Journal of Information & Computational Science 9: 4 (2012) 915–929
Available at http://www.joics.com

Secure Two-party Protocols on Planar Convex Hulls ⋆

Bo Yang [a,b,*], Zhiyi Shao [a], Wenzheng Zhang [c]

[a] *School of Computer Science, Shaanxi Normal University, Xi'an 710062, China*
[b] *College of Informatics, South China Agricultural University, Guangzhou 510642, China*
[c] *National Laboratory for Modern Communications, Chengdu 610041, China*

Abstract

Convex hulls are fundamental problems in computational geometry. Secure multi-party geometric computation is a type of specific secure multi-party computation problems, and has found various applications in many areas as military, computer graphics, etc. In this paper we give some more efficient secure protocols on planar convex hulls, including the point inclusion problem, the intersection and the union of two convex hulls. In addition, we give some basic protocols in secure computational geometry, including angle of elevation, intersection of two line segments, and secure binary search protocol. We envisage these basic protocols will be useful in further study to secure computational geometry.

Keywords: Secure Multi-party Computation; Computational Geometry; Convex Hull; Protocol

1 Introduction

Secure Multi-party Computation (SMC) problems deal with the following situation: Two (or many) parties want to jointly perform a computation. Each needs to contribute its private input to this computation, but neither should disclose its private inputs to the others, or any third party. According to the theoretical SMC studies, all of the SMC problems can be solved in theory by using the circuit evaluation protocol [1]. But using this general solution for special cases of multi-party computation can be impractical; special solutions must be developed for special cases for efficiency reasons. Motivated by this observation, researchers have started to look for special solutions for each specific SMC problem. Secure multi-party geometric computation is a type of specific secure multi-party computation problems, and has found various applications in many areas as military, computer graphics, etc., its study was initiated by Atallah et al. [2] with their work on secure point inclusion problem and polygonal intersection problem. Their protocol for the point inclusion problem is applicable to simple polygonal domain and has round complexity $O(n)$ where n is the number of edges of the polygon. Later, in [3], the point inclusion problem for

⋆This work is supported by the National Natural Science Foundation of China under Grant 60973134, 61173164, 61003232, and the Natural Science Foundation of Guangdong Province under Grant 10351806001000000.
*Corresponding author.
Email address: byang@snnu.edu.cn (Bo Yang).

April 2012

circular domain was studied. However, their solution is not secure in the sense that each party obtains additional information regarding the location of the other party's object. Moreover, their solution is highly inefficient. In [4], a more efficient protocol for the point inclusion problem in a circular domain was proposed, but the protocol is applicable only to integer points in a plane. In [5], the point inclusion problem in a star-shaped domain and a more general polygonal domain were considered. Two protocols for the star shaped domain have round complexities $O(n)$ and $O(logn)$ respectively, and a protocol for more general polygonal domain has round complexity $O(n)$, where n is the number of vertices. However, in each round, two secure scalar product protocols and two millionaire protocols are needed in the first protocol, two secure scalar product protocols are needed in the second protocol and the third protocol, the complexities of three protocols are $O(2n)$ times the addition of the one of a secure scalar product protocol and the one of the millionaire protocol, $O(2logn)$ times the one of a secure scalar product protocol, $O(2n)$ times the one of a secure scalar product protocol, respectively.

In this paper we give some more efficient secure protocols on convex hulls, including the point inclusion problem, the intersection and the union of two convex hulls.

In the following, we will use some fundamental concepts and existing protocols, including the semi-honest model, millionaire protocol, secure scalar product protocol, homomorphic encryption schemes, commutative encryption schemes, the relative location of a point to a directed line segment, secure location of a point to a directed line segment protocol, secure computation of two points distance. For these fundamental concepts and protocols, the reader is referred to [6].

2 Some Fundamental Protocols in Computational Geometry

In this section, we will further give two basic protocols in secure computational geometry, secure computation of angle of elevation and secure intersection of two line segments protocol.

2.1 Secure Computation of Angle of Elevation

We assume that Alice has a private point $P_0(x_0, y_0)$ and Bob has a private line segment with two endpoints $P_1(x_1, y_1), P_2(x_2, y_2)$. The angle of elevation that $P_0(x_0, y_0)$ makes with $\overrightarrow{P_1P_2}$ as the reference edge is which between the vectors $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_1P_0}$, denoted as the scalar product $\theta = \arccos \frac{\overrightarrow{P_1P_2} \cdot \overrightarrow{P_1P_0}}{|\overrightarrow{P_1P_2}| \cdot |\overrightarrow{P_1P_0}|}$.

The protocol is as follows

Protocol 1 (Secure angle of elevation protocol)

Inputs: Alice has a private point $P_0(x_0, y_0)$ and Bob has a private line segment with two endpoints $P_1(x_1, y_1), P_2(x_2, y_2)$.

Outputs: Alice gets the angle of elevation θ.

(1) Alice takes a vector $X = (x_0, y_0, 1)$, Bob takes a vector $Y = (\frac{x_2 - x_1}{d}, \frac{y_2 - y_1}{d}, \frac{-x_1(x_2 - x_1) - y_1(y_2 - y_1)}{d})$, and a random number v, in which $d = |\overrightarrow{P_1P_2}|$.

(2) Alice engages in a secure scalar product protocol with Bob, and gets $u = X \cdot Y + v$.

(3) Bob sends v to Alice.

(4) Alice engages in a secure two points distance protocol with Bob, and gets $d_0 = |\overrightarrow{P_1P_0}|$.

(4) Alice computes the angle as $\theta = \arccos\frac{u-v}{|\overrightarrow{P_1P_0}|}$.

Theorem 1 The Protocol 1 is correct, secure, and the complexity is twofold the one of a secure scalar product protocol.

Proof. **Correctness:** Because $u-v = X\cdot Y = \frac{1}{d}[x_0(x_2-x_1)+y_0(y_2-y_1)-x_1(x_2-x_1)-y_1(y_2-y_1)] = \frac{1}{d}[(x_2-x_1)(x_0-x_1)+(y_2-y_1)(y_0-y_1)] = \frac{1}{d}[(x_2-x_1,y_2-y_1)\cdot(x_0-x_1,y_0-y_1)] = \frac{\overrightarrow{P_1P_2}\cdot\overrightarrow{P_1P_0}}{|\overrightarrow{P_1P_2}|}$. Therefore $\theta = \arccos\frac{\overrightarrow{P_1P_2}\cdot\overrightarrow{P_1P_0}}{|\overrightarrow{P_1P_2}|\cdot|\overrightarrow{P_1P_0}|} = \arccos\frac{u-v}{|\overrightarrow{P_1P_0}|}$.

Security:

In definition of simulation paradigm(Definition 1 section 2.1 of [6]), $x = \{P_0(x_0,y_0)\}, y = \{P_1(x_1,y_1), P_2(x_2,y_2)\}, f(x,y)$ $= \theta, view_1^{\Pi}(x,y) = \{P_0(x_0,y_0),\theta,d_0,u,v\}$ satisfying $\cos\theta = \frac{u-v}{d_0}$, $f_2(x,y) = output_2^{\Pi}(x,y)$. We construct a simulator S_1 as follows:

S_1 receives $\{P_0(x_0,y_0),\theta\}$ as its inputs, and proceeds by

(1) S_1 selects randomly and uniformly a point $P_1'(x_1',y_1')$, and computes $d_0' = |\overrightarrow{P_0P_1'}|$.

(2) S_1 constructs a vector $Y' = (\cos\alpha, \sin\alpha, -x_1'\cos\alpha - y_1'\sin\alpha)$ with α as unknown variable.

(3) S_1 establishes an equation $\cos\theta = \frac{X\cdot Y'}{d_0'}$, in which $X = (x_0,y_0,1)$, solves this equation to get $\cos\alpha, \sin\alpha$, therefore, gets the vector Y'.

(4) S_1 selects randomly and uniformly a number v' and computes $u' = X\cdot Y' + v'$.

(5) S_1 outputs $S_1(x,f(x,y)) = \{P_0(x_0,y_0),\theta,$ $d_0',u',v'\}$, in which θ, d_0', u', v' satisfy $\cos\theta = \frac{u'-v'}{d_0'}$.

It is obvious $\{S_1(x,f_1(x,y))\} \equiv^C \{view_1^{\Pi}(x,y)\}$.

For Bob, he does not obtain any output, and will learn nothing about P_0.

Complexity: In step (2) and step (4), secure scalar product protocol is performed twice, therefore the complexity of the protocol is twofold the one of secure scalar product protocol.

2.2 Secure Computation of Intersection of Two Line Segments

We assume that two users Alice and Bob respectively hold a private line segment L_1, L_2 with endpoints $P_1(x_1,y_1), P_2(x_2,y_2)$ and $P_1'(x_1',y_1'), P_2'(x_2',y_2')$. Both of them want to determine whether L_1 and L_2 intersect, to determine the intersection point if they intersect.

The protocol is as follows.

Protocol 2 (Secure intersection of two line segments protocol)
Inputs: Alice has a private line segment L_1 with endpoints $P_1(x_1,y_1), P_2(x_2,y_2)$, and Bob has a

private line segment L_2 with endpoints $P_1'(x_1', y_1'), P_2'(x_2', y_2')$.

Outputs: Whether or not L_1 and L_2 intersect, and the intersection point if they intersect.

We divide the protocol as two subprotocols.

Subprotocol 2.1

(1) Alice takes vectors $X_1 = ((y_1-y_2)^2, (x_2-x_1)(y_1-y_2), (x_2-x_1)^2, (y_1-y_2)(x_1y_2-x_2y_1), (x_2-x_1)(x_1y_2-x_2y_1), (x_1y_2-x_2y_1)^2)$, $X_2 = (x_1x_2, x_1y_2+y_1x_2, y_1y_2, x_1+x_2, y_1+y_2, 1)$, and Bob takes vectors $Y_1 = (x_1'x_2', x_1'y_2'+y_1'x_2', y_1'y_2', x_1'+x_2', y_1'+y_2', 1)$, $Y_2 = ((y_1'-y_2')^2, (x_2'-x_1')(y_1'-y_2'), (x_2'-x_1')^2, (y_1'-y_2')(x_1'y_2'-x_2'y_1'), (x_2'-x_1')(x_1'y_2'-x_2'y_1'), (x_1'y_2'-x_2'y_1')^2)$, and two random numbers v_1, v_2.

(2) Alice engages in a secure scalar product protocol with Bob twice, and gets $u_1 = X_1 \cdot Y_1 + v_1$ and $u_2 = X_2 \cdot Y_2 + v_2$.

(3) Alice engages in a millionaire protocol with Bob twice to determine which is larger between u_1 and v_1, u_2 and v_2, respectively. If $u_1 \leq v_1$ and $u_2 \leq v_2$, L_1 and L_2 intersect, Alice communicates the result to Bob, and goes to the Subprotocol 2.2 with Bob; else, Alice communicates the result to Bob, and returns.

Subprotocol 2.2

(4) Alice takes vectors $A_1 = (x_2 - x_1, y_2 - y_1), A_2 = (-x_1, y_1, 1)$, Bob takes vectors $B_1 = (y_2' - y_1', -(x_2' - x_1')), B_2 = (y_2' - y_1', x_2' - x_1', x_1'(y_2' - y_1') - y_1'(x_2' - x_1'))$, and two random numbers v_1', v_2'. Alice engages in a secure scalar product protocol with Bob twice, gets $u_1' = A_1 \cdot B_1 + v_1'$ and $u_2' = A_2 \cdot B_2 + v_2'$.

(5) Bob sends v_1', v_2' to Alice.

(6) Alice computes $t_1 = \frac{u_2'-v_2'}{u_1'-v_1'}$, and obtains the intersection point as $(x_1 + t_1(x_2 - x_1), y_1 + t_1(y_2 - y_1))$.

(7) Alice communicates the result to Bob.

Theorem 2 The Protocol 2 is correct, secure, and the complexity is the addition of four times the one of a secure scalar product protocol with two times the one of a millionaire protocol.

Proof. **Correctness:** If L_1 and L_2 intersect, P_1, P_2, the endpoints of L_1, lie on opposite sides of L_2. As the same time, P_1', P_2', the endpoints of L_2, lie on opposite sides of L_1. Therefore $D(P_1', P_2', P_1) \cdot D(P_1', P_2', P_2) \leq 0$ and $D(P_1, P_2, P_1') \cdot D(P_1, P_2, P_2') \leq 0$. It is easy to verify that in step(1)$\sim$(3), $X_1 \cdot Y_1 = D(P_1, P_2, P_1') \cdot D(P_1, P_2, P_2')$ and $X_2 \cdot Y_2 = D(P_1', P_2', P_1) \cdot D(P_1', P_2', P_2)$. Therefore, in step (3), Alice can correctly judge whether L_1 and L_2 intersect.

The intersection point can be computed by a parametric equation securely. The parametric equations of L_1 and L_2 are $s_1(t_1) = \overrightarrow{OP_1} + t_1\overrightarrow{P_1P_2} = (x_1 + t_1(x_2 - x_1), y_1 + t_1(y_2 - y_1))$ and $s_2(t_2) = \overrightarrow{OP_1'} + t_2\overrightarrow{P_1'P_2'} = (x_1' + t_2(x_2' - x_1'), y_1' + t_2(y_2' - y_1'))$, respectively. Solve the simultaneous equations, we can obtain the parameter of the intersection point as $t_1 = \frac{-x_1(y_2'-y_1')+y_1(x_2'-x_1')+x_1'(y_2'-y_1')-y_1'(x_2'-x_1')}{(x_2-x_1)(y_2'-y_1')-(x_2'-x_1')(y_2-y_1)} = \frac{A_2 \cdot B_2}{A_1 \cdot B_1} = \frac{u_2'-v_2'}{u_1'-v_1'}$. Therefore Alice gets the intersection point as $(x_1 + t_1(x_2 - x_1), y_1 + t_1(y_2 - y_1))$

correctly.

Security: The security of Subprotocol 2.1 is obvious from the security of secure scalar product protocol and millionaire protocol. We prove the security of Subprotocol 2.2 as follows.

In definition of simulation paradigm, $x = \{P_1(x_1, y_1), P_2(x_2, y_2)\}$,
$y = \{P_1'(x_1', y_1'), P_2'(x_2', y_2')\}$,
$f(x, y) = \{(x_0, y_0)\}$, $view_1^{\Pi}(x, y) = \{P_1(x_1, y_1), P_2(x_2, y_2), (x_0, y_0), t_1, u_1', v_1', u_2', v_2'\}$, satisfying $t_1 = \frac{u_2' - v_2'}{u_1' - v_1'}$. $f_2(x, y) = output_2^{\Pi}(x, y)$.

We construct a simulator S_1 as follows:
S_1 receives $\{P_1(x_1, y_1), P_2(x_2, y_2), (x_0, y_0)\}$ as its inputs, and proceeds by

(1) S_1 selects randomly and uniformly a point $P_1''(x_1'', y_1'') \notin \overrightarrow{P_1P_2}$.

(2) S_1 establishes a system of equations

$$\begin{cases} \frac{y - y_1''}{x - x_1''} = \frac{y_0 - y_1''}{x_0 - x_1''} \\ x_0 = t_1(x_2 - x_1) + x_1 \\ t_1(A_1 \cdot B_1') = A_2 \cdot B_2' \end{cases} \tag{1}$$

in which $B_1' = (y - y_1'', -(x - x_1'')), B_2' = (y - y_1'', x - x_1'', x_1''(y - y_1'') - y_1''(x - x_1''))$.

It is easy to prove that the system of equations is equivalent to $\frac{y - y_1''}{x - x_1''} = \frac{y_0 - y_1''}{x_0 - x_1''}$, thus, step (2) can be substituted by

(2) S_1 establishes an equation of line

$$\frac{y - y_1''}{x - x_1''} = \frac{y_0 - y_1''}{x_0 - x_1''} \tag{2}$$

selects randomly and uniformly a point $P_2''(x_2'', y_2'')$ in this line, satisfying $D(P_1, P_2, P_1'') \cdot D(P_1, P_2, P_2'') \leq 0$.

(3) S_1 constructs two vectors B_1', B_2' from P_1'' and P_2'', as the same way that Bob constructs B_1, B_2. S_1 computes $A_1 \cdot B_1', A_2 \cdot B_2', t_1 = \frac{A_2 \cdot B_2'}{A_1 \cdot B_1'}$.

(4) S_1 selects randomly and uniformly two numbers v_1'', v_2'', and computes $u_1'' = A_1 \cdot B_1' + v_1'', u_2'' = A_2 \cdot B_2' + v_2''$

(5) S_1 outputs $\{P_1(x_1, y_1), P_2(x_2, y_2), (x_0, y_0),$
$t_1, u_1'', v_1'', u_2'', v_2''\}$, in which $t_1, u_1'', v_1'', u_2'',$
v_2'' satisfy $t_1 = \frac{u_2'' - v_2''}{u_1'' - v_1''}$ because of the equivalence of equation (1) and equation (2).

It is obvious $\{S_1(x, f_1(x, y))\} \equiv^C \{view_1^{\Pi}(x, y)\}$.

The simulator S_2 can be constructed analogously.

This protocol can not be performed twice, else Alice will get four equations with variables x_1', y_1', x_2', y_2', solves these equations, and obtains $P_1'(x_1', y_1'), P_2'(x_2', y_2')$.

In fact, from the geometry signification of this problem, if Alice gets two intersection points of L_1 with L_2, she will get the equation of line on which L_2 lies.

Complexity: In the protocol, four secure scalar product protocols and two millionaire protocols are performed, therefore the complexity is the addition of four times the one of a secure scalar product protocol with two times one of a millionaire protocol.

2.3 Secure Binary Search Protocol

Given a sequence $S = \{x_1, \cdots, x_n\}$ of n non-decreasing datum items, to determine the position of an item x in S, the binary search process is performed as follows.

$low = 1; hign = n;$

while $low \leq high$ do

$mid = \lfloor \frac{low+high}{2} \rfloor;$

case

$x > x_{mid} : low = mid + 1;$

$x = x_{mid}$: return;

$x < x_{mid} : high = mid - 1;$

end.

The secure binary search protocol is needed if we are concerned with privacy. Let x be a private item held by Alice, S is a private non-decreasing sequence held by Bob. Alice wants to determine the position of x in S, but does not want reveal her private item to Bob, nor does Bob to Alice. We design a protocol based on the method of [25].

Let E be Alice's semantically secure public-key encryption scheme with additional homomorphic property, D is the decryption scheme correspondingly. For a cryptosystem to be semantically secure, it must be infeasible for a computationally-bounded adversary to derive significant information about a message (plaintext) when given only its ciphertext and the corresponding public encryption key.

The protocol is as follows.

Protocol 3 (Secure binary search protocol)
Inputs: Alice has a private item x and Bob has a private non-decreasing sequence $S = \{x_1, \cdots, x_n\}$.
Outputs: Alice gets the position of x in S or the range of x between two items of S.

(1) Alice computes $c = E(x)$ and sends it to Bob.

(2) Bob selects n tri-tuple integers (u_i, v_i, w_i), satisfying $|v_i - w_i| < u_i$ and $u_i > 0$, for $i = 1, \cdots, n$. If Paillier's public-key cryptosystem is used, n tri-tuple integers (u_i, v_i, w_i) must further satisfy $u_i N + v_i < N^2, u_i N + w_i < N^2$, in which N is modulus, such that $x < N$ and $x_i < N (i = 1, \cdots, n)$. Bob computes $(X_i, Y_i) = (c^{u_i} E(v_i), E(u_i x_i + w_i))$ and sends them to Alice.

(3) Alice computes

$low = 1; hign = n;$

while $low \leq high$ do
$mid = \lfloor \frac{low+high}{2} \rfloor$;
$D(X_{mid}) = u_{mid}x + v_{mid}$;
$D(Y_{mid}) = u_{mid}x_{mid} + w_{mid}$;
case
$D(X_{mid}) \geq D(Y_{mid})$:
$low = mid + 1$;
$D(X_{mid}) < D(Y_{mid})$:
$high = mid - 1$;
end;
return mid.

Theorem 3 The Protocol 3 is correct, secure, and the round complexity is $O(logn)$.

Proof. **Correctness:** The correctness follows from [25]. From $D(X_{mid}) \leq N^2, D(Y_{mid}) \leq N^2$, and $\frac{D(X_{mid})-D(Y_{mid})}{u_{mid}} = x - x_{mid} + \frac{v_{mid}-w_{mid}}{u_{mid}}$, $-1 < \frac{v_{mid}-w_{mid}}{u_{mid}} < 1, u_{mid} > 0$, we get that $D(X_{mid}) - D(Y_{mid})$ has the same sign as $x - x_{mid}$. So, if $D(X_{mid}) \geq D(Y_{mid})$, then $x \geq x_{mid}$, x must be in right half of S, and $low = mid+1$, else $x \leq x_{mid}$, x must be in left half of S, and $high = mid-1$. Therefore, the correctness immediately follows from the correctness of binary search without privacy.
Security: In the protocol, Bob gets $c = E(x)$, from the security of encryption scheme E, Bob can not get any information of x, including which side x is in S.

For Alice, from $(X_i, Y_i) = (c^{u_i}E(v_i), E(u_ix_i + w_i))$, she can not get x_i, for $i = 1, \cdots, n$, this is because $D(X_i) = u_ix + v_i, D(Y_i) = u_ix_i + w_i$, in which u_i, v_i, w_i are randomly selected by Bob, x_i is masked by u_i and w_i.
Complexity: From the round complexity of binary search, the round complexity of Protocol 5 is $O(logn)$.

3 Some Secure Protocols on Planar Convex Hulls

3.1 Secure Point Inclusion Protocol

We assume that Alice has a private point Q, Bob has a private convex hull P with vertices $P_1, \cdots, P_n$ in counterclockwise order. They want to determine whether Q is included in P cooperatively, but neither wants to reveal its private data to the other.

Bob can do precomputation on his convex hull P as following:

Takes the first edge $\overrightarrow{P_1P_2}$ as the reference edge, computes all angles of vertices with respect to the reference edge and gets a non-decreasing sequence of $(\theta_3, \cdots, \theta_n)$, in which $\theta_i = \arccos \frac{\overrightarrow{P_1P_2} \cdot \overrightarrow{P_1P_i}}{|\overrightarrow{P_1P_2}| \cdot |\overrightarrow{P_1P_i}|}$.

If the privacy is not concerned with, the protocol can be performed as follows.
Protocol 4 (The point inclusion protocol without privacy) [26, 27]

(1) Calculate the angle of elevation that Q makes with $\overrightarrow{P_1P_2}$ as reference edge, which denoted as θ_Q.

(2) If $\theta_Q < 0$ or $\theta_Q \geq \pi$, then report that Q is outside the convex hull; else, go to next step.

(3) If $0 \leq \theta_Q < \pi$, perform a binary search for largest $\theta_i \leq \theta_Q$, with two possibilities

- $\theta_Q > \theta_n$: Q is outside the convex hull.
- $\theta_i \leq \theta_Q < \theta_{i+1}$ for some i: the point lies in the sector bounded by the half-infinite rays from P_1 to P_i and P_{i+1} (include on boundary $\overrightarrow{P_1P_i}$). To further determine Q is in the left or right of $\overrightarrow{P_iP_{i+1}}$, judge the sign of the determinant $D(P_i, P_{i+1}, Q)$. If $D(P_i, P_{i+1}, Q) < 0$, Q is outside the convex hull; else, Q is inside the convex hull(include on boundary $\overrightarrow{P_iP_{i+1}}$).

If the privacy is concerned with, the protocol can be performed as follows. In which, (E_A, D_A), (E_B, D_B) with homomorphic property and commutative property are semantically secure encryption and decryption pairs of Alice and Bob. For a vector $Z = (z_1, \cdots, z_n)$, we define $E(Z) = (E(z_1), \cdots, E(z_n))$, $D(Z) = (D(z_1), \cdots, D(z_n))$.

Protocol 5 (Secure point inclusion protocol)

Inputs: Alice has a private point $Q(x_0, y_0)$, Bob has a private convex hull P with vertices $P_1(x_1, y_1), \cdots, P_n(x_n, y_n)$ in counterclockwise order and a non-decreasing sequence of $(\theta_3, \cdots, \theta_n)$, in which θ_i is the angle of vertex P_i with respect to the reference edge $\overrightarrow{P_1P_2}$.

Output: Whether or not Q is included in P.

(1) Alice engages in a secure angle of elevation protocol with Bob, and gets the angle of elevation that Q makes with $\overrightarrow{P_1P_2}$ as reference edge, which denoted as θ_Q.

(2) Alice checks whether or not $\theta_Q < 0$ or $\theta_Q \geq \pi$, if it is yes, Q is outside P, Alice communicates the result to Bob; else, go to next step.

(3) Alice engages in a secure binary search protocol for j, such that $\theta_j \leq \theta_Q < \theta_{j+1}$, with Bob.

(4) Alice takes $A = (x_0, y_0, 1)$, computes $E_A(A)$ and sends it to Bob.

(5) Bob takes $B_i = (y_i - y_{i+1}, x_{i+1} - x_i, x_i y_{i+1} - x_{i+1} y_i)$, for $i = 2, \cdots, n$, in which $n+1$ is taken as 1, the same below, computes $E_B(E_A(A))$, $E_B(B_i)(i = 2, \cdots, n)$ and sends them to Alice.

(6) Alice computes $D_A(E_B(E_A(A))) = E_B(A)$.

(7) Alice computes $E_B(A) * E_B(B_j) = E_B(AB_j)$, and sends Bob $E_A(E_B(AB_j))$.

(8) Bob computes and sends Alice $D_B(E_A(E_B(AB_j))) = E_A(AB_j)$.

(9) Alice computes $D_A(E_A(AB_j)) = AB_j$, and checks its sign. If the sign is negative, Q is outside P; else, Q is inside P(include on boundary $\overrightarrow{P_jP_{j+1}}$).

(10) Alice communicates the result to Bob.

Theorem 4 The Protocol 5 is correct, secure, the communication cost and the computational cost are $O(nd)$ and $O(n \log N)$ modular multiplications respectively, where d is the number of bits needed to represent any number in the input, and N is modulus in Paillier's homomorphic encryption scheme.

Proof. **Correctness:** In step (1), from the correctness of secure angle of elevation protocol, Alice gets the angle of elevation that Q makes with $\overrightarrow{P_1P_2}$ as reference edge. In steps (2)$\sim$(3), if $0 \leq \theta_Q < \pi$, then from the correctness of the secure binary search protocol, Alice gets j which satisfies $\theta_j \leq \theta_Q < \theta_{j+1}$, that is, Q lies in the sector bounded by half-infinite rays from P_1 to P_j and P_{j+1} (include on boundary $\overrightarrow{P_1P_j}$).
In step (4)$\sim$(9), Alice and Bob determine in which side of $\overrightarrow{P_jP_{j+1}}$ Q lies cooperatively. In step (4), by her point Q, Alice takes a vector $A = (x_0, y_0, 1)$. In step (5), since he does not know which sector Q lies in, Bob takes $n-1$ vectors $B_i = (y_i - y_{i+1}, x_{i+1} - x_i, x_i y_{i+1} - x_{i+1} y_i)$, for $i = 2, \cdots, n$, by his vertices $P_2(x_2, y_2), \cdots, P_n(x_n, y_n)$, and sends Alice $E_B(B_i)(i = 2, \cdots, n)$. It is easily verified that AB_i is equal to the determinant $D(P_i, P_{i+1}, Q)$. Alice knows which B_i is the one that she needs, so she chooses and gets $E_B(B_j)$ in step (7). From the property of the homomorphic encryption scheme, through steps (4) $\sim$ (9), Alice gets AB_j, i.e. $D(P_j, P_{j+1}, Q)$. If $D(P_j, P_{j+1}, Q) < 0$, then Q lies in the right of directed line segment $\overrightarrow{P_jP_{j+1}}$, i.e. Q is outside P. Therefore, Alice can get correct result.
Security: From the security of secure angle of elevation protocol and the security of secure binary search protocol, the protocol in step (1)$\sim$(3) is secure. We construct two simulators from step (4) on.

S_1 receives $\{Q(x_0, y_0), A \cdot B_j\}$ as its inputs, that is, S_1 knows $\{Q(x_0, y_0)\}$ lies in which wedge, and proceeds by

(1) S_1 selects randomly and uniformly a number x'_{j+1} and a series of points $P'_1(x'_1, y'_1), \cdots, P'_j(x'_j, y'_j)$, $P'_{j+2}(x'_{j+2}, y'_{j+2}), \cdots, P'_n(x'_n, y'_n)$.

(2) S_1 establishes an equation $x_0(y'_j - y'_{j+1}) + y_0(x'_{j+1} - x'_j) + x'_j y'_{j+1} - x'_{j+1} y'_j = A \cdot B_j$, in which y'_{j+1} is as unknown variable, solve this equation, and get y'_{j+1} and point $P'_{j+1}(x'_{j+1}, y'_{j+1})$.

(3) S_1 constructs a series of vectors $B'_i = (y'_i - y'_{i+1}, x'_{i+1} - x'_i, x'_i y'_{i+1} - x'_{i+1} y'_i)$, for $i = 1, \cdots, n$, computes $E_B(B'_i)(i = 2, \cdots, n)$ and $E_B(A \cdot B'_j)$.

(4) S_1 outputs $\{S_1(x, f_1(x, y))\} = \{Q(x_0, y_0), A \cdot B_j, E_B(B'_1), \cdots, E_B(B'_n)\}$

Because $\{view_1^{\Pi}(x, y)\} = \{Q(x_0, y_0), A \cdot B_j, E_B(B_1), \cdots, E_B(B_n)\}$, from the semantically security of E_B, $\{E_B(B'_1), \cdots, E_B(B'_n)\} \equiv^C \{E_B(B_1), \cdots, E_B(B_n)\}$, and from the construction of S_1, $A \cdot B'_j = A \cdot B_j$, therefore we have $\{S_1(x, f_1(x, y))\} \equiv^C \{view_1^{\Pi}(x, y)\}$.

The construction of S_2 is as follows.

S_2 receives $P_1(x_1, y_1), \cdots, P_n(x_n, y_n)$ and a symbol λ as its inputs, in which λ is used to indicate whether or not Q is included in P. Let $\lambda = 1$, if Q is included in P; else $\lambda = 0$. $\{view_2^{\Pi}(x, y)\} = \{P_1(x_1, y_1), \cdots, P_n(x_n, y_n), \lambda, E_A(A), E_A(A \cdot B_j)\}$. S_2 proceeds by

(1) S_2 selects randomly and uniformly two neighbor points $P_{i'}(x_{i'}, y_{i'}), P_{i'+1}(x_{i'+1}, y_{i'+1})$, constructs a vector $B'_{i'} = (y_{i'} - y_{i'+1}, x_{i'+1} - x_{i'}, x_{i'} y_{i'+1} - x_{i'+1} y_{i'})$.

(2) S_2 selects randomly and uniformly a point $Q'(x'_0, y'_0)$ satisfying $A' \cdot B'_{i'} > 0$ if $\lambda = 1$, else $A' \cdot B'_{i'} < 0$, in which $A' = (x'_0, y'_0, 1)$.

(3) S_2 computes $E_A(A'), E_A(A' \cdot B'_{i'})$ and outputs $\{S_2(y, f_2(x, y))\} = \{P_1(x_1, y_1), \cdots, P_n(x_n, y_n), \lambda, E_A(A'), E_A(A' \cdot B'_{i'})\}$

From the semantically security of E_A, $E_A(A)$ and $E_A(A')$, $E_A(A \cdot B_j)\}$ and $E_A(A' \cdot B'_{i'})\}$ are indistinguishable, respectively. We have $\{S_2(y, f_2(x, y))\} \equiv^C \{view_2^{\Pi}(x, y)\}$

Complexity:

Communication cost: The communication cost constitutes the one of a secure angle of elevation protocol (which is equivalent to the one of two secure scalar product protocols from theorem 2) and exchange of $n + 4$ data, thus the communication cost is $8\mu nd + (n + 4)d = O(nd)$, where d is the number of bits needed to represent any number in the input.

Computational cost: The computational cost constitutes the one of a secure angle of elevation protocol, $2n$ encryptions and $2 \log n$ decryptions in a secure binary search protocol in Step (3), and $n+6$ encryption or decryptions in Step (4) $\sim$ (9). Thus the computational cost is $8\mu n \log N + 2(3n + 2 \log n + 6) \log N = O(n \log N)$ modular multiplications, where N is modulus in Paillier's homomorphic encryption scheme if Paillier's homomorphic encryption scheme is used.

Compared with the protocol in [5]: In [5], the second protocol in which binary search was used has round complexity $O(logn)$, but in each round, a secure scalar product protocols is needed. Thus, the communication cost and computational cost are $4nd + (\log n)(2 + 4\mu nd) = O(nd \log n)$ and $8n \log N + (3 + 4\mu n \log N) \log n = O(n \log n \log N)$ modular multiplications, respectively.

So, the communication cost and computational cost in Protocol 7 is $\frac{1}{\log n}$ of the one in [5], respectively.

3.2 Secure Intersection of Two Convex Hulls

We assume that Alice has a private convex hull P with vertices $P_1, \cdots, P_m$ in counterclockwise order, Bob has a private convex hull Q with vertices $Q_1, \cdots, Q_n$ in counterclockwise order. Both want to determine the intersection of two convex hulls; however neither of them wants to disclose private information to the other.

The vertices of intersection of two convex hulls P and Q consist of three types of vertices:

- The vertices in P which are included in Q.
- The vertices in Q which are included in P.
- The intersections of P's boundaries and Q's boundaries.

The first two types of vertices can be determined by calling a secure point inclusion protocol. To determine third type of vertices, we need design a new secure protocol, our protocol is based on the idea of [28]. The idea is to have the edges "chase" each other in such a way that the intersection points will all be found. This means that neither edge will get too far ahead of the other.

Let $\overrightarrow{P_iP_{i+1}}$ and $\overrightarrow{Q_jQ_{j+1}}$ be two edges which are being checked, if $\overrightarrow{Q_jQ_{j+1}}$ "aims toward" $\overrightarrow{P_iP_{i+1}}$, but does not cross it, then we have j to advance in order to "close in" on a possible intersection with P. Similarly, i to advance.

Let $H(e)$ be the open halfplane to the left of edge e, $(\overrightarrow{P_iP_{i+1}} \times \overrightarrow{Q_jQ_{j+1}})_z$ be the z coordinate of the cross product $\overrightarrow{P_iP_{i+1}} \times \overrightarrow{Q_jQ_{j+1}}$. The advance rules is given in the Table 1.

in which, $Q_{j+1} \in H(\overrightarrow{P_iP_{i+1}})$ if and only if Q_{j+1} lies in the left of directed line segment $\overrightarrow{P_iP_{i+1}}$, i.e. $D(P_i, P_{i+1}, Q_{j+1}) > 0$. Similarly, to judge other three halfplane conditions.

Table 1: The advance rules of edge

$(\overrightarrow{P_iP_{i+1}} \times \overrightarrow{Q_jQ_{j+1}})_z$	Halfplane	Advance Rule
> 0	$Q_{j+1} \in H(\overrightarrow{P_iP_{i+1}})$	$i = i + 1$
> 0	$Q_{j+1} \notin H(\overrightarrow{P_iP_{i+1}})$	$j = j + 1$
< 0	$P_{i+1} \in H(\overrightarrow{Q_jQ_{j+1}})$	$j = j + 1$
< 0	$P_{i+1} \notin H(\overrightarrow{Q_jQ_{j+1}})$	$i = i + 1$

The secure advance rules protocol for two users A and B is given as follows.

Protocol 6 Adv(A, B, (P_i, P_{i+1}), (Q_j, Q_{j+1}))

Inputs: A has an edge (P_i, P_{i+1}) on a private convex P, B has an edge (Q_j, Q_{j+1}) on another private convex Q.

Outputs: The result of advancing along $\overrightarrow{P_iP_{i+1}}$ or along $\overrightarrow{Q_jQ_{j+1}}$.

A engages in a secure cross product with B to determine the sign of $(\overrightarrow{P_iP_{i+1}} \times \overrightarrow{Q_jQ_{j+1}})_z$.

If "$(\overrightarrow{P_iP_{i+1}} \times \overrightarrow{Q_jQ_{j+1}})_z > 0$",

- A engages in a secure scalar product with B to determine the sign of $D(P_i, P_{i+1}, Q_{j+1})$.
  If "$D(P_i, P_{i+1}, Q_{j+1}) > 0$",
  $i = i + 1$, return;

  Else

  $j = j + 1$, return;

Else;

- B engages in a secure scalar product with A to determine the sign of $D(Q_j, Q_{j+1}, P_{i+1})$,
  If "$D(Q_j, Q_{j+1}, P_{i+1}) > 0$",

  $j = j + 1$, return;

  Else

  $i = i + 1$, return;

Theorem 5 The Protocol 6 is correct, secure, and the complexity is twofold the one of a secure scalar product protocol.

Proof. The correctness and security are obvious.

Complexity: In the protocol, a secure cross product, which is equivalent to a secure scalar product protocol, and a secure scalar product protocol are performed, therefore the complexity is twofold the one of a secure scalar product protocol.

To apply to the secure intersection of two convex hulls protocol after a secure point inclusion protocol being performed for two convex hulls P and Q, we can add two steps in the secure intersection of two line segments protocol (Protocol 2), from the property of convex hull, as follows.

(1) If both P_1 and P_2 are included in convex hull Q, Alice reports the result of "no intersect".

(2) If both P_1' and P_2' are included in convex hull P, Bob reports the result of "no intersect".

Protocol 7 (Secure intersection of two convex hulls protocol)
Inputs: Alice has a private convex hull P with vertices $P_1, \cdots, P_m$ in counterclockwise order, Bob has a private convex hull Q with vertices $Q_1, \cdots, Q_n$ in counterclockwise order.
Outputs: The intersection points of P and Q.

$i = 1; j = 1;$
while $(i < m)||(j < n)\&\&(i < 2m)\&\&(j < 2n)$ do
Alice engages in a secure line segment intersection protocol with Bob to determine whether or not $\overrightarrow{P_iP_{i+1}}$ intersects with $\overrightarrow{Q_jQ_{j+1}}$.
If it is yes, then

(1.1) return the intersection point;

(1.2) $i = i + 1$;

(1.3) Bob engages in a secure line segment intersection protocol with Alice to determine whether or not $\overrightarrow{Q_jQ_{j+1}}$ intersects with $\overrightarrow{P_iP_{i+1}}$.
If it is yes, then

(1.3.1) return the intersection point;

(1.3.2) $j = j + 1$;

Else

(1.3.3) call Adv(Bob, Alice, (Q_j, Q_{j+1}), (P_i, P_{i+1}));

(1.3.4) goto (1.3);

Else

(1.4) call Adv(Alice, Bob, (P_i, P_{i+1}), (Q_j, Q_{j+1}));

end{while}.

Theorem 6 The Protocol 7 is correct, secure, and the complexity is $4(m + n)$ times the one of a secure scalar product protocol.

Proof. **Correctness:** In each loop, firstly Alice judges whether or not $\overrightarrow{P_iP_{i+1}}$ intersects with $\overrightarrow{Q_jQ_{j+1}}$, if it is yes, from the correctness of a secure line segment intersection protocol, Alice gets the intersection point, has i to advance, and gets next edge. Then Bob judges whether or not $\overrightarrow{Q_jQ_{j+1}}$ intersects with $\overrightarrow{P_iP_{i+1}}$ in Step (1.3), if it is yes, from the correctness of a secure line segment intersection protocol, Bob gets the intersection point, has j to advance, and gets next edge. If $\overrightarrow{Q_jQ_{j+1}}$ does not intersect with $\overrightarrow{P_iP_{i+1}}$, Bob performs the protocol Adv (Bob, Alice, (Q_j, Q_{j+1}), (P_i, P_{i+1})) with Alice to determine which i or j to advance.

In loop, if $\overrightarrow{P_iP_{i+1}}$ does not intersect with $\overrightarrow{Q_jQ_{j+1}}$, Alice performs the protocol Adv (Alice, Bob, (P_i, P_{i+1}), (Q_j, Q_{j+1})) with Bob to determine which i or j to advance.

Therefore, the edges in two convex hulls are checked alternately, and all intersection points are obtained.

Security: In each loop, for edge $\overrightarrow{P_iP_{i+1}}$ held by Alice and edge $\overrightarrow{Q_jQ_{j+1}}$ held by Bob, from the security of a secure line segment intersection protocol and security of a secure advance rules protocol, Bob will learn nothing about P_i and P_{i+1}, Alice will learn nothing about Q_j and Q_{j+1}.
Complexity: The while loop is performed at most $m + n$ times. In each loop, at most two secure line segment intersection protocols and one secure advance rules protocol are performed, from Theorems 3 and 6, the complexity of each loop is four times the one of the secure scalar product protocol. Therefore, the complexity of Protocol 9 is $4(m + n)$ times the one of a secure scalar product protocol.

3.3 Secure Union of Two Convex Hulls

Two users, Alice and Bob, have a private convex hull P and Q respectively. They want to determine the union of two convex hulls; however neither wants to disclose private information to the other.

The vertices of union of two convex hulls P and Q consist of three types of vertex:

- The vertices in P which are not included in Q.
- The vertices in Q which are not included in P.
- The intersections of P's boundaries and Q's boundaries.

The first two types of vertices can be determined by calling a secure point inclusion protocol, the third type of vertices can be determined by calling protocol 9.

4 Conclusion

We have given protocols for secure computation of angle of elevation, intersection of two line segments, secure binary search protocol, and the point inclusion problem on convex hull, the intersection and the union of two convex hulls, analysized their correctness, security and complexities. Except secure binary search protocol, other protocols' complexities depend on the one of a secure scalar product protocol used. A secure scalar product protocol has served as one of the basic building blocks for many other secure protocols, the need for more efficient and more practical solutions for this problem still remains.

References

[1] O. Goldreich. Secure Multi-party Computation (Working Draft), 1998. (http://citeseer.ist.psu.edu /goldreich98secure.html)

[2] M. J. Atallah, W. Du. Secure multiparty computational geometry, Lecture Notes in Computer Science, Proceedings, LNCS 2125, 165-179, Algorithms and Data Structures: 7th International Workshop, WADS 2001, Providence, RI, USA, August 8-10, 2001

[3] S. D. Li, Y. Q. Dai. Secure two-party computational geometry, Journal of Computer Science and Technology, 20(2), 2005, 258-263

[4] Y. L. Luo, L. S. Huang, H Zhong. Secure two-party point-circle inclusion problem, Journal of Computer Science and Technology, 22(1), 2007, 88-91

[5] Tony Thomas. Secure two-party protocols for point inclusion problem, International Journal of Network Security, Vol. 9, No. 1, July 1-7, 2009

[6] Bo Yang, Aidong Sun, Wenzheng Zhang. Secure two-party protocols on planar circles. Journal of Information & Computational Science, Vol. 8, No. 1, 2011, 29-40

[7] O. Goldreich. Foundations of Cryptography: Basic Applications. Cambridge University Press, London, 2004

[8] A. C. Yao. Protocols for secure computations. in: Proc. the 23rd Annual IEEE Symposium on Foundations of Computer Science, Chicago, 1982, 160-164

[9] A. C. Yao. How to generate and exchange secrets. in: Proc. 27th Ann. IEEE Symp. Foundations of Computer Science, 1986, 162-167

[10] O. Goldreich, S. Micali, A. Wigderson. How to play any mental game. in: Proc. 19th Ann. ACM Symp. Theory of Computing, 1987, 218-229

[11] Marc Fischlin. A cost-effective pay-per-multiplication comparison method for millionaires. In Proc. the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA, Springer-Verlag, 2001, 457-472

[12] Ian F. Blake, Vladimir Kolesnikov. Strong conditional oblivious transfer and computing on intervals. In Proc. Advances in Cryptology - ASIACRYPT'04, Springer-Verlag, 2004, 515-529

[13] Hsiao-Ying Lin, Wen-Guey Tzeng. An efficient solution to the Millionaires' problem based on homomorphic encryption. Journal of Computer Science and Technology, 2005, 3531 of LNCS: 456-466

[14] S. D. Li, D. S. Wang, Y. Q. Dai, P. Luo. Symmetric cryptographic solution to Yao's millionaires' problem and an evaluation of secure multiparty computations. Information Sciences, 178 (1), 2008, 244-255

[15] W. Du, Z. Zhan. Building decision tree classifier on private data. in IEEE ICDM Workshop Proceedings, Volume 14 in the Conferences in Research and Practice in Information Technology Series, Australian Computer Society, Sydney, Australia, 2002, 1-8

[16] J. Vaidya, C. C. Clifton. Privacy preserving association rule mining in vertically partitioned data. in: Proc. The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, SIGKDD, ACM Press, Edmonton, Canada, 2002, 639-644

[17] W. Du, M. Atallah. Privacy-preserving cooperative statistical analysis. in: Proc. of the 17th Annual Computer Security Applications Conference (ACSAC), ACM SIGSAC, IEEE Computer Society, New Orleans, Louisiana, 2001, 102-110

[18] A. Artak, E. C. Vladimir. A new efficient privacy-preserving scalar product protocol. in: Proc. 6th Australasian Data Mining Conference (AusDM'07), Gold Coast, Australia. 209-214

[19] B. Goethals, S. Laur, H. Lipmaa, T. Mielikainen. On private scalar product computation for privacy-preserving data mining, in the 7th Annual International Conference in Information Security and Cryptology (ICISC), LNCS 3506, 2004, 104-120

[20] Z. Yang, R. N. Wright, H. Subramaniam. Experimental analysis of a privacy-preserving scalar product protocol, International Journal of Computer Systems Science and Engineering, vol. 21, no. 1, 2006, 47-52

[21] P. Paillier. Public-key cryptosystems based on composite degree residue classes, Advances in Cryptology, Eurocrypt'99, LNCS 1592, 1999, 223-238

[22] D. Naccache, J. Stern. A new cryptosystem based on higher residues, Proceedings of the 5th ACM Conference on Computer and Communications Security, 1998, 59-66

[23] T. Okamoto, S. Uchiyama. An efficient public-key cryptosystem, Advances in Cryptology, Eurocrypt'98, 1998, 308-318

[24] Jianer Chen. Computational Geometry: Methods and Applications, Texas AM, 1996

[25] Bo Qin, Hui Qin, Kefu Zhou, Xiaofeng Wang, Yumin Wang. Millionaire's protocol with constant complexity, Journal of Xi'an University of Technology, 21(2), 2005, 149-152 (in Chinese)

[26] Enoch Lau. Computational Geometry: An Odyssey. COMP4045 Assignments. SID 200415765

[27] Bo Zhao, Xisheng He, Mingxun Jiang. Convex Hull in 2D, http://www.tcs.fudan.edu.cn/rudolf /Courses/Algorithms/Alg_cs_07w /Webprojects/Zhaobo_hull/index.html

[28] Joseph O'Rourke. Computational Geometry in C (Second Edition), Cambridge University Press, September 1998