doi:10.3772/j.issn.1006-6748.2015.04.014

Secure planar convex hull protocol for large-scaled point sets in semi-honest model[1]

Sun Maohua(孙茂华)[2]*, Zhu Hongliang**, Li Qi**
(*Information School, Capital University of Economics and Business, Beijing 100070, P. R. China)
(**School of Computer, Beijing University of Posts and Telecommunications, Beijing 100876, P. R. China)

Abstract

Efficiency and scalability are still the bottleneck for secure multi-party computation geometry (SMCG). In this work a secure planar convex hull (SPCH) protocol for large-scaled point sets in semi-honest model has been proposed efficiently to solve the above problems. Firstly, a novel privacy-preserving point-inclusion (PPPI) protocol is designed based on the classic homomorphic encryption and secure cross product protocol, and it is demonstrated that the complexity of PPPI protocol is independent of the vertex size of the input convex hull. And then on the basis of the novel PPPI protocol, an effective SPCH protocol is presented. Analysis shows that this SPCH protocol has a good performance for large-scaled point sets compared with previous solutions. Moreover, analysis finds that the complexity of our SPCH protocol relies on the size of the points on the outermost layer of the input point sets only.

Key words: secure multi-party computation, secure multi-party computational geometry (SMCG), secure planar convex hull protocol (SPCH), privacy-preserving point-inclusion protocol (PPPI), semi-honest model

0 Introduction

With the rapid expansion of smart phone and tablet markets, location based service (LBS) becomes more and more popular. LBS uses information on the geographical position of mobile devices and it has a large number of users in social network. Although the LBS applications have significant benefits, some of them reveal privacy of users which may attract security risk. For example, revealing that a master is not at home may be a risk if that information is discovered by thieves. Several technical approaches exist to protect privacy in LBS. One way that formalizes privacy requirements for LBS is done by secure multi-party computation (SMC) protocols. SMC enables mutually suspicious parties to compute a joint function on their private input in a manner that at the end parties know nothing useful and except the result. SMC was introduced by Yao[1], further extended by Goldreich, et al.[2] and other researchers. A series of works have considered the design of more efficient SMC protocols in all kinds of application area.

As a special application area of SMC, SMCG aims to protect the input data in a geometric function computation. SMCG was proposed by Atallah, et al.[3] in 2001. Since its instruction, SMCG has been extensively studied because it can be wildly used in many applications e.g. the privacy issue in LBS applications. In this paper, the SMCG line is followed to protect the privacy information in the geometry function computation, which mainly focuses on two classical SMCG problems, namely PPPI and SPCH.

PPPI problem: Alice has a convex hull $P = \{p_0, p_1, \cdots, p_n\}$, where p_0 is the bottom-left point and other points are sorted anticlockwise. Point p_i can be described as $p_i = (x_i, y_i)$ for $0 \leqslant i \leqslant n$. Bob has a point $q = (m, n)$. Alice and Bob wish to estimate whether q locates in P without revealing to each other anything intended. Concretely, Alice cannot get the value of q and Bob cannot get anything about P.

SPCH problem: Alice has a point set A, Bob has a point set B. Alice and Bob wish to jointly find the convex hull for these $A \cup B$ points. However, neither Alice nor Bob wish to disclose any more information to each other than what can be derived from the result.

[1] Supported by the Young Scientists Program of CUEB (No. 2014XJQ016, 00791462722337), National Natural Science Foundation of China (No. 61302087), Young Scientific Research Starting Foundation of CUEB and Improve Scientific Research Foundation of Beijing Education.

[2] To whom correspondence should be addressed. E-mail: starjingxiang@sina.com

Received on Oct. 23, 2014

PPPI and SPCH are the classical and best studied problems in SMCG. Since their introduction, several techniques have been used to realize PPPI and SPCH protocols. Solutions to PPPI and SPCH include:

Circuit-Based Solutions A naive solution to PPPI or SPCH is using the circuit-based SMC, which is a generic secure computation protocol that allows the secure evaluation of arbitrary functions, expressed as Arithmetic or Boolean circuits. The most classical circuit-based SMC protocol called GMW was introduced by Goldreich, et al[2]. Their solution allows evaluation of arbitrary functions between two parties in the semi-honest model. Improved GMW protocols are proposed in Refs[4-6] recently. Although any polynomial-time multi-party computation can be done by circuit-based solutions, this generic approach is sometimes impractical due to its complexity. Recent research on PPPI and SPCH focuses on finding more efficient privacy-preserving custom algorithms.

Custom PPPI Solutions The first custom PPPI protocol was proposed by Atallah, et al.[3]. Li, et al.[7] proposed an approximate secure multi-party graph inclusion protocol based on Monte Carlo approach and Cantor encoding. Luo, et al.[8] proposed a PPPI protocol to determine whether a point was inside a convex polygon based on the secure cross product protocol; the computational complexity of their protocol is $O(tnlogn + tn^2)$ where t denotes the vertex size in the convex hull. Based on additive homomorphic encryption, Liu et al. developed a privacy-preserving point-line relation determination protocol[9] and the computational complexity of their protocol was $O(tlogn + tn^2)$. As evaluating the positional relationship between a point and every edge in the convex hull is an intuition to address the PPPI problem, it is found that the computation and communication complexity of the previous protocols using this intuition rely on the vertex size of the input polygon. When the vertex size of the input polygon is large, the previous protocols are less efficient.

Custom SPCH Solutions Lu, et al.[10] provided a SPCH scheme based on Graham algorithm with the computational complexity $O(rN^3 + N\log N)$ where N denotes the size of the input points set. Based on the Euclid-distance Measure scheme, Wang, et al.[11] presented an approximate solution to the SPCH problem with computational complexity $O(N^3 + N^2\log N)$. Hans, et al.[12] constructed an improved SPCH protocol with complexity $O(N\log N)$, unfortunately it was shown in secure in Ref.[13]. Wang, et al. presented a SMC protocol for two party convex hull construction with quadratic communication complexity[14]. In addition, Li, et al.[15] presented a quadratic SMC protocol for approximately three-dimensional convex hulls. Like the existing custom PPPI protocols, it is found find that the complexity of these SPCH protocols relies on the size of input point sets. When the input point sets are large, the protocols are less efficient.

In this work, two effective SMCG protocols including a PPPI protocol and a SPCH protocol are proposed in the semi-honest model. In the PPPI protocol, parties do not need to determine the positional relationship between the input point and every edge in the convex hull; they only need to determine the positional relationship between the input point and one edge which is called the nearest edge. When the size of the input convex hull is large, the proposed protocol is more efficient. The SPCH protocol is designed based on the incremental method. As the parties compute the convex hull of their input point sets respectively in the preprocessing stage, the complexity of the SPCH protocol is only related to the size of the points in the outermost layer. Compared with the previous SPCH protocols, the proposed protocol is faster when the input point sets are large scaled.

Analysis shows that the new protocols are secure in the semi-honest mode. The semi-honest model assumes that parties follow the protocol correctly, and there is no efficient adversary that can extract more information from the transcript of the protocol execution than what is revealed from the output. The scheme secure is not given against malicious adversaries for three reasons. First, the semi-honest model is secure enough when it is hard to tamper the protocol software. This is just right fit for our settings. Second, most of the existing SMCG protocols[7-15] and many advanced SMC protocols[3-18] are proposed in the semi-honest model. The choice of semi-honest model follows the previous work. Third, it is an independent object of interest in SMC to convert protocols in semi-honest model to malicious model[19-28]. These conversions can be used to change the proposed protocols' secure model if necessary. The security analysis of our protocols uses the definition of the security in semi-honest model given by Goldreich[29]. This analysis method is widely used in SMC protocols.

The paper is organized as follows. Section 1 described the preliminaries. Section 2 depicts the proposed PPPI protocol. The SPCH protocol is presented in Section 3. Conclusion is drawn in Section 4.

1 Preliminaries

1.1 Millionaire protocol

In 1980's, Yao[1] proposed a constant-round protocol called Millionaire Protocol (MP) to securely compare two private input data owned by the two participants separately. In the proposed protocol, MP is used as the underlying block. In the rest of the paper, MP is used (x, y) to denote the Millionaire Protocol with the input data x and y. The return of MP(x, y) is defined as

$$Return = \begin{cases} 1, & x > y \\ 0, & x = y \\ -1, & x < y \end{cases}$$

1.2 Homomorphic encryption

Homomorphic encryption allows us to compute in the cipher text as we do in the plain text. For additive homomorphic encryption, the additive cipher text $a \oplus b$ can be got easily by the cipher text $\bar{a}$ and $\bar{b}$. An efficient and widely used additive homomorphic encryption scheme is Paillier's cryptosystem[31]. The security of Paillier's cryptosystem relies on the decisional composite residuosity assumption.

1.3 Secure cross product protocol

Cross product operation is a fundamental primitive in computational geometry. In secure cross product setting, Alice has a private point while Bob has a private line segment p_1p_2. Alice and Bob wish to compute the sign of cross product $p_1p_0 \times p_1p_2$ without revealing their private information. Secure cross product problem and the solution were first introduced by Luo, et al.[30]. In their protocol, Alice and Bob separately generate a four-dimensional vector locally, then they call the secure scalar product protocol, at last they get the sign of cross product through a carefully designed computation based on the scalar product. A random integer is introduced in the protocol to protect the private input information of Alice and Bob.

In Luo's protocol, Alice and Bob compute the sign of $p_1p_0 \times p_1p_2$ when Alice has the point p_0 and Bob has line p_1p_2. The sign of cross product when Alice has points p_0, p_1 and Bob has point p_2 is needed to be evaluated securely in many settings. That is to say, the points may belong to different owners. According to different point owners, different secure cross product protocols should be used. It is found that these protocols have the same principle to the one in Ref. [30], but they have different intermediate variables. The variables used in different settings are shown in Appendix B.

In the rest of the paper, SCP_PL $(p_0, \overline{p_1p_2})$ is used to denote secure cross product protocol parameterized by $(p_0, \overline{p_1p_2})$ for a point and a line.

2 PPPI Protocol

2.1 Protocol design

The definition of PPPI problem is described in the Introduction. To avoid evaluating the positional relationship between the q and every edge in P, the following idea is used: in the planar field, three cases exist in the relationship between a convex and a point:

(1) If $n < y_0$, then $q \notin P$. Such as point A with the convex hull $P = \{p_0, p_1, \cdots, p_4\}$ in Fig. 1.

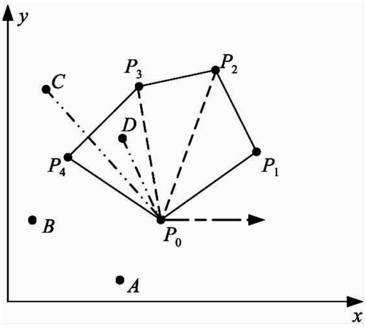

Fig. 1 Relationship between a point and a convex

(2) If $n = y_0$, it is needed to estimate whether $q = p_0$. If $q = p_0$, then point q is the bottom-left point of P. If $q \neq p_0$, then $q \notin P$. Such as point B with the convex hull $P = \{p_0, p_1, \cdots, p_4\}$ in Fig. 1.

(3) If $n > y_0$, it is assumed that q locates between p_0p_{i-1} and${}_{p0}p_i$. Now, it is needed to estimate the relationship between q and $p_{i-1}p_i$. **the nearest edge** in the convex hull is called P for point q. If q locates at the left side of $p_{i-1}p_i$, then $q \in P$. Such as point D with the convex hull in Fig. 1. When q locates at the right side of $p_{i-1}p_i$, $q \notin P$, such as point C with convex hull $P = \{p_0, p_1, \cdots, p_4\}$ in Fig. 1.

Based on the idea above, firstly MP and additive homomorphic encryption scheme are used to find the nearest edge in P for q; Secondly, SCP_PL is used to determine the relationship between q and the nearest edge which reflect the relationship between q and P. The PPPI protocol is described in Table 1. In the rest of our paper, $\boldsymbol{A}_i$^$\boldsymbol{B}_i$ denotes that Alice and Bob compute the function jointly in step i; $\boldsymbol{A}_i | \boldsymbol{B}_i$ denotes that Alice and Bob compute the function seperately in step i; $\boldsymbol{A}_i$ denotes that Alice computes the function alone in step i.

Table 1 PPPI Protocol

```
Algorithm 1 PPPI ($q$, $P$)
Input: A point $q$ and a convex hull $P$.
Output: True if $q \in P$, False otherwise.
Preprocessing:
(1) Alice calls the $KeyGen_{\varepsilon}$ of Paillier's cryptosystem to generate the key pairs ($pk$, $sk$). $pk$ denotes the public key and $sk$ denotes the private key.
(2) Alice computes the set $A = \{a_0, a_1, \cdots, a_n, a_{n+1}\}$. $a_i$ is the slope of $\overline{p_0 p_i}$.
Processing:
$A_1$^$B_1$:
  $t = MP(n, y_0)$;
  if ($t == -1$)
     Return False; //when $n < y_0$, $q \notin P$
  else if( $t == 0$ )
  { if( $MP(m, x_0) == 0$);
       Return True; //$q = p_0$
  else
       Return False; // $q \notin P$ }
  else
  go to step 2;
$A_2$:
  Compute: $E(-x_0)$, $E(-y_0)$;
  Send(Alice→Bob, $E(-x_0)$, $E(-y_0)$);
$B_3$:
  Generate: $r$ // Bob randomly chooses an integer $r$.
  Compute: $[E(-x_0)]^r E(rm) = E(rm - rx_0)$;
  Compute: $[E(-y_0)]^r E(rn) = E(rn - ry_0)$;
  Send(Bob→Alice, $E(rm - rx_0)$, $E(rn - ry_0)$);
$A_4$^ $B_4$:
  Alice Computes: $k = \frac{D(E(rn - ry_0))}{D(E(rm - rx_0))}$;
  if ($a_{i-1} < \alpha < a_i$)
     returns (SCP _ PL ($q, \overline{p_l p_{l+1}}$) );
  else if ($a = a_i$)
  {   if ($MP(m, x_{i-1}) \times MP(x_i, m) \times MP(n, y_{i-1}) \times MP(y_i, nm) == 1$)
         return True;
      else
         return False; }
```

2.2 Security analysis

Theorem 1 Assuming the underlying millionaire protocol, homomorphic encryption scheme and SCP _ PL protocol are secure in semi-honest model, the proposed PPPI protocol securely evaluates the relationship between the input point and convex hull in the presence of semi-honest adversaries.

Proof: As the underlying millionaire protocol is secure in semi-honest model, our protocol is secure if $n \leqslant y_0$. If $n > y_0$, the security of our protocol is analyzed as following:

Alice inputs the point set $P = \{p_0, p_1, \cdots, p_n\}$, Bob inputs the point set $q = (m, n)$. The protocol outputs $output^{\Pi} = t$ where $t = True$ refers to $q \in P$ and $t = False$ refers to $q \notin P$.

Bob's view The view of Bob during the execution of π is

$view_{Bob}^{\Pi}(P, q) = (q, pk, r, E(-x_0), E(-y_0), E(rm - rx_0), E(rn - ry_0), view_{Bob}^{MP}, view_{Bob}^{SCP_{PL}}, t)$.

A simulator is construsted to receive Bob's private input, output and simulate the view of Bob in the protocol. Firstly, generates random integers r', x'_0 and y'_0. Secondly, Sim_B encrypts x'_0 and y'_0 with the public key pk, the ciphertext denoted as $E(x'_0)$ and $E(y'_0)$. Thirdly, Sim_B computes the following cipher text with the public key pk:

$$E'(r'm - r'x_0) = [E(-x_0)]^{r'} E(r'm)$$

$$E'(r'n - r'y_0) = [E(-y_0)]^{r'} E(r'n)$$

Then, Sim_B simulates following step 4 of the protocol. At this step, Sim_B gets $view_{Sim_B}^{MP}$ or $view_{Sim_B}^{SCP\_PL}$. Finally, Sim_B outputs the simulated view:

$view_{Sim_B}^{\Pi}(P, q) = (q, pk, r', E(x'_0), E(y'_0), E'(r'm - r'x_0), E'(r'n - r'y_0), view_{Sim_B}^{MP}, view_{Sim_B}^{SCP_{PL}}, t)$.

As the underlying homomorphic encryption scheme, millionaire protocol and SCP _ PL protocol are secure in semi-honest model, the conclusion is that

$$E(x'_0) \overset{C}{\equiv} E(x_0), \; E(y'_0) \overset{C}{\equiv} E(y_0)$$

$$E'(r'm - r'x_0) \overset{C}{\equiv} E(rm - rx_0)$$

$$E'(r'n - r'y_0) \overset{C}{\equiv} E(rn - ry_0)$$

$$view_{Bob}^{MP} \overset{C}{\equiv} view_{Sim_B}^{MP}, \; view_{Bob}^{SCP\_PL} \overset{C}{\equiv} view_{Sim_B}^{SCP\_PL}$$

Thus it is concluded the simulated view is distinguishable from the real view:

$$view_{Bob}^{\Pi}(P, q) \overset{C}{\equiv} view_{Sim_B}^{\Pi}(P, q)$$

The same simulator can be created for Alice.

2.3 Algorithm complexity

In SMCG, many protocols use underlying building blocks without specifying the protocol detail. Notations defined in Table 6 is used to compare the complexity. To compare the computational complexity concretely, the underlying protocols are referred as follows: Paillier's cryptosystem is used as the additive homomorphic encryption scheme; the protocol presented in Ref. [18] is used to solve the millionaire problem; Luo's solution[30] is used as the underlying SCP _ PL protocol.

The communicational complexity is concluded in Table 2 (t is the vertex size in the convex hull). It shows that the complexity of protocols in Refs[8,9] relies on the vertex size of the convex hull while the proposed protocol is not. When the number of the vertex in the convex hull is huge, our protocol is much better than the previous protocols.

Table 2 Algorithm complexity

	Computational complexity	Communicational complexity	Computational complexity (Instantiation)
the proposed protocol	$rT_h + 2T_d + 6T_a + T_c$	$5C_a + C_c + 4$	$O(r\log n + n^2)$
Ref. [8]	$4T_a + t\ T_c$	$4C_a + t\ C_c$	$O(tn\log n + tn^2)$
Ref. [9]	$t(4T_h + 3T_d + T_a)$	$t(4C_h + 3C_d + C_a)$	$O(t\log n + tn^2)$

3 Secure planar convex hull protocol

3.1 Protocol design

To reduce the interactive computation, Alice and Bob evaluate the convex hull of his/her point set locally at the preprocessing phase. This method is also used in the previous works[32,33]. $M = \{m_1, \cdots, m_k\}$ and $N = \{n_1, n_2, \cdots, n_t\}$ stand for the vertex set of convex hull of Alice's and Bob's point set respectively. Now, Alice and Bob can use incremental method to evaluate the convex hull of set M and N.

When evaluating the convex hull of point p and convex hull P, there are two cases as shown in Fig. 2:

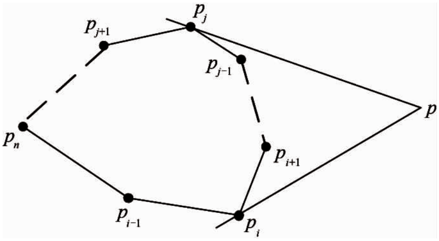

Fig. 2 A convex hull and a point outside

(1) If p locates in P, the new convex hull remains to be P.

(2) If p locates out of P, Alice and Bob need to evaluate the two tangent points and use p to replace all the vertexes between the two tangent points in P. The method to evaluate the two tangent points is shown as follows. If $\overline{p_{l-1}p_l} \times \overline{p_lp} > 0$ and $\overline{p_lp_{l+1}} \times \overline{p_lp} < 0$, p_i is the left tangent point. If $\overline{p_{J-1}p_J} \times \overline{p_Jp} < 0$ and $\overline{p_Jp_{J+1}} \times \overline{p_Jp} > 0$, p_i is the right tangent point.

When evaluating the cross product of p and $\overline{p_lp_{l+1}}$, points p_i and p_{i+1} may belong to Alice or Bob. According to the different point owner, different secure cross product protocols should be used. These protocols have the same principle to the one in Ref. [30], but they have different intermediate variables as shown in Table 5.

The SPCH protocol is described in Table 3.

Table 3 SPCH protocol

```
Algorithm2 SPCH($A$,$B$)
Input: Alice inputs her private point set $A$. Bob inputs his private point set $B$.
Output: The vertex set $P_{new}$.
Preprocessing:
  $A_0 = CHP(A)$; //CHP is the convex hull evaluation function.
  $B_0 = CHP(B)$;
  $m = |A_0|$; //$|x|$ refers to the size of $x$.
  $n = |B_0|$;
  $A_0 = \{a_1, a_2, \cdots, a_m\}$; //$a_i$ is the vertex of $A_0$.
  $B_0 = \{b_1, b_2, \cdots, b_n\}$;
Processing:
for $i = 1$ to $m$ do
{   if(PPPI $(a_i, B_{i-1})$ = = True)
      {$B_i = B_{i-1}$;
      continue; //break out of the circulation
      }
  $t_i = |B_{i-1}|$;
  $b_{t+1} = b_1$;
  $A_1$^$B_1$:
      for $j = 1$ to $t$ do
      { if ($b_j \in CHP(B)$ & $b_{j+1} \in CHP(B)$)
              $s_i = SCP\_PL(a_i, \overline{b_jb_{j+1}})$;
      if ($b_j \in CHP(A)$ & $b_{j+1} \in CHP(B)$)
              $s_i = SCP\_PL\_1(a_i, \overline{b_jb_{j+1}})$;
      if ($b_j \in CHP(B)$ & $b_{j+1} \in CHP(A)$)
              $s_i = SCP\_PL\_2(a_i, \overline{b_jb_{j+1}})$;
      if ($b_j \in CHP(A)$ & $b_{j+1} \in CHP(A)$)
              { $s_i = \overline{b_ia_i} \times \overline{b_jb_{j+1}}$);
              Send(Alice→Bob, $s_i$);}
  $A_2$|$B_2$:
    for $k = 1$ to $t - 1$ do
    {if ($S_k \geqslant 0$&&$S_{k+1} \leqslant 0$)
          $b_{left} = b_{k+1}$;
    if ($S_k \leqslant 0$&&$S_{k+1} \geqslant 0$)
```

```
        $b_{\text{right}} = b_{k+1}$; }
$A_3$:
    Delete($b_{\text{left}}$, $b_{\text{right}}$);
    Insert($b_{\text{left}}$, $b_{\text{right}}$, $a_i$); //insert $a_i$ into $b_{\text{left}}$ and $b_{\text{right}}$
    $B_i = \{b_1, b_2, \cdots, b_{\text{left}}, a_i, b_{\text{right}}, \cdots, b_t\}$;
}
    $P_{\text{new}} = B_m$;
```

3.2 Security analysis

Theorem 2 Assuming the underlying protocol is secure in semi-honest model, the SPCH protocol securely evaluates the convex hull of the input point sets in the presence of semi-honest adversaries.

Proof: It starts from the case that Bob is corrupted. Simulator S is created to receive Bob's private input, output and Bob's view in the protocol. According to the protocol, Bob's view is $view_i^{\Pi}(A, B) = (B, B_i, S_i, t_i)$ *where* $i = 1, 2, \cdots, m$. S can recreate Alice's first point a'_1 according to B_1 where the positional relationship of a'_1 and every edge in B_0 equals to the one of a_1 and every edge in B_0. Then S continually recreates Alice's second point a'_2 according to B_2. In the same way, S can recreate $a'_i (i = 3, 4, \cdots, m)$. Now S recreates a set $A' = \{a'_1, a'_2, \cdots, a'_m\}$. As the positional relationship between A' and Bob's convex hull is the same with the relationship between Alice's points and Bob's convex hull, every point in A' has the same tangents with Alice's. The resulted convex hull set of A' and B is the same with A and B. That is to say, the adversary gets nothing valuable from Bob's input, output and Bob's view in the protocol.

The same simulator can be created for Alice.

3.3 Algorithm complexity

In the processing stage, PPPI protocol is called for m times. For the worst case that all the vertexes of Alice's convex hull locate at the outside of Bob's convex hull, SCP _ PL Protocol is used for mn times. So the computational complexity of the proposed protocol is $mnT_c + mT_p$, and the communication complexity is $mC_p + mnC_c$. The notations used here are defined in Appendix B.

The computational and communicational complexity comparison of Lu's, Wang's and the proposed protocol are shown in Table 4. It shows that the complexity of protocols in Refs[10,11] relies on the input point set size. The proposed protocol's complexity only relies on the size of the input points' convex hull vertex set. So, when the points in the two party's input point set are dense, the proposed protocol is more efficient than the previous works.

Table 4 Algorithm complexity

	Computational complexity	Communicational complexity	Computational complexity (Instantiation)
The proposed protocol	$mnT_c + mT_p$	$mC_p + mnC_c$	$O(mn^3 + rmlogn)$
Ref. [10]	$T_a + 2(M + N)T_c$	$C_a + 2(M + N)C_c$	$O(rN^3 + NlogN)$
Ref. [11]	$(M + N)(T_c + T_a)$	$(M + N)T_a$	$O(N^3 + MNlogN + N^2 logN)$

(M and N denote the set size of Alice and Bob. m and n denote the size of Alice's and Bob's convex hull vertex set.)

To compare the complexity visually, the computational complexity of Lu's, Wang's and the proposed protocol are charted in Fig. 3. For the sake of comparison, it is assumed that $M = N$, $m = n$, $M = 100m$, $r = 50$. The same conclusion can be got with the theoretical analysis above.

4 Conclusion

Privacy-preserving point-inclusion and secure planar convex hull are the classical problems in SMCG. In this work, a novel PPPI protocol has been designed based on the classic homomorphic encryption and secure cross product protocol. Analysis shows that novel PPPI is highly efficient because the complexity is not

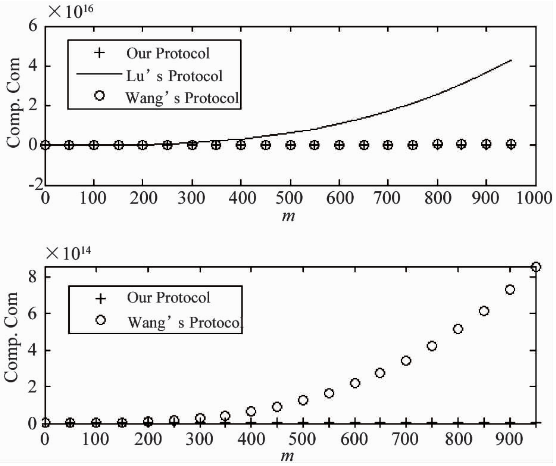

Fig. 3 Comparisons of Different SPCH protocols

related to the vertex size of the convex hull. Based on the novel PPPI protocol, an effective SPCH protocol has been presented. Analysis finds that the complexity of this SPCH protocol only relies on the size of the points in the outermost layer of the input point sets, and it has a good performance for large-scaled point sets compared with the previous solutions.

The proposed protocols are secure in semi-honest model, and the security of the protocols has been demonstrated by Goldreich method. The real world implementation and protocols secure against malicious adversary will be the goal in the future.

References

[1] Yao A C. Protocols for secure computations. In Proceedings of 23th Annual IEEE Symposium on Foundations of Computer Science, Chicago, USA, 1982. 160-164

[2] Goldreich O, Micali S, Wigderson A. How to play any mental game. In: Proceedings of the 19th Annual CAN Symposium on Theory of Computing, 1987. 218-229

[3] Atallah M J, Du W L. Secure multi-party computational geometry. *International Workshop on Algorithms and Data Structures*, 2004, 2125: 165-179

[4] Choi S G, Hwang K W, Kaza J, et al. Secure multi-party computation of Boolean circuits with application to privacy in online marketplaces. In: Proceedings of the Cryptographer's Track at the RSA Conference on Topics in Cryptology, San Francisco, USA, 2012. 416-432

[5] Nielsen J B, Nordholt P S, Orlandi C, et al. A new approach to practical active-secure two-party computation. In: Proceedings of the Advances in Cryptology-CRYPTO 2012, volume 7417 of LNCS, 2012. 681-700

[6] Schncider T, Zohner M. GMW vs Yao? Efficient secure two-party computation with low depth circuits. In: Proceedings of the Financial Cryptography and Data Security, volume 7859 of LNCS, 2013. 275-292

[7] Li S D, Si T G, Dai Y Q. Secure multi-party computation of set-inclusion and graph-inclusion. *Journal of Computer Research and Development*, 2005, 42(10): 1647-1653

[8] Luo Y L, Huang L S, Zhong H, et al. A secure protocol for determining whether a point is inside a convex polygon. *Chinese Journal of Electronic*, 2006, 15(4): 578-582

[9] Liu W, Luo S S, Chen P. Privacy-preserving point-line relation determination protocol and its application. *Journal of Beijing University of Posts and Telecommunications*, 2008, 31(2): 72-75

[10] Lu S F, Luo Y L. Privacy-preserving in graham algorithm for finding convex hulls. *Computer Engineering and Applications*, 2008, 44(36): 130-133

[11] Wang Q, Huang L S. Privacy- preserving protocols for finding the convex hulls. In: Proceedings of the 3rd International Conference on Availability, Reliability and Security, Barcelona, Spain, 2008. 727-732

[12] Hans S, Addenpalli S C, Gupta A, et al. On privacy preserving convex hull. In: Proceedings of the International Conference on Availability, Reliability and Security, Los Alamitos, USA, 2009. 187-192

[13] Eppstein D, Goodrich M T, Tamassia R. Privacy-preserving data-oblivious geometric algorithm for geographic data. In: Proceedings of the 18th ACM SIGSPATIAL International Conference Advances in Geographic Information Systems, San Jose, USA, 2010. 13-22

[14] Wang Q, Zhang Y. A convex hull algorithm for planar point set based on privacy protecting. In: Proceedings of the International Workshop on Education Technology and Computer Science, Wuhan, China, 2009. 434-437

[15] Li D, Huang L S, Yang W, et al. A practical three-dimensional privacy-preserving approximate convex hulls protocol. In: Proceedings of the 2008 Japan-China Joint Workshop on Frontier of Computer Science and Technology, Nagasahi, Japan, 2008. 17-23

[16] Pinkas B, Schneider T, Zohner M. Faster private set intersection based on OT extension. In: Proceedings of the 23rd USENIX Security Symposium (USENIX Security' 14), San Diego, USA, 2014. 447

[17] Mohassel P, Sadeghian S. How to hide circuits in MPC an efficient framework for private function evaluation. In: EUROCRYPT, volume 7881 of LNCS, 2013. 557-574

[18] Gennaro R, Hazay C, Jeffrey S. S. Automata Evaluation and Text search protocols with simulation based security. In Public Key Cryptography, volume 6056 of LNCS, 2010. 332-350

[19] Barni M, Failla P, Kolesnikov V, et al. Secure evaluation of private linear branching programs with medical applications. In: Proceedings of the 14th European Symposium on Research in Computer Security, volume 5789 of LNCS, 2009. 424-439

[20] Katz J, Ostrovsky R. Round-optimal secure two-party computation. In: CRYPTO, volume 3125 of LNCS, 2004. 335-354

[21] Pass R. Bounded-concurrent secure multi-party computation with a dishonest majority. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing, 2004. 232-241

[22] Katz J, Ostrovsky R, Adam S. Round efficiency of multiparty computation with a dishonest majority. In EUROCRYPT, volume 2656 of LNCS, 2003. 578-595

[23] Damgård I, Ishai Y. Constant-round multiparty computation using a black-box pseudorandom generator. In CRYPTO, volume 3621 of LNCS, 2005. 378-394

[24] Aumann Y, Lindell Y. Security against covert adversaries: efficient protocols for realistic adversaries. In: Proceedings of the 4th Theory of Cryptography Conference, volume 4392 of LNCS, 2007. 137-156

[25] Goyal V, Mohassel P, Smith A. Efficient two party and multi party computation against covert adversaries. In EUROCRYPT, volume 4965 of LNCS, 2008. 289-306

[26] Lindell Y, Pinkas B. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In EUROCRYPT, volume 4515 of LNCS, 2007. 52-78

[27] Malkhi D, Nisan N, Pinkas B, et al. Fairplay - a secure two-party computation system. In: Proceedings of the 13th Conference on USENIX Security Symposium, volume 13, 2004. 09-13

[28] Sella J B, Orlandi C. LEGO for two-party secure computation. In Theory of Cryptography Conference, volume 5444 of LNCS, 2009. 368-386

[29] Goldreich O. The Foundations of Cryptography-Volume 2, Basic Applications. Cambridge University Press, 2004

[30] Luo Y L, Huang L S, Wei J J, et al. Privacy-preserving cross product protocol and its application. *Chinese Journal of Computers*, 2007, 30(2):248-254

[31] Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In: EUROCRYPT, volume 1592 of LNCS, 1999. 223-238

[32] Nielsen J B, Nordholt P S, Orlandi C, et al. A new approach to practical active-secure two-party computation. In CRYPTO, volume 7417 of LNCS, 2012. 681-700

[33] Damgard I, Pastro V, Smart N, et al. Multiparty computation from somewhat homomorphic encryption. In CRYPTO, volume 7417 of LNCS, 2012. 643-662

Appendix A

Table 5 Intermediate variables among SCP _ PL

Protocol Name	Alice	Bob	s_1	s_2
SCP _ PL	q	p_i, p_{i+1}	$(y, -x, -y, x)$	$(x_i, y_i, -x_{i+1}, y_{i+1})$
SCP _ PL _ 1	q, p_i	p_{i+1}	(x_i, y_i, x, y)	$(y_{i+1}, -x_{i+1}, x_{i+1}, -y_{i+1})$
SCP _ PL _ 2	q, p_{i+1}	p_i	$(y_{i+1}, -x_{i+1}, x, -y)$	(x_i, y_i, y_i, x_i)

Appendix B

Table 6 Notation used in the paper

	Notation	Protocol
Computational Complexity	T_a	Millionaire Protocol
	T_c	SCP _ PL Protocol
	T_h	Paillier's encryption
	T_d	Paillier's decryption
	T_p	PPPI
Communicational Complexity	C_a	Millionaire Protocol
	C_c	SCP _ PL
	C_p	PPPI

Sun Maohua, born in 1986. She received her Ph. D degree from Beijing University of Posts and Telecommunications in 2013. She also received her Bachelor's degree from Shandong University in 2008. Her research interests include secure multi-party computation and information security.