Research on the Problem of Privacy-Preserving Closest Pair

Cheng Chao1, Luo Yonglong*1 and Cheng Wenjuan2
1Department of Computer Science, Anhui Normal University, Wuhu, China, 241003
2School of Computer & Information, Hefei University of Technology, Hefei, China, 230009
Email: ylluo@ustc.edu.cn

Abstract—The problem of closest pair is a basic problem of computational geometry. This problem was studied while considering the privacy and a protocol was proposed to solve it only considering the level of coordinate axis. The protocol ranked the numbers of two parties firstly. After that each party got the orders of his own numbers. Then the two participants used the protocol of private comparison protocol to get the closet pair according to the relative positions of numbers. We analyzed the correctness, the complexity and compared with previous. This protocol will leak some information, but don’t need the third party. This enhances the security of protocol.

Index Terms—Computational geometry; Secure two-party computation; Closest pair; Private comparison

I. INTRODUCTION

With the increasing progress of human society, the protection of the private information has become increasingly subject to public attention. For example, two countries respectively set up \( n \) intelligent stations of their own in the third country. Now they want to create a network among their sites to facilitate mutual cooperation. They also want to establish the shortest route which can not only save money but also shrink the possibility of information leakage. However, the two countries are unwilling to disclose the location of other intelligences apart from the most closet two sites. How to protect their own private information in cooperatively calculating is an urgent problem needs to be resolved.

The above-mentioned problem is a special kind of secure multi-party computation (SMC) problem [1]. Some special problems need to be solved by using special methods to get high efficiency [2]. A number of specific issues of the SMC has received a great deal of research results [5 ~ 7] [10~12]. We consider the two parties in a secure computing environment for the above-mentioned problem. The calculation of the two sides of the closet pair also belongs to privacy-preserving computational geometry (PPGG) problem.

Atallah M. J [3] first proposed the concept of PPGG and introduced some computational geometry problems such as point-inclusion, intersect-determination of polygons, the closest pair and the convex hull problem. It had given some initial solutions. Du Wenliang [4] designed a protocol to judge whether the point is in a circle or in an ellipse. It also introduced the problem of closet pair but it didn’t give a specific agreement. After that, many scholars studied on PPGG and designed many efficient protocols. But the problem of closest pair has been seldom studied. A protocol was proposed in [8] to solve this problem but it needed the help of the third party. But in reality, the entirely credible third party doesn’t exit. At present, the research data is scarce, so this paper mainly studied this problem and proposed a protocol to solve it.

In this paper, we assume the two parties are semi-honest. They will strictly enforce the agreement. They will not exist when the protocol is implementing or insert malicious information. But they may collect the information whatever they can get and infer the other side’s information.

II. PRELIMINARIES
A. Definitions

Definition 1. Computation Model

Secure two-party computation is a kind of secure distributed computation protocol. In this protocol, Alice and Bob respectively have their own inputs \(x\) and \(y\). They want to compute \(f(x,y)\) cooperatively. But they don’t want to leak input of their own. Generally speaking, Alice and Bob first disguise their inputs. So their inputs are transferred into new data. Then they compute on the new data. At last, the middle result of computation is reverted to the result of original problem.

Definition 2. Privacy-Preserving Closest Pair [8]

Alice has \(n\) points in a certain plane, \(A=(a_1,a_2,...,a_n)\), \(a_i=(x_i,y_i), i=1,2,...,n\). Bob has \(m\) points in this certain plane, \(B=(b_1,b_2,...,b_m), b_j=(x_j,y_j), j=1,2,...,m\). They want to get the closest pair among their points. But they don’t want to leak their own input.

In this paper, we only consider the points on \(x\) axes. So the problem of secure two-party ranking can be described as follows: Alice has \(n\) points \((a_1,a_2,...,a_n)\), \(a_1<a_2<...<a_n\). Bob has \(m\) points \((b_1,b_2,...,b_m)\), \(b_1<b_2<...<b_m\). They want to compute the closest pair among their points cooperatively.

B. Private Comparison Protocol

Secret comparison protocol is one of SMC’s basic operations. This protocol plays very important role in designing high efficiency SMC protocol. Scientists have made a lot of results [13~16] after many year’s research. A high efficient private comparison protocol DCP based on cross product protocol was designed in [9]. This protocol needs 3 rounds, 8 encryptions and 4 decryptions.

III. Secure Two-party Ranking Protocol

Secure two-party ranking problem can be described as follows: Alice has \(n\) data \((a_1,a_2,...,a_n), a_1<a_2<...<a_n\). Bob has \(m\) data \((b_1,b_2,...,b_m), b_1<b_2<...<b_m\). They want to rank these \(m+n\) data and get the order of each data in the ascending sequence without leaking the inputs.

In order to get the order of each data, we need to compare each data with other data and count how many data is smaller than it. So we need to compare each \(a_i\) with \(b_j (j=1,2,...,m)\). \(r_a\) and \(r_b\) respectively count how many data is smaller than \(a_i\) and \(b_j\).

Based on the above-mentioned, the protocol is described as follows:

(security) In Step 1 Bob sends the number of data to Alice. And when Step 2 was finished, Alice and Bob will get the order of their own data. The same time Alice will also know the range of some Bob’s data when the orders are not continuous. So this protocol will leak the number of data of Bob and the range.

| TABLE 1 |
| Secure Two-party Ranking Protocol |

**Protocol 1 STRP (Secure Two-party Ranking Protocol)**

- **Input:** Alice inputs \((a_1,a_2,...,a_n), a_1<a_2<...<a_n\), and Bob inputs \((b_1,b_2,...,b_m), b_1<b_2<...<b_m\).
- **Output:** Alice gets \((r_{a1},r_{a2},...,r_{an})\) and Bob gets \((r_{b1},r_{b2},...,r_{bm})\).
- **Step 1:** \(r_a=0, r_b=0\), Bob sends \(m\) to Alice.
- **Step 2:** \(i\leftarrow 1;\)
  - While \(i\leq n\) do
    - \(j\leftarrow 1;\)
    - While \(j\leq m\) do
      - Alice invokes DCP \((a_i,b_j)\)
      - If \(a_i<b_j\) then \(r_{aj}++;\)
    - \(r_a=r_a++;\)
  - \(r_a=r_a++;\)
- **Step 3:** \(j\leftarrow 1;\)
  - While \(j\leq m\) do
    - \(r_b=r_{bj}++;\)

**Table 1 (complexity)** This protocol mainly invokes secret comparison protocol DCP \(n*m\) times. So the round complexity counts on the DCP which this paper used. So protocol 1 needs \(8mn\) encryptions and \(4mn\) decryptions. The round complexity is \(3mn+1\).

IV. The Closest Pair Protocol

We first invoke protocol 1 STRP to rank these \(m+n\) data. So the participants get the orders of their own data in the ascending sequence. Then Alice examines whether there has data of Bob before or behind \(a_i\) in turn. If there has some one, Alice sends the order of this data to Bob and Bob sends data which has been changed to Alice. Alice gets the smallest distance at present according to...
invoke the secret comparison DCP and records the order of relative pair. After Alice accomplish the above thing \( n \) times two parties can get the closest pair’s orders.

Next we take an example to illustrate the implementation of this protocol:

Assume Alice and Bob respectively have ten points. The numbers of these points are listed in TABLE II.

The process of performing the protocol is as follows: (①②③④ is one loop process)

![Fig.1 The performing process of PPCPP](image_url)

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
<th>Random numbers</th>
<th>New Bob’s data</th>
<th>the orders of Alice after rank</th>
<th>The orders of Bob after rank</th>
<th>New Bob’s data after rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 = 1 )</td>
<td>( b_1 = 5 )</td>
<td>( v_1 )</td>
<td>( 5 + v_1 )</td>
<td>( R_{a1} = 1 )</td>
<td>( R_{b1} = 2 )</td>
<td>( 5 + V_{b1} )</td>
</tr>
<tr>
<td>( a_2 = 9 )</td>
<td>( b_2 = 12 )</td>
<td>( v_2 )</td>
<td>( 12 + v_2 )</td>
<td>( R_{a2} = 3 )</td>
<td>( R_{b2} = 4 )</td>
<td>( 12 + V_{b2} )</td>
</tr>
<tr>
<td>( a_3 = 22 )</td>
<td>( b_3 = 18 )</td>
<td>( v_3 )</td>
<td>( 18 + v_3 )</td>
<td>( R_{a3} = 6 )</td>
<td>( R_{b3} = 5 )</td>
<td>( 18 + V_{b3} )</td>
</tr>
<tr>
<td>( a_4 = 45 )</td>
<td>( b_4 = 30 )</td>
<td>( v_4 )</td>
<td>( 30 + v_4 )</td>
<td>( R_{a4} = 9 )</td>
<td>( R_{b4} = 7 )</td>
<td>( 30 + V_{b4} )</td>
</tr>
<tr>
<td>( a_5 = 69 )</td>
<td>( b_5 = 36 )</td>
<td>( v_5 )</td>
<td>( 36 + v_5 )</td>
<td>( R_{a5} = 12 )</td>
<td>( R_{b5} = 8 )</td>
<td>( 36 + V_{b5} )</td>
</tr>
<tr>
<td>( a_6 = 80 )</td>
<td>( b_6 = 49 )</td>
<td>( v_6 )</td>
<td>( 49 + v_6 )</td>
<td>( R_{a6} = 13 )</td>
<td>( R_{b6} = 10 )</td>
<td>( 49 + V_{b6} )</td>
</tr>
<tr>
<td>( a_7 = 100 )</td>
<td>( b_7 = 68 )</td>
<td>( v_7 )</td>
<td>( 68 + v_7 )</td>
<td>( R_{a7} = 15 )</td>
<td>( R_{b7} = 11 )</td>
<td>( 68 + V_{b7} )</td>
</tr>
<tr>
<td>( a_8 = 120 )</td>
<td>( b_8 = 89 )</td>
<td>( v_8 )</td>
<td>( 89 + v_8 )</td>
<td>( R_{a8} = 16 )</td>
<td>( R_{b8} = 14 )</td>
<td>( 89 + V_{b8} )</td>
</tr>
</tbody>
</table>

② Alice computes \( temp = d - (5 + V_{b1} - 1) = 1.0*10^{10} \) and compare it with Bob’s \( V_{b1} \) . temp > \( V_{b1} \) . So \( P_a = R_{a1}, P_b = R_{b1} \)
According to the above-mentioned idea, the protocol is described as follows:

**Security**  This protocol invokes protocol STRP in Step 1. Protocol STPR will leak the number of Bob’s data and the range of Bob’s data. In Step 2, Bob adds his own numbers with random numbers so it will not leak the concrete data. In Step 3, we mainly invoke DCP. This protocol is secure. At the same time Bob has added his own numbers with random numbers so in Step 3 it will not leak information. So PPCPPP is secure.

**Complexity**  In Step 1, STRP is invoked one time. The round complexity of protocol STRP is $3mn + 1$. In Step 3, protocol DCP is invoked $\min\{m,n\}$ times at most. The number of communication is $\min\{m,n\}$ at most.

### Table III

Privacy-Preserving Closest-Pair of Points Protocol

<table>
<thead>
<tr>
<th>Protocol 2 PPCPPP (Privacy-Preserving Closest-Pair of Points Protocol)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Alice inputs $(a_1, a_2, \ldots, a_n)$, $a_i &lt; a_{i+1}$ and. Bob input $(b_1, b_2, \ldots, b_m)$, $b_i &lt; b_{i+1} \ldots &lt; b_m$ (a, b \neq \ldots \neq a, b \neq \ldots \neq b)</td>
</tr>
<tr>
<td><strong>Output:</strong> $P_a, P_b$. ($P_a$ represents the order of Alice’ point in the closet pair and $P_b$ represents the order of Bob’s point in the closet pair)</td>
</tr>
<tr>
<td><strong>Step 1:</strong> Alice does $d \leftarrow 1.0 \times 10^{10}$, $P_a \leftarrow 0, P_b \leftarrow 0$. STRP $(a_1, a_2, \ldots, a_n; b_1, b_2, \ldots, b_m)$; Bob does $V \leftarrow 0$.</td>
</tr>
<tr>
<td><strong>Step 2:</strong> Bob chooses $m$ random numbers $v_1, v_2, \ldots, v_m$, so Bob’s numbers become $b_1 + v_1, b_2 + v_2, \ldots, b_m + v_m$.</td>
</tr>
</tbody>
</table>
| **Step 3:** $i \leftarrow 1$
| While $i \leq n$ do |
| Alice examines whether there is a Bob’s data before $a_i$ |
| If there has a data, then Alice sends $F_a \leftarrow -1$ to Bob and Bob sends $d_{raw} - 1 + V_{raw} - 1$ to Alice; // $F_a$ represents the order of $a_i$ in $m+n$ data |
| Alice computes temp $= d(a_i (d_{raw} - 1 + V_{raw} - 1))$ and invokes DCP (temp, $V + V_{raw} - 1$); |
| if temp $> V + V_{raw} - 1$
| then $d = a_i (d_{raw} - 1 + V_{raw} - 1), P_a = F_a, P_b = F_a - 1$; |
| Alice sends $-1$ to Bob, Bob do $V = V_{raw} - 1$. |
| Alice examines whether there is a Bob’s data after $a_i$ |
| If there has data, then Alice sends $F_a \leftarrow +1$ to Bob then Bob sends $d_{raw} + 1 + V_{raw} + 1$ to Alice; |
| Alice computes temp $= d((d_{raw} + 1 + V_{raw} + 1) - a_i)$ and invokes DCP (temp, $V - V_{raw} + 1$); |
| if temp $> V - V_{raw} - 1$
| then $d = (d_{raw} + 1 + V_{raw} + 1) - a_i, P_a = F_a, P_b = F_a + 1$; |
| Alice sends $1$ to Bob, Bob do $V = V_{raw} - 1$. |
| Step 4: Alice sends $p_a$ and $p_b$ to Bob |
| }//end of Protocol 2. |

in Step 3, the round complexity is $3\min\{m,n\}$ at most. In Step 4, the number of communication is 1. So the round complexity of PPCPPP is $3mn + 3\min\{m,n\}$ at most.

This protocol invokes protocol STRP in Step 1. STRP needs $8mn$ encryptions and $4mn$ decryptions. In Step 3, protocol DCP is invoked $\min\{m,n\}$ times at most. So Step 3 needs $8\min\{m,n\}$ encryptions and $4\min\{m,n\}$ decryptions. So PPCPPP needs $8mn + 8\min\{m,n\}$ encryptions and $4mn + 4\min\{m,n\}$ decryptions.

V. Conclusion

In this paper, we mainly discussed the problem of
closest pair and proposed a privacy-preserving protocol when the points are all on x axes. We first ranked all of the points then invoked secret comparison protocol to get the orders of closest pair. It doesn’t need the third party as previous paper. It promotes the security of the protocol.

REFERENCES


