

A Practical Solution for Privacy-preserving Approximate Convex Hulls Problem

Dong Li, Liusheng Huang, Wei Yang, Youwen Zhu, Yonglong Luo, Lingjun Li, Zhili Chen.

(National High Performance Computing Center at Hefei, Department of Computer Science and Technology, University of Science and Technology of China, Hefei, 230027, P. R. China)
(E-mail: lixido@mail.ustc.edu.cn, [lshuang, qubit}@ustc.edu.cn](mailto:lshuang,qubit}@ustc.edu.cn), zhuyw@mail.ustc.edu.cn, ylluo@ustc.edu.cn, [lqli, zlchen3@mail.ustc.edu.cn](mailto:lqli,zlchen3@mail.ustc.edu.cn))

Abstract

Convex Hulls Problem is a special case of Privacy-preserving Geometry problems in the inquiry of Secure Multi-Party Computation (SMC). It can be applied in military, commercial and many other fields. However, because of the definition's inherent defect, current schemes will inevitably disclose the points on the vertices. In this paper, we proposed the concept of privacy-preserving approximate convex hulls problem and provide a practical protocol which is more secure and efficient than previous convex hulls protocols. We also show that it can be applied to finding the approximate intersection area of two private convex hulls.

1. Introduction

In this era, more and more computations need the private input from several parties, who do not want to share their private data. Secure Multi-party Computation (SMC) [1-2], dealing with the problem that two or more parties want to jointly perform a computation, can provide useful solutions in such scenarios: each party needs to contribute its private inputs for computation, but no party wants to disclose its private input to the other parties. SMC was introduced by A. C. Yao [1] in 1982. Generally speaking, SMC deals with computing any probabilistic functions on any inputs, in a dis-

tributed network where each participant holds one of the inputs, ensuring independence of the inputs, correctness of the computation, and that no more information is revealed to a participant in the computation than can be inferred from the participant's input and output.

Now there have been lots of theoretic results of SMC [3-6]. However, using the solutions derived by these theoretic results for special cases of SMC may be impractical. As a result, special solutions should be developed for efficiency reasons in various applications [2].

Privacy-Preserving Geometric Computation (PPGC) [9] is an important sub-issue of secure multi-party computation problem. Wenliang Du and Zhijun Zhan [5] had developed a framework for all the SMC problems. In the field of PPGC, they listed five problems: Intersection, Point-Inclusion, Range Searching, Closest Pair and Convex Hulls. The convex hulls problem can be described like this: Alice and Bob want to jointly find the convex hulls secretly for the two point-sets that each have on the plane. Figure 1 is an example of convex hull. Atallah *et al.* [9] first proposed the Convex Hulls problem as follows:

Convex Hulls Problem: Alice has M points in the plane; Bob has N points in the plane. Alice and Bob want to jointly find the convex hulls for these $M+N$ points; however, neither Alice nor Bob wants to disclose any more information to the other party than what could be derived from the result.

Based on the two algorithms proposed by Chand & Kapur [10] and Preparata & Shamos [11] and basic SMC protocols, Wang *et al.* [14] presented two privacy-preserving protocols to find the convex hulls.

This work was supported by the National Natural Science Foundation of China (No. 60773032 & 60703071), the Ph.D. Program Foundation of Ministry of Education of China (No. 20060358014), the Natural Science Foundation of Jiangsu Province of China (No BK2007060), and the Anhui Provincial Natural Science Foundation (No. 070412043).

However, their scheme will inevitably disclose the private points on the vertices of the convex hull because of the defect of the definition. For example, in figure 1 we can see that there is one point exposed on each vertex of the convex hull. In some practical applications, such as two military troops want to jointly find the area covered by their monitoring points, the two parties don't want to disclose these points, and they would accept an approximate area rather than disclose their private information. To correct this shortcoming, in this paper, we propose a new protocol in PPGC, called Privacy-preserving Approximate Convex Hull Protocol. Compared with the protocols in [14], our protocol is more secure and more efficient. It can be applied to most of the applications using the Convex Hulls protocols previously.

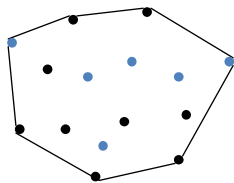


Figure 1. An example of convex hull

The paper is organized as follows: Section 2 discusses the preliminaries and basic definitions used in this paper. Section 3 introduces the concept of approximate polygon and the protocol we developed. Then section 4 discusses the protocol in detail by the experiment result and comparison. Finally section 5 concludes the paper.

2. Preliminaries and definitions

In this section, we introduce some necessarily preliminaries and definitions.

2.1 Secure Two-party Computation

Secure Two-Party Computation (STC) is a special case of SMC when there are only two participants. In this paper, we call the two participants in STC Alice and Bob. Alice inputs her private data x and Bob inputs y respectively, then they jointly compute the function $f(x, y)$, but neither one is willing to disclose his own private input to the other or any other third party. The Convex Hulls problem discussed below is a special instance of STC problem.

2.2 Flipping coin

Alice and Bob can use the following protocol which is proposed by Manuel Blum [15], to determine a random number θ .

We assume that x is a k -bit integer. We can execute the protocol below to get a k -bit integer θ , which is random for Alice and Bob. This protocol needs a Strong anti-collision hash function $H(x)$.

Flip-Coin protocol:

- a) Alice computes $H(x)$, and then sends it to Bob.
- b) Bob guesses k -bit of x , denoted as y , then sends y to Alice.
- c) If Bob's guess for the i^{th} bit of x is right, then Alice sets the i^{th} bit of θ is 1, else sets the i^{th} bit of θ zero.
- d) Alice sends x and θ to Bob then Bob can check the result.

The protocol is very simple but secure enough that Alice and Bob can determine the random integer θ .

2.3 Secure comparison protocols

The purpose of this protocol is to compare two private numbers. The private comparison problem is first introduced by Yao [1] and referred to as Yao's Millionaire Problem: two parties want to determine who is richer without disclosing anything else about their wealth. The solution proposed by Yao [1] is exponential communication complexity numbers involved, using an un-trusted third party. Cachin proposed a solution [3] based on the φ -hiding assumption also using an un-trusted third party. But the communication complexity of Cachin's scheme is $O(l)$, while l is the number of bits of each input number. Qin *et al.* [13] proposed a protocol based on the combination of a public-key cryptosystem of the homomorphic encryption and the theoretic construction relying on the φ -hiding assumption. Yao's Millionaire Problem Protocol is an important base of the approximate convex hull protocol we proposed. In this paper, we use it to find the larger one or the smaller one from two values while keep the other one secret.

We use $SC - Max(\theta_1, \theta_2)$ to denote that we get the larger one from θ_1 and θ_2 without disclosing the smaller one by secure comparison protocol.

We use $SC - Min(\theta_1, \theta_2)$ to denote that we get the smaller one from θ_1 and θ_2 without disclosing the larger one by secure comparison protocol.

2.4 Definitions

We first propose the definition about the Privacy-preserving Approximate Convex Hulls Problem.

Definition 1: Privacy-preserving Approximate Convex Hulls Problem:

Alice has m points on the plane; Bob has n points on the plane. They want to jointly find a polygon that approximates the convex hulls containing these $m+n$ points; however, neither Alice nor Bob wants to disclose any more information to the other party than what could be derived from the result.

In the following, we call the convex hull defined in [9] as original convex hull, which is the minimum approximate convex hull. For a give point set, there exists infinite approximate convex hulls, and so it is necessary to define the similarity measurement, a standard to evaluate the similarity of an approximate convex hull and the original convex hull. From definition 1 we know the approximate convex hull's area is larger than the original convex hull's, and so it is reasonable to define the ratio of their areas as the similarity measurement. If we use S_A to denote the area of the approximate convex hull, and use S_B to denote the area of the original convex hull then we can define the similarity measurement Δ as follows:

Definition 2: Similarity measurement of approximate convex hull:

$$\Delta = \frac{S_B}{S_A}.$$

Obviously $0 \leq \Delta \leq 1$. When Δ closer to 1, the approximate convex hull will be more similar to the original convex hull, and then it is likely to expose more private points on its vertices. But when Δ is closer to 0, it means the approximate convex hull's area is several times of the original convex hull's. In such case, the cost of hiding the points on the vertices of the original convex hull is too high. In practical applications, Δ can be defined by the requirements of the application.

3. Approximate convex hulls protocol

In section 3.1, we will give several notations that will be used to present our protocol. In section 3.2, we will describe the protocol in detail, and in section 3.3, we will show that the method we use to find the approximate can also be used to find the approximate intersection area of two private convex hulls.

3.1 Notations.

We give some notations here. They will be used in the description of our protocol in section 3.2.

- 1) Alice's private point set, if she has m points:

$$A = \{\text{point } a_i | i=1 \dots m\}$$

- Bob's private points set, if he has n points:

$$B = \{\text{point } b_i | i=1 \dots n\}$$

Every point on the plane can be identified by two coordinate values: the x -coordinate value and y -coordinate value.

- 2) Reference point: the auxiliary points, we denote these reference point set:

$$P = \{\text{point } p_i | i=1 \dots t\}$$

The number of reference point t is defined in the protocol.

- 3) C_1 is a circle that contains all their private points. Circle C_2 is larger than C_1 , the radius of C_2 is determined in the protocol. All the reference points lie on C_2 . C_1 and C_2 are concentric circles, and O is their center.
- 4) The angle of reference point: we use O as the origin of coordinates. Then the angle of reference point p is the angle between vector \overrightarrow{Op} and the positive direction of the x -coordinate.
- 5) The reference vector: it is a vector that tangent with C_2 at reference point p . The direction is anti-clockwise.

3.2 Privacy-preserving Approximate Convex Hulls Protocol (PACHP)

In this section, we propose our protocol based on the definitions above.

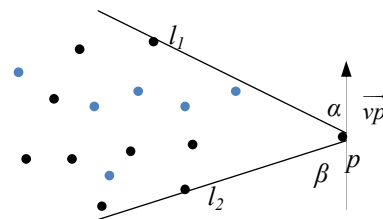


Figure 2. $\angle PAB$ is the angle we get for reference point p

The main idea of this protocol can be described as follows: Alice and Bob first find circle C_1 that contains all their private points; then they agreed on the number and position of the reference points. For each reference

point they draw a minimum angle clamping all their private points. Figure 2 describes an angle formed by reference point p and two lines, which are l_1 and l_2 passing through p . The intersection area of all the angles is the approximate convex hull. The following is the detailed steps of our protocol.

Input: Alice's private points set A , Bob's private points set B .

Output: Approximate convex hull G .

Protocol PACHP:

Begin

a) Alice computes point $P_A = \frac{1}{m} \sum_{i=1}^m a_i$, and sends P_A to

Bob. Bob computes $P_B = \frac{1}{n} \sum_{i=1}^n b_i$, and sends P_B to

Alice. The plus computation is to sum the x-coordinate values y-coordinate values together respectively.

b) They compute the point $O = \frac{P_A + P_B}{2}$, i.e. O is the

middle of P_A and P_B . Alice computes the maximum distance of her points to O , let's denote it d_A , and Bob computes d_B respectively.

Then they jointly compute:

$$d = SC - \text{Max}(d_A, d_B)$$

In this step, Alice and Bob get circle C_1 whose center is O and radius is d . Then they select O as the origin point of the Cartesian coordinate. They can construct a coordinate system O - xy .

c) Alice and Bob reach an agreement on the number of reference point t and $R(R > d)$, the radius of C_2 . Using the *Flipping - Coin* protocol, Alice and Bob can determine θ_1 , the angle of the first reference point. The location of point p_1 can be determined on circle C_2 . Then reference point p_i lies on C_2 with the angle $\theta_i = \theta_1 + \frac{2\pi}{n}$. Then we can also get the reference array $\overline{vp_i}$ $i=1 \dots t$.

d) For each reference point p_i ($i=1 \dots t$)

Begin

Alice computes

$$\alpha_1 = \max_{k=1..m} (\angle(\overline{p_i a_k}, \overline{vp_i}))$$

$$\beta_1 = \min_{k=1..m} (\angle(\overline{p_i a_k}, \overline{vp_i}))$$

Bob computes

$$\alpha_2 = \max_{k=1..n} (\angle(\overline{p_i b_k}, \overline{vp_i}))$$

$$\beta_2 = \min_{k=1..n} (\angle(\overline{p_i b_k}, \overline{vp_i}))$$

Then they compute the angles by secure compare protocol:

$$\alpha = SC - \text{Max}(\alpha_1, \alpha_2)$$

$$\beta = SC - \text{Min}(\beta_1, \beta_2)$$

We get two lines l_1 and l_2 with angles α and β between vector $\overline{vp_i}$ respectively. Angle $Angle_i$ can be formed by vertex p_i , l_1 and l_2 (like the angle in figure 2).

End

e) Alice and bob compute the intersection area for all the angles $Angle_i$, $i=1 \dots t$. The intersection is the approximate convex hull G .

End of PACHP

The correctness and security of PACHP are analyzed in the following two theorems.

Theorem 1 (Correctness): G is an approximate convex hull.

Proof: All Alice's and Bob's points lie in polygon G which we finally get by PACHP. The protocol is correct if we can prove that all Alice's and Bob's private points are in G . In step d of PACHP, all the points are in $Angle_i$, $i=1 \dots t$. G is the intersection of the angles, so all the points are in G . G is an approximate convex hull.

Theorem 2 (Security): Neither Alice nor Bob will disclose any private points except the intersection of more than three lines.

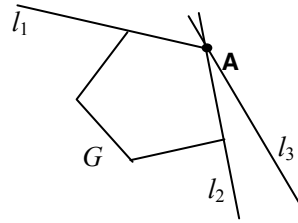


Figure 3. Proof of security

Proof: In our protocol, there are $2t$ lines exposed. In the algorithm, we can conclude that there is no less than one point on each exposed line. If there are more than three lines intersect at one point as in figure 3, which shows three lines l_1 , l_2 and l_3 intersect on point A ,

we can identify that A is a private point because A is the only point both on the edge of G and l_3

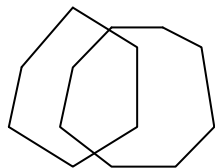


Figure 4. The position of two convex hulls

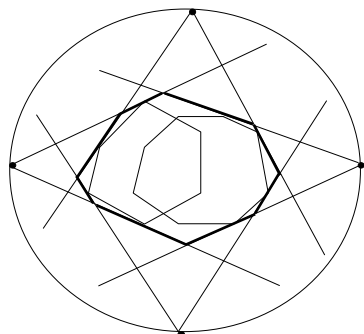


Figure 5. After executing PACHP

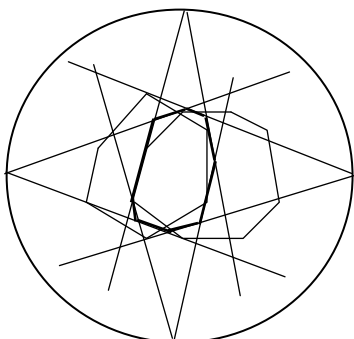


Figure 6. Find the intersection area

We can see the procedure directly by two figures. Figure 4 illustrates the position of the two polygons, and they are Alice's and Bob's private convex hulls. After PACHP, we get Figure 5, in which the polygon with bold edges is the output of our protocol. Only use four reference points used here.

3.3 Get the approximate intersection area of two polygons.

The protocol developed in section 3.2 can be modified for other applications. It can be used to find the intersection polygon of two private convex hulls. Consider such a scenario that Alice and Bob have a private convex hull respectively, they want to find whether there the two convex hulls intersect. If the two convex

hulls intersect, they can get an approximate intersection polygon. We can solve this problem just change the private selecting strategy in step d of PACHP:

$$\alpha = SC - \text{Min}(\alpha_1, \alpha_2)$$

$$\beta = SC - \text{Max}(\beta_1, \beta_2)$$

if $\alpha > \beta$ in any step, return the result that there is no intersection.

The result can be illustrated in Figure 6 for the same two convex hulls in Figure 4, the polygon with bold edges is our result.

4. Evaluation

We evaluate the effectiveness and security of the approximate convex hull generated by our protocol comparing with the convex hulls protocol in [14].

Computation Complexity Analysis: The main computational cost is employing the secure comparison protocol, denoted as $O(D)$. For the PPCHP_GW provided in [14], the complexity is $O\{(m+n)hD\}$ if the hull has h edges. For the PPCHP_QH in [14], the complexity is $O\{(m+n)D \log(m+n)\}$. If we use t reference points in PACHP, there are $2t+1$ times of secure comparison protocol used. 1 times in step b , and $2t$ times in step d . We only employ the flipping coin protocol once. For Alice, the computational cost is $O(mt)$. For Bob, the computational cost is $O(nt)$. The total cost is $O\{(m+n)Dt\}$. In our experiment, we find that we can achieve a given similarity measurement while t keeps constant. Then our protocol is more efficient than the protocols in [14].

Table 1. The average similarity measurement

$t \backslash R/d$	1.1	1.5	2	3
2	0.551	0.470	0.386	0.280
4	0.739	0.692	0.618	0.482
8	0.703	0.675	0.672	0.663
10	0.709	0.752	0.665	0.686
20	0.751	0.817	0.798	0.822

We simulated the protocol for different t and R/d . We use 100 random points each time and simulate the protocol 100 times to get the average similarity measurement. According the result in table 1, if we want to generate a convex hull that the similarity value is about

0.7, we can set $R=1.1d\sim 1.5d$ and the number of reference points is 4. These parameters can be determined by the requirements of applications.

In practical usage, for example, in military cooperation based on the locations on their maps, we can use the border of the map to arrange the reference points. In this way we can simplify the protocol, making it more efficient without compromising the security. Also if Alice and Bob classify their private points, it will be easier for them to find α and β in step d of our protocol.

Security Analysis: By the protocols in [14], the convex hulls we get will expose all the points on the vertices, however, in our protocol, these points are on the edges and they cannot be identified except there are more than three lines intersecting on one point. Even in the worst case, if all private points are exposed on the vertices of the approximate convex hull, in this case, it is the same with the original convex hull. Based on the fact that the original convex hull is just the worst case of approximate convex hull, the security of approximate convex hull is always better than original convex hull.

5. Conclusion and future work

Convex Hulls problem plays a significant role in the field of secure multi-party computational geometry. In this paper, we have defined the Approximate Convex Hulls Problem and proposed a practical protocol that can conceal more private information. We also discussed the computational complexity and security of our protocol.

Though it is more secure and efficient than the protocols proposed in [14], there still leaves some spaces needed to be investigated. For example, there are still some points exposed in this protocol. Besides, the protocol for three-dimension space is more complex. In our future work, we will strive to design a better way to solve this problem.

6. References

- [1] A. C. Yao. "Protocols for secure computations". In: Proc. 23rd Annual IEEE Symposium on Foundations of Computer Science. Los Alamitos: IEEE Computer Society Press, 1982. 160-164.
- [2] O. Goldreich, *Secure multi-party computation*, (manuscript version 1.3), 2002. [online] <http://theory.lcs.mit.edu/ doed>.
- [3] C. Cachin, "Efficient private bidding and auctions with an oblivious third party", Proc. of the 6th ACM conference on Computer and Communications Security, Singapore, pp.120-127, 1999.
- [4] S. Goldwasser, "Secure Multi-party computations: Past and present", Proc. of the 16th Annual ACM Symposium on Principles of Distributed Computing, Santa Barbara, CA, USA, pp.1-6. 1997.
- [5] W. L. Du, M. J. Atallah, "Secure multi-party computation problems and their applications: A review and open problems", Proc. of New Security Paradigms Workshop, Cloudcroft, New Mexico, USA, pp.11-20, 2001.
- [6] W. L. Du, M. J. Atallah, F. Kerschbaum, Protocols for secure remote database access with approximate matching, Proc. of the First Workshop on Security and Privacy in E-Commerce, Nov. 2000.
- [7] LUO Yong-Long, HUANG Liu-Sheng, et al., "A Secure Protocol for Determining Whether a Point is Inside a Convex Polygon", *Chinese Journal of Electronics*, Vol.15, No.4, Oct. 2006.
- [8] LUO Yong-Long, HUANG Liu-Sheng, et al., "Secure Two-Party Point-Circle Inclusion Problem", *Journal of Computer Science & Technology*, Jan. 2007, Vol.22, No.1, pp.88-91.
- [9] M. J. Atallah, W. L. Du, Secure multi-party computational geometry, Lecture Notes in Computer Science, Vol. 2125, Springer Verlag. Proc. of 7th International Workshop on Algorithms and Data Structures (WADS), pp.165-179, 2001.
- [10] Donald R. Chand, Sham S. Kapur, "An algorithm for Convex Polytopes", *Journal of the Association for Computing Machinery*, Vol. 17, No.1, January 1970, pp.78-86.
- [11] Franco P. Preparata, Michael I. Shamos, *Computational Geometry: an introduction*, Springer-Verlag New York, Inc. 1985, ISBN 0-387-96131-3.
- [12] W. L. Du, Y. S. Han and S. G. Chen, "Privacy-preserving multi-variety statistical analysis: linear regression and classification", Proc. of the 4th SIAM International Conference on Data Mining, pp.222-233, 2004.
- [13] Qin Jing, Zhang Zhenfeng, Feng Dengguo, et al., "A protocol of comparing information without leaking". *Journal of Software*, 2004, 15(3): 421-427 (in Chinese).
- [14] WANG Qi, LUO Yong-Long, HUANG Liu-Sheng, "Privacy-preserving Protocol for Finding the Convex Hulls", *The Third International Conference on Availability, Reliability and Security*, 2008
- [15] M. Blum, *Coin Flipping by Telephone*, IEEE COMP-CON 1982, pp. 133-137.