based on Reference [10], we analyze the possible attack behaviors of malicious participants.

(2) To address the issue of the malicious behaviors of

*Corresponding Author: Gang Xu; E-mail: gx@ncut.edu.cn
DOI: 10.53106/160792642024052503002

adversaries, the MPC protocol under the malicious model is designed using the cut-and-choose method and zero-knowledge proof, which has lower computational complexity and higher efficiency.

(3) The proposed protocol is secure under the malicious model, proved by the real/ideal model paradigm. Among the existing malicious model protocols, this protocol securely realizes the secure judgment of rational number relationships under the malicious model.

2 Related Work

The problem of comparing secure data is to obtain the result without revealing the private data. This problem is the basic module for many MPC problems.

Reference [9] modified the solution of the Millionaires' problem based on the Paillier encryption algorithm, which can resist malicious behaviors but cannot judge the relationship of rational numbers. Reference [11] proposed a card-based cryptographic protocol by introducing private permutations for private operation storage instead of a shuffle operation, but it needs the number of cards related to the comparison data, which can be considered that the computational overhead is too high and cannot resist malicious adversaries. Reference [12] designed a solution to the Millionaires' problem under the semi-honest model based on the Decisional Diffie-Hellman hypothesis. However, it cannot resist malicious adversaries. Reference [13] designed a Millionaires' problem with a 0-1 coding rule based on the ElGamal encryption algorithm, but its computational cost is large. In addition, it cannot resist malicious adversaries. Reference [14] designed secure judgment protocols for two equal rational numbers based on the Paillier encryption algorithm, and also gave a secure judgment protocol for equality of rational numbers under the malicious model, but cannot judge rational numbers.

Therefore, this paper proposes the secure relationship judgment protocol of rational numbers under the malicious model. The designed protocol is efficient and can judge $x < y, x > y$, or $x = y$ once time. Compared with the previous protocols, the protocol proposed in this paper is more efficient and can resist malicious attacks.

3 Preliminary Knowledge

3.1 Zero-Knowledge Proof Based on the Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC). Neal Koblitz [15] proposed a method for using elliptic curves in cryptography fields in 1987. The security of ECC comes from the discrete logarithm difficulty problem, which has many advantages. For example, in some cases, using a smaller key can achieve higher security than other algorithms. For example, the elliptic curve equation on the finite field Z_p is $y^2 = x^3 + ax + b$, there are two points P and Q on the curve. The crossing point P and Q is a straight line intersection elliptic curve on point R'. Then draw a straight line through R' perpendicular to the X-axis, and the focus of the elliptic curve is R. The elliptic curve is shown in Figure 1.

It can be shown that at any point P and $Q \in E_P(a, b)$ in the set $E_P(a, b)$ there are:

(1) $P + Q = R$;

(2) $P + O = P$;

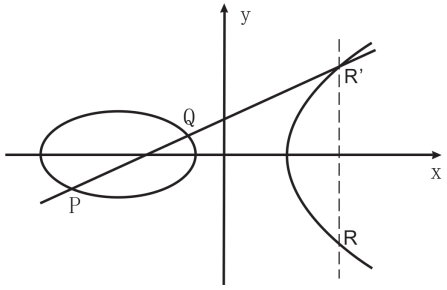

Figure 1. Elliptic curve

(3) Repeated addition is the definition of multiplication, such as $3P = P + P + P$.

The Zero-Knowledge Proof Based ECC [16]. In this protocol, hash functions are used to encrypt a public value as an e value to prevent cheating. The program framework is shown in Figure 2.

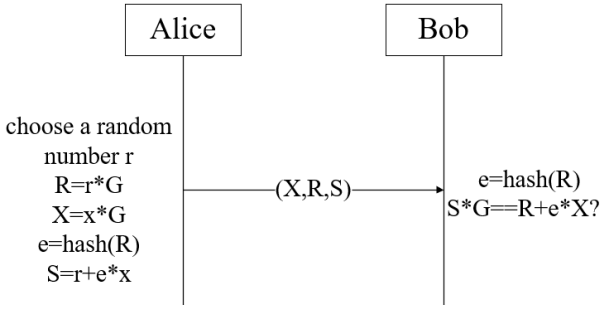

Figure 2. Non-interactive zero knowledge proof

First, an elliptic curve E_p is selected by Alice, and a point on the curve is selected as the base point, denoted as G. Alice selects a random number r and calculates $R = r * G$, $X = x * G$, $e = hash(R)$, $S = r + e * x$, generates a proof (R, X, S), and sends it to Bob. Bob receives the proof that calculates $e = hash(R)$ and then verifies that $S * G \stackrel{?}{=} R + e * X$ is true or not, determining whether Alice knows the message x. The proof is as follows:

$$S * G = (r + e * x) * G = r * G + e * x * G = R + e * X.$$

In the above verification process, Alice uses $e = hash(R)$ to construct the e value, if Alice wants to cheat, she needs to construct R to meet $R = S * G - e * X$, from the elliptic curve of the discrete logarithm difficult problem, through R and G to calculate the random number r, the probability can be ignored, so she cannot cheat.

3.2 Security Definition under the Malicious Model

The widely accepted definition of security for MPC protocols under the malicious model is given by Goldreich in the Reference [17], which is the real/ideal model paradigm.

The Ideal Protocol with TTP. P_1, P_2 have private data x, y, and they invoke a trusted third party (TTP) to calculate

the function $f(x, y) = (f_1(x, y), f_2(x, y))$. After the protocol is finished, P_1 and P_2 get $f_1(x, y)$ and $f_2(x, y)$ respectively without revealing their private data. The protocol is as follows:

(1) The data is provided to TTP. If the participant P_i is honest, P_i always provides real data. If the participant P_i is a malicious participant, depending on x or y. The strategy adopted by P_i is that P_i does not execute the protocol (no data sent) or send false data to TTP during execution.

(2) TTP sends the result to P_1. If TTP receives (x, y), he will calculate $f(x, y)$ and sends $f_1(x, y)$ to P_1. If not, the special symbol $\perp$ is sent to P_1.

(3) TTP sends the result to P_2. If P_1 is not a semi-honest participant, P_1 can ignore TTP after receiving $f_1(x, y)$. Otherwise, the symbol $\perp$ is sent to P_2. If not, P_2 will receive $f_2(x, y)$ from TTP.

Because participants can only get $f_i(x, y)$ from TTP, the above ideal protocol is the most secure. If the security of the actual protocol is the same as that of the ideal protocol, the actual protocol can also be considered secure.

There is a probabilistic polynomial-time (PPT) function $F: \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$, and the first and second elements of $F(x, y)$ are represented by $F_1(x, y)$ and $F_2(x, y)$, respectively. Let $\overline{B} = (B_1, B_2)$ be a pair of PPT algorithms for participant policies in an ideal protocol. If there is at least one $B_i (i \in \{1, 2\})$ during the execution of the protocol for all u, z, r, v with $B_i(u, z, r) = u$, $B_i(u, z, r) = v$. For B_i, his input is u, the auxiliary input is z, the selected random number is r, and the local output $F_i()$ obtained from TTP is denoted as v. Such $\overline{B}(B_1, B_2)$ is considered acceptable. The auxiliary information z is held by the participant in the ideal model, and the adversary uniformly selects a random number r to jointly execute $F(x, y)$ with policy B, the procedure that is noted as $IDEAL_{F,\overline{B}(z)}(x, y) = \gamma(x, y, z, r)$, $\gamma(x, y, z, r)$ can be defined as follows:

- If P_1 is honest, then:

$\gamma(x, y, z, r) = (f_1(x, y'), B_2(y, z, r, f_2(x, y')))$, where $y' = B_2(y, z, r)$.

- If P_2 is honest, then:

$$\gamma(x,y,z,r) = \begin{cases} (B_1(x,z,r,f_1(x',y),\perp),\perp), & if\ B_1(x,z,r,f_1(x',y)) = \perp, \\ (B_1(x,z,r,f_1(x',y)),f_2(x',y)), & else. \end{cases}$$

in both cases $x' = B_1(x, z, r)$.

Definition 1 *Security of the Malicious Model.* $\prod$ is denoted as a two-party protocol for calculating F. $\overline{A} = (A_1, A_2)$ are two PPT algorithms that represent the strategies of participants in the real model. If there is at least one $A_i (i \in \{1, 2\})$ in the protocol execution that is consistent with the policy specified by $\prod$, $A = (A_1, A_2)$ is acceptable for $\prod$. In particular, A_i ignores its auxiliary input. If (x, y) is the input, z is the auxiliary input, $REAL_{\Pi,A(z)}(x, y)$ is denoted as the process of implementing protocol $\prod$ using strategy A. When $A_1 = (x, z)$ and $A_2 = (y, z)$ interact, the generated output sequence is defined as $REAL_{\Pi,A(z)}(x, y)$.

For any policy pair $A = (A_1, A_2)$ acceptable in the actual protocol, if the corresponding acceptable $\overline{B} = (B_1, B_2)$ can be found in an ideal protocol, satisfy the following formula:

$$\{IDEAL_{F,\overline{B}(z)}(x,y)\}_{x,y,z} \overset{c}{\equiv} \{REAL_{\Pi,A(z)}(x,y)\}_{x,y,z}, \quad \textbf{(1)}$$

it can be called $\prod$ securely computes the function F.

3.3 The MPC Protocol for Comparing Rational Numbers under the Semi-honest Model

3.3.1 Problem Description and Solution Ideas

Most of the data being compared in real life are rational numbers. Reference [10] proposed a method to compare the size of rational numbers confidentially by using the calculation formula of the triangle area. The detailed explanation is as follows. Suppose there is a triangle $\Delta P_0P_1P_2$ which is composed of three points $P_0(x_0, y_0)$, $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$. Its area calculation formula can be expressed as:

$$S_{\Delta P_0P_1P_2} = \frac{1}{2}\begin{vmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} = \tfrac{1}{2}\left[y_0(x_2 - x_1) + x_0(y_1 - y_2) + x_1y_2 - x_2y_1\right]. \quad \textbf{(2)}$$

The sign of the triangle area calculated by the Formula **(2)** is related to the arrangement order of the points $P_0(x_0, y_0)$, $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$. If the points $P_0(x_0, y_0)$, $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ are arranged counterclockwise, the area value is positive according to the formula; if the points $P_0(x_0, y_0)$, $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ are arranged in a clockwise direction, then the triangle area value is negative.

3.3.2 Specific Protocol

According to Formula **(2)**, the positive or negative value of the triangle area is related to the order of vertex arrangement. In Protocol 1, two data m and n to be compared are encoded on the vertex coordinate values of a certain triangle, and the sign of the triangle area value can be securely calculated to determine the size relationship between m and n [10].

For the convenience of description, define the following functions:

$$sign(\lambda) = \begin{cases} 1, & \lambda \text{ is a positive number;} \\ -1, & \lambda \text{ is a negative number;} \\ 0, & \lambda = 0. \end{cases} \quad \textbf{(3)}$$

$$F(m,n) = \begin{cases} 1, & m > n; \\ -1, & m < n; \\ 0, & m = n. \end{cases} \quad \textbf{(4)}$$

Protocol 1 cannot resist malicious attacks. By analyzing the malicious behaviors which may be executed in the protocol, and aiming at malicious behaviors, design the secure comparison protocol of rational numbers against malicious adversaries based on the cut-and-choose method and other cryptographic tools.

Protocol 1. The MPC protocol for comparing rational numbers under the semi-honest model

Input: Alice holds data m, Bob holds data n.
Output: $F(m, n)$.

step 1: The rational number x_0 is selected by Alice and Bob as abscissa. They constructs the two vertices $P_0(x_0, m)$ and $P_1(x_0, n)$ with private data m and n as ordinates respectively.

step 2: Bob selects a rational number x_2, satisfies $x_2 < x_0$, and selects a random number y_2 to construct another vertex $P_2(x_2, y_2)$.

step 3: The three vertices $P_0(x_0, m)$, $P_1(x_0, n)$ and $P_2(x_2, y_2)$ form a triangle $\Delta P_0P_1P_2$, as shown in Figure 3. The following steps calculate the area of a triangle: Bob randomly selects a positive rational number r and calculates: $a = r(n - y_2)$, $b = r(x_2 - x_0)$ and $c = r(x_0y_2 - x_2n)$. Bob sends a, b, c to Alice.

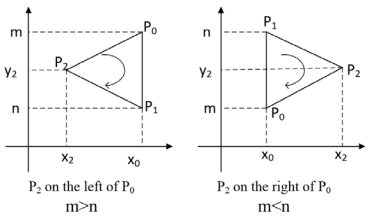

Figure 3. Compare m and n by triangle area symbols

step 4: Alice calculates $\lambda = (ax_0 + bm + c)$. Alice tells Bob to calculate the $sign(\lambda)$ of the result λ.

step 5: Bob can deduce $F(m, n)$ through $sign(\lambda)$:
If $sign(\lambda) = -1$, then $P_0 \rightarrow P_1 \rightarrow P_2$ is clockwise, with $F(m, n) = 1$ $(m > n)$;
If $sign(\lambda) = 1$, then $P_0 \rightarrow P_1 \rightarrow P_2$ is counterclockwise, with $F(m, n) = -1$ $(m < n)$;
If $sign(\lambda) = 0$, then $F(m, n) = 0$ (m, n).

step 6: Bob sent $F(m, n)$ to Alice.
The protocol ends.

4 The MPC Protocol for Comparing Rational Numbers under the Malicious Model

4.1 The Idea of Solutions

Under the semi-honest model, Protocol 1 is secure as long as the participants do not implement malicious attacks. In this section, malicious attacks that may be executed by the malicious adversary are analyzed. According to the malicious behaviors, the corresponding preventive measures are designed, so that the malicious behaviors of the adversary cannot be implemented or can be found immediately.

Under the malicious model, the following three kinds of malicious behaviors made by the participant P_i cannot be avoided in the ideal protocol, so we do not consider: (1) participants use false data instead of input; (2) either party chooses not to execute the protocol; (3) either party terminates the protocol at any time. In addition, they may implement the following malicious attacks:

- Possible malicious behaviors of Alice (assuming that Bob is honest at this time): Alice tells Bob the false $sign(\lambda)$ in step 4, which makes Bob get a false result;
- Possible malicious behaviors of Bob (assuming that Alice is honest at this time): There are the following situations: (1) In step 3, Bob sends false a, b, and c to Alice; (2) In step 6, Bob knows $F(m, n)$ and then tells Alice the false result, then Alice will get the false result.

For the possible malicious behaviors of Alice, Bob gets the false $sign(\lambda)$, then he also gets the false $F(m, n)$. However, Alice knows $F(m, n)$, then just needs to reverse the sign to get the correct result. Malicious behavior may be successful.

For Bob's possible malicious behavior, (1) Alice gets false a, b and c, then the calculated $sign(\lambda)$ is also false, Bob cannot get the correct result, so deception is unsuccessful; (2) Bob tells Alice false $F(m, n)$ that it is possible to cheat successfully.

4.2 Specific Protocol

For the above malicious behavior, cryptography tools such as the zero-knowledge proof are used to find or avert the possible malicious behaviors that may be carried out in Protocol 1. Finally, both parties get the result at the same time, both parties are equal to ensure the fairness of the protocol.

Protocol 2 is designed to resist malicious adversaries, and the detailed steps are as follows:

Protocol 2. The MPC protocol for comparing rational numbers under the malicious model

Input: Alice holds data m, Bob holds data n.
Output: $F(m, n)$.

Preparation: The random numbers selected in steps 4, 8, 9 are either positive or negative. Given an ECC E_p and base point G, Alice randomly selects an integer sk_a on the integer field Z_q as the private key and obtains the public key $PK_a = sk_a * G$. Bob randomly selects an integer sk_b on Z_q as the private key and obtains the public key $PK_b = sk_b * G$. The G and the respectively public keys may be published.

step 1: The rational number x_0 is selected by Alice and Bob as abscissa. They constructs the two vertices $P_0(x_0, m)$ and $P_1(x_0, n)$ with private data m and n as ordinates respectively.

step 2: The rational number x_2' selected by Alice, satisfies $x_2' < x_0$, and selects a random number y_2' to form another vertex $P_2'(x_2', y_2')$.

step 3: Both the random number y_2 and the rational number x_2 satisfying $x_2 < x_0$ are selected by Bob to construct the vertex $P_2(x_2, y_2)$.

step 4: Alice and Bob choose m positive random numbers r_{ai}, r_{bi} i $(1, \ldots, m)$, calculate respectively

$$\{\alpha_{1a}^i, \alpha_{2a}^i, \alpha_{3a}^i\} = \{r_{ai}(m - y_2'),\ r_{ai}(x_2' - x_0),\ r_{ai}(x_0y_2' - x_2'm)\},$$

$$\{\beta_{1b}^i,\ \beta_{2b}^i,\ \beta_{3b}^i\} = \{r_{bi}(n - y_2),\ r_{bi}(x_2 - x_0),\ r_{bi}(x_0y_2 - x_2n)\},$$

and publish $\{\alpha_{1a}^i, \alpha_{2a}^i, \alpha_{3a}^i\}$, $\{\beta_{1b}^i, \beta_{2b}^i, \beta_{3b}^i\}$.

step 5: According to the cut-and-choose method, Alice selects randomly m / 2 groups $\{\beta_{1b}^i, \beta_{2b}^i, \beta_{3b}^i\}$ from m groups $\{\beta_{1b}^i, \beta_{2b}^i, \beta_{3b}^i\}$, requires Bob to publish $r_{bi}y_2$ and $r_{bi}x_2 n$. Alice verifies that $r_{bi}x_0y_2 - r_{bi}x_2 n \stackrel{?}{=} \beta_{3b}^i$. Passing the verification is the premise of performing the next step, if not, the protocol will be stopped.

step 6: Bob selects randomly m / 2 groups $\{\alpha_{1a}^i, \alpha_{2a}^i, \alpha_{3a}^i\}$ from m groups $\{\alpha_{1a}^i, \alpha_{2a}^i, \alpha_{3a}^i\}$, requires Alice to publish $r_{ai}y_2'$ and $r_{ai}x_2' m$. Bob verifies that $r_{ai}x_0y_2' - r_{ai}x_2' m \stackrel{?}{=} \alpha_{3a}^i$. Passing the verification is the premise of performing the next step, if not, the protocol will be stopped.

step 7: Alice and Bob randomly choose one $\{\beta_{1b}^j, \beta_{2b}^j, \beta_{3b}^j\}$ and $\{\alpha_{1a}^j, \alpha_{2a}^j, \alpha_{3a}^j\}$ from the rest $\{\beta_{1b}^i, \beta_{2b}^i, \beta_{3b}^i\}$ and $\{\alpha_{1a}^i, \alpha_{2a}^i, \alpha_{3a}^i\}$.

step 8: It is agreed that the k digits number shall be retained after the decimal point of the calculated result of λ, and that the original result shall be expanded 10^k fold. According to the zero-knowledge proof: Alice chooses the positive random number a to calculate $\lambda_B = a(\beta_{1b}^j x_0 + \beta_{2b}^j m + \beta_{3b}^j\}$, then $\lambda_B' = \lambda_B \times 10^k$, selects a random number r_A, and calculates $R_A = r_A * G$, $X_A = \lambda_B' * G$, $e_a = hash(R_A)$, $S_A = r_A + e_a * \lambda_B'$. Alice sends (R_A, X_A, S_A) to Bob.

step 9: Bob chooses the positive random number b to calculate $\lambda_A = b(\alpha_{1a}^j x_0 + \alpha_{2a}^j n + \alpha_{3a}^j\}$, then $\lambda_A' = \lambda_A \times 10^k$, selects a random number r_B, and calculates $R_B = r_B * G$, $X_B = \lambda_A' * G$, $e_b = hash(R_B)$, $S_B = r_B + e_b * \lambda_A'$. Bob sends (R_B, X_B, S_B) to Alice.

step 10: Bob calculates $e_a = hash(R_A)$ after receiving the certificate, and verify whether $S_A * G \stackrel{?}{=} R_A + e_a * X_A$.

step 11: Alice calculates $e_b = hash(R_B)$ after receiving the certificate, and verify whether $S_B * G \stackrel{?}{=} R_B + e_b * X_B$. And they send the calculated results to each other.

step 12: Both parties prove that the calculation is correct through 8-11 steps of the zero-knowledge proof, that is, verify $S_A * G \stackrel{?}{=} R_A + e_a * X_A$ and $S_B * G \stackrel{?}{=} R_B + e_b * X_B$. If one of them fails to pass the proof, the party who fails to pass the proof is malicious.

step 13: If both parties pass the proof, they will tell each other λ_B' and λ_A', according to the sign of the result of the calculation $sign(\lambda_A')$ and $sign(\lambda_B')$, Alice and Bob know $F(m, n)$.
The protocol ends.

4.3 Correctness Analysis

(1) The first four steps in Protocol 2 are the process of calculating partial factors in the triangle area formula by Alice and Bob after selecting their auxiliary points. In the process, both parties will not get valuable information about each other's auxiliary points and private data.

(2) The purpose of steps 5-7 in this protocol is to determine whether there is a malicious adversary between the two parties. Verify only the correctness of α_{3a}^i and β_{3b}^i to avoid information disclosure. If one of them provides the false $\{\alpha_{1a}^i, \alpha_{2a}^i, \alpha_{3a}^i\}$ or $\{\beta_{1b}^i, \beta_{2b}^i, \beta_{3b}^i\}$, he will not get the right result by himself according to the following steps.

(3) In step 8, keep the k digits number after the decimal point of the value of λ and it shall be expanded 10^k fold to ensure that λ is an integer. We can find the deception behavior by the zero-knowledge proof based on the hash function and ECC, and the final judgment result is based on the symbols of λ_A' and λ_B', and the expansion does not change the sign of positive or negative, so the effect of the expansion on the numerical accuracy is negligible.

(4) In steps 8-11, malicious participants can be found. For instance, Alice uses the hash function to encrypt the public value R_A in step 8 to construct e_a, if Alice wants to cheat, she needs to construct R_A to satisfy $R_A = S_A * G - e_a * X_A = S_A * G - hash(R_A) * X_A$. According to the difficulty of the discrete logarithm of ECC, Alice is difficult to construct such R_A to eliminate the impact of X_A equivalence and make it permanent, so Alice cannot cheat. The same is true for Bob.

(5) In step 13, Alice and Bob obtained λ_A' and λ_B' respectively, which can verify the authenticity. For example, Alice calculates $\lambda_A' * G \stackrel{?}{=} X_B$, and if it is equal, it is the correct result.

4.4 Security Proof

Security analysis. In Protocol 2, the two parties share the same status and perform the same attacks, so Alice's possible malicious behaviors as an example are analyzed.

(1) Step 4 of Protocol 2 requires that Alice selects positive random numbers and calculates $\{\alpha_{1a}^i, \alpha_{2a}^i, \alpha_{3a}^i\}$ using the same m, x_2', y_2', and x_0. However, if she uses different m, x_2', y_2', and x_0, Bob cannot find it in step 6. It may lead to that the $\{\alpha_{1a}^j, \alpha_{2a}^j, \alpha_{3a}^j\}$ selected by Bob in step 7 is calculated with false data. This situation is equivalent to Alice changing her input, and the ideal protocol can't avoid this situation, so it will not be considered. If Alice deceives in this step, the λ_A calculated in step 9 is also false, which will lead Alice not to get correct results.

(2) The protocol requires Alice and Bob to agree that the random numbers r_{ai}, r_{bi}, a and b are either positive or negative (to counteract the impact of random numbers on the positive and negative value of the triangle area). If Alice chooses r_{ai} and a in violation of the protocol, this situation is equivalent to providing false input, which cannot be avoided in the ideal protocol and will not be considered.

(3) In step 11, Alice needs to prove (R_B, X_B, S_B) with the zero-knowledge proof. After obtaining λ_A', she can judge whether λ_A' is correct according to the proof obtained in step 9. This step cannot be deceived.

(4) After Alice gets λ_A', she can deduce to get λ_A, but even if λ_A is the exact value, Alice cannot know Bob's private data n (for $\lambda_A = b(\alpha_{1a}^j x_0 + \alpha_{2a}^j n + \alpha_{3a}^j\}$, there are two unknowns b and n in an equation).

We use the real/ideal model paradigm to prove that Protocol 2 is secure.

Theorem 1: *Under the malicious model, Protocol 2 (denoted as* $\prod$*) is secure.*

Proof: To prove that $\prod$ can securely calculate the PPT function, we need to find the acceptable policy pair $\overline{A} = (A_1, A_2)$ adopted by both parties when $\prod$ is executed so that the acceptable policy pair $\overline{B} = (B_1, B_2)$ in the ideal protocol satisfy **(1)** in Definition 1.

If the protocol is to be secure and feasible, we will not allow both parties to be dishonest at the same time. The following are two situations: (1) A_1 is honest, A_2 is dishonest; (2) A_1 is dishonest, A_2 is honest.

Case One: A_1 is honest, A_2 is dishonest. When A_1 is honest, execute $\prod$, then:

$$REAL_{\Pi,\overline{A}}(m,n) = \{F(m, A_2(n)), A_2((\alpha_{1a}^i, \alpha_{2a}^i, \alpha_{3a}^i), \lambda_B', S)\},$$

where S is the sequence of messages generated by A_2 in the process of the zero-knowledge proof.

A_1 performs $\prod$ honestly, in which B_1 is determined by the protocol and it performs the protocol correctly according to the protocol steps. At this point, we only need to convert A_2 into B_2, which means we should find an acceptable strategy under the ideal model for $\overline{B} = (B_1, B_2)$ so that its output is indistinguishable from $REAL_{\Pi,\overline{A}}(m,n)$ calculations. (Also note: The ideal model malicious adversary B_2 does not know how the actual protocol adversary A_2 will make decisions when facing a certain problem. We should determine the B_2's behavior according to the behavior of A_2.)

(1) In the ideal model, B_1 behaves according to the protocol, he sends the true m to TTP (when B_1 receives the message, he also allows TTP to send the message to B_2, that is, B_2 must receive the message eventually). B_2 is not honest, so his message to TTP depends on the B_2's strategy, because it is consistent with A_2, so we call A_2 to get B_2's strategy. In summary, we know that the information that B_2 sends to TTP is $A_2(n)$, and the information that B_2 gets from TTP is F (m, $A_2(n)$) (B_1 also gets this result).

(2) Next, B_2 uses F (m, $A_2(n)$) to obtain a $view_{B_2}^F(m, A_2(n))$, which is indistinguishable from the $view_{B_2}^{\Pi}(m, A_2(n))$ obtained by A_2 when actually executing the protocol. And it is handed over to A_2, and the output of A_2 can be obtained. We want B_2 to use his input to assume that the other party's input satisfies the result to execute the protocol, the following is the specific process:

a. B_2 randomly selects m' as the other party's input to get $F(m', A_2(n))=F(m, A_2(n))$. Let B_2 dresses up as A_1 with A_2 to perform the protocol $\prod$;
b. B_2 sends the message $\{A_2(\alpha_{1a}^i', \alpha_{2a}^i', \alpha_{3a}^i')\}$ obtained in step 4 to A_2;
c. In step 5, when A_2 publishes the message, B_2 verifies;
d. In step 6, the data required by A_2 is published by B_2;
e. B_2 and A_2 perform the rest steps, get the corresponding λ_B'' and enable A_2 to prove $S_A * G \stackrel{?}{=} R_A + e_a * X_A$ according to the zero-knowledge proof in the protocol, so as to prove λ_B'' is correct. All message sequences issued during the certification process are marked as S';

(3) B_2 uses $((\alpha_{1a}^i', \alpha_{2a}^i', \alpha_{3a}^i'), \lambda_B'', S')$ to call A_2's strategy, outputs $A_2((\alpha_{1a}^i', \alpha_{2a}^i', \alpha_{3a}^i'), \lambda_B'', S')$, and we can get $IDEAL_{F,\overline{B}}(m,n) = \{F(m, A_2(n)), A_2((\alpha_{1a}^i', \alpha_{2a}^i', \alpha_{3a}^i'), \lambda_B'', S')\}$.

For A_2, the same algorithm is used in steps 4-11 of the protocol, so $(\alpha_{1a}^i, \alpha_{2a}^i, \alpha_{3a}^i) \stackrel{c}{\equiv} (\alpha_{1a}^i', \alpha_{2a}^i', \alpha_{3a}^i')$, $\lambda_B' \stackrel{c}{\equiv} \lambda_B''$, and the zero-knowledge proof can guarantee $S \stackrel{c}{\equiv} S'$, then it is easy to get: $\{IDEAL_{F,\overline{B}}(m,n)\} \stackrel{c}{\equiv} \{REAL_{\Pi,\overline{A}}(m,n)\}$.

Case Two: A_1 is dishonest, A_2 is honest. At this situation, the strategy of B_2 is determined by the protocol. Real model opponent A_1 needs to be converted into an ideal adversary B_1. The output of A_1 when the protocol $\prod$ is executed is completely determined by A_1's strategy and the $view_{A_1}^{\Pi}$ it obtains, then there are two cases:

- The result is not published by A_1 or the zero-knowledge proof (deemed as A_1 aborts the protocol) fails to pass. In this case, TTP sends $\perp$ to A_2, there is $REAL_{\Pi,\overline{A}}(m,n) = \{A_1((\beta_{1b}^i, \beta_{2b}^i, \beta_{3b}^i), \lambda_A', S), \perp\}$;
- Conversely, TTP sends $F(A_1(m), n)$ to A_2, then: $REAL_{\Pi,\overline{A}}(m,n) = \{A_1((\beta_{1b}^i, \beta_{2b}^i, \beta_{3b}^i), \lambda_A', S), F(A_1(m), n)\}$, where S is the sequence of messages received by A_1 during the zero-knowledge proof process.

(1) A_2 is honest, then B_2 in the ideal model also executes honestly and outputs the correct result according to the protocol. At this point, A_1 only needs to be converted into B_1, which means it should find an acceptable strategy under the ideal model for $\overline{B} = (B_1, B_2)$ so that its output is indistinguishable from $REAL_{\Pi,\overline{A}}(m,n)$ calculations.

(2) B_1 requires A_1 to obtain the input information $A_1(m)$ during the real protocol execution and sends $A_1(m)$ to TTP to obtain F ($A_1(m)$, n). Then B_1 makes use of F ($A_1(m)$, n) to invoke the strategy of A_1 and provide all messages expected by A_1, so that we can get $view_{B_1}^F(A_1(m), n)$, we want to make it indistinguishable from the $view_{A_1}^{\Pi}(A_1(m), n)$ calculation by A_1 when executing the real protocol, and give it to A_1 to obtain the output of A_1. B_1 want to use his input to assume that the other party's input satisfies the result to execute the protocol. The following is the specific process:

a. B_1 randomly selects an n' satisfying F ($A_1(m)$, n') = F ($A_1(m)$, n) and simulates the protocol step with n'.
b. That is, B_1 dresses up as A_2 and A_1 to execute protocol $\prod$.
c. B_1 sends the information $\{A_1(\beta_{1b}^i', \beta_{2b}^i', \beta_{3b}^i')\}$ required to be published in step 4 of the protocol to A_1;

d. In step 5, B_1 publishes the validation information required by A_1;
e. In step 6, when A_1 publishes the information required for verification, B_1 verifies;
f. B_1 and A_1 perform the rest steps of the protocol to obtain the corresponding λ_A''. According to the zero-knowledge proof in the protocol, A_1 can prove $S_B * G \stackrel{?}{=} R_B + e_b * X_B$, so as to prove to A_1 that λ_A'' is correct. All message sequences issued during the certification process are marked as S';

(3) At this point, B_1 gets the message $((\beta_{1b}^i{}', \beta_{2b}^i{}', \beta_{3b}^i{}'), \lambda_A'', S')$. Because the information A_1 needs when actually executing the protocol has been obtained, and its output can be determined according to the A_1's strategy, we will not consider the possible subsequent malicious behavior of A_1. The only difference in the final result is whether A_2 can receive $F(A_1(m), n)$.

(4) B_1 leverages $((\beta_{1b}^i{}', \beta_{2b}^i{}', \beta_{3b}^i{}'), \lambda_A'', S')$ to require A_1 and outputs $A_1((\beta_{1b}^i{}', \beta_{2b}^i{}', \beta_{3b}^i{}'), \lambda_A'', S')$. There are two situations:

- In the ideal model, when B_1 notifies TTP that it will not send the final results to B_2, then: $IDEAL_{F,\overline{B}}(m,n)\} = \{A_1((\beta_{1b}^i{}', \beta_{2b}^i{}', \beta_{3b}^i{}'), \lambda_A'', S'), \perp\}$;
- In the ideal model, when B_1 notifies TTP that it will not send the final results to B_2, it can obtain: $IDEAL_{F,\overline{B}}(m,n)\} = \{A_1((\beta_{1b}^i{}', \beta_{2b}^i{}', \beta_{3b}^i{}'), \lambda_A'', S'), F(A_1(m), n)\}$.

In both cases, the output of A_2 and B_2 are the same and just prove that $((\beta_{1b}^i{}', \beta_{2b}^i{}', \beta_{3b}^i{}'), \lambda_A'', S')$ and $((\beta_{1b}^i, \beta_{2b}^i, \beta_{3b}^i), \lambda_A', S)$ calculations are indistinguishable. Obviously, for A_1, the same algorithm is used in steps 4-11 of the protocol, so $(\beta_{1b}^i, \beta_{2b}^i, \beta_{3b}^i) \stackrel{c}{\equiv} (\beta_{1b}^i{}', \beta_{2b}^i{}', \beta_{3b}^i{}')$, $\lambda_A' \stackrel{c}{\equiv} \lambda_A''$ and the zero-knowledge proof can ensure $S \stackrel{c}{\equiv} S'$, then: $\{IDEAL_{F,\overline{B}}(m,n)\} \stackrel{c}{\equiv} \{REAL_{\Pi,\overline{A}}(m,n)\}$.

In summary, any acceptable PPT algorithm in the real protocol can find an acceptable strategy in the ideal model for $\overline{A} = (A_1, A_2)$, and the acceptable strategy for $\overline{B} = (B_1, B_2)$ satisfies $\{IDEAL_{F,\overline{B}}(m,n)\} \stackrel{c}{\equiv} \{REAL_{\Pi,\overline{A}}(m,n)\}$, therefore, under the malicious model, Protocol 2 is secure.

5 Analysis and Comparison of Protocol Efficiency

5.1 Computational Complexity

In Reference [9], $10m\lg N + 2$ times modular multiplication is required (where m represents the modular exponents generated by the participants having m groups, and N represents the modular number of the Paillier encryption protocol). In Reference [13], $[(2m + 3) \lg P + 5m]$ times modular multiplication is to be performed. In Reference [14], $(b_1 + b_2 + \lambda)$ times modular multiplication is to be performed (b_1 and b_2 are the numbers of bits of both input data).

The computational complexity of Protocol 2 mainly includes: Alice and Bob have performed three ECC encryption operations, $(3m + \frac{m}{2} + 4)$ ordinary addition, $(3m + \frac{m}{2} \times 7 + 6)$ ordinary multiplication respectively. Because the computational complexity of ordinary multiplication and addition is low and can be ignored compared with modular multiplication. A total of 12 modular multiplication operations are performed, in which m represents the number of verification data groups generated during the cut-and-choose process.

5.2 Communication Complexity

In Reference [9], the number of interaction rounds is 3. In Reference [13], the number of interaction rounds is 3. In Reference [14], the number of interaction rounds is $3m - 1$. Protocol 2 has four rounds of communication. Table 1 compares the overall performance of the protocol.

Table 1. Protocols' performance comparison

Protocol	Computational complexity (modular multiplication)	Communication rounds	Resist the malicious adversary
Reference [9]	$10m \lg N + 2$	3	Yes
Reference [13]	$(2m+3)\lg P + 5m$	3	No
Reference [14]	$(b_1 + b_2 + \lambda)$	$3m - 1$	Yes
Protocol 2	12	4	Yes

Reference [9] can resist malicious attacks, but it has high computational complexity and cannot compare rational numbers. Reference [13] cannot resist malicious adversaries, and the protocol efficiency is also low. The computational complexity of Reference [14] is slightly lower, but it can only be judged whether the rational numbers are equal or not, and the size relationship is not obtained. Among the existing protocols, Protocol 2 has the lowest computational complexity and fewer communication rounds, and the efficiency is the highest compared to other malicious models. Moreover, it can compare rational numbers. In conclusion, Protocol 2 is more efficient when the number of communication rounds does not differ significantly.

Note: the cut-and-choose method is widely used in protocols under the malicious model. Nevertheless, for Protocol 2, the zero-knowledge proof is the main source of its computational complexity. Service outsourcing or preprocessing can be used to further improve the efficiency of Protocol 2.

5.3 Experimental Simulation

In order to intuitively compare the complexity of each protocol, experimental simulations are carried out in references [9, 13-14], and Protocol 2. The experimental environment is as follows: processor Intel(R) Core(TM) i5-6300HQ CPU @ 2.30GHz 2.30 GHz, memory 8GB, windows 10 (64 bit) operating system, programming language select python.

In the experiment, let the modulus of the Paillier algorithm and ElGamal algorithm be the same, ignoring the time of protocol preprocessing. During the experiment, each participant holds a rational number in the range of 0 to 100 and carries out multiple experiments on each protocol under four modulus of 128, 256, 512, and 1024 bit respectively. Ten results are randomly selected from 100 test results, and the average execution time is taken to draw the following Figure 4.

The experimental result indicates that the average time consumption of Protocol 2 in different moduli is lower than protocols in references [9, 13], and [14]. Protocol 2 can compare rational numbers, which cannot be done by other protocols. While resisting malicious adversaries, Protocol 2 is more efficient than the protocols of references [9] and [14], and the greater the modulus, the greater the advantage. For Protocol 2, the computational complexity is mainly due to the increased zero-knowledge proof and cut-and-choose method. If these calculations are outsourced, the efficiency of the protocol will be greatly improved. Protocol 2 is more efficient and practical.

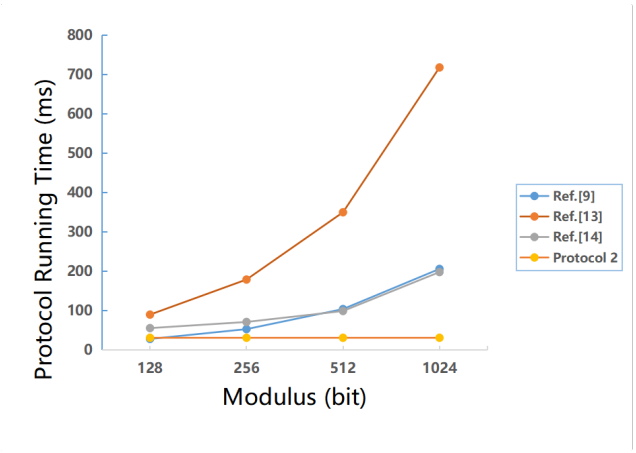

Figure 4. Comparison of running time with different modulus

6 Conclusion

The Millionaires' problem is the cornerstone of MPC, and many MPC protocols are built on this basis. The Millionaires' Protocol achieves confidentiality in comparison of data and is the fundamental protocol for solving many other MPC problems such as geometric position relationship determination, electronic auction, and graphic similarity determination. There are many malicious participants in real life, however, few existing protocols are feasible under the malicious model, and the efficiency limits the application of security protocols (Almost all universal protocols that can be applied to all scenarios come at the cost of sacrificing efficiency). The MPC protocol for comparing rational numbers under the malicious model proposed in this paper (Protocol 2) cannot only resist malicious adversary attacks, but also take into account the size comparison of rational numbers, and compared with the existing protocols greatly improves efficiency. Therefore, the protocol proposed is efficient and secure, with practical value. In the future, we will improve some MPC protocols in specific scenarios based on this protocol. In addition, studying how to improve the efficiency of Protocol 2 is also a focus of our future research.

Acknowledgement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

We the undersigned declare that this manuscript entitled "Confidentially Compare Rational Numbers under the Malicious Model" is original, has not been published before and is not currently being considered for publication elsewhere. All the authors listed have approved the manuscript that is enclosed.

This research was funded by the National Natural Science Foundation of China (72293583, 72293580), Inner Mongolia Natural Science Foundation (2021MS06006), 2023 Inner Mongolia Young Science and Technology Talents Support Project (NJYT23106), 2022 Basic Scientific Research Project of Direct Universities of Inner Mongolia (2022-101), 2022 Fund Project of Central Government Guiding Local Science and Technology Development (2022ZY0024), 2022 Chinese Academy of Sciences "Western Light" Talent Training Program "Western Young Scholars" Project (22040601), Open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (SKLNST-2023-1-08), 2023 Inner Mongolia Archives Technology Project (2023-16), the 14th Five Year Plan of Education and Science of Inner Mongolia (NGJGH2021167), Inner Mongolia Science and Technology Major Project (2019ZD025), 2022 Inner Mongolia Postgraduate Education and Teaching Reform Project (JGSZ2022037), Inner Mongolia Postgraduate Scientific Research Innovation Project (S20231164Z), Research and Application Project of Big Data Privacy Security Computing System (2023); Tianjin Renai College & Tianjin University Teacher Joint Development Fund Cooperation Project (FZ231001); Baotou Rare Earth High tech Zone Enterprise Youth Science and Technology Innovation "1+1" Action Plan Project (202309); Yunnan Key Laboratory of Blockchain Application Technology (202305AG340008, YNB202301).

References

[1] D. H. Vu, T. D. Luong, T. B. Ho, An efficient approach for secure multi-party computation without authenticated channel, *Information Sciences*, Vol. 527, pp. 356-368, July, 2020.

[2] G. Elkoumy, S. A. Fahrenkrog-Petersen, M. Dumas, P. Laud, A. Pankova, M. Weidlich, Secure multi-party computation for inter-organizational process mining, *Enterprise, Business-Process and Information Systems Modeling*, Grenoble, France, 2020, pp. 166-181.

[3] O. Goldreich, Secure multi-party computation, *Manuscript Preliminary Version*, pp. 1-110, June, 1998.

[4] A. C. Yao, Protocols for Secure Computations, *23rd IEEE Annual Symposium on Foundations of Computer Science*, Chicago, IL, USA, 1982, pp. 160-164.

[5] H. Zhong, Y. Sang, Y. Zhang, Z. Xi, Secure multi-party computation on blockchain: An overview, *Parallel Architectures, Algorithms and Programming (PAAP)*, Guangzhou, China, 2019, pp. 452-460.

[6] F. Benhamouda, S. Halevi, T. Halevi, Supporting private data on hyperledger fabric with secure multiparty computation, *IBM Journal of Research and Development*, Vol. 63, No. 2/3, pp. 3:1-3:8, March-May, 2019.

[7] S. Li, N. Mu, J. Le, S. Liao, Privacy preserving frequent itemset mining: Maximizing data utility based on database reconstruction, *Computers & Security*, Vol. 84, pp. 17-34, July, 2019.

[8] Z. Chen, S. Li, L. Chen, Q. Huang, W. Zhang, Fully privacy-preserving determination of point-range relationship, *Scientia Sinica Informationis*, Vol. 48, No. 2, pp. 187-204, 2018.

[9] S. Li, W. Wang, R. Du, Protocol for millionaires' problem in malicious models (in chinese), *Scientia Sinica Informationis*, Vol. 51, No. 1, pp. 75-88, June, 2021.

[10] X. Liu, S. Li, J. Liu, X. Chen, G. Xu, Secure multiparty computation of a comparison problem, *SpringerPlus*, Vol. 5, No. 1, pp. 1-17, September, 2016.

[11] T. Nakai, Y. Misawa, Y. Tokushige, M. Iwamoto, K. Ohta, How to solve millionaires' problem with two kinds of cards, *New Generation Computing*, Vol. 39, No. 1, pp. 73-96, April, 2021.

[12] M. Liu, P. Nanda, X. Zhang, C. Yang, S. Yu, J. Li, Asymmetric commutative encryption scheme based efficient solution to the millionaires' problem, *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, New York, USA, 2018, pp. 990-995.

[13] Z. Li, L. Chen, Z. Chen, Y. Liu, T. Gao, Design and applications of efficient protocol of millionaires' problem based on 1-r encoding, *Journal of Cryptologic Research*, Vol. 6, No. 1, pp. 50-60, February, 2019.

[14] S. Li, R. Du, Y. Yang, Q. Wei, Privately determining equality of rational numbers, *Acta Electronica Sinica*, Vol. 48, No. 10, pp. 1933-1937, October, 2020.

[15] N. Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation*, Vol. 48, No. 177, pp. 203-209, January, 1987.

[16] C. Luo, *Research on utxo model blockchain privacy protection method based on zero knowledge proof*, Master' s Thesis, Beijing Jiaotong University, Beijing, China, 2021.

[17] O. Goldreich. *Foundations of Cryptography: Volume 2: Basic Applications*, Cambridge University Press, 2009.

Biographies

Xin Liu was born in 1983 and is an associate professor. He received Doctorate degree in Computer Software and Theory from Shaanxi Normal University in 2017. His research interests are in the areas of information security, communication technology, and secure multiparty computation. Email: lx2001.lx@163.com

Xiaomeng Liu was born in 1998 and is a graduate student. Her research interests are cryptography and secure multi-party computation. Email: 15552086354@163.com

Dan Luo was born in 1980 and is a Doctor. His research interests are intelligent information processing, machine learning and pattern recognition. Email: rdjeans@gmail.com

Gang Xu received the Ph.D. degree from Beijing University of Posts and Telecommunications in 2018. His current research interests include blockchain, quantum cryptography and quantum network coding. Email: gx@ncut.edu.cn

Xiu-Bo Chen received the Ph.D. degree from Beijing University of Posts and Telecommunications in 2009. She is currently a professor in the school of cyberspace security at Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include cryptography, blockchain and information security. Email: xb_chen@bupt.edu.cn