A secure multi-party computation solution to intersection problems of sets and rectangles*

LI Shundong[1]**, DAI Yiqi[2], WANG Daoshun[2] and LUO Ping[2]

(1. Department of Computer Science and Technology, Beijing Normal University, Beijing 100875, China; 2. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Received July 19, 2005; revised September 19, 2005

Abstract Secure multi-party computation (SMC) is a research focus in international cryptographic community. At present, there is no SMC solution to the intersection problem of sets. In this paper, we first propose a SMC solution to this problem. Applying Cantor encoding method to computational geometry problems, and based on the solution to set-intersection problem, we further propose solutions to points inclusion problem and intersection problem of rectangles and further prove their privacy-preserving property with widely accepted simulation paradigm. Compared with the known solutions, these new solutions are of less computational complexity and less communication complexity, and have obvious superiority in computational and communication complexity.

Keywords: Cantor encoding, secure multi-party computation, set-intersection, rectangle-intersection, computational complexity.

Secure multi-party computation (SMC) is recently a research focus in international cryptographic community[1]. It was first introduced by Yao[2], and extended by Goldreich, Micali and Wigderson[3,4] et al. They proved that the general SMC problem is solvable[2—4] in theory, but using the solutions derived by these general results for special cases of multi-party computation is impractical, and special solutions should be developed for special cases for effciency reasons[3,4]. SMC has many applications in electronic world. Goldwasser has predicted that "The field of SMC is today where public-key cryptography was ten years ago, namely an extremely powerful tool and rich theory whose real-life usage is at this time only beginning but will become in the future an integral part of our computing reality[5]."

Goldreich's observation and Goldwasser's prediction motivated people to search for specific SMC problems as well as to search for their solutions. These problems include comparing two numbers[2,6], privacy-preserving data mining[7], comparing information without leaking it[6], privacy-preserving scientific computation[8], secure remote database access[9], secure auction[10], privacy-preserving multivariate statistical analysis[11] etc. SMC and their applications have been described in other review articles[5,12].

Secure multi-party computational geometry is a new research field of SMC. Problems studied include inclusion problem of polygons, intersection problems of polygons, convex hulls of some private points[12,13], the distance of two private points, relative location of private circles or ellipses[14]. This paper will focus on the following two problems:

Problem 1 (Set intersection problem). Company A has a customer set $A = \{a_1, a_2, \cdots, a_m\}$, and bank B has a blacklist of bad credit customers $B = \{b_1, b_2, \cdots, b_n\}$. Company A and bank B want to jointly determine how many customers of company A are in the blacklist of bad credit customers without revealing A and B, that is to jointly compute the cardinality of $A \cap B$, denoted by $|A \cap B|$, without disclosing their elements. The most important case is that $m = 1$.

Problem 2 (Rectangle intersection problem[13]). Alice has a rectangle G_1, and Bob has a rectangle G_2; they both want to determine privately whether or not G_1 and G_2 intersect without revealing where the intersection occurs. Atallah et al. have investigated this problem about convex polygons, its interesting applications, and SMC.

Of course, rectangles are a special kind of polygons, so solution for polygons proposed in [13] can also be used to solve Problem 2. But using the solu-

* Supported by National Natural Science Foundation of China (Grant No. 90304014) and Hi-tech Research and Development Program of China (Grant No. 2005AA114160)

** To whom correspondence should be addressed. E-mail: shundong@mail.tsinghua.edu.cn

tions derived from polygons for the case of rectangles is impractical for efficiency. The very special nature of rectangles whose sides are parallel to two orthogonal directions suggests that perhaps more efficient SMC solutions can be developed to solve such highly structured problems.

In this study, we apply Cantor encoding method[15] to solve secure computational geometry problems, and obtain the following results:

(i) In order to solve the intersection problem and point-inclusion problem of rectangles, we first propose an efficient solution to set-intersection problem, which has some independent use in SMC.

(ii) Based on the solution to set-inclusion problems and Cantor encoding, we propose a solution to points inclusion problem of rectangles.

(iii) Based on the solution to set-intersection problem and Cantor encoding, we further propose a solution to intersection problem of rectangles.

These solutions have much less computational complexity and communication complexity than that of solutions derived from polygons, and their privacy-preserving properties are proved by the well-known simulation paradigm.

1 Related work

1.1 Determining whether or not two numbers are equal

Commutative encryption scheme[6]. We call an encryption scheme commutative if it satisfies that

$$E_a(E_b(x)) = E_b(E_a(x)), \quad (1)$$

where $E_a(x)$ denotes encrypting number x with key a.

Solution for determining whether or not two numbers are equal. Fagin et al.[6] have proposed a solution to this problem. If Alice and Bob want to determine $x \stackrel{?}{=} y$ privately, they can proceed as follows.

1) They negotiate a commutative encryption scheme E, and choose their keys a, b respectively.

2) Alice computes $E_a(x)$, and Bob computes $E_b(y)$.

3) Alice and Bob exchange $E_a(x)$ and $E_b(y)$.

4) Alice and Bob compute $E_a(E_b(y))$ and $E_b(E_a(x))$ respectively. Alice sends $E_a(E_b(y))$ to Bob.

5) Bob determines $E_b(E_a(x)) \stackrel{?}{=} E_a(E_b(y))$. If $E_b(E_a(x)) = E_a(E_b(y))$, then $x = y$; otherwise $x \neq y$.

1.2 Point-inclusion and intersection problems of rectangles

Motivation. Country A decides to bomb a rectangular region in country C. Country B, which is an ally of country A, has very important national benefits (political, economic or military) in country C. A does not want to hurt ally's benefits, neither does it want to disclose its bombing region, and B does not want to let A know where its benefits locate at unless this bomb will seriously hurt its benefits. So country A and B need to determine whether or not B's benefit locations are included in the bombing region without revealing the bombing region and benefit locations. This problem can be phrased formally as follows.

Points Inclusion Problem. Alice knows a point set $P = \{P_1(x_1, y_1), P_2(x_2, y_2), \cdots, P_m(x_m, y_m)\}$; Bob knows a rectangle G whose bottom left coordinates (resp., top right) is $G_l(x_l, y_l)$ (resp., $G_r(x_r, y_r)$). They want to determine whether or not P is included in G without revealing P and G.

After a costly market research, company A decides to expand its market share in some rectangular region. However A is aware of that another competing company B is also planning to expand its market share in some rectangular region. Strategically, A and B do not want to compete each other in the same region, so they want to know whether or not they have a region of overlap. Of course, they do not want to disclose their location information because not only this information costs both companies a lot of money, but it can also cause significant damage to the company if it were disclosed to other parties. This problem can be phrased as Problem 2.

In this application, it is the privacy of location of the rectangles, but not their shapes and area that we are concerned about. The known solutions are as follows.

Solution to points inclusion problem[13] **(Solution 1).**

For $i = 1, 2, \cdots, m$, Alice and Bob determine

that,

1) whether or not $x_l \leqslant x_i \leqslant x_r$;

2) whether or not $y_l \leqslant y_i \leqslant y_r$;

3) if $(x_l \leqslant x_i \leqslant x_r) \wedge (y_l \leqslant y_i \leqslant y_r)$, then $P_i \in G$, otherwise $P_i \notin G$.

Due to the requirement of privacy-preserving, steps 2) and 3) must be done secretly with invocation of the protocol for millionaires' problem. It needs to invoke millionaires' protocol 4 times to determine whether or not $P_i \in G$. The protocol is very difficult to execute for its high computational complexity.

Solution to rectangle-intersection problem[13] **(Solution 2).**

The solution for determining whether or not rectangle G_1 intersects G_2 proceeds as follows:

For $i = 1, 2, 3, 4$, Alice and Bob privately determine whether the i-th edge of G_1 intersects j-th ($j = 1, 2, 3, 4$) edge of G_2.

If there is a pair (i, j) such that the i-th edge of G_1 intersects j-th ($j = 1, 2, 3, 4$) edge of G_2, then G_1 intersects G_2, otherwise they do not intersect. It needs to invoke millionaires' protocol 4 times to determine whether or not one edge intersects another edge, so it needs to invoke millionaires' protocol 64 times to determine whether or not a rectangle intersects another rectangle. Millionaires' protocol is based on public key algorithm, and it is very inefficient compared with the protocols based on symmetric key algorithm. Moreover, it cannot solve the problems in the case that G_1 is totally inside G_2 or G_2 is totally inside G_1.

2 Preliminaries

2.1 Encoding coordinates of a point

To solve points inclusion problem of rectangle, we need a function $h: N \times N \to N$, where N is the set of natural numbers. Define $h(x, y)$ as

$$h(x,y) = \begin{cases} h(0,0) = 0 \\ h(0,1) = 1 \\ h(x,1) = h(x-1,1) + 1 \\ h(x,y) = h(x,y-1) + x + y. \end{cases} \quad (2)$$

It constructs a 1-1 mapping between a pair of natural numbers and a single natural number. The encoding table is shown in Table 1, where a column is corresponding to an x, and a row is corresponding to a y.

Table 1. Cantor encoding

	0	1	2	3	4	5	6	7	8	…
0	0	1	3	6	10	15	21	28	36	…
1	2	4	7	11	16	22	29	37	46	…
2	5	8	12	17	23	30	38	47	57	…
3	9	13	18	24	31	39	48	58	69	…
4	14	19	25	32	40	49	59	70	82	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

This encoding is called Cantor encoding[15] which is used to encode the points with natural number coordinates in this paper.

2.2 Privacy-preserving property

Semi-honest parties[4]. Our work assumes that all parties are semi-honest. Generally speaking, a semi-honest party is one who follows the protocol properly with the exception that it keeps a record of all its intermediate computations and might try to derive other parties' private inputs from the record. Proposition 2.1.3 in [4] says that if π is a protocol for computing a functionality f, then π privately computes f if and only if π securely computes f in a semi-honest model. Ref.[4] also shows that any protocol, secure in the semi-honest model, can be transformed into a protocol which is secure against any feasible adversarial behavior. So all SMC solutions focus on semi-honest parties, so are our solutions in this paper.

Definition 1[4]. Suppose that the parties participating the secure two-party computation are Alice and Bob.

(i) Let $f = (f_1, f_2)$ be a secure two-party computational probabilistic polynomial-time function, and denote the protocol for computing f by π. Alice (resp., Bob), holding input x (resp., y), wishes to obtain the first (resp., second) element in $f(x, y)$, denoted by $f_1(x, y)$ (resp., $f_2(x, y)$).

(ii) The view of Alice(resp., Bob) during an execution of π on input (x, y) denoted by $view_1^{\pi}(x, y)$ (resp., $view_2^{\pi}(x, y)$) is $(x, r^1, m_1^1, \cdots, m_t^1)$ (resp., $(y, r^2, m_1^2, \cdots, m_t^2)$), where r^1 (r^2) represents the outcome of Alice's (resp., Bob's) internal coin tosses, and m_i^1 (resp., m_i^2) represents the i-th message Alice (resp., Bob) has received.

(iii) The output of Alice (resp., Bob) during an execution of π on (x, y) is denoted by $output_1^{\pi}(x, y)$ (resp., $output_2^{\pi}(x, y)$).

SMC with respect to semi-honest parties. We say that π privately computes f, if there exist polynomial-time algorithms denoted by S_1 and S_2 such that

$$\{(S_1(x, f_1(x, y)), f_2(x, y))\}_{x,y\in\{0,1\}^*} \overset{c}{\equiv} \{(view_1^{\pi}(x, y), output_2^{\pi}(x, y))\}_{x,y\in\{0,1\}^*}, \quad (3)$$

$$\{(f_1(x, y), S_2(y, f_2(x, y)))\}_{x,y\in\{0,1\}^*} \overset{c}{\equiv} \{(output_1^{\pi}(x, y), view_2^{\pi}(x, y))\}_{x,y\in\{0,1\}^*}, \quad (4)$$

where $\overset{c}{\equiv}$ denotes computational indistinguishability.

3 Solutions

Basic idea. This section aims at solving the SMC problems of point-inclusion and intersection of rectangles. The thought is very straightforward:

Rectangle G is an infinite set of points on a plane, and point set P consists of m points on the same plane. If $P_i \in G$, then P_i is inside G. The key problem here is that G is an infinite set, and in order to determine whether or not $P_i \in G$, it must be transformed into a finite set.

Rectangles G_1 and G_2 are two infinite sets of points on a plane, and if their intersection set $G_1 \cap G_2 \neq \phi$, then G_1 intersects G_2; otherwise, G_1 does not intersect G_2. In order to privately compute whether or not $P_i \in G$ and $G_1 \cap G_2 \overset{?}{=} \phi$, rectangles must be transformed into finite sets. So the following work will focus on how to transform an infinite set into a finite set. This can be realized by Cantor encoding and some special technique which will be discussed later, and thus induces the problems of rectangle to the problems of sets.

3.1 Solution to set intersection problem

Convention. What we mean by saying encrypting a set is that encrypt all its elements one by one, that is, if $A = \{a_1, a_2, \cdots, a_n\}$, then $E(A) = \{E(a_1), E(a_2), \cdots, E(a_n)\}$.

We start with solving the set intersection problem, and then solve the SMC problem of point inclusion and intersection of rectangles.

Solution 3. Set-intersection Problem Protocol.

Inputs: Set $A = \{a_1, a_2, \cdots, a_m\}$ and set $B = \{b_1, b_2, \cdots, b_n\}$.

Output: $|A \cap B|$.

i) Alice and Bob negotiate a commutative encryption scheme and their private keys a, b.

ii) Alice and Bob compute $E_a(A) = \{E_a(a_1), E_a(a_2), \cdots, E_a(a_m)\}$ and $E_b(B) = \{E_b(b_1), E_b(b_2), \cdots, E_b(b_n)\}$ respectively. And then exchange their results $E_a(A), E_b(B)$.

iii) Alice computes $E_a(E_b(B)) = \{E_a(E_b(b_1)), E_a(E_b(b_2)), \cdots, E_a(E_b(b_n))\}$; and sends the result to Bob.

iv) Bob computes $E_b(E_a(A)) = \{E_b(E_a(a_1)), E_b(E_a(a_2)), \cdots, E_b(E_a(a_m))\}$, and $|A \cap B|$.

Clearly, $|E_b(E_a(A)) \cap E_a(E_b(B))| = |A \cap B|$. If $|E_b(E_a(A)) \cap E_a(E_b(B))| = \min(m, n)$, then either $A \subset B$ or $A \supset B$; if $|E_b(E_a(A)) \cap E_a(E_b(B))| = k (0 < k < \min(m, n))$, then $A \cap B \neq \phi$; if $|E_b(E_a(A)) \cap E_a(E_b(B))| = 0$, then $A \cap B = \phi$.

If $\min(m, n) = 1$, this scheme degenerates to a SMC scheme determining whether or not a special element belongs to a given set, that is the set-inclusion problem.

3.2 Solution to rectangle inclusion problem

The thought of this solution is also straightforward. Partition the rectangle into grids, and then make coordinate transformation such that coordinates of all grid nodes are pairs of natural numbers. In the new coordinate system, the coordinates of P are $\{P_1(x_1', y_1'), P_2(x_2', y_2'), \cdots, P_m(x_m', y_m')\}$, and the coordinates of bottom left (resp., top right) vertex of G is $G_l(x_l', y_l')$ (resp., $G_l(x_r', y_r')$). We represent every grid with the coordinates of its bottom left node. If a point $P_i(x_i', y_i') (0 < i \leqslant m)$ is inside rectangle G, then it must be inside or on a grid, and thus $([x_i']; [y_i'])$ equals bottom left vertex coordinate of this grid. The whole rectangle can be represented by a finite set G' of all its grid's bottom left node coordinates. Alice and Bob use $h(x, y)$ to respectively transform sets P, G into natural number sets $T(P)$, $T(G)$. This problem is thus induced to the problem of set intersection problem, and if $|T(P) \cap T(G)|$

$=i(0\leqslant i\leqslant m)$, then there are i points of P inside G.

Assume, without loss of generality, that all the points of P and G are in the first quadrant. The location relation between these points and the rectangle is shown in Fig. 1, and Cantor encoding method and the solution are shown in Fig. 2. Alice and Bob can transform G into a natural number set $T(G)$ as follows.

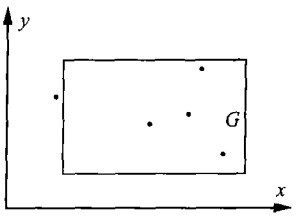

Fig. 1. The relation between points and the rectangle.

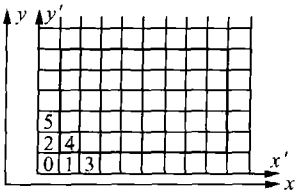

Fig. 2. The illustration of the solution to point-inclusion problem.

1) Alice randomly chooses two integral numbers p, q $(2\leqslant p\leqslant 10, -10\leqslant q\leqslant 10)$, and computes

$$u_x = \frac{x_r - x_l}{p},$$

$$u_y = \frac{y_r - y_l}{p},$$

$$(x_l - q\cdot u_x, y_l - q\cdot u_y). \tag{5}$$

2) Alice and Bob take point $(x_l - q\cdot u_x, y_l - q\cdot u_y)$ as the origin, and take u_x, u_y as the coordinate unit to build a new coordinate system. Clearly, this transformation does not affect the relative location of P with respect to G. In the new coordinate system, the coordinates of all vertices of rectangle G are pairs of natural numbers.

3) Suppose, in the new coordinate system, that the coordinates of P are $(x_i', y_i')(1\leqslant i\leqslant m)$, and the coordinates of the bottom left (resp., top right) vertex of G are (x_l', y_l')(resp., (x_r', y_r')).

4) Partition rectangle G into grids with lines $x' = \lfloor x_l' \rfloor, \lfloor x_l' \rfloor + 1, \cdots, \lfloor x_r' \rfloor - 1, \lfloor x_r' \rfloor$ and $y' = \lfloor x_l' \rfloor, \lfloor y_l' \rfloor + 1, \cdots, \lfloor y_r' \rfloor - 1, \lfloor y_r' \rfloor$, where $\lfloor x \rfloor$ denotes the largest integer that is smaller than x. These line segments partition G into p^2 grids.

5) Represent grid $(\lfloor x \rfloor < x \leqslant \lfloor x \rfloor + 1, \lfloor y \rfloor < y \leqslant \lfloor y \rfloor + 1)$ by its bottom left natural coordinates $(\lfloor x \rfloor, \lfloor y \rfloor)$. Thus these p^2 grids are represented by the set

$$G' = \begin{Bmatrix} (\lfloor x_l' \rfloor, \lfloor y_l' \rfloor + 0), & (\lfloor x_l' \rfloor + 1, \lfloor y_l' \rfloor + 0), & \cdots, & (\lfloor x_r' \rfloor, \lfloor y_l' \rfloor + 0), \\ (\lfloor x_l' \rfloor, \lfloor y_l' \rfloor + 1), & (\lfloor x_l' \rfloor + 1, \lfloor y_l' \rfloor + 1), & \cdots, & (\lfloor x_r' \rfloor, \lfloor y_l' \rfloor + 1), \\ \cdots, & \cdots, & \cdots, & \cdots, \\ (\lfloor x_l' \rfloor, \lfloor y_r' \rfloor + 0), & (\lfloor x_l' \rfloor + 1, \lfloor y_r' \rfloor + 0), & \cdots, & (\lfloor x_r' \rfloor, \lfloor y_r' \rfloor + 0) \end{Bmatrix} \tag{6}$$

and represent (x_i', y_i') $(1\leqslant i\leqslant m)$ by $(\lfloor x_i' \rfloor, \lfloor y_i' \rfloor)$ $(1\leqslant i\leqslant m)$, the following set can be obtained:

$$P' = \{(\lfloor x_1' \rfloor, \lfloor y_1' \rfloor), (\lfloor x_2' \rfloor, \lfloor y_2' \rfloor), \cdots, (\lfloor x_m' \rfloor, \lfloor y_m' \rfloor)\}. \tag{7}$$

6) Using Cantor encoding, Alice and Bob respectively transform natural number pair set P' and G' into natural number sets $T(P)$ and $T(G)$ which are shown as in Fig. 2, where

$$T(P) = \{h(\lfloor x_1' \rfloor, \lfloor y_1' \rfloor), \cdots, h(\lfloor x_m' \rfloor, \lfloor y_m' \rfloor)\},$$
$$T(G) = \{h(\lfloor x_l' \rfloor, \lfloor y_l' \rfloor), \cdots, h(\lfloor x_r' \rfloor, \lfloor y_r' \rfloor)\}. \tag{8}$$

Clearly, if $P_i \in G$, then $h(\lfloor x_i' \rfloor, \lfloor y_i' \rfloor) \in T(G)$. So, we have the following solution.

Solution 4. Solution to points inclusion problem.

Inputs: Point set P and rectangle G.

Output: The points number of P that are inside G.

1) Using the above approach, Alice, Bob transform P, G into natural number sets $T(P), T(G)$.

2) Using Solution 3, Alice and Bob privately compute $|T(P)\cap T(G)|$. If $|T(P)\cap T(G)| = k$ $(0\leqslant k\leqslant m)$, then there are $k(0\leqslant k\leqslant m)$ points of P inside G.

This solution has a little flaw. If there are grids including more than one point, Alice and Bob cannot know the exact number of points inside G; but for solving rectangle intersection problem, this flaw does not affect obtaining correct conclusion.

3.3 Solution to intersection problem of rectangles

We only consider the cases in which the edges of rectangles are parallel to x axis or y axis. Assume that rectangles G_1 and G_2 are in the first quadrant, moreover, suppose that the coordinates of bottom left (resp., top right) vertex of $G_i(i=1,2)$ and area are $G_{il}(x_{il}, y_{il})$(resp., $G_{ir}(x_{ir}, y_{ir})$)and $A(G_i)$.

There are six basic cases when two rectangles intersect, which are shown in Fig. 3. The solution proposed in Ref. [13] does not work in cases (1) and (6). If $A(G_1)>A(G_2)$, then case (6) can be excluded. If $A(G_1)=A(G_2)$, cases (1) and (6) can both be excluded. In order to decrease computational complexity of the known solution in Ref. [13] and solve the problems of cases (1) and (6), we propose the following solution.

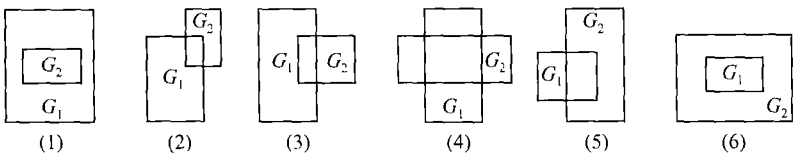

Fig. 3. Six basic cases when two rectangles intersect.

After case (6) is excluded, 5 cases left can be classified into 2 classes. The first class includes (1), (2) and (3), in which at least one vertex of G_2 is inside G_1; the second class includes (4) and (5), which does not have any obvious properties. While detailed analysis reveals the following properties. Fix the following points of rectangle G_2:

$$\left\{G_2(x_{2l}+u_x, y_{2l}), G_2(x_{2l}+2u_x, y_{2l}), \cdots, G_2\left(x_{2l}+\left[\frac{x_{2r}-x_{2l}}{u_x}\right]u_x, y_{2l}\right)\right\},$$

$$\left\{G_2(x_{2l}+u_x, y_{2r}), G_2(x_{2l}+2u_x, y_{2r}), \cdots, G_2\left(x_{2l}+\left[\frac{x_{2r}-x_{2l}}{u_x}\right]u_x, y_{2r}\right)\right\},$$

$$\left\{G_2(x_{2l}, y_{2l}+u_y), G_2(x_{2l}, y_{2l}+2u_y), \cdots, G_2\left(x_{2l}, y_{2l}+\left[\frac{y_{2r}-y_{2l}}{u_y}\right]u_y\right)\right\},$$

$$\left\{G_2(x_{2r}, y_{2l}+u_y), G_2(x_{2r}, y_{2l}+2u_y), \cdots, G_2\left(x_{2r}, y_{2l}+\left[\frac{y_{2r}-y_{2l}}{u_y}\right]u_y\right)\right\}.$$

The number of these points is $2\left[\frac{x_{2r}-x_{2l}}{u_x}\right]+2\left[\frac{y_{2r}-y_{2l}}{u_y}\right]$. In this class, there is at least one of above points inside G_1. Denote these points as a point set P, and 4 vertices of G_2 as set R. Cases (1)—(5) have a same property, that is there are at least one point of P inside G_1. So, we have the following solution.

Solution 5. Intersection problem of rectangles.

Inputs Rectangles G_1, G_2.

Output Whether or not G_1, G_2 intersect.

(i) Using Millionaire's protocol, Alice and Bob privately compare $A(G_1)$ and $A(G_2)$. Without loss of generality, suppose that $A(G_1)>A(G_2)$.

(ii) Alice and Bob use Solution 4 to determine $|T(R)\cap T(G_1)|$. If $|T(R)\cap T(G_1)|>0$, Alice and Bob conclude that G_1 intersects G_2 which terminates the solution.

(iii) Using the above mentioned method, Alice and Bob consist sets $T(P)$, $T(G_1)$. They privately compute $|T(P)\cap T(G_1)|$ with Solution 4. If $|T(P)\cap T(G_1)|>0$, Alice and Bob know that G_1, G_2 intersect; G_1, G_2 do not intersect otherwise.

If Alice and Bob are not concerned about the leaking of the area information of the rectangles, and they just want to know whether or not the two rectangles intersect without revealing their rectangles, then they can even choose that $p=1$. The solution will be much easier.

4 Privacy-preserving property

Solutions 4 and 5 are built on Solution 3, so the security of Solution 3 underlies the security of Solutions 4 and 5. Intuitively, this solution is private under the assumption that the public key encryption scheme is secure. We have the following conclusion:

Theorem 1. Solution 3 to set intersection problem, denoted by π, is private.

Proof. We prove this theorem by showing two simulators S_1, S_2 for simulating $view_1^{\pi}(A, B)$ and $view_2^{\pi}(A, B)$ such that formulae (3) and (4) hold.

Notice that during the execution of Solution 4, the *view* and the *output* of Alice and Bob are

$$view_1^{\pi}(A,B) = \{A, r^1, E_a(A), E_b(B), E_a(E_b(B)), E_b(E_a(A))\},$$

$$view_2^{\pi}(A,B) = \{B, r^2, E_b(B), E_a(A), E_b(E_a(A)), E_a(E_b(B))\},$$

$$output_1^{\pi}(A,B) = output_2^{\pi}(A,B) = f_1(A,B) = f_2(A,B) = |A \cap B|,$$

respectively. Suppose that $|A \cap B| = k$. S_1 proceeds as follows.

1) S_1 takes $(A, f_1(A,B)) = (A, |A \cap B|) = (A, k)$ as input, and chooses a commutative encryption scheme E and keys a and b' according to the outcome of its internal coin tosses $(r^1)'$. S_1 generates a set $B' = \{b_1', b_2', \cdots, b_n'\}$, in which there are k pairs of (i, j) such that $a_{i_1} = b_{j_1}', \cdots, a_{i_k} = b_{j_k}'$.

2) S_1 computes $E_a(A) = \{E_a(a_1), E_a(a_2), \cdots, E_a(a_m)\}$, and $E_{b'}(B') = \{E_{b'}(b_1'), E_{b'}(b_2'), \cdots, E_{b'}(b_n')\}$.

3) S_1 computes $E_a(E_{b'}(B'))$, $E_{b'}(E_a(A))$ and $|E_a(E_{b'}(B')) \cap E_{b'}(E_a(A))|$.

By the construction of B', there must exist k pairs of (i, j) such that $E_{b'}(E_a(a_{i_1})) = E_a(E_{b'}(b_{j_1}'), \cdots, E_{b'}(E_a(a_{i_k})) = E_a(E_{b'}(b_{j_k}')$, and thus it concludes that $|E_a(E_{b'}(B')) \cap E_{b'}(E_a(A))| = |A \cap B| = k$. Let $S_1(A, f_1(A,B)) = \{A, (r^1)', E_a(A), E_{b'}(B), E_a(E_{b'}(B')), E_{b'}(E_a(A))\}$, then

$$|S_1(A, f_1(A,B)), f_2(A,B)| = |A, (r^1)', E_a(A), E_{b'}(B), E_a(E_{b'}(B')), E_{b'}(E_a(A)), k|,$$

and

$$|view_1^{\pi}(A,B), output_2^{\pi}(A,B)| = |A, r^1, E_a(A), E_b(B), E_a(E_b(B)), E_b(E_a(A)), k|,$$

hence

$$|(S_1(A, f_1(A,B)), f_2(A,B))| \stackrel{c}{\equiv} |(view_1^{\pi}(A,B), output_2^{\pi}(A,B))|.$$

Similarly, we can show another simulator S_2 such that

$$|(f_1(A,B), S_2(A, f_2(A,B)))| \stackrel{c}{\equiv} |(output_1^{\pi}(A,B), view_2^{\pi}(A,B))|.$$

The theorem follows. □

5 Comparison

Computational complexity. For computational complexity, we consider only the most expensive modular exponentiation of computing $a^b \bmod n$. The other operations, such as hashing, single multiplication, division and extended-OR(XOR) are neglected for their little time-consuming.

To determine the inclusion problem of m points, Solution 1 needs to invoke the protocol for millionaires' problem $4m$ times. Let

$$(\max(x), \max(y)) = (\max(x_1, x_2, \cdots, x_k, x_l, x_r), \max(y_1, y_2, \cdots, y_k, y_l, y_r)),$$

and $l = \max([\max(x)], [\max(y)])$, then each invocation of millionaires' protocol needs $l + 1$ times modular exponentiation operation, l times modular operation, and at least $\frac{1}{2}l^2 + l$ times verification. For more details about protocol for millionaires' problem, see Ref. [18]. Each invocation of protocol for millionaires' problem needs $4m(l + 1)$, denoted by $O(m \cdot l)$ times modular exponentiation operation. If $m = 10$, $l = 200$, each invocation needs at least $40(200 + 1) = 8040$ modular exponentiations.

Solution 4 only needs $2p^2 + 2m$, where p is a constant determined by Alice, and is independent of the scale of problem, denoted by $O(p^2)$ modular exponentiations. If we choose $p = 5$, solution 4 only needs $2 \cdot 5^2 + 2 \cdot 10 = 120$ modular exponentiations.

To determine whether or not two rectangles intersect, Solution 2 needs to determine whether or not two edges intersect 16 times. To determine whether or not two edges intersect needs to invoke the protocol for millionaires' problem 4 times, which needs 201 modular exponentiations (suppose that $l = 200$). So Solution 2 needs $16 \times 4 \times 201 = 12864$ modular exponentiations.

There are p^2 points in set $T(G_1)$, and there are $\left(2\left[\frac{x_{2r} - x_{2l}}{u_{1r} - u_{1l}}\right] + 2\left[\frac{y_{2r} - y_{2l}}{u_{1r} - u_{1l}}\right]\right) + 4$ points in $T(P)$. Solution 5 needs only

$$2p^2 + 4\left[\frac{x_{2r} - x_{2l}}{u_x}\right] + 4\left[\frac{y_{2r} - y_{2l}}{u_y}\right] + 8 = 2p^2 + 4p\left(\left[\frac{x_{2r} - x_{2l}}{u_{1r} - u_{1l}}\right] + \left[\frac{y_{2r} - y_{2l}}{u_{1r} - u_{1l}}\right]\right) + 8$$

modular exponentiations to determine whether two rectangles intersect, where p is a constant on the option of Alice and is independent of the size of prob-

lem. Its computational complexity is $O(p^2)$. If $p=5$, and $\left(\left[\frac{x_{2r}-x_{2l}}{u_{1r}-u_{1l}}\right]+\left[\frac{y_{2r}-y_{2l}}{u_{1r}-u_{1l}}\right]\right)=10$, Solution 5 needs $2\cdot 5^2+4\times 5\times 10+8=256$ modular exponentiations. Adding the modular exponentiations of an invocation of millionaires' protocol, the solution total needs 8296 modular exponentiations. Furthermore, Solution 5 can reveal whether or not there is an inclusion relation between two rectangles, which cannot be revealed by Solution 2.

Communication complexity. Solution 1 needs to invoke protocol for millionaires' problem $4m$ times, each invocation needs 3 rounds of communications, hence the communication complexity of Solution 1 is $12m(O(m))$ rounds. Solution 4 needs $5(O(1))$ rounds of communication. Its communication complexity is much less than that of Solution 1.

Solution 2 needs to invoke protocol for millionaires' problem 64 times to determine whether or not two rectangles intersect, and each invocation needs 3 rounds of communication. So Solution 2 determines whether or not two rectangles intersect with communication complexity of 192 rounds. Solution 5 only needs 6 rounds of communication to determine whether or not two rectangles intersect. Table 2 shows the comparison of computational and communication complexities of Solutions 1, 2, 4, 5,

Table 2. Comparison of Solutions 1, 2, 4, 5 ($m=4$, $l=100$)

	Solution 1	Solution 4	Solution 2	Solution 5
Computational complexity	8040	120	12864	8296
Communication complexity	$4m$	5	192	6

and demonstrates that new solutions surpass known solutions in performance of both computational complexity and communication complexity. Moreover, Solution 4 can solve the SMC problem of point-inclusion of parallelogram, and in this case, a non-rectangular coordinate system needs to be established.

6 Conclusion

We have developed a solution to the problem of set intersection that has not been investigated before. Based on this solution, the SMC solutions to problems of points inclusion and intersection of rectangles are further proposed, and their privacy-preserving properties are proved with simulation paradigm. In the future study, we will investigate SMC solutions to the points inclusion problem that can overcome the flaw in our Solution 4. For the limitation of pages and our knowledge, the property analysis is not deep enough.

References

1 Cachin C. and Camenisch J. LNCS 3027: Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2—6, 2004, 1—55, 419—473.

2 Yao A. Protocols for secure computations. In: Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science. Chicago, USA, Nov.3—5, 1982, 160—164.

3 Goldreich O., Micali S. and Wigderson A. How to play ANY mental game. In: Proceedings of the 19th Annual ACM Conference on Theory of Computing. New York, USA, May 28—30, 1987, 218—229.

4 Goldreich O. Secure multi-party computation (working draft). Available at http://www.wisdom.weizmann.ac.il/%7Eoded/pp.html [2005-06-01].

5 Goldwasser S. Multi-party computations: Past and present. In: Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing. Santa Barbara, California, USA, August 21—24, 1997, 1—6.

6 Fagin R., Naor M. and Winkler P. Comparing information without leaking it. Communications of the ACM, 1996, 39(5): 77—85.

7 Lindell Y. and Pinkas B. Privacy preserving data mining. Journal of Cryptology, 2002, 15(3): 177—206.

8 Du W.L. and Atallah M.J. Privacy-preserving cooperative scientific computations. In: Proceedings of the 14th IEEE Computer Security Workshop. Nova Scotia, Canada, June 11—13, 2001, 273—282.

9 Du W.L. and Atallah M.J. Protocol for secure remote database access with approximate matching. In the 7th ACM Conference on Computer and Communication Security, The First Workshop on Security and Privacy in E-Commerce; Athens, Greece, Nov. 2000, available at http://www.cis.syr.edu/ wedu/Research/publication.html [2005-02-25]

10 Cachin C. Efficient private bidding and auction with an obvious third party. In: Proceedings of the 6th ACM Conference on Computer and Communication Security. Singapore, Nov.1—4, 1999, 120—127.

11 Du W.L., Han Y.H.S. and Chen S.G. Privacy-preserving multivariate statistical analysis: linear regression and classification. In: Proceedings of 2004 SIAM International Conference on Data Mining. Nova Scotia, Canada, June 11—13, 2001, 273—282.

12 Du W.L. and Atallah M.J. Secure multi-party computation problems and their applications: A review and open problems. In: Proceedings of New Security Paradigms Workshop 2001. Cloudcroft, New Mexico, USA, Sep.11—13, 2001, 11—20.

13 Atallah, J. and Du, W.L. Secure multi-Party computational geometry. In: Seventh International Workshop on Algorithms and Data Structures (WADS2001). Providence, Rhode Island, USA, Aug. 8—10, 2001, 165—179.

14 LI S.D. and DAI Y.Q. Secure two-party computational geometry. Journal of Computer Science and Technology, 2005, 20(2): 258—263.

15 Zhang L.A. Introduction to Computability and Computational Complexity (in Chinese). Beijing: Peking University Press, 2003, 42—43.