Secure Judgment of Point and Line Relationship Against Malicious Adversaries and Its Applications

Xin Liu[1,2], Yang Xu[1], Gang Xu[3*], Xiu-Bo Chen[4], Yu-Ling Chen[5]

[1] School of Information Engineering, Inner Mongolia University of Science and Technology, China
[2] Computer College, Shaanxi Normal University, China
[3] School of Information Science and Technology, North China University of Technology, China
[4] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China
[5] College of Computer Science and Technology, Guizhou University, China
lx2001.lx@163.com, 1016051241@qq.com, gx@ncut.edu.cn, flyover100@163.com, Ylchen3@gzu.edu.cn

Abstract

With the rapid development of the Internet and information technology, the problem of zero-trust networks has become increasingly prominent, and secure multi-party computation has become a research hotspot to solve the problem of zero-trust networks. The secure judgment of point and line relationship is an important research branch of secure computing set geometry. However, most of resent secure computing protocols of point and line relationship are designed in the semi-honest model and cannot resist malicious attacks. Therefore, this paper analyzes the possible malicious adversary behaviors and designs a secure protocol in the malicious model. In this paper, the Paillier cryptosystem, zero-knowledge proof, and cut-choose method are used to resist malicious behavior, and the real/ideal model paradigm method is used to prove the security of the protocol. Compared with the existing solutions, the malicious model protocol is still efficient and widely used in real applications.

Keywords: Secure multi-party computation, Point and line relationship, Malicious model, Real/ideal model paradigm

1 Introduction

With the rise of information technology, people enjoy the convenience it brings. At the same time, the following problems of network zero trust and privacy security also begin to attract people's attention. The idea of secure multi-party computing (MPC) was put forward by Yao [1] in 1982. It is an important technology for information privacy protection. In recent years, many cryptologists have devoted themselves to the research in this direction. Since 1987, Goldreich and others have conducted in-depth research on MPC [2-3]. The research problems mainly include the following aspects: the Millionaire problem [1, 4-7], confidential computational geometry problem [8-10], confidential data mining problem [11-13], confidential statistical analysis [14-15], confidential set relationship judgment [16-17], etc. The models of MPC computing include semi-honest model and malicious model. Research issues include the establishment of security model, security analysis, general protocol design, calculation protocol of specific problems, analysis and comparison of protocol efficiency, etc. [18-22].

At present, almost all the MPC problems solved by public key encryption algorithms are in the semi-honest model, and there are few protocols against malicious enemies, but the MPC protocol against malicious enemies is more in line with the needs of practical applications.

The secure judgment of point and line relationship is an important research branch of MPC set geometry. It has important applications in military, aerospace, business, life and other fields. In Reference [23], by comparing the idea that whether the slopes of the straight lines formed by two points in three points are equal, the relationship between points and line segments can be judged confidentially. Reference [24] solved the points inclusion problem by using inner product protocol and hash function through the transformation idea of triangular area problem. Based on Paillier cryptosystem. Reference [25] designed two efficient rational number interval secret calculation protocols and applied them to the problem of point and line determination. Reference [26] designed a protocol that could safely calculate two private points and straight lines based on the Paillier cryptosystem. However, the above protocols are in the semi-honest model and cannot resist malicious opponents.

In view of the shortcomings and possible malicious behaviors of the above secure judgment semi-honest model protocols of point and line relationship protocol, we use some cryptography tools to resist them, and finally propose the MPC protocol of point and line relationship against malicious adversaries. After analysis, the proposed protocol has more efficient computing efficiency and security. The main contributions of this paper are as follows:

(1) Firstly, based on the Paillier cryptosystem, a secure judgment protocol of point and line relationship in the semi-honest model is introduced and analyzed.

(2) Aiming at the situation that the semi-honest model protocol may be attacked by some malicious attacks, the secure judgment protocol against malicious adversaries is designed with the help of the cryptographic tools such as the zero-knowledge proof and cut-choose method.

(3) The correctness and performance of the protocol in the malicious model are analyzed. The real/ideal model paradigm method is used to prove that the protocol is secure in the malicious model. Finally, the application scenarios of the protocol are proposed.

*Corresponding Author: Gang Xu; E-mail: gx@ncut.edu.cn
DOI: 10.53106/160792642022092305010

2 Related Knowledge

2.1 Paillier Cryptosystem

The Paillier cryptosystem [27] is introduced as follows: firstly, select two large prime numbers p and q to guarantee $\gcd\left(pq,(p-1)(q-1)\right)=1$; secondly, calculate $N=pq$ and $\lambda(N)=lcm(p-1,q-1)$; then choose a random number $g(g\in Z_{N^2}^{*})$, such that the order of integral division is satisfied; define $L(x)=\frac{x-1}{N}$, and calculate $\mu=\left(L(g^{\lambda}\bmod N^2)\right)^{-1}\bmod N$; finally, the public key is obtained: (N,g), private key: (λ,μ).

Encryption:

$c=g^m\cdot r^N \bmod N^2$.

Decryption:

$m=L(c^{\lambda}\bmod N^2)\cdot\mu \bmod N$.

Additive homomorphism:

$E(m_1)=c_1=g^{m_1}r_1^N \bmod N^2$, $E(m_2)=c_2=g^{m_2}r_2^N \bmod N^2$, then $E(m_1+m_2)=c_1c_2=g^{m_1+m_2}(r_1r_2)^N \bmod N^2$.

2.2 Proving the Equality of Discrete Logarithms

Zero-knowledge proof is a common cryptography tool in MPC, which proves a fact to the other party without disclosing private data. Reference [7] designed a solution to prove the equality of discrete logarithms in the malicious model.

Let G be a cyclic group with the unknown order m, g is its generator, h is an element of G, and $a=g^x$, $\beta=h^x$. Prove $\log_g a=\log_h\beta$ but not x.

The proof is as follows:

(1) Bob randomly selects a random number r in G, calculates $X=g^r, Y=h^r, e=H(g,h,a,\beta,X,Y)$, where H is a single hash function, and sends r to Alice.

(2) Alice calculates $y=r+e\times x, g^y, h^y$, and sends g^y, h^y to Bob.

(3) Because $g^y=g^{r+ex}=g^r(g^x)^e=g^ra^e=Xa^e$, and $h^y=h^{r+ex}=h^r(h^x)^e=g^r\beta^e=Y\beta^e$, so $g^y/a^e=X$, $h^y/\beta^e=Y$, then $H(g,h,a,\beta,g^y/a^e,h^y/\beta^e)=H(g,h,a,\beta,X,Y)=e$. Bob verifies that g^y, h^y meet $H(g,h,a,\beta,g^y/a^e,h^y/\beta^e)=e$.

2.3 Security in the Malicious Model

To prove that a protocol is secure in the malicious model, we must make the protocol meet the security definition in the malicious model. The real/ideal model paradigm was proposed by Goldreich [3].

Ideas: Alice and Bob have data x and y respectively. They calculate $f(x,y)=\left(f_1(x,y),f_2(x,y)\right)$ with the help of trusted third party (TTP). After the execution of the protocol, both parties get $f_1(x,y)$ and $f_2(x,y)$ respectively, but do not disclose x and y. The steps are as follows:

(1) Alice's and Bob's inputs x and y respectively;

(2) The honest party will always provide the correct x, y to TTP. Malicious participants may decide not to execute the protocol, or provide wrong x' or y' when executing the protocol;

(3) After the TTP obtains the inputs (x,y), it calculates $f(x,y)$ and sends $f_1(x,y)$ to Alice, otherwise it sends a special symbol $\perp$ to Alice;

(4) If Alice is a malicious participant, it may ignore the TTP after receiving $f_1(x,y)$. At this time, TTP sends $\perp$ to Bob; Otherwise, TTP will send $f_2(x,y)$ to Bob.

Because both participants cannot get other information except their own $f_i(x,y)$ from TTP, the ideal model protocol is secure. If the real model protocol can achieve the same security, then the real model protocol is secure.

In the ideal model protocol, participants have auxiliary information z and take the process of jointly calculating $F(x,y)$ by strategy $\overline{B}$ as $IDEAL_{F,\overline{B}(z)}(x,y)$, which is defined as that the enemy evenly selects a random number r to make $IDEAL_{F,\overline{B}(z)}(x,y)=\gamma(x,y,z,r)$, where $\gamma(x,y,z,r)$ is defined as follows (Note: if both parties in the malicious model are malicious, it is impossible to design a secure protocol, which will not be considered.):

● If Alice 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 Bob is honest,

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

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

Definition 1 Security of malicious model protocol.

If an acceptable policy pair $\overline{A}=(A_1,A_2)$ can be found in the real model protocol Π, there is an acceptable policy pair $\overline{B}=(B_1,B_2)$ in the ideal model protocol, making the $\{IDEAL_{F,\overline{B}(z)}(x,y)\}_{x,y,z}\overset{c}{\equiv}\{REAL_{\Pi,\overline{A}(z)}(x,y)\}_{x,y,z}$. That is to say Π securely calculates F.

3 A Secure Protocol for the Point and Line Relationship in the Semi-honest Model

The semi-honest model is the foundation of the malicious model. Most MPC protocols in the malicious model are designed in the semi-honest model. In Reference [28], one party encrypts his elements using Paillier cryptosystem, sends the results to the other party, and the other party decrypts the results and compares them, to judge whether the point is on a straight line. Protocol 1 for details is as follows.

Protocol 1: The MPC protocol for the point and line relationship in the semi-honest model

Input: Alice's input line L: $y=kx+b$, and David's input point $P_0(x_0,y_0)$.

Output: Whether P_0 is on the line L .

Preparation: David uses the Paillier cryptosystem to generate the public key (g, N) and private key λ .

Specific steps:

(1) David chooses a random number r_1 and encrypts x_0 to get $c_1 = g^{x_0} r_1^N \bmod N^2$, and sends c_1 to Alice;

(2) Alice chooses a random number r_2 and encrypts c_1 to compute the following result: $c_2 = c_1^k g^b r_2^N \bmod N^2 = g^{kx_0+b}(r_1^k r_2)^N \bmod N^2$, sends c_2 to David;

(3) David deciphers $v = kx_0 + b$ to get $v = kx_0 + b$, and knows that the point P_0 is on the line L ;

(4) David tells Alice the result.

The protocol ends.

Protocol 1 is designed in the semi-honest model. If one party performs a malicious behavior, the other party will not get the correct result. Therefore, the corresponding security MPC protocol is designed for the possible malicious behaviors in the following section.

4 Proposed MPC Protocol for the Point and Line Relationship in the Malicious Model

Solution idea: To design a MPC protocol in the malicious model, we need to analyze the possible malicious behaviors in the semi-honest model, and then design a MPC protocol in the malicious model. So this section designs a malicious model protocol for the possible malicious behaviors in Protocol 1. The result of the protocol implementation is that the malicious behavior can be stopped or found.

There are three kinds of malicious behaviors that can not be prevented in the ideal model protocol: ① refuse to participate in the protocol; ② provide false input, or replace one's input; ③ stop protocol in the middle of the process, that is, after getting the information, one party needs to prevent other participants from obtaining their information. The ideal model protocol can not prevent these malicious behaviors, so the real model protocol does not consider these situations.

Also, when executing the Protocol 1, Alice and David may carry out the following malicious behaviors:

(1) David's possible malicious behavior (assuming that Alice is honest at this time) includes two situations: ① when selecting a random number r_1 to compute x_0, David does not select the real random number to analyze Alice's data later. No matter what number David chooses, as long as Alice chooses a random number, David can't get Alice's data. ② After decryption, he tells Alice the wrong value of v, which makes Alice get the wrong conclusion.

(2) The possible malicious behavior of Alice (assuming David is honest at this time): ① the r_2 selected by Alice when calculating $c_2 = c_1^k g^b r_2^N \bmod N^2 = g^{kx_0+b}(r_1^k r_2)^N \bmod N^2$ is not a random number. David can't get any information from r_2, because decryption eliminates the impact of r_2. ② For the malicious behavior of Alice telling David a wrong value of v after decryption, David can use the zero-knowledge proof method to prove v is the correct result to Alice in the process of protocol execution. Furthermore, the protocol can not be decrypted unilaterally, that is to say, both parties need to decrypt together to get the final result. Protocol 2 is as follows.

Protocol 2: The MPC protocol for the point and line relationship in the malicious model.

Input: Alice's input line L: $y = kx + b$, David's input point $P_0(x_0, y_0)$.

Output: Whether point P_0 on the line L .

Preparation: Alice and David use the Paillier cryptosystem to generate public keys (g_a, N_a) and (g_d, N_d), private keys λ_a and λ_d, respectively.

Specific steps:

(1) Alice selects m random numbers a_i ($i = 1, \ldots, m$), calculates $c_{1a}^i = g_a^{a_i b} \bmod N_a^2$, $c_{2a}^i = g_a^{a_i k} \bmod N_a^2$ and $c_{3a}^i = g_a^{-a_i} \bmod N_a^2$, and publishes $(c_{1a}^i, c_{2a}^i, c_{3a}^i)$.

(2) Bob selects m random numbers d_i ($i = 1, \ldots, m$), calculates $c_{1d}^i = g_d^{d_i x_0} \bmod N_d^2$, $c_{2d}^i = g_d^{d_i} \bmod N_d^2$ and $c_{3d}^i = g_d^{-d_i y_0} \bmod N_d^2$, and publishes $(c_{1d}^i, c_{2d}^i, c_{3d}^i)$.

(3) Using the idea of cut-choose method, Alice randomly selects $m/2$ groups of $(c_{1d}^i, c_{2d}^i, c_{3d}^i)$ from m groups of $(c_{1d}^i, c_{2d}^i, c_{3d}^i)$, and asks David to disclose the corresponding $d_i x_0$ and $d_i y_0$, and verifies $c_{1d}{}^i = g_d^{d_i x_0} \bmod N_d^2$ and $c_{3d}{}^i = g_d^{-d_i y_0} \bmod N_d^2$.
David randomly selected $m/2$ group $(c_{1a}^i, c_{2a}^i, c_{3a}^i)$ from m group $(c_{1a}^i, c_{2a}^i, c_{3a}^i)$, and asked Alice to disclose the corresponding $a_i b$ and $a_i k$, and verify $c_{1a}^i = g_a^{a_i b} \bmod N_a^2$ and $c_{2a}^i = g_a^{a_i k} \bmod N_a^2$.
If the verification is passed, the next step will be executed, otherwise the protocol will be stopped.

(4) Alice randomly selects one $(c_{1d}^i, c_{2d}^i, c_{3d}^i)$ from the remaining $(c_{1d}^i, c_{2d}^i, c_{3d}^i)$ and selects a random number $a \in Z_d^*$ with $a \neq 0$, $r_1 \in Z_a^*$. Calculate $c_d = (c_{1d}^i)^{ak}(c_{2d}^i)^{ab}(c_{3d}^i)^a r_1^{N_d} \bmod N_d^2 = g_d^{ad_i(kx_0+b-y_0)} r_1^{N_d} \bmod N_d^2$ and publish c_d .

(5) David randomly selects one $(c_{1a}^j, c_{2a}^j, c_{3a}^j)$ from the remaining $(c_{1a}^i, c_{2a}^i, c_{3a}^i)$ and selects a random number $d \in Z_a^*$ with $d \neq 0$, $r_2 \in Z_d^*$ .Calculate $c_a = (c_{1a}{}^j)^d (c_{2a}{}^j)^{dx_0} (c_{3a}{}^j)^{dy_0} r_2^{N_a} \bmod N_a^2 = g_a^{a_j d(kx_0+b-y_0)} r_2^{N_a} \bmod N_a^2$ and publish c_a .

(6) Alice calculates $m_a = c_a^{\lambda_a} \bmod N_a^2$, David calculates $m_d = c_d^{\lambda_d} \bmod N_d^2$, and sends m_a and m_d to each other.

(7) Both parties use the zero-knowledge proof to prove that the calculation is correct, that is, to prove $\log_{c_d} m_d = \log_{g_d} u$ and $\log_{c_a} m_a = \log_{g_a} v$. The party

who fails to pass the proof is malicious.

(8) If both parties pass the proof, Alice can get $d_i a(kx_0 + b - y_0)$ by calculating $L(m_d)/L(u)$ and judge whether its value is 0. David can get $a_j d(kx_0 + b - y_0)$ by calculating $L(m_a)/L(v)$ and judge whether its value is 0. If the result is 0, then point P_0 is on line L; otherwise, point P_0 is not on line L.

The protocol ends.

Correctness analysis: Through Protocol 2, both parties can fairly know whether point P_0 is on line L without disclosing their own information. The analysis is as follows:

(1) The $(c_{1a}^i, c_{2a}^i, c_{3a}^i)$ and $(c_{1d}^i, c_{2d}^i, c_{3d}^i)$ published in step 1 and 2 don't disclose information, because they add their own random numbers. The purpose of step 3 is to verify whether the correct ciphertext is published by both parties.

(2) In step 4 and step 5, Alice and David calculate respectively:

$$\begin{aligned}c_d &= (c_{1d}^i)^{ak}(c_{2d}^i)^{ab}(c_{3d}^i)^a r_1^{N_d} \bmod N_d^2\\ &= (g_d^{d_i x_0} \bmod N_d^2)^{ak}(g_d^{d_i} \bmod N_d^2)^{ab}\\ &\quad\times(g_d^{-d_i y_0} \bmod N_d^2)^a r_1^{N_d} \bmod N_d^2\\ &= g_d^{ad_i kx_0 + ad_i b - ad_i y_0} r_1^{N_d} \bmod N_d^2\\ &= g_d^{ad_i(kx_0+b-y_0)} r_1^{N_d} \bmod N_d^2\end{aligned}$$

$$\begin{aligned}c_a &= (c_{1a}{}^j)^d (c_{2a}{}^j)^{dx_0}(c_{3a}{}^j)^{dy_0} r_2^{N_a} \bmod N_a^2\\ &= (g_a^{a_j b} \bmod N_a^2)^d (g_a^{a_j k} \bmod N_a^2)^{dx_0}\\ &\quad\times(g_a^{-a_j} \bmod N_a^2)^{dy_0} r_2^{N_a} \bmod N_a^2\\ &= g_a^{a_j db + a_j dkx_0 - a_j dy_0} \bmod N_a^2\\ &= g_a^{a_j d(kx_0+b-y_0)} r_2^{N_a} \bmod N_a^2\end{aligned}$$

(3) In steps 7 and 8, the zero-knowledge proof is used to verify whether the two parties have made malicious attacks:

$$\begin{aligned}m &= \frac{L(m_d)}{L(u)} = \frac{L(c_d^{\lambda_d} \bmod N_d^2)}{L(g_d^{\lambda_d} \bmod N_d^2)} = \frac{L(g_d^m r_1^{N_d})^{\lambda_d} \bmod N_d^2}{L(1+N_d)^{\lambda_d} \bmod N_d^2}\\ &= \frac{L(1+N_d)^{m\lambda_d} \bmod N_d^2}{L((1+N_d\lambda_d) \bmod N_d^2} = \frac{\frac{1+N_d m\lambda_d - 1}{N_d}}{\frac{1+N_d\lambda_d - 1}{N_d}} = \frac{m\lambda_d}{\lambda_d}\\ &= d_i a(kx_0 + b - y_0) \overset{?}{=} 0.\end{aligned}$$

$$\begin{aligned}m &= \frac{L(m_a)}{L(v)} = \frac{L(c_a^{\lambda_a} \bmod N_a^2)}{L(g_a^{\lambda_a} \bmod N_a^2)} = \frac{L(g_a^m r_2^{N_a})^{\lambda_a} \bmod N_a^2}{L(1+N_a)^{\lambda_a} \bmod N_a^2}\\ &= \frac{L(1+N_a)^{m\lambda_a} \bmod N_a^2}{L((1+N_a\lambda_a) \bmod N_a^2} = \frac{\frac{1+N_a m\lambda_a - 1}{N_a}}{\frac{1+N_a\lambda_a - 1}{N_a}} = \frac{m\lambda_a}{\lambda_a}\\ &= a_j d(kx_0 + b - y_0) \overset{?}{=} 0.\end{aligned}$$

If $m=0$, then point P_0 is on line L, otherwise, point P_0 is not on line L.

In the whole process, no secret information is disclosed, and both parties can calculate the result, which avoids the unfairness caused by one party's calculation results telling the other party, and completes the judgment of the relationship between point and line in the malicious model, and the correctness is proved.

5 Security Proof

Security analysis: the status and operation of both parties in Protocol 2 are exactly equal, so we only analyze the possible malicious behaviors of David and its impact on Alice's data privacy.

(1) In step 2, if David uses different (x_0, y_0) in the process of calculating $(c_{1d}^i, c_{2d}^i, c_{3d}^i)$, Alice can't find it in the verification in step 3. As a result, the (x_0, y_0) in the $(c_{1d}^i, c_{2d}^i, c_{3d}^i)$ selected by Alice in step 5 is not a real number. This situation belongs to the situation of false input, which is also unavoidable in the ideal model and will not be considered.

(2) In the process of protocol, the only malicious behavior that David can successfully implement is that a random number d_i he selects does not meet the requirements. It is not found in the verification process in step 3, and it is just selected by Alice in step 5. In this way, Alice will not get the correct result.

If David wants to cheat through the above situation, his best choice is to set a group of $(c_{1d}^i, c_{2d}^i, c_{3d}^i)$ that does not meet the requirements in m groups of $(c_{1d}^i, c_{2d}^i, c_{3d}^i)$, and the other $m-1$ groups meet the requirements. In this way, the probability of successful cheating is the largest, equal to $1/m$, and the probability of success in other cases is smaller. If $m = 20$, and only one group does not meet the requirements, the probability of successful cheating is $\frac{C_{19}^{10}}{C_{20}^{10}} \times \frac{1}{10} = \frac{1}{200}$. If 10 groups do not meet the requirements, the probability of successful cheating is smaller or even minimal, that is: $\frac{C_{10}^{10}}{C_{20}^{10}} \times \frac{1}{2} = 2.7 \times 10^{-7}$. If David sets more than $m/2$ groups that do not meet the requirements, it will be found in the verification phase.

Alice may carry out malicious attacks and the probability of successful cheating is the same. Therefore, the protocol is secure. Next, we use the real/ideal model paradigm to prove that the protocol is secure in the malicious model.

Theorem 1: The MPC protocol for the point and line relationship is secure in the malicious model.

Prove: To prove that the protocol $\prod$ is secure in the malicious model, we must be able to convert the acceptable policy pair $\overline{A} = (A_1, A_2)$ in the real model protocol when executing $\prod$ into the corresponding policy pair $\overline{B} = (B_1, B_2)$ in the ideal model protocol, so that the output calculation of A_1, A_2 in $\prod$ cannot be distinguished from that of B_1, B_2 in the ideal protocol. Because both parties are not allowed to be dishonest at the same time in the malicious model, we deal with the two cases: A_1 is honest or A_2 is honest, respectively.

(1) A_1 is honest, A_2 is dishonest.

B_1 is determined by A_1, and he will execute the protocol according to the requirements of the protocol. In this case:

$$REAL_{\Pi,\bar{A}}\left[(k,b),(x_0,y_0)\right] = \left\{F\left[(k,b),A_2(x_0,y_0)\right],A_2\left((c_{1d}^i,c_{2d}^i,c_{3d}^i),m_d,S\right)\right\} \quad \textbf{(1)}$$

Where, F is the output result, S is the message sequence received by A_2 in the process of zero-knowledge proof, $1 \le i \le m$.

① In the ideal model, because B_1 imitates the behavior of honest A_1, it will send the real numbers k and b to TTP. What dishonest B_2 will send to TTP depends on A_2 's decision. B_2 sends (x_0,y_0) to A_2, then gets $A_2(x_0,y_0)$ from A_2, B_2 sends $A_2(x_0,y_0)$ to TTP, and gets $F\left[(k,b),A_2(x_0,y_0)\right]$ from TTP.

② Now, B_2 should use the $F\left[(k,b),A_2(x_0,y_0)\right]$ to try to obtain $view_{B_2}^F\left[(k,b),A_2(x_0,y_0)\right]$ that is indistinguishable from $view_{A_2}^{\Pi}\left[(k,b),A_2(x_0,y_0)\right]$ obtained by A_2 when actually executing the protocol, and give the $view_{B_2}^F\left[(k,b),A_2(x_0,y_0)\right]$ to A_2 to output.

- B_2 randomly select k' and b' so that $F\left[(k',b'),A_2(x_0,y_0)\right]=F\left[(k,b),A_2(x_0,y_0)\right]$. B_2 simulates the protocol with k' and b', that is, B_2 acts as A_1 with A_2 to execute the protocol and sends all messages required by step 1 of the protocol to A_2.
- B_2 publishes the information required to be published by A_1 in step 4 of the protocol.
- B_2 and A_2 get the corresponding m_d' for the remaining part of the execution protocol, prove to A_2 that the λ_d used in m_d' calculation is correct by using the zero-knowledge proof, and record the message sequence S' sent in the proof process.

③ B_2 calls A_2 with $\left((c_{1d}^{i'},c_{2d}^{i'},c_{3d}^{i'}),m_d',S'\right)$. Output $A_2\left((c_{1d}^{i'},c_{2d}^{i'},c_{3d}^{i'}),m_d',S'\right)$. In this way, we get:

$$IDEAL_{F,\bar{B}}\left[(k,b),(x_0,y_0)\right] = \left\{F\left[(k,b),A_2(x_0,y_0)\right],A_2\left((c_{1d}^{i'},c_{2d}^{i'},c_{3d}^{i'}),m_d',S'\right)\right\} \quad \textbf{(2)}$$

For A_2, $c_{1d}^i \overset{C}{\equiv} c_{1d}^{i'}$, $c_{2d}^i \overset{C}{\equiv} c_{2d}^{i'}$, $m_d \overset{C}{\equiv} m_d'$ (the first two $\overset{C}{\equiv}$ because they both encrypt the ciphertext with the same probability encryption algorithm, and the latter $\overset{C}{\equiv}$ because they are calculated by random numbers $d_i[kA(x_0)+b-A(y_0)]$ and $d_i'[kA(x_0)+b-A(y_0)]$, while d_i, d_i' is indistinguishable). At the same time, there is zero knowledge proof to ensure $S \overset{C}{\equiv} S'$, so there is

$$REAL_{\Pi\bar{A}}\left[(k,b),(x_0,y_0)\right] \overset{C}{\equiv} IDEAL_{F,\bar{B}}\left[(k,b),(x_0,y_0)\right] \quad \textbf{(3)}$$

(2) A_2 is honest, A_1 is dishonest.

B_2 is determined by A_2, and he will execute the protocol according to the requirements of the protocol. A_1 's decision has the following two situations:

① When executing Π, A_1 does not publish the results or does not pass zero knowledge proof, thus:

$$REAL_{\Pi\bar{A}}\left[(k,b),(x_0,y_0)\right] = \left\{A_1\left[(c_{1a}^i,c_{2a}^i,c_{3a}^i),m_a,S\right],\perp\right\} \quad \textbf{(4)}$$

Where S is the message sequence received by A_1 in the process of zero knowledge proof, $1 \le i \le m$.

② If A_1 publishes the results and passes the zero knowledge proof, A_2 will receive $F\left[A_1(k,b),(x_0,y_0)\right]$. at this time:

$$REAL_{\Pi\bar{A}}\left[(k,b),(x_0,y_0)\right] = \left\{A_1\left[(c_{1a}^i,c_{2a}^i,c_{3a}^i),m_a,S\right],F\left[A_1(k,b),(x_0,y_0)\right]\right\} \quad \textbf{(5)}$$

If A_1 publishes his own decryption results in the real model protocol and passes the final zero-knowledge proof, B_2 will get $F\left[A_1(k,b),(x_0,y_0)\right]$ in the ideal model. If A_1 does not publish the result or fails to pass the corresponding zero knowledge proof in the real model, B_1 tells TTP not to send the result to B_2 in the ideal model, and B_2 will get $\perp$.

B_1 finally obtains $view_{B_1}^F\left[A_1(k,b),(x_0,y_0)\right]$, which is indistinguishable from the $view_{A_1}^{\Pi}\left[A_1(k,b),(x_0,y_0)\right]$ obtained by A_1 when executing the real model protocol.

① B_1 randomly selects a group of x_0' and y_0' to make $F\left[A_1(k,b),(x_0',y_0')\right]=F\left[A_1(k,b),(x_0,y_0)\right]$, B_1 makes an protocol with A_1 by the role of A_2, publishes the corresponding (g_a',N_a',v'), and generates the corresponding $(c_{1a}^{i'},c_{2a}^{i'},c_{3a}^{i'})$ according to the x_0' and y_0'.

② B_1, acting as A_2, accepts the $(c_{1a}^{i'},c_{2a}^{i'},c_{3a}^{i'})$ selected by A_1 for verification in step 4.

③ B_1, acting as A_2, calculates m_a' and sends it to A_1, and proves that the calculated m_a' is correct by the zero-knowledge proof, that is, $m_a' = c_a'^{\lambda_a'} \bmod N_a'^2$.

④ Finally, B_1 obtains the $\left((c_{1a}^{i'},c_{2a}^{i'},c_{3a}^{i'}),m_a',S'\right)$ (we do not consider the subsequent malicious behavior of A_1, because A_1 has obtained the information required for the actual implementation of the protocol. He may not decrypt the c_d' or conduct the zero-knowledge proof in the end.)

⑤ A_1 outputs what B_1 outputs, that is, $\left((c_{1a}^{i'},c_{2a}^{i'},c_{3a}^{i'}),m_a',S'\right)$.

In the ideal model, when B_1 tells TTP not to send results to B_2:

$$IDEAL_{F\bar{B}}\left[(k,b),(x_0,y_0)\right]=\left\{A_1\left[\left(c_{1a}^{i'},c_{2a}^{i'},c_{3a}^{i'}\right),m_a',S'\right],\perp\right\} \quad \textbf{(6)}$$

In the ideal model, when B_1 tells TTP to send results to B_2 :

$$IDEAL_{F\bar{B}}\left[(k,b),(x_0,y_0)\right] = \left\{A_1\left[\left(c_{1a}^{i'},c_{2a}^{i'},c_{3a}^{i'}\right),m_a',S'\right],F\left[A_1(k,b),(x_0,y_0)\right]\right\} \quad \textbf{(7)}$$

In either case, the outputs of A_2 and B_2 are the same in the real and ideal model, as long as it is proved that $\left((c_{1a}^{i'},c_{2a}^{i'},c_{3a}^{i'}),m_a',S'\right)$ and $\left((c_{1a}^{i},c_{2a}^{i},c_{3a}^{i}),m_a,S\right)$ are computationally indistinguishable. Because $(c_{1a}^{i'},c_{2a}^{i'},c_{3a}^{i'})$ and $(c_{1a}^{i},c_{2a}^{i},c_{3a}^{i})$ are ciphertext encrypted by the same secure Paillier cryptosystem, their calculation is indistinguishable. m_a and m_a' are random numbers multiplied by constants, and then encrypted and modulo exponential operation. The calculation is also indistinguishable. Zero-knowledge proof guarantees $S\overset{C}{\equiv}S'$. So:

$$REAL_{\Pi\bar{A}}\left[(k,b),(x_0,y_0)\right]\overset{C}{\equiv}IDEAL_{F,\bar{B}}\left[(k,b),(x_0,y_0)\right] \quad \textbf{(8)}$$

To sum up, in the real model protocol, for $\bar{A}=(A_1,A_2)$, there is a $\bar{B}=(B_1,B_2)$ in the ideal model:

$$IDEAL_{F,\bar{B}}\left[(k,b),(x_0,y_0)\right]\overset{C}{\equiv}\left\{REAL_{\Pi,A}\left[(k,b),(x_0,y_0)\right]\right\} \quad \textbf{(9)}$$

Therefore, Protocol 2 is secure in the malicious model.

6 Efficiency Analysis

In terms of efficiency analysis, we compare the efficiency and overall performance with relevant references. In Reference [23], firstly, by comparing whether the slopes of the straight lines formed by two points of three points are equal, the relationship between points and line segments can be judged confidentially. Reference [24] solved the point inclusion problem by using inner product protocol and hash function through the transformation idea of triangular area problem.Based on Paillier cryptosystem, Reference [25] designed two efficient rational number interval secret calculation protocols and applied them to the problem of point and line determination. Reference [26] designed a protocol that can securely calculate two private points and straight lines based on the Paillier variant cryptosystem. However, the existing protocols are in the semi-honest model and can not resist malicious opponents. The proposed Protocol 2 can resist malicious adversary attacks. Table 1 is a comparison of the overall performance of Protocol 2 with Reference [23-26].

Table 1. Overall performance comparison

Protocol	Fair	Cryptography tools	Attacker model
Reference [23]	No	Paillier	Semi-honest model
Reference [24]	No	Inner product protocol, hash function	Semi-honest model
Reference [25]	No	Paillier	Semi-honest model
Reference [26]	No	Paillier variant	Semi-honest model
Protocol 2	Yes	Paillier, Zero-knowledge proof, cut-choose method	Malicious model

When analyzing the efficiency of a protocol, we usually consider two aspects: computational complexity and communication complexity. When measuring the computational complexity, because the complexity of modular exponentiation is much higher than that of other operations, it generally ignores other operational complexity of protocol execution and only considers the numbers of modular exponentiation operations with the highest computational cost.

Computational complexity: Reference [23] carries out two times Paillier cryptosystem operations and one time Paillier decryption operation, so a total of six modulus exponentiation operations are carried out. Reference [24] has a total of one inner product protocol. Reference [25] has carried out 4 times Paillier encryption operations, one time Paillier decryption operation and 3 times ciphertext modulus index operations, so a total of 13 times modulus exponentiation operations have been carried out. Reference [26] carries out $2(l_1+l)$ times Paillier variant encryption operations and l_1+l times Paillier variant decryption operations, so a total of $6(l_1+l)$ times modulus exponentiation operations are carried out (both l_1, l are the number of random ciphertexts selected by the author).

In Protocol 2, Alice and David start to perform one modulo exponential operation respectively; subsequently, the two parties carried out $3m$ times modulus exponentiation operations respectively; if both parties verify the $m/2$ modular index, $3m/2$ modulus exponentiation operation will be carried out respectively, and $3m$ modulus exponentiation operation will be carried out in total. Both parties will prove the zero-knowledge of discrete logarithm once respectively, and 6 modular exponential operations are required each time, with a total of 12 modular exponential operations. Two Paillier decryption operations need two modulus exponentiation operations. To sum up, a total of $9m+16$ modular exponential operations are required (By analysis, generally $m=20$ is enough).

From the above analysis, it can be seen that the computational complexity of Protocol 2 proposed in this paper has increased, but not much. Protocol 2 is not comparable with other protocols, because of different application scenarios and security, Protocol 2 can resist malicious behavior. so Protocol 2 has wide applications. See Table 2 for details.

Communication complexity: Reference [23] carried out 2 rounds of communication. Reference [24] carried out 3 rounds of communication. Reference [25] carried out 2 rounds of communication. Reference [26] carried out l_1+l+4 rounds of communication. In Protocol 2, Alice and David made 8 rounds of communication.

Table 2. Efficiency comparison of different protocols

Protocol	Computational complexity (modulus exponentiation)	Communication rounds	Resist malicious adversaries
Reference [23]	6	2	×
Reference [24]	1 inner product protocol	3	×
Reference [25]	13	2	×
Reference [26]	$6(l_1+l)$	l_1+l+4	×
Protocol 2	$9m+16$	8	√

Experimental simulation: In order to show the efficiency of our protocol, we compare Protocol 2 with references [23-26]. The specific experimental environment is windows10 (64 bit) operating system, Intel (R) core (TM) i7-5500u CPU @ 2.40GHz processor and 8.00gb memory. The experiment is carried out with Python language.

Figure 1 is a comparison diagram of the time consumption of each protocol with the increase of modulus. In the experiment, the modulus of Paillier encryption algorithm is the same, and the preprocessing time of each protocol is ignored in the experiment. Calculate the average execution time of five protocols under 128, 256, 512 and 1024 bit modulus respectively. Among them, the ordinate represents the time consumed (ms) and the abscissa represents different modulus (bit) (in Reference [24], we set the modulus of each time as the different time taken to complete an inner product protocol.) As can be seen from the figure, the execution time of Protocol 2 is better than references [23] and [26] when the modulus ≤ 512. Protocol 2 is better than Reference [24] when the modulus > 512. And the execution time of Protocol 2 tends to increase steadily with the increase of modulus.

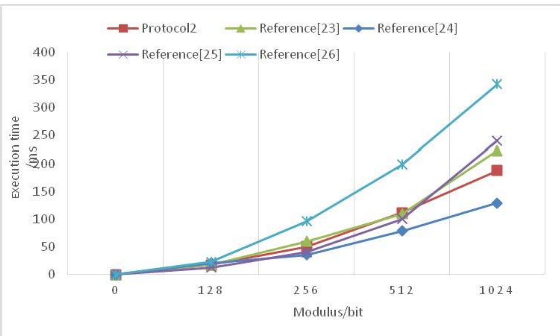

Figure 1. Comparison of execution time of different modules

To sum up, Protocol 2 designed in this paper not only maintains excellent communication efficiency, but also can resist malicious enemy attacks. The security of the protocol is greatly improved and more practical. For the first time, the MPC protocol for the point and line relationship in the malicious model is proposed. This protocol is optimal compared with the existing protocols in terms of computational efficiency, communication efficiency and performance. (**Note:** For the malicious model protocol, bit commitment, cut-choose method and zero- knowledge proof are generally used to force malicious participants to act like semi-honest participants. The increase of bitcoin commitment, cut-choose method and zero-knowledge proof will greatly increase the computational complexity and reduce the execution efficiency, which makes the efficiency of malicious model protocol and semi-honest model protocol impossible to compare. We can use preprocessing or compute outsourcing to improve efficiency, and these two methods are feasible in our protocol. This part of the calculation has nothing to do with confidential data. It can be precomputed or outsourced, which can at least double the efficiency.)

7 Applications

The secure computational geometry problems have always been a hot issue in MPC. The secure determination of points and lines has important applications in aviation, land, marine and other military fields (as shown in Figure 2).

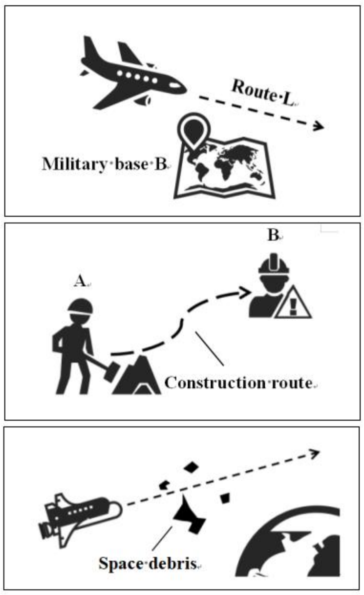

Figure 2. Application examples

Example 1: The route of airline A wants to confirm whether it will break into the military base B. In order not to affect the operation of military base, they want to cooperate to calculate whether the route intersects with the base without disclosing their route position and base position.

Example 2: Two construction teams A and B hope to cooperate in the development of a certain area. If A and B want to ensure that the construction site of A is not on the construction route of B without disclosing their construction plan to each other in advance.

Example 3: In the process of human research and development of space, different organizations have developed their own space debris distribution map, and both sides ensure that their own aircraft will not collide with the space debris of other organizations in the flight process of their own flight routes without disclosing their own data information to each other.

8 Conclusion

Securely computing the relationship of the point and line is widely used. Most of the current MPC protocols are designed in the semi-honest model. On the premise of analyzing the possible malicious behaviors, this paper designs a MPC protocol of the relationship between point and line in the malicious model by combining the zero-knowledge proof and cut-choose methods, which can resist malicious attacks and make our scheme more fair and feasible. The protocol remains efficient through the efficiency analysis and has better security by real/ideal model paradigm proof. In the future, we will work out more MPC protocols in the malicious model, which will make the actual MPC protocols both effective and secure.

Acknowledgments

Xin Liu would like to acknowledge the support from Inner Mongolia Discipline Inspection and Supervision Big Data Laboratory Open Project Fund (IMDBD202020), Inner Mongolia Natural Science Foundation (2021MS06006), 2022 Basic Scientific Research Project of Direct Universities of Inner Mongolia (20220101), 2022 Fund Project of Central Government Guiding Local Science and Technology Development (20220175), Baotou Kundulun District Science and Technology Plan Project (YF2020013), 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 (20220213), the 2022 Ministry of Education Central and Western China Young Backbone Teachers and Domestic Visiting Scholars Program (20220393). Gang Xu would like to acknowledge the support from NSFC (92046001, 61962009), Basic Scientific Research Business Fee Project of Beijing Municipal Commission of Education (110052972027), Research Startup Fund Project of North China University of Technology (110051360002).

References

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

[2] O. Goldreich, S. Micali, A. Wigderson, How to Play Any Mental Game, *Proceedings of The Nineteenth Annual ACM Symposium on Theory of Computing*, New York, USA, 1987, pp. 218-229.

[3] O. Goldreich, *Foundations of Cryptography- Volume 2: Basic Applications*, Ph. D. Cambridge University Press, London, UK, 2009.

[4] A. Dm, B. Yih, C. Tm, Practical Card-based Implementations of Yao's millionaire Protocol, *Theoretical Computer Science*, Vol. 803, pp. 207-221, January, 2020.

[5] M. Liu, Y. Luo, P. Nanda, S. Yu, J. Zhang, Efficient Solution to The Millionaires' problem Based on Asymmetric Commutative Encryption Scheme, *Computational Intelligence*, Vol. 35, No. 3. pp. 555-576, August, 2019.

[6] S.-D. Li, M.-Y. Zhang, An Efficient Solution to the Blind Millionaires' Problem, *Chinese Journal of Computers*, Vol. 43, No. 9, pp. 1-14, September, 2020.

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

[8] D. W. H. A. Silva, C. P. Araujo, E. Chow, B. S. Barillas, A New Approach Towards Fully Homomorphic Encryption Over Geometric Algebra, *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference*, New York, USA, 2019, pp. 241-249.

[9] Q. Wei, S. Li, W. Wang, Y. Yang, Privacy-preserving Computational Geometry, *International Journal of Network Security*, Vol. 21, No. 6, pp. 1071-1080, November, 2019.

[10] U. Martínez-Penas, Communication Efficient and Strongly Secure Secret Sharing Schemes Based on Algebraic Geometry Codes, *IEEE Transactions on Information Theory*, Vol. 64, No. 6, pp. 4191-4206, June, 2018.

[11] J. Liu, Y. Tian, Y. Zhou, Y. Xiao, N. Ansari, Privacy Preserving Distributed Data Mining Based on Secure Multi-party Computation, *Computer Communications*, Vol. 153, pp. 208-216, March, 2020.

[12] J. Li, H. Huang, Faster Secure Data Mining via Distributed Homomorphic Encryption, *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Virtual Event, CA, USA, 2020, pp. 2706-2714.

[13] Y. Yamamoto, M. Oguchi, Distributed Secure Data Mining with Updating Database Using Fully Homomorphic Encryption, *International Conference on Ubiquitous Information Management and Communication*, Phuket, Thailand, 2019, pp. 937-949.

[14] S. Nishoni, A. Tenis, Secure Communication With Data Analysis and Auditing Using Bilinear Key Aggregate Cryptosystem in Cloud Computing, *Materials Today: Proceedings*, Vol. 24, No. 4, pp. 2358-2365, 2020.

[15] N. Almutairi, F. Coenen, K. Dures, A Cryptographic Ensemble for Secure Third Party Data Analysis:

Collaborative Data Clustering without Data Owner Participation, *Data & Knowledge Engineering*, Vol. 126, Article No. 101734, March, 2020.
[16] M.-Y. Ma, Z. Liu, Y. Xu, L. Wu, Private-preserving determination problem of integer-interval positional relationship, *Journal of Computer Applications*, Vol. 40, No. 9, pp. 2657-2664, September, 2020.
[17] J. Zhang, *Research on Application Protocols of Secure Multi-Party Geometry and Sets Computation,* Ph. D. Beijing Jiaotong University, Beijing, China, 2020.
[18] K. He, L. Yang, J. Hong, J. Jiang, J. Wu, X. Dong, Z. Liang, PrivC—A Framework for Efficient Secure Two-Party Computation, *International Conference on Security and Privacy in Communication Systems*, Orlando, FL, USA, 2019. pp. 394-407.
[19] Q. Feng, D. He, Z. Liu, H. Wang, K.-K. R. Choo, Secure NLP: A System for Multi-Party Privacy-Preserving Natural Language Processing, *IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 3709 - 3721, May, 2020.
[20] Z. Gheid, Y. Challal, X. Yi, A. Derhab, Efficient and Privacy-aware Multi-party Classification Protocol for Human Activity Recognition, *Journal of Network and Computer Applications*, Vol. 98, pp. 84-96, November, 2017.
[21] C. Zhao, S. Zhao, M. Zhao, Z. Chen, C.-Z. Gao, H. Li, Y.-A. Tan, Secure Multi-Party Computation: Theory, Practice and Applications, *Information Sciences*, Vol. 476, pp. 357-372, February, 2019.
[22] J. Kolberg, P. Drozdowski, M, Gomez-Barrero, C, Rathgeb, C. Busch, Efficiency Analysis of Post-quantum-secure Face Template Protection Schemes Based on Homomorphic Encryption, *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*, Darmstadt, Germany, 2020, pp. 1-4.
[23] X.-J. Zuo, X.-L. Yang, S.-D. Li, Privately Determining Protocol on Three Points Are Collinear and Its Applications, *Journal of Cryptography Research*, Vol. 3, No. 3, pp. 238-248, 2016.
[24] J. Dou, W. Wang, S. Li, Privately Determining Interval Location Relation, *Chinese Journal of Computers*, Vol. 42, No. 5, pp. 1031-1044, May, 2019.
[25] Y. Guo, S. Zhou, J. Dou, S. Li, D. Wang, Efficient Privacy-Preserving Interval Computation and Its Applications, *Chinese Journal of Computers*, Vol. 40, No. 7, pp. 1664-1679, July, 2017.
[26] L. M. Gong, S. D. Li, J. W. Dou, Y. M. Guo, D. S. Wang, Homomorphic Encryption Scheme and A Protocol on Secure Computing a Line by Two Private Points, *Journal of software*, Vol. 28, No. 12, pp. 3274-3292, December, 2017.
[27] P. Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, *International Conference on the Theory and Applications of Cryptographic Techniques*, Prague, Czech Republic, 1999, pp. 223-238.
[28] X. Liu, *Research on Several Problems of Secure Multiparty Computing*, Ph. D. Shaanxi Normal University, Xi'an, China, 2017.

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

Yang Xu was born in 1996 and is a graduate student. His research interests are cryptography and secure multi-party computation.
Email: 1016051241@qq.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: flyover100@163.com

Yu-Ling Chen received the M.S. degree from Guizhou University in 2009. She is currently an associate professor in Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang, China. Her recent research interests include cryptography and information security. Email: Ylchen3@gzu.edu.cn