Privacy-preserving Computational Geometry

Qiong Wei[1], Shundong Li[1], Wenli Wang[2], and Yanjing Yang[1]
(Corresponding author: Shundong Li)
School of Computer Science, Shaanxi Normal University, Xi'an 710119, China[1]
No.199, South Chang'an Road, Yanta District, Xi'an 710062
School of Mathematics and Information Science, Shaanxi Normal University, Xi'an 710119, China[2]
(Email: weiqiong@snnu.edu.cn)
(Received Nov. 20, 2018; Revised and Accepted May 18, 2019; First Online June 15, 2019)

Abstract

Secure multi-party computation (SMC) is a research hotspot in the field of international cryptography. Privacy-preserving computational geometry (PPCG) is the main branch of SMC. In this paper, we first design a protocol for deciding intersection of line segments, which can be used to determine whether polygons intersect. Then, we design a protocol to privately compute distance from a point to a plane, which is applicable to rational numbers or integers. We theoretically analyze the correctness, security and efficiency of the protocols. Based on the distance protocol, we construct a protocol to compute volume of a tetrahedron, and other two protocols to determine position relation between a line and a plane, and that between two planes. Finally, we analyze the efficiency of the protocol for determining position relation between a line and a plane and verify the analysis with experimental simulation.

Keywords: Cryptography; Distance from a Point to a Plane; Position Relation; Privacy-Preserving Computational Geometry; Secure Multi-party Computation

1 Introduction

SMC is a collaborative private computation between a group of non-trusted parties. It is an important technology of privacy protection in the information society and a research hotspot in the international cryptography field. SMC enables participants with private data to cooperate with each other in some joint computations without revealing their private data, thus enabling people to maximize the use of private data without compromising the privacy of the data. Therefore, SMC is widely used in data mining [20], data query [2], outsourcing computing [7] and so on [8,9]. SMC was first proposed by professor Yao [22], a Turing prize winner. Goldreich, Micali and others have done a lot of researches on the basis of Yao's work, which laid a theoretical foundation for SMC [19]. Goldwasser also predicted that SMC would become a vital part of computing science.

PPCG is an important area of SMC, which mainly studies the information security in the geometric cooperative computation. Du *et al.* [4] first introduced PPCG problem including point inclusion problem, intersection problem of two segments, intersection problem of two convex polygons, and convex hull problem of several secret points, and proposed solutions to these problems. Zhang *et al.* [23] studied the point inclusion problem. Liu *et al.* [12] studied the computation of triangle area in plane. Zuo *et al.* [24] solved the problem of whether three points are collinear based on Paillier homomorphic encryption scheme. Li *et al.* [10] proposed a secure solution to determine whether two graphics are similar. Luo *et al.* [13] studied the problem of determining the relationship between two lines, between a line and a plane, and between two planes. However, as the scheme calls for multiple basic protocols, such as comparing equal protocol, inner product protocol and data corresponding proportional protocol, the communication and computation costs of this scheme are very high. Li *et al.* [11] solved the determination of spatial position relation by computing the volume and height of tetrahedron. The scheme was efficient, but its ratio relation was disclosed when comparing the height of different tetrahedrons. Yang *et al.* [21] solved the decision problem of intersection between a straight line and a plane, in which the socialist millionaire protocol was invoked. The scheme requires multiple operations of encryption and decryption and cannot solve the rational number problem. Chen *et al.* [3] solved the problem of privately determining the position relations of various geometric objects in space. This scheme mainly uses the Boneh homomorphic encryption and inner product protocols, and outsources complex computing tasks to the cloud. However, the protocol brings additional high computational costs. If the computing is not outsourced to the cloud, the solution will be very inefficient. In addition, the scheme also converts the determination of the position relationship between a straight line and a plane and between two planes into an angle problem, thus revealing the angle information.

Sun *et al.* studied the problem of deciding intersec-

tion of line segments [18]. The scheme proposed by Sun *et al.* [18] invokes a complex millionaire protocol and discloses the endpoint information of these segments when looking for the intersection point. Luo *et al.* [14] and other existing schemes invoke many basic protocols, so the computational complexity is higher.

In this paper, we design a new protocol to determine whether two segments intersect, which solves the problems existing in previous schemes and can be used to determine the intersection of polygons. As for the secure computation problem of the distance from a point to a plane, the existing research uses inner product protocol in [13] to solve it, which has a high computational complexity. We design a secure computation protocol for distance from a point to a plane. On the basis of the distance problem, we design secure and efficient protocols to privately determine the position relation between a line and a plane, and between two planes in space.

Privately determining position relations of geometric objects is of great importance in practical applications. Consider the following two scenarios. Scenario 1: During the war, country A and country B are going to build a railway in country C, but the construction route will be kept secret until the railway is completed. In order to prevent future train collisions, countries A and B hope to determine whether the two routes will intersect without disclosing their own routes, so as to negotiate in advance and avoid accidents. Scenario 2: Two airlines have designed routes L_1 and L_2 between A and B. In order to ensure the safety of the routes, they need to determine whether the two routes will intersect, but in order not to lose the economic interests of the two airlines, they should not disclose their respective route information. Therefore, they want to determine whether L_1 and L_2 will intersect without disclosing their own route. In reality, many problems can be reduced to privately determine the position relations of geometric objects, so this problem has important research significance and value.

Our contributions: The main contributions of this paper are as follows.

1) In order to determine whether two segments intersect, we design a protocol based on the Paillier encryption algorithm, which avoids calling millionaire protocol and improves the efficiency;
2) In order to compute the distance from a point to a plane privately, we design an efficient protocol based on the Paillier homomorphic encryption algorithm, which solves the problem that the Paillier algorithm cannot directly encrypt non-integers, so that the protocol can solve not only the integer problem, but also the rational number problem;
3) By using the protocol for distance from a point to a plane, we further discuss and solve the problem of privately computing the volume of tetrahedron, the problem of privately determining position relation between a line and a plane, and between two planes;
4) In this paper, only a small amount of encryption and decryption operations are needed in the protocol to privately determine position relation between a line and a plane, and between two planes. In addition, the angle between line and plane will not be disclosed when the line intersects plane and the two planes intersect. Therefore, our protocol is secure and efficient.

Paper organization: The rest of this paper is organized as follows: Section 2 introduces some preliminaries. Section 3 presents a protocol for secure line segment intersection problem. Section 4 gives a protocol for secure distance from a point to a plane. Section 5 gives some applications of the protocol for the secure distance from a point to a plane. Section 6 analyses the efficiency of our protocols. Section 7 concludes the paper.

2 Preliminaries

2.1 Security

Semi-honest parties [6]. The protocols and securities proposed in this study are all based on a semi-honest model. A semi-honest party will follow the prescribed protocol exactly, but he may record the results of all intermediate computations and try to derive other parties' private inputs from the record. Goldreich has proved that, a protocol which can privately compute a function f in the semi-honest model can be complied, by introducing a bit commitment macro, into another protocol which can compute the function f in the malicious model. The semi-honest model is not only an important methodological tool but also provides a good model in many settings. It suffices to prove that a protocol is secure in the semi-honest model.

Two-party computation. Two-party computation represents a randomized computation process that maps a random input pair to an output pair: $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$. This implies that given an input pair (x, y), the function will output two random variables $(f_1(x, y), f_2(x, y))$. The function is denoted by $f : (x, y) \to (f_1(x, y), f_2(x, y))$.

Privacy by simulation [17]. At present, simulations are used widely to prove the security of SMC protocols by simulating the execution process of SMC. The mathematical expression for the simulation is as follows.

Alice and Bob want to compute function f privately. Assume that $f = (f_1, f_2) : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times$

$\{0,1\}^*$ is a probabilistic polynomial time function and that π represents a two-party protocol for computing f. When the input is (x,y), the message sequence obtained from the execution of protocol π is denoted by $view_i^{\pi}(x,y) = (x, r^i, m_1^i, m_2^i, \cdots, m_t^i)$, where i represents the ith participant, r^i represents the random number generated by the ith participant, and m_j^i represents the jth message that the ith participant obtains. The output of the participant is denoted by $output_i^{\pi}(x,y)(i=1,2)$.

Definition 1. *For a function $f(x,y)$, π privately computes f if probabilistic polynomial time algorithms S_1 and S_2 exist such that*

$$\{S_1(x, f_1(x,y))\}_{x,y} \stackrel{c}{\equiv} \{view_1^{\pi}(x,y)\}_{x,y} \quad (1)$$

$$\{S_2(y, f_2(x,y))\}_{x,y} \stackrel{c}{\equiv} \{view_2^{\pi}(x,y)\}_{x,y} \quad (2)$$

where $\stackrel{c}{\equiv}$ represents the computational indistinguishability.

2.2 Homomorphic Encryption

Homomorphic encryption [5] plays a crucial role in SMC and cloud computing security. One of the most important properties of homomorphic encryption is that it can perform some operations on the ciphertext without knowing the decryption key to ensure the privacy of the plaintext. Additively and multiplicatively homomorphic encryption are two general homomorphic types used in current researches. Paillier [16] designs an additively homomorphic encryption scheme that satisfies

$$\begin{aligned} E(m_1) \cdot E(m_2) &= E(m_1 + m_2) \\ (E(m_1))^{m_2} &= E(m_1 \cdot m_2). \end{aligned}$$

The Paillier cryptosystem can be constructed as follows.

Setup. Given a security parameter k, let $N = p \times q$, where p and q are two large primes, $\lambda = lcm(p-1, q-1)$ is the least common multiple of $p-1$ and $q-1$. Choose a $g \in Z_N^*$ at random such that $\gcd(L(g^{\lambda} \bmod N^2), N) = 1$, where $L(x) = \frac{x-1}{N}$. The public key of the cryptosystem is (g, N), and the private key is λ.

Encryption. To encrypt message $m < N$, choose a random number $r < N$, and compute

$$c = E(m) = g^m r^N \mod N^2$$

Decryption. Compute

$$m = D(c) = \frac{L(c^{\lambda} \bmod N^2)}{L(g^{\lambda} \bmod N^2)} \mod N^2$$

2.3 Distance from a Point to a Plane

Given a plane $Ax + By + Cz + D = 0$, the distance from point $P_0(x_0, y_0, z_0)$ to the plane can be denoted by:

$$d = \frac{|Ax_0 + By_0 + Cz_0 + D|}{\sqrt{A^2 + B^2 + C^2}} \quad (3)$$

2.4 Volume of Tetrahedron

Let the bottom area of the tetrahedron be S and the height be d. The volume of an arbitrary tetrahedron can be denoted by:

$$V = \frac{1}{3} Sd. \quad (4)$$

3 Privately Determine whether Two Segments Intersect

Suppose that Alice and Bob have segments L_1 and L_2, respectively. They want to know whether L_1 intersects with L_2 without disclosing any other information.

3.1 Basic Principle

Suppose that the two endpoints of L_1 and L_2 are $P_1(x_1, y_1), P_2(x_2, y_2)$ and $P_3(x_3, y_3), P_4(x_4, y_4)$, respectively, and the equations of the two straight line L_1 and L_2 are $l_1 : y = f_1(x) = k_1 x + b_1$, $l_2 : y = f_2(x) = k_2 x + b_2$, respectively. For L_1 and L_2, first determine whether L_1 intersects with straight line l_2, that is, whether P_1 and P_2 are on both sides of l_2. If they are on the same side, then the two segments do not intersect. Otherwise, continue to determine whether P_3 and P_4 are on both sides of l_1. If they are on both sides, then L_1 intersects with L_2, otherwise it does not. Therefore, the problem is converted into determining whether a segment intersects with a straight line. If $(y_1 - f_2(x_1)) \times (y_2 - f_2(x_2)) \le 0$, then P_1 and P_2 are located on both sides of the straight line l_2 (including a point on the line). (When the slope of the straight line does not exist, the line is a vertical line of $x = C$, where C is a constant, just determine whether $(x_1 - x) \times (x_2 - x) \le 0$. This paper only considers general situations.) Therefore, we compute

$$\begin{aligned} m &= (y_1 - f_2(x_1)) \times (y_2 - f_2(x_2)) \\ &= u_1 + U_1 + u_2 U_2 + u_3 U_3 + u_4 U_4 + u_5 U_5 \end{aligned}$$

where $u_1 = y_1 y_2, u_2 = x_2 y_1 + x_1 y_2, u_3 = y_1 + y_2, u_4 = x_1 x_2, u_5 = x_1 + x_2, U_1 = {b_2}^2, U_2 = -k_2, U_3 = -b_2, U_4 = {k_2}^2, U_5 = k_2 b_2$. If $m \le 0$, then P_1, P_2 are on both sides of l_2.

Similarly, if $(y_3 - f_1(x_3)) \times (y_4 - f_1(x_4)) \le 0$, then P_3 and P_4 are located on both sides of l_1. Therefore, we compute

$$\begin{aligned} n &= (y_3 - f_1(x_3)) \times (y_4 - f_1(x_4)) \\ &= v_1 + V_1 + v_2 V_2 + v_3 V_3 + v_4 V_4 + v_5 V_5 \end{aligned}$$

where $v_1 = y_3 y_4, v_2 = -(x_3 y_4 + x_4 y_3), v_3 = -(y_3 + y_4), v_4 = x_3 x_4, v_5 = x_3 + x_4, V_1 = {b_1}^2, V_2 = k_1, V_3 = b_1, V_4 = {k_1}^2, V_5 = k_1 b_1$. If $n \le 0$, then the two endpoints P_3, P_4 are on both sides of l_1.

L_1 intersects with L_2 if and only if $m \le 0$ and $n \le 0$.

In addition, an important step in determining whether two polygons intersect is to determine whether they have

a set of intersecting edges. Therefore, the method to determine whether two segments intersect can also be used to determine whether two polygons intersect.

For simple exposition, we define

$$P(L_1, L_2) = \begin{cases} 0, & L_1 \text{ intersects with } L_2 \\ 1, & \text{otherwise} \end{cases}$$

Proposition 1. *For the Paillier cryptosystem, $N = p \times q$, where p and q are two large primes. Suppose that $0 \leq u, v < N/2$, $C = E(u)E(N - v)$ and $w = D(C)$, we have the following conclusions: $u = v$ if and only if $w = 0$; $u > v$ if and only if $0 < w < N/2$; $u < v$ if and only if $w > N/2$.*

Proof. According to the definition and additive homomorphism of the Paillier encryption algorithm we know that $C = E(u)E(N - v) = E(N + u - v \bmod N)$. Then we decrypt C with private key and get $w = D(C) = (u - v) \bmod N$.

1) when $u = v$, $w = D(C) = (u - v) \bmod N = 0$.
2) when $u > v$, because $0 < u - v < N/2$, $w = (u - v) \bmod N = u - v < N/2$.
3) when $u < v$, because $-N/2 < u - v < 0$, $w = (u - v) \bmod N = N + u - v > N/2$.

Because $w = D(C) = (u-v) \bmod N \in Z_N$ and $0 \leq u, v < N/2$, w can only get one of the three results: $w = 0, 0 < w < N/2$ or $N/2 < w < N$. This completes the proof of the proposition. □

3.2 Protocol Design

Protocol 1: Privately determine whether two segments intersect.

Inputs: Private segments L_1 and L_2.

Output: $P(L_1, L_2)$.

1) Alice generates the public key and private key of the Paillier encryption scheme, and tells the public key to Bob.
2) Alice takes the two endpoints $P_1(x_1, y_1), P_2(x_2, y_2)$ of segment L_1, and computes $u_1 = y_1y_2, u_2 = x_2y_1 + x_1y_2, u_3 = y_1 + y_2, u_4 = x_1x_2, u_5 = x_1 + x_2$. Then, Alice encrypts u_1, u_2, u_3, u_4, u_5 with public key to get $E(u_1), E(u_2), E(u_3), E(u_4), E(u_5)$, and sends the ciphertexts to Bob.
3) Bob first computes $U_1 = {b_2}^2$, $U_2 = -k_2$, $U_3 = -b_2$, $U_4 = {k_2}^2$, $U_5 = k_2b_2$, and then computes $Z_1 = E(u_1)E(U_1)E(u_2)^{U_2}E(u_3)^{U_3}E(u_4)^{U_4}E(u_5)^{U_5}E(N)$. Bob sends Z_1 to Alice.
4) Alice decrypts Z_1 to get z_1.
5) If $z_1 \in (0, N/2)$, then segments L_1 and L_2 do not intersect. Alice outputs $P(L_1, L_2) = 1$. Otherwise, they proceed with the following steps.
6) Alice computes $V_1 = {b_1}^2$, $V_2 = k_1$, $V_3 = b_1$, $V_4 = {k_1}^2$, $V_5 = k_1b_1$, and encrypts V_1, V_2, V_3, V_4, V_5 to get $E(V_1), E(V_2), E(V_3), E(V_4), E(V_5)$. Alice sends ciphertexts to Bob.
7) Bob first computes $v_1 = y_3y_4$, $v_2 = -(x_3y_4 + x_4y_3)$, $v_3 = -(y_3 + y_4)$, $v_4 = x_3x_4$, $v_5 = x_3 + x_4$, and then computes $Z_2 = E(V_1)E(v_1)E(V_2)^{v_2}E(V_3)^{v_3}E(V_4)^{v_4}E(V_5)^{v_5} \cdot E(N)$. Bob sends Z_2 to Alice.
8) Alice decrypts Z_2 to get z_2.
9) If $z_2 \in (0, N/2)$, then L_1 and L_2 do not intersect. Alice outputs $P(L_1, L_2) = 1$. Otherwise, Alice outputs $P(L_1, L_2) = 0$.

3.3 Correctness

Theorem 1. *Protocol 1 can correctly determine whether two segments intersect.*

Proof. According to the additive homomorphism of the Paillier encryption algorithm, we can get $Z_1 = E(u_1)E(U_1)E(u_2)^{U_2}E(u_3)^{U_3}E(u_4)^{U_4}E(u_5)^{U_5}E(N) = E(u_1U_1 + u_2U_2 + u_3U_3 + u_4U_4 + u_5U_5 + N) = E(m + N)$. Alice decrypts Z_1 to get z_1. According to Proposition 1 we can see that: if $m > 0$, then $(m + N) \bmod N < N/2$; if $m = 0$, then $(m + N) \bmod N = 0$; otherwise, $(m + N) \bmod N > N/2$. Therefore, Alice can determine whether m is positive or negative based on z_1 according to Proposition 1. According to the calculation principle, we can correctly determine whether the two endpoints P_1 and P_2 of L_1 are on both sides of the straight line l_2 by judging whether m is positive or negative. Similarly, Alice can determine whether the two endpoints P_3 and P_4 of L_2 are on both sides of the straight line l_1 by judging whether n is positive or negative. Therefore, Protocol 1 can correctly determine whether two segments intersect. This completes the proof of the theorem. □

3.4 Security

By the simulation paradigm, we can prove that Protocol 1 is secure. The theorem is proved by constructing simulators S_1 and S_2 such that Equations (1) and (2) hold.

Theorem 2. *Protocol 1 for determining whether two segments intersect is secure.*

Proof.

1) S_1 selects an arbitrary segment L_2' (with coordinates of $P_3'(x_3', y_3'), P_4'(x_4', y_4')$ and linear equation $y' = f_2'(x) = k_2'x + b_2'$) such that $P(L_1, L_2') = P(L_1, L_2)$.
2) S_1 computes the linear equation $y' = f_2'(x) = k_2'x + b_2'$ of L_2' according to P_3', P_4'.
3) S_1 computes $U_1' = {b_2'}^2$, $U_2' = -k_2'$, $U_3' = -b_2'$, $U_4' = {b_2'}^2$, $U_5' = k_2'b_2'$ and computes $Z_1' = E(u_1)E(U_1')E(u_2)^{U_2'}E(u_3)^{U_3'}E(u_4)^{U_4'}E(u_5)^{U_5'}E(N)$.

4) S_1 decrypts Z_1' to obtain z_1'. If the protocol is terminated at this time,

$$view_1^{\pi}(L_1, L_2) = \{L_1, Z_1, P(L_1, L_2)\}.$$

The information sequence generated in the simulation process is: $S_1(L_1, f_1(L_1, L_2)) = \{L_1, Z_1', P(L_1, L_2')\}$.

According to the computation process, for Alice: $Z_1' \stackrel{c}{\equiv} Z_1$, and $P(L_1, L_2') = P(L_1, L_2)$, therefore

$$\{S_1(L_1, f_1(L_1, L_2))\} \stackrel{c}{\equiv} \{view_1^{\pi}(L_1, L_2)\}.$$

Simulator S_2 can be constructed like this, and the following formula holds

$$\{S_2(L_2, f_2(L_1, L_2))\} \stackrel{c}{\equiv} \{view_2^{\pi}(L_1, L_2)\}.$$

If it is impossible to determine whether the two segments intersect at this point, then the second part of the protocol needs to be performed. The following simulation process is as follows:

5) S_1 computes $v_1' = y_3'y_4'$, $v_2' = -(x_3'y_4' + x_4'y_3')$, $v_3' = -(y_3'+y_4')$, $v_4' = x_3'x_4'$, $v_5' = x_3'+x_4'$, and computes $Z_2' = E(V_1)E(v_1')E(V_2)^{v_2'}E(V_3)^{v_3'}E(V_4)^{v_4'}E(V_5)^{v_5'}E(N)$.

6) S_1 decrypts Z_2' to get z_2'.

During the execution of Protocol 1,

$$view_1^{\pi}(L_1, L_2) = \{L_1, Z_1, Z_2, P(L_1, L_2)\}.$$

The information sequence generated in the simulation process is:

$$S_1(L_1, f_1(L_1, L_2)) = \{L_1, Z_1', Z_2', P(L_1, L_2')\}.$$

From the above part of proof, we can see: $Z_1' \stackrel{c}{\equiv} Z_1$; according to the computation process, for Alice: $Z_2' \stackrel{c}{\equiv} Z_2$ and $P(L_1, L_2') = P(L_1, L_2)$, therefore

$$\{S_1(L_1, f_1(L_1, L_2))\} \stackrel{c}{\equiv} \{view_1^{\pi}(L_1, L_2)\}.$$

Similarly, simulator S_2 can be constructed like this, and the following formula holds

$$\{S_2(L_2, f_2(L_1, L_2))\} \stackrel{c}{\equiv} \{view_2^{\pi}(L_1, L_2)\}.$$

This completes the proof of the theorem. □

4 Privately Compute the Distance from a Point to a Plane

Suppose that Alice has a plane π: $Ax + By + Cz + D = 0$ and Bob has a point $P_0(x_0, y_0, z_0)$. They want to know the distance between P_0 and π without disclosing other information about the point and the plane.

4.1 Basic Principle

According to Equation (3), the distance from a point to a plane can be computed directly, but there is no secrecy in doing so. Therefore, we design a secure Protocol 2 to compute the distance. Protocol 2 is mainly implemented by using the Paillier homomorphic encryption algorithm. Suppose that A, B, C, and D in Equation (3) are all rational numbers, since the Paillier encryption algorithm cannot directly encrypt rational numbers, we can transform them into integers for processing. Thus, we multiply A, B, C, and D by the least common multiple m of their denominator (if A, B, C, D are all integers, $m = 1$), and do the same for x_0, y_0, z_0. (If x_0, y_0, z_0 are integers, the least common multiple of denominator $n = 1$).

4.2 Protocol Design

Protocol 2: Privately compute distance from a point to a plane.

Inputs: Private plane π: $Ax + By + Cz + D = 0$ and point $P_0(x_0, y_0, z_0)$.

Output: The distance d from P_0 to π.

1) Alice generates the public key and private key of the Paillier homomorphic encryption scheme, and tells the public key to Bob.

2) Alice computes the least common multiple m of four rational denominators A, B, C, and D (when A, B, C, D are integers, $m = 1$), then computes $A' = A \cdot m$, $B' = B \cdot m$, $C' = C \cdot m$, $D' = D \cdot m$. Alice encrypts plane π with public key to obtain $E(\pi) = (E(A'), E(B'), E(C'))$, and sends $E(\pi)$ to Bob.

3) Bob finds out the least common multiple n of three rational denominators (when x_0, y_0, z_0 are integers, $n = 1$), and chooses random numbers r_1, r_2 to compute $x_0' = x_0 \cdot r_1 n$, $y_0' = y_0 \cdot r_1 n$, $z_0' = z_0 \cdot r_1 n$, then Bob computes $T = E(A')^{x_0'} \cdot E(B')^{y_0'} \cdot E(C')^{z_0'} \cdot {r_2}^N \bmod N^2$. Bob sends T and $r_1 n$ to Alice.

4) Alice decrypts T with private key and gets $T' = D(T) = A'x_0' + B'y_0' + C'z_0'$. Then Alice computes $d = \frac{|T' + r_1 n D'|}{m r_1 n \cdot \sqrt{A^2+B^2+C^2}}$, and tells Bob the result.

4.3 Correctness

Theorem 3. *Protocol 2 can correctly get the distance from a point to a plane.*

Proof. Since each rational number can be expressed as a fraction, Alice turns the rational number into an integer by multiplying A, B, C, D by the least common multiple m of their denominator. Similarly, Bob multiplies x_0, y_0, z_0 by the least common multiple n of their denominators. According to the homomorphism of the Paillier

encryption algorithm,

$$\begin{aligned} T &= E(A')^{x_0'} \cdot E(B')^{y_0'} \cdot E(C')^{z_0'} \cdot {r_2}^N \bmod N^2 \\ &= E(A'x_0' + B'y_0' + C'z_0') \end{aligned}$$

Alice decrypts the value of $A'x_0' + B'y_0' + C'z_0'$, then computes:

$$\begin{aligned} d &= \frac{|T' + r_1 n D'|}{mr_1 n \cdot \sqrt{A^2 + B^2 + C^2}} \\ &= \frac{|Amr_1 nx_0 + Bmr_1 ny_0 + Cmr_1 nz_0 + Dmr_1 n|}{mr_1 n \cdot \sqrt{A^2 + B^2 + C^2}} \\ &= \frac{|Ax_0 + By_0 + Cz_0 + D|}{\sqrt{A^2 + B^2 + C^2}} \end{aligned}$$

It can be seen from the above formula that m, n do not affect the final result. This completes the proof of the theorem. □

4.4 Security

The security of Protocol 2 is based on the security of the Paillier homomorphic encryption algorithm, which has semantic security. By the simulation paradigm, we can prove that Protocol 2 is secure.

Theorem 4. *Protocol 2 for computing the distance from a point to a plane is secure.*

Proof. The theorem is proved by constructing simulators S_1 and S_2 that make Equations (1) and (2) hold. □

1) S_1 accepts input $(\pi, f_1(\pi, P_0))$, and selects a point $P_1(x_1, y_1, z_1)$ such that $f_1(\pi, P_1) = f_1(\pi, P_0)$.

2) S_1 computes the least common multiple n' of the denominator of rational numbers x_1, y_1, z_1 and chooses a random number r_1' to compute $x_1' = x_1 r_1' n', y_1' = y_1 r_1' n', z_1' = z_1 r_1' n'$. Then S_1 chooses a random number r_2' to compute $T_1 = E(A')^{x_1'} \cdot E(B')^{y_1'} \cdot E(C')^{z_1'} {r_2'}^N \bmod N^2$. S_1 encrypts T_1 to get T_1', and finally computes $d' = \frac{|T_1' + r_1' n' D_1'|}{mr_1' n' \cdot \sqrt{A^2+B^2+C^2}}$.

$$view_1^{\pi}(\pi, P_0) = \{\pi, r_1 n, T, d\}$$

The information sequence generated in the simulation process is: $S_1(\pi, f_1(\pi, P_0)) = \{\pi, r_1' n', T_1', d'\}$,

By definition and the semantic security of the homomorphic encryption scheme, $f_1(\pi, P_1) = f_1(\pi, P_0)$, $T_1 \overset{c}{\equiv} T$, $d' = d$. Therefore,

$$\{S_1(\pi, f_1(\pi, P_0))\} \overset{c}{\equiv} \{view_1^{\pi}(\pi, P_0)\}$$

Similarly, we can construct S_2 such that

$$\{S_2(P_0, f_2(\pi, P_0))\} \overset{c}{\equiv} \{view_2^{\pi}(\pi, P_0)\}$$

This completes the proof of the theorem.

5 Application

Now, we can use the distance protocol to privately compute the volume of tetrahedron and determine the position relation between a line and a plane and between two planes.

5.1 Privately Compute the Volume of Tetrahedron

Suppose that Alice has several points in a plane π: $Ax + By + Cz + D = 0$, and Bob has a point $P_0(x_0, y_0, z_0)$(p_0 is not on the plane π). These points and P_0 constitute a tetrahedron. Alice and Bob want to compute the volume of the tetrahedron without disclosing any information about P_0 and π. The key to solve the problem is to get the height of tetrahedron, that is, the distance from point P_0 to plane π. Finally, the volume of tetrahedron can be calculated according to Equation (4).

Protocol 3: Privately compute the volume of tetrahedron.

Inputs: Private plane π: $Ax + By + Cz + D = 0$, private point $P_0(x_0, y_0, z_0)$.

Output: The volume V of tetrahedron formed by point P_0 and other points in π.

1) Alice and Bob invoke Protocol 1 to compute the distance d from $P_0(x_0, y_0, z_0)$ to π.

2) Alice obtains d and computes the area S of the bottom surface, then computes $V = \frac{1}{3}Sd$. Alice sends the result to Bob.

5.2 Privately Determine the Position Relations between a Straight Line and a Plane

Suppose that Alice has a plane π: $Ax + By + Cz + D = 0$ and Bob has a straight line L. They want to know the position relationship between L and π without disclosing any information about π and L.

5.2.1 Basic Principle

Choosing two different points $P_1(x_1, y_1, z_1), P_2(x_2, y_2, z_2)$ on the line L and comparing distances d_1 and d_2 from these two points to the plane. If $d_1 \neq d_2$, then line L intersects with plane π. If $d_1 = d_2 = 0$, then L is in π. If $d_1 = d_2 \neq 0$, then L is parallel to π. It is also important to note that when a straight line intersects a plane, the specific value of the distance cannot be computed directly, because this will disclose the angle between the line and the plane. Therefore, the relationship of the distance between different points to the plane should be kept secret in the protocol.

For simple exposition, we define

$$P(L,\pi)=\begin{cases}0, & L \text{ is in the } \pi\\ 1, & L \text{ is parallel to } \pi\\ 2, & L \text{ intersects with } \pi\end{cases}$$

5.2.2 Protocol Design

Protocol 4: Privately determine the position relation between a line and a plane.

Inputs: Private plane π: $Ax+By+Cz+D=0$ and straight line L.

Output: $P(L,\pi)$.

1) Alice generates the public key and private key of the Paillier homomorphic encryption scheme, and tells the public key to Bob.

2) Alice encrypts plane π with public key to obtain $E(\pi)=(E(A),E(B),E(C))$, and sends $E(\pi)$ to Bob.

3) Bob chooses two points $P_1(x_1,y_1,z_1), P_2(x_2,y_2,z_2)$ on the line L, then computes $t_1=E(A)^{x_1}\cdot E(B)^{y_1}\cdot E(C)^{z_1}$, $t_2=E(A)^{x_2}\cdot E(B)^{y_2}\cdot E(C)^{z_2}$, and computes $T_1=t_1\cdot t_2^{-1}$. Bob sends T_1 to Alice.

4) Alice decrypts T_1 to obtain $T_1'=D(T_1)=Ax_1+By_1+Cz_1-Ax_2-By_2-Cz_2$. If $T_1'\neq 0$, then $d_1\neq d_2$, L intersects π, Alice outputs $P(L,\pi)=2$. The protocol terminates. Otherwise, they continue to perform the next step.

5) Bob sends t_1 to Alice.

6) Alice decrypts t_1 to obtain $t_1'=Ax_1+By_1+Cz_1$, then computes $d_1=d_2=\frac{t_1'+D}{\sqrt{A^2+B^2+C^2}}$. If $d_1=d_2=0$, L is in the π. Alice outputs $P(L,\pi)=0$. If $d_1=d_2\neq 0$, then L is parallel to π. Alice outputs $P(L,\pi)=1$.

5.3 Privately Determine the Position Relation between Two Planes

Suppose that Alice has a plane π_1: $A_1x+B_1y+C_1z+D_1=0$ and Bob has a plane π_2: $A_2x+B_2y+C_2z+D_2=0$. They want to know the position relation between π_1 and π_2 without disclosing any information about the two planes.

5.3.1 Basic Principle

Since two intersecting lines can determine a plane, we choose two intersecting lines L_1 and L_2 in the plane π_2, and then the problem of determining the position relation between two planes is transformed into the problem of determining the position relation between a line and a plane. Therefore, we can call protocol 4 to solve this problem. For simplicity, select the intersection point P_0 of two intersecting lines as one of the points. In addition, select the other point in two straight lines. If the distances d_0,d_1 from two points P_0,P_1 on the line L_1 to the plane π_1 and the distances d_0,d_2 from points P_0,P_2 on the line L_2 to the plane π_1 satisfy that $d_0=d_1=d_2=0$, then π_1 coincides with π_2. If $d_0=d_1=d_2\neq 0$, then π_1 is parallel to π_2. Otherwise, π_1 intersects with π_2.

For simple exposition, we define:

$$P(\pi_1,\pi_2)=\begin{cases}0, & \pi_1 \text{ and } \pi_2 \text{ coincide}\\ 1, & \pi_1 \text{ is parallel to } \pi_2\\ 2, & \pi_1 \text{ intersects with } \pi_2\end{cases}$$

5.3.2 Protocol Design

Protocol 5: Privately determine the position relation between two planes.

Inputs: Private plane π_1: $A_1x+B_1y+C_1z+D_1=0$ and π_2: $A_2x+B_2y+C_2z+D_2=0$.

Output: $P(\pi_1,\pi_2)$.

1) Bob chooses two intersecting lines L_1 and L_2 in the plane π_2.

2) Alice and Bob invoke Protocol 4 to compare the distances d_0,d_1 from two points P_0,P_1 on the line L_1 to the plane π_1 and the distances d_0,d_2 from two points P_0,P_2 on the line L_2 to the plane π_1.

3) If $d_0=d_1=d_2=0$, then π_1 coincides with π_2, Alice outputs $P(\pi_1,\pi_2)=0$. If $d_0=d_1=d_2\neq 0$, then π_1 is parallel to π_2, Alice outputs $P(\pi_1,\pi_2)=1$. Otherwise, π_1 intersects with π_2, Alice outputs $P(\pi_1,\pi_2)=2$.

6 Performance Analysis

6.1 Efficiency Analysis

Computational complexity analysis. At present, the solutions to privately determine whether two segments intersect need to invoke the complex millionaire protocol, oblivious transfer, inner product protocol, etc. One of the most efficient schemes is the protocol in [18], which uses the Paillier encryption algorithm and needs to be encrypted twice and decrypted 4 times. Encryption or decryption using the Paillier encryption algorithm requires 2 modular exponentiations at a time. In addition, the millionaire protocol based on the Paillier homomorphic encryption scheme [1] is invoked 3 times, and each invocation requires $4n$ modular exponentiations (n is the bit length of the input data). Ignoring multiplication and addition operations, the total computational overheads are $4n+12$ modular exponentiations. In this paper, Protocol 1 needs to be encrypted 12 times and decrypted twice, so the total computational overheads are 28 modular exponentiations. From the analysis above, we can see that with the growth of data, the efficiency of our scheme has obvious advantages. (The modular exponentiation is M_e.)

For the research of privacy-preserving computation of the distance from a point to a plane, the scheme in [13] needs to invoke the inner product protocol based on oblivious transfer. Assuming that the security parameter is m. To execute an inner product protocol needs to invoke the 1-out-of-k oblivious transfer m times, i.e. $\lg k$ 1-out-of-k oblivious transfer, i.e. $2m \lg k$ modular exponentiations. According to the practical significance, only when $m > 5$ and $k > 8$ can the scheme achieve the basic security level. Thus, the scheme requires at least 30 modular exponentiations. In this paper, we design Protocol 2 for computing the distance from a point to a plane. Protocol 2 needs to be encrypted 3 times and decrypted once, so the total computational overheads are 8 modular exponentiations.

Communication complexity analysis. The measure of communication complexity is the number of bits of information exchanged in the protocol or the number of communication rounds. In the study of SMC, the number of communication rounds is usually used. The scheme in [18] requires $4 + 3c$ rounds of communication where c represents the number of rounds of millionaire protocol. The communication complexity of Protocol 1 in this paper is 2 rounds. [13] calls m times inner product protocol based on oblivious transfer. The communication complexity is m rounds. The communication complexity of Protocol 2 is 2 rounds. The computational and communication complexity are shown in Table 1.

In addition, we give an analysis of the efficiency and security of Protocol 4 in the application part and compare it with the related protocols in [3, 11, 13, 21]. [13] and [3] invoke the inner product protocol, and [13] and [21] invoke the data ratio protocol. In the whole process of execution, the most expensive computation cost is modular exponentiation. [11] mainly uses multiplication. The total number of inner product protocol called in each scheme, the number of modular exponentiations required by the user, and the number of multiplication operations are taken as indicators to measure the complexity of computation, and the others are ignored. The modular exponentiation is M_e and multiplication is M.

Privately determining the position relations between a line and a plane in [13] (Protocol 6): The protocol invokes the inner product protocol twice and the data ratio protocol twice, and the inner product protocol uses the oblivious transfer method in [15]. Assuming the security parameter is m, according to the analysis in the original paper, it needs at least $30m$ modular exponentiations, so the computational cost of the protocol is $30mM_e$. [11] (Protocol 3): This protocol mainly performs matrix operations, and the total number of multiplication operations is $36M$. However, the protocol will disclose the ratio between the distance from different points on the line to the plane. [21] (Protocol 4): This scheme mainly uses the Paillier homomorphic encryption algorithm, and calls data ratios protocol twice. The total computational overheads are $35M_e$. [3] (Protocol 5): The scheme invokes inner product protocol twice and outsources them to cloud computing, and invokes data ratio protocol once. The total modular exponentiations are $15M_e$. However, when the line and plane are intersected, this scheme will disclose the angle between the line and the plane. In this paper (Protocol 4), the total number of modular exponentiations is $16M_e$ and the total number of multiplication operations is $4M$.

As for the problem of privately determining position relation between a line and a plane, the comparisons are shown in Table 2.

6.2 Experimental Test and Analysis

We verify the computational complexity by simulating the time taken to perform Protocol 2, and compare it with the existing scheme in [13]. In addition, we test the time used in Protocol 4 to determine the position relation between a line and a plane. [3] shifts this decision problem to cloud computing platforms and outsources the complex computation to the cloud, which results in additional high economic costs. What's more, in the case of traditional participant interaction, the computational complexity of this scheme is still very high, and it is found that both the schemes in [11] and [3] have information leakage. Therefore, we choose [21] which is the most efficient solution and without information leakage to compare with Protocol 4 in the mode of participant interaction.

Our test environment: Windows 10 64 bit operating system. The processor is Intel (R) Core (TM) i5-6600 CPU @3.30 HZ, and memory is 8GB. We program in JAVA language.

Experimental method. We randomly selected 20 sets of data, conducted 2000 simulation experiments on each set value, and calculated the average of experimental results. Figure 1 depicts the comparison of the execution time between Protocol 2 and the scheme in [13]. Figure 2 describes the comparison of the implementation time between Protocol 4 and the scheme in [21].

The experimental results show that the average execution time of Protocol 2 is between 15 and 25 milliseconds, which is much more efficient than the method in [13]. At the same time, Protocol 4 can guarantee the security with high efficiency. To sum up, the computation cost and computational complexity of our protocols are relatively low.

7 Conclusion

PPCG has always been an important issue in cryptography and SMC. Based on the Paillier homomorphic encryption scheme, we first proposed a secure Protocol 1 to determine whether two segments intersect. By using the

Table 1: Comparison of computational and communication complexity between our protocols and existing schemes

	[18]	Protocol 1	[13]	Protocol 2
Computational Complexity	$(4n+12)M_e$	$28M_e$	$2m \lg k > 30M_e$	$8M_e$
Communication Complexity	2 rounds	$4+3c$ rounds	m rounds	2 rounds

Table 2: Efficiency and performance comparison between Protocol 4 and existing protocols

	Computation overhead	Inner product protocol	Data ratio protocol	Information leakage?
Protocol 6 in [13]	$30mM_e$	twice	twice	No
Protocol 3 in [11]	$36M$	-	-	Yes
Protocol 4 in [21]	$35M_e$	-	twice	No
Protocol 5 in [3]	$15M_e$	twice (cloud computing)	once	Yes
Protocol 4	$16M_e+4M$	-	-	No

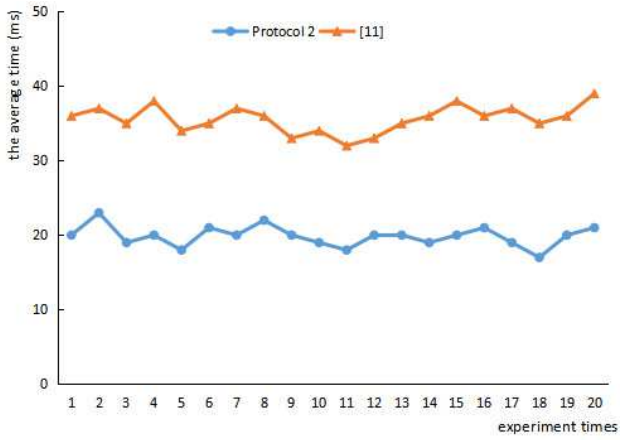

Figure 1: Comparison of execution time between Protocol 2 and [13]

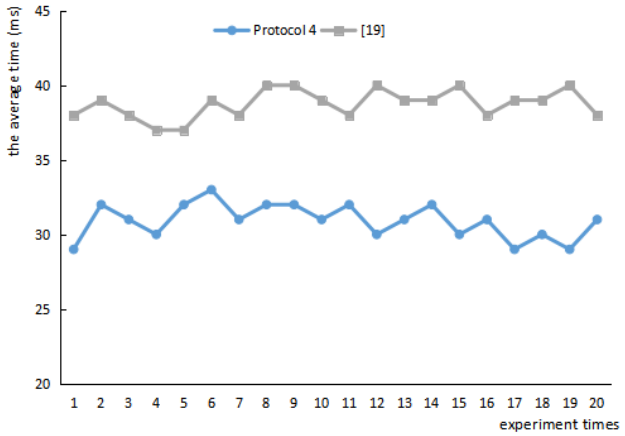

Figure 2: Comparison of execution time between Protocol 4 and [21]

principle of Protocol 1, we can determine whether polygons intersect. Then, we propose a secure Protocol 2 to compute the distance from a point to a plane. Protocol 2 is not only suitable for integers, but also for rational numbers. The correctness of Protocol 1 and Protocol 2 are analyzed and proved, and the security of the two protocols is proved by simulation paradigm. Next, by using the distance from a point to a plane, we solve the problem of privately computing volume of a tetrahedron, the problem of privately determining position relation between a line and a plane, and the problem of privately determining position relation between two planes in space. Finally, we prove that Protocol 4 is not only efficient but also secure by comparing with the existing protocols for privately determining the position relation between a line and a plane in space. The problems studied in this paper have important practical significance for research and application of SMC, and our solutions for these problems are secure in the semi-honest model. In the future study, we will focus on the SMC of various computational geometry problems in the malicious models.

Acknowledgments

The authors would like to thank the anonymous reviewers for detailed and valuable comments. This work is supported by the National Natural Science Foundation of China (Grant no. 61272435).

References

[1] I. F. Blake, V. Kolesnikov, "Strong conditional oblivious transfer and computing on intervals," in *Advances in Cryptology-Asiacrypt, International Conference on the Theory and Application of Cryptology and Information Security*, pp. 515–529, 2004.

[2] R. Campos and A. Jatowt, "Survey of temporal information retrieval and related applications," *Acm Computing Surveys*, vol. 47, no. 2, pp. 1–41, 2014.

[3] Z. H. Chen, S. D. Li, Q. Huang, Y. Ding and M. Sun, "Privacy-preserving determination of spatial location-relation in cloud computing," *Chinese Journal of Computers*, vol. 39, no. 137, pp. 351–363, 2017.

[4] W. Du and M. J. Atallah, "Secure multi-party computation problems and their applications: A review and open problems," *Proceedings of New Security Paradigms workshop New York: ACM Press*, pp. 13–22, 2001. ISBN:1-58113-457-6.

[5] R. Frederick, "Core concept: Homomorphic encryption," *Proceedings of the National Academy of Science*, vol. 112, no. 28, pp. 8515–8516, 2015.

[6] O. Goldreich, "Foundations of cryptography: Casic applications," *Journal of the Acm*, vol. 10, no. 509, pp. 359–364, 2004.

[7] F. Kerschbaum, "Privacy-preserving computation," *Springer Berlin Heidelberg*, vol. 8319, pp. 41–54, 2012.

[8] C. T. Li, M. S. Hwang, Y. P. Chu, "A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks", *Computer Communications*, vol. 31, no. 12, pp. 2803-2814, July 2008.

[9] C. T. Li, M. S. Hwang, Y. P. Chu, "Further improvement on a novel privacy preserving authentication and access control scheme for pervasive computing environments", *Computer Communications*, vol. 31, no. 18, pp. 4255–4258, Dec. 2008.

[10] S. D. Li, X. L. Yang, X. J. Zuo, *et al.*, "Privacy protecting similitude determination for Graphics Similarity," *Chinese Journal of Electronics*, vol. 45, no. 9, pp. 2184–2189, 2017.

[11] S. D. Li, C. Y. Wu, D. S. Wang, and Y. Q. Dai, "Secure multiparty computation of solid geometric problems and their applications," *Information Sciences An International Journal*, vol. 282, pp. 401–413, 2014.

[12] L. Liu, X. Chen, and W. Lou, "Secure three-party computational protocols for triangle area," *International Journal of Information Security*, vol. 15, no. 1, pp. 1–13, 2016.

[13] Y. L. Luo, L. S. Huang, W. W. Jing, and W. J. Xu, "Privacy protection in the relative position determination for two spatial geometric," *Journal of Computer Research and Development*, vol. 43, no. 3, pp. 410–416, 2006.

[14] Y. L. Luo, L. S. Huang, W. W. Jing, W. J. Xu, and G. L. Chen, "Privacy-preserving cross product protocol and its applications," *Chinese Journal of Computers*, vol. 30, no. 2, pp. 248–254, 2007.

[15] M. Naor, B. Pinkas, "Oblivious transfer and polynomial evaluation," *ACM Symposium on Theory of Computing*, pp. 245–254, 1999. ISBN:1-58113-067-8.

[16] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, 1999.

[17] B. Reimer, R. Fried, B. Mehler, *et al.*, "Brief report: Examining driving behavior in young adults with high functioning autism spectrum disorders: A pilot study using a driving simulation paradigm," *Journal of Autism and Developmental Disorders*, vol. 43, no. 9, pp. 2211–2217, 2013.

[18] M. H. Sun, S. S. Luo, Y. Xin, and Y. X. Yang, "Secure two-party line segments intersection scheme and its application in privacy-preserving convex hull intersection," *Journal on Communications*, vol. 34, no. 1, pp. 30–42, 2013.

[19] M. Vishakha and R. Sharma, "Review paper on cryptography," *International Journal of Research*, vol. 2, no. 5, pp. 141–142, 2015.

[20] J. Yang, J. S. Zhao and J. P. Zhang, "A privacy preservation method for high dimensional data mining," *Acta Electronica Sinica*, vol. 41, no. 11, pp. 2187–2192, 2008.

[21] X. L. Yang, S. D. Li and X. J. Zuo, "Secure multiparty geometry computation," *Journal of Cryptologic Research*, vol. 3, no. 1, pp. 33–41, 2016.

[22] A. C. Yao, "Protocols for secure computations," in *IEEE Computer Science*, pp. 160–164, 1982.

[23] J. Zhang, S. S. Luo, Y. X. Yang, and Y. Xin, "Research on the privacy-preserving point-in-polygon protocol," *Journal on communications*, vol. 37, no. 4, pp. 87–95, 2016.

[24] X. J. Zuo, X. L. Yang, and S. D. Li, "Privately determining protocol on three points are collinear and its applications," *Journal of Cryptologic Research*, vol. 3, no. 3, pp. 238–248, 2016.

Biography

Qiong Wei was born in 1994. She is currently pursuing the M.S. degree with School of Computer Science in Shaanxi Normal University. Her research interests focus on secure multi-party computation and information security.

Shundong Li was born in 1963. He received the Ph.D. degree in Department of Computer Science and Technology from Xi'an Jiaotong University in 2003. He is now a Professor with School of Computer Science in Shaanxi Normal University. His research interests focus on modern cryptography and secure multi-party computation.

Wenli Wang was born in 1991. She is currently pursuing the M.S. degree with School of Mathematics and Information Science in Shaanxi Normal University. Her research interests focus on applied mathematics and cryptography.

Yanjing Yang was born in 1995. She is currently pursuing the M.S. degree with School of Computer Science in Shaanxi Normal University. Her research interests focus on secure multi-party computation and information security.