Advances in Computer Science Research, volume 88
International Conference on Computer, Network, Communication and Information Systems (CNCI 2019)

An Efficient Protocol for Privately Determining the Relationship between Two Rectangles

Wenli Wang[1, a], Xuhong Liu[1, b]

[1]School of Mathematics and information Science, Shaanxi Normal University, Xi'an 710119, China.

[a]wenliwang@snnu.edu.cn, [b]xuhong@snnu.edu.cn

Abstract. Secure multiparty computation *(smc*) is a research focus in the international cryptographic community, while secure multiparty geometry computation is an important field of *smc*. In this paper, we study an important secure computational geometric problem, that is, to privately determine the location relation between two rectangles. The existing protocols for this problem are of low efficiency. To efficiently solve this problem, we use sorting technique to design a protocol for privately determining the location relation between two rectangles. Finally, we analyze the theoretical efficiency to show that our protocol is efficient.

Keywords: Secure multi-party computation, computational geometry, the relation between two rectangles.

1. Introduction

In recent years, information leakage events occurred frequently, so it is imperative to protect information privacy in a distrust network environment. The privacy requirements make the secure multiparty computation (*smc*) became a focus of the international cryptographic community.

Smc is a technology by which two or more participants can cooperatively perform some computations on their private data without disclosing the privacy of their private data. It was initially introduced by Yao [1] for two party cases and were thoroughly studied by Goldreich and Cramer et al.[2,3] extended *smc* to multiple party cases. Privacy-preserving computational geometry is an important branch of *smc*. Many geometric problems have been studied, such as the relation between line and hyperbola [4], the relation between line and planes [5], the relation between two planes [6], the relation between two straight lines [7], intersect between two convex polygons [8].

For the problem of privately determining the location relation between two rectangles, we only consider the cases in which the edges of rectangles are parallel to x axis or y axis. In order to describe clearly, we use A_1, A_2 to denote two rectangles. Luo [9] solved the polygons intersection problem based on protocol of two line segments intersection. Unfortunately, this protocol cannot solve the problems in the case that the rectangle $A_1(A_2)$ is totally inside $A_2(A_1)$. Then, to reduce the computation overhead and solve the problem in case that A_1 is totally in A_2 or vice versa, Li [10] studied and proposed a new protocol. In their approach, each party used the Cantor encoding to transform a rectangle into a set and then identify whether the intersection cardinality of two sets is zero based on the commutative encryption scheme. If the area of the rectangle is too large, then the set elements will increase, which may result in higher computational complexity.

In order to overcome the disadvantages of the schemes mentioned above, we reduce the problem of determining the relation between two rectangles to determining the relationship between two intervals. Our contributions are as follows.

(1) We introduce a new coding method and use the ElGamal threshold encryption scheme to compute the order of the vertex.

(2) We use the method of sorting for vertex and propose a protocol for determining the location relation between two rectangles. The protocol proposed is efficient and can also determine the location relation in case $A_1(A_2)$ is totally inside $A_2(A_1)$.

2. Preliminaries

2.1 Security Model.

Privacy by simulation paradigm: Suppose that $f=(f_1,f_2)$ is a probabilistic polynomial-time function and π is a protocol that computes f. Alice who holds x, and Bob who holds y, want to cooperatively compute $f(x,y)=(f_1(x,y),f_2(x,y))$ without disclosing the data x,y.

In the process of π, Alice, Bob get messages sequence $view_1^{\pi}(x,y)=(x,r^1,m_1^1,\cdots,m_t^1)$, $view_2^{\pi}(x,y)=(y,r^2,m_1^2,\cdots,m_t^2)$, respectively. Where $r^1(r^2)$ is the result of Alice's (Bob's) internal coin tosses and $m_i^1(m_i^2)$ is the i-th message that Alice (Bob) received.

Definition 1(privacy w.r.t semi-honest behavior): For a deterministic function f,π privately computes f if there exist probabilistic polynomial-time algorithms, denoted as S_1,S_2 such that

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

$$\{S_2(y,f_2(x,y))\}_{x,y}\overset{c}{\equiv}\{view_2^{\pi}(x,y)\}_{x,y}. \tag{2}$$

Where $\overset{c}{\equiv}$ denotes computational indistinguishable, $view_1^{\pi}(x,y)$ and $view_2^{\pi}(x,y)$ are related random variables that are defined as functions of the same random execution.

In order to prove that a two-party computation protocol is secure, we must construct simulators S_1 and S_2 such that (1) and (2) holds.

2.2 The Threshold Encryption Scheme.

The threshold encryption scheme is an important tool in secure multiparty computation to against collusion attacks. Let n be the number of parties, if at least t party can decrypt ciphertext effectively, and less than t party cannot get useful information, such a system is called a (t,n)-threshold cryptosystem. In this paper, we use (2,2)-threshold trivial cryptosystem. In the following, we construct the threshold encryption scheme by using ElGamal encryption system:

KeyGen: On input a security parameter k, the *KeyGen* algorithm generates a large prime p and a generator g. Each participant $P_i(1\le i\le n)$ randomly chooses a number k_i as a private key share and computes $h_i=g^{k_i} \bmod p$. The public key is

$$h\equiv\prod_{i=1}^{n}g^{k_i} \bmod p\equiv g^{\sum_{i=1}^{n}k_i} \bmod p.$$

Encrypt: Taking M and h as inputs and selecting a random number r, computes

$$E(M)=(c_1,c_2)=(g^r \bmod p, Mh^r \bmod p).$$

Decrypt: This algorithm takes $E(M)=(c_1,c_2)$ as input and computes

$$M=\frac{c_2}{\prod_{i=1}^{n}c_1^{k_i}} \bmod p.$$

3. Determining the Location Relation between Two Rectangles

The problem of privately determining the location relation between two rectangles can be described as follows. Alice has $A_1=I_1\times J_1(I_1=[x_1,x_2],J_1=[y_1,y_2])$ and Bob has $A_2=I_2\times J_2(I_2=[x_3,x_4],J_2=[y_3,y_4])(x_i,y_j\in U=\{u_1,u_2,\cdots,u_n\})$. They want to know the location relation between their rectangles A_1 and A_2 without disclosing the information of rectangles. For simplicity, we only consider that the edges of rectangle are of non-coincidence.

According to the character of rectangle, it is not hard to obtain the following facts:

(1) If I_1 intersects I_2 and J_1 intersects J_2, then, A_1 intersects A_2.

(2) If neither I_1 intersects I_2, nor J_1 intersects J_2, then, A_1 separates A_2.

If we get the orders $(x_i)_{ord},(y_j)_{ord}$ of x_i, y_j, we can further easily determine the location relation from the order value. If I_1 intersects I_2, $((x_1)_{ord},(x_2)_{ord})$, $((x_3)_{ord},(x_4)_{ord})$ at least one group is non-adjacent. For example, $(x_1)_{ord}=1,(x_2)_{ord}=3,(x_3)_{ord}=2,(x_4)_{ord}=4$, then $\sum_{i=2,4}(x_i)_{ord}-\sum_{i=1,3}(x_i)_{ord}=4$. If I_1 separates I_2, both of them are adjacent, for example, $(x_1)_{ord}=1,(x_2)_{ord}=2$, $(x_3)_{ord}=3,(x_4)_{ord}=4$, then $\sum_{i=2,4}(x_i)_{ord}-\sum_{i=1,3}(x_i)_{ord}=2$. Similarly, we will get $\sum_{j=2,4}(y_j)_{ord}-\sum_{j=1,3}(y_j)_{ord}=4$, if J_1 intersects J_2; otherwise, $\sum_{j=2,4}(y_j)_{ord}-\sum_{j=1,3}(y_j)_{ord}=2$. Therefore, we can determine the location relation between two rectangles according to the value of $\sum_{i,j=2,4}((x_i)_{ord}+(y_j)_{ord})-\sum_{i,j=1,3}((x_i)_{ord}+(y_j)_{ord})$.

Before compute the order of x_i, y_j, we first introduce a building block: $1\text{-}r$ Encoding, which is used to encode a set to a $1\text{-}r$ vector. The principle for encoding a set $\{x_1,x_2\}$ to a $1\text{-}r$ vector $\hat{X}_1=(\hat{x}_{11},\hat{x}_{12},\cdots,\hat{x}_{1n})$ is as follows: if $u_a\in X(a\in[1,n])$, then $\hat{x}_a=r$; otherwise, $\hat{x}_a=1$. Similarly, Bob encode his set $\{x_3,x_4\}$ to $\hat{X}_2=(\hat{x}_{21},\hat{x}_{22},\cdots,\hat{x}_{2n})$. Similarly, they can encode $\{y_1,y_2\}$ $\{y_3,y_4\}$ as $\hat{Y}_1=(\hat{y}_{11},\hat{y}_{12},\cdots,\hat{y}_{1n}),\hat{Y}_2=(\hat{y}_{21},\hat{y}_{22},\cdots,\hat{y}_{2n})$.

Then, they represent the vector $\hat{X}_1,\hat{X}_2$ and $\hat{Y}_1,\hat{Y}_2$ by the following matrix X and Y. Where

$$X=\begin{pmatrix}\hat{X}_1\\ \hat{X}_2\end{pmatrix}=\begin{pmatrix}\hat{x}_{11} & \cdots & \hat{x}_{1n}\\ \hat{x}_{2n} & \cdots & \hat{x}_{2n}\end{pmatrix},\ Y=\begin{pmatrix}\hat{Y}_1\\ \hat{Y}_2\end{pmatrix}=\begin{pmatrix}\hat{y}_{11} & \cdots & \hat{y}_{1n}\\ \hat{y}_{2n} & \cdots & \hat{y}_{2n}\end{pmatrix}.$$

Here, we assume that $x_i=u_{a_i}(a_i\le n)$ is true. To compute the order of x_i, they computes the product h_i of the elements in the former a_i columns of the matrix X. Due to the particularity of elements, h_i can be represented as $h_i=r^{d_{x_i}}$, that is the exponentiation d_{x_i} is the order of x_i. Similarly, they can compute the product g_i of the elements of the matrix Y to obtain the order d_{y_i} of y_i.

Theorem 1. According to the above calculation, Alice and Bob can correctly compute the order of x_i, y_j.

Proof. For $x_i=u_{a_i}(a_i\le n)$, in order to compute its order, they compute the product of the elements in the former a_i columns of the matrix X. According to the above encoding, only the position of x_i and data smaller than x_i in the former a_i columns are encoded as r, and the rest are encoded as 1. Therefore, the exponentiation d_{x_i} indicates how many elements are smaller than x_i, that is, the order of x_i. Similarly, they also can correctly compute the order d_{y_i} of y_i.

In order not to disclose additional information, the value of $(x_i)_{ord},(y_j)_{ord}(1\le i,j\le 4)$ should be private. In other words, we just compute the final result $\sum_{i,j=2,4}((x_i)_{ord}+(y_j)_{ord})-\sum_{i,j=1,3}((x_i)_{ord}+(y_j)_{ord})$ to privately compute the relationship between two rectangles. In order to describe clearly, we define

$$P(A_1,A_2)=\begin{cases}1, & \text{if rectangle } A_1 \text{ intersect } A_2;\\ 0, & \text{if rectangle } A_1 \text{ separate } A_2.\end{cases}$$

Protocol 1: A protocol for determining the location relation between two rectangles

Input: Alice inputs rectangle $A_1=I_1\times J_1$, Bob inputs rectangle $A_2=I_2\times J_2$.

Output: $P(A_1,A_2)$.

1. Alice and Bob select the public parameter g,p for the ElGamal threshold encryption schemes. Then, they randomly choose number k_1,k_2 as a private key share and compute $h_i=g^{k_i}\bmod p(i=1,2)$. The public key is

$$h\equiv h_1h_2\equiv g^{k_1}g^{k_2}\bmod p\equiv g^{k_1+k_2}\bmod p.$$

2. Alice encodes her set $\{x_1,x_2\},\{y_1,y_2\}$ as $\hat{X}_1=(\hat{x}_{11},\hat{x}_{12},\cdots,\hat{x}_{1n})$ and $\hat{Y}_1=(\hat{y}_{11},\hat{y}_{12},\cdots,\hat{y}_{1n})$, then encrypts $\hat{X}_1,\hat{Y}_1$ as $E(\hat{X}_1)=(E(\hat{x}_{11})\ \cdots\ E(\hat{x}_{1n})),E(\hat{Y}_1)=(E(\hat{y}_{11}),\cdots,E(\hat{y}_{1n}))$.

3. Similarly, Bob encodes $\{x_3, x_4\}, \{y_3, y_4\}$ as $\hat{X}_2 = (\hat{x}_{21}, \cdots, \hat{x}_{2n})$ $\hat{Y}_2 = (\hat{y}_{21}, \hat{y}_{22}, \cdots, \hat{y}_{2n})$, and then encrypts $\hat{X}_2, \hat{Y}_2$ as $E(\hat{X}_2) = (E(\hat{x}_{21}) \ \cdots \ E(\hat{x}_{2n})), E(\hat{Y}_2) = (E(\hat{y}_{21}) \ \cdots \ E(\hat{y}_{2n}))$.

4. Alice and Bob publishes $E(\hat{X}_1), E(\hat{Y}_1)$ and $E(\hat{X}_2), E(\hat{Y}_2)$, respectively. So, all parties get the matrix $E(X), E(Y)$.

5. According to her data, Alice executes the following:

(a) According to the value of x_1, x_2, y_1, y_2, Alice computes the product $H_1 = E(h_1), H_2 = E(h_2)$ and $G_1 = E(g_1), G_2 = E(g_2)$ of the elements in the matrix $E(X)$ and $E(Y)$. Then, further computes $\alpha_{11} = H_2 H_1^{-1} G_2 G_1^{-1}$, where $H_1^{-1}(G_1^{-1})$ is the multiplicative inverse of $H_1(G_1)$.

(b) Chooses a random number r_1 to compute $\alpha_{12} = (g^{r_1} \bmod p, h^{r_1} \bmod p)$.

(c) Sends $\alpha_1 = \alpha_{11}\alpha_{12}$ to Bob.

6. Similarly, Bob executes the following:

(a) According to the value of x_1, x_2, y_1, y_2, Bob computes the product $H_3 = E(h_3), H_3 = E(h_3)$ and $G_3 = E(g_3), G_3 = E(g_3)$ of the elements in the matrix $E(X)$ and $E(Y)$. Then, further computes $\alpha_{21} = H_4 H_3^{-1} G_4 G_3^{-1}$, where $H_3^{-1}(G_3^{-1})$ is the multiplicative inverse of $H_3(G_3)$.

(b) Chooses a random number r_2 to compute $\alpha_{22} = (g^{r_2} \bmod p, h^{r_2} \bmod p)$.

(c) Computes $c = \alpha_1\alpha_2 = \alpha_{21}\alpha_{22} = (c_1, c_2)$ and then sends $(c_1^{k_2}, (c_1, c_2))$ to Alice.

7. Alice computes $c_1^{k_1}$ and $\hat{c} = \frac{c_2}{c_1^{k_1} c_1^{k_2}} = r^d$. If $d = 8$, outputs $P(A_1, A_2) = 1$; otherwise, outputs $P(A_1, A_2) = 0$.

3.1 Correctness and Security.

According to the calculation principle, we can correctly determine the relation from the value $\sum_{i,j=2,4} ((x_i)_{ord} + (y_j)_{ord}) - \sum_{i,j=1,3} ((x_i)_{ord} + (y_j)_{ord})$. In addition, we can correctly calculate the order of x_i, y_i by calculating H_i, G_i according to Theorem 1. Hence, Protocol 1 can correctly determine the location relation between two rectangles.

Theorem 2: Protocol 1 can securely determine the location relation between two rectangles.

Proof. The security of Protocol 1 is based on the security of threshold decryption algorithm. Each party only publishes the encrypted information $E(\hat{X}_1), E(\hat{Y}_1)$ and $E(\hat{X}_2), E(\hat{Y}_2)$ in the calculation process. According to the probabilistic threshold encryption algorithm is semantically secure, without the cooperation of all participants, any message encrypted with the public key is computationally indistinguishable. Therefore, the *view* obtained when the protocol is actually executed is computationally indistinguishable with the sequence of information obtained by simulate with inputs $\{x_1', x_2'\}, \{y_1', y_2'\}$ or $\{x_3', x_4'\}, \{y_3', y_4'\}$ that keep $P(A_1, A_2)$ unchanged. Thus,

$$\{S_1(A_1, f_1(A_1, A_2))\}_{A_1, A_2} \overset{c}{\equiv} \{view_1^{\pi}(A_1, A_2)\}_{A_1, A_2};$$

$$\{S_2(A_2, f_2(A_1, A_2))\}_{A_1, A_2} \overset{c}{\equiv} \{view_2^{\pi}(A_1, A_2)\}_{A_1, A_2}.$$

4. Efficiency Analysis

Computational complexity: in this section, we only compare our protocols with Li's scheme. For computational complexity, we only consider the most expensive modular exponentiation.

Li's scheme transforms the rectangle $A_1(A_2)$ into a set $T(A_1)(T(A_2))$ by Cantor encoding. The execution of Li's protocol needs to invoke the protocol for millionaire. Therefore, the modular exponentiation was increasing and the protocol totally needs at least 1000 modular exponentiations. Our Protocol 1, both Alice and Bob needs $2n+1$ ($|U| = n(n > 8)$) encryptions and 1 decryptions. For ElGamal encryption schemes, each encryption takes 2 modular multiplications and each decryption takes 1 modular multiplication. When the number of elements n of U is 30, our protocols only need 246 modular multiplications.

Communication Complexity: Communication complexity, i.e. communication rounds, is an important factor for evaluating secure multiparty computation solutions. Li's scheme needs 6 rounds, while our protocol requires 2 rounds. Table 1 summarizes the comparison.

Table 1 Comparison of the Computational and Communication Complexity

Numble	Li's scheme	Protocol 1
Computational Complexity	1000	246
Communication Complexity	6	2

Through the above analysis, the theoretical efficiency indicated that our protocol 1 is efficient.

5. Conclusion

In this paper, we use sorting technique to design an efficient protocol for privately determining the location relation between two rectangles. In future research, we will study the problem of the relationship between two rectangles in the malicious model and other secure multiparty geometry problems.

References

[1]. Yao, A.C. Protocols for secure computations. The 23rd IEEE Symposium on Foundations of Computer Science. Chicago, 1982, p. 160-164.

[2]. Cramer R, Damgard IB, Nielsen JB. Secure Multiparty Computation and Secret Sharing. Cambridge University Press, 2015, p. 1-373.

[3]. Goldreich O. The Fundamental of Cryptography: Basic Applications. Cambridge University Press, 2004, p. 1-342.

[4]. He F, Wang T. Research and Application of Secure Multi-Party Computation in Several Computational Geometry Problems. 2013 International Conference on Industrial Control and Electronics Engineering. China, 2012, p. 1434-1437.

[5]. Li SD, Wu CY, Wang DS, et al. Secure multiparty computation of solid geometric problems and their applications. Information Sciences. Vol. 282 (2014) No. 10, p. 401-413.

[6]. Chen ZH, Li SD, Huang Q, et al. Privacy-Preserving Determination of Spatial Location -Relation in Cloud Computing. Chinese journal of computers. Vol. 40 (2017) No. 2, p. 351-363.

[7]. Yang XL, Li SD, Zuo XJ. Secure multi-party geometry computation. Journal of Cryptologic Research. Vol. 3 (2016) No. 1, p. 33-41.

[8]. Li SD, Wang DS, Dai YQ. Efficient Secure Multiparty Computational Geometry. Chinese Journal of Electronics. Vol. 19 (2010) No. 2, p. 324-328.

[9]. Luo YL, Huang LS, Xu WJ, et al. A Protocol for Privacy-Preserving Intersect-Determination of Two Polygons. ACTA ELECTRONICA SNICA. Vol. 35 (2007) No. 4, p. 685-691.

[10]. Li SD, Dai YQ, Wang DS, et al. A secure multi-party computation solution to intersection problems of sets and rectangles. Natural Science: Materials International. Vol. 16 (2006) No. 5, p. 538-545.