Privacy-preserving Similarity Sorting in Multi-party Model

Yifei Yao[1] and Fanhua Yu[1]
(Corresponding author: Yifei Yao)

College of Computer Science and Technology, Changchun Normal University[1]
No. 677, Changji North Road, Erdao District, Changchun 130032, China
(Email: yao_yifei@126.com)

(Received May. 29, 2016; revised and accepted Aug. 25 & Sep. 3, 2016)

Abstract

In social network, it is conceivable that a rational execution sequence does good to cooperative mission, especially for a large number of participants. However, there are many difficulties for multi-party computation, the most important of which is privacy. In this paper, secure multi-party computation technology and dimensionality reduction are chosen to design a privacy-preserving protocol, which sorts m people according to their similarity. In a n dimensional system, the secure protocol's time complexity is $O(mn+n+m\log m)$ and communication complexity is $O(m)$. Detailed analysis about security and applicability are also presented in this paper. In addition, the protocol can be improved in security at the cost of complexity, with an arbitration agreement designed against fraud.

Keywords: Dimensionality Reduction; Privacy-preserving Computation; Secure Multi-party Computation; Similarity Sort Algorithm

1 Introduction

In the age of big data, complex information is emerging endlessly, and traditional algorithms are facing challenges of high dimensional data. Meanwhile, disclosure of sensitive information becomes the major deterrent for the growth of social network [1]. For these issues, special schemes have been proposed in many domains, such as designing privacy-aware systems in a cloud environment [9] and defining privacy protection mechanisms for mobile social networks [11, 12]. In contrast to developing approaches against corruption attacks, arbitral protocols is also a good choice for fairness and privacy preserving.

With the rapid development of communication technology, security turns more and more essential, which makes secure protocols designed to solve basic problem popular [14]. In former applications, people always collect distributed information together and turn to a trust third party (TTP) for solution. But the demand of privacy makes it hard to find such an agency trusted by all the participants. Actually, each party wants the result correct, avoiding leaking his information to the others. Secure multi-party computation makes cooperative calculation privately and prevents participants' data from leaking [5]. Privacy-preserving techniques provide methods to calculate functions with the input of private information [18]. It turns out to be attractive because it can benefit people in security [4].

One significant technical challenge in social network is sorting. With an execution sequence for the participants, they will work more efficiently and fairly. In addition, sorting algorithm is the basis of many fields such as data analysis and database systems, which is a core step of many algorithms and of significance both in theory and practice. Guan Wang pushes all knowledge and influence of input values down to small black-box circuits avoiding the significant computational overheads, he uses Yao's garbled paradigm as reference, but his method only works on two party system. Doctor Jónsson proposed a secure sort algorithm which can be used as a building-block with $O(n\log n)$ comparisons in $O(\log n)$ rounds. Even it can be built upon any secret sharing scheme supporting multiplication and addition, complexity is high without any dimensionality reduction [8]. Because it is not easy to construct the optimal branching program for a complex function, Bingsheng Zhang designed several constant-round 0-error oblivious sorting algorithms together with some useful applications. In paper [2], Dan Bogdanov and his partners compared several published sorting methods. They evaluated the theoretical performance and discussed the practical implications of the different approaches. After that, Dan Bogdanov's group improved two earlier designs based on sorting networks and quick sort with the capability of sorting matrices. They also proposed two new designs - a naive comparison-based sort with a low round count and an oblivious radix sort algorithm that does not require any private comparisons in [3]. Koki Hamada proposed a simple and general approach of converting non-data-oblivious comparison sort algorithm. Although his

method improved the running time compared to existing protocols in experiments, it can be only used in a certain field as the author described [7]. Then in 2014, Koki Hamada improved his work, he used a new technique called "shuffle-and-reveal" for an $O(n \log n)$ communication complexity result. But it is also restricted for a constant number of parties and a field with a constant size [6].

In this paper, we propose a protocol to achieve a reasonable sequence for m partners in n dimensional system, and then analyze its security, complexity and applicability. The paper is organized as follows. In Section 2 we describe preliminaries. The privacy preserving similarity sorting protocol is introduced in Section 3. Then Section 4 discusses the protocol's complexity, security and applicability. Two kinds of improvement measure are applied in Section 5 and arbitration procedure which is used in case of fraud is discussed in Section 6. The further work together with a conclusion is proposed at last.

2 Preliminaries

2.1 Secure Multi-party Computation (SMC)

SMC is a kind of distributed calculation, it needs each party's private data as input, then broadcasts the final result without leaking anyone's privacy. SMC technology was proposed in 1982 [15], and it comes into more and more domains such as social networking services [17], ad hoc networks [13], computation geometry [16], data mining [10] and so on.

A third-party who is trusted by the whole group can help them do the privacy-preserving work, he can get enough information to complete the calculation and publish the result. But the hypothesis of trusted third party is unsafe and less realistic. It is known that any secure computation problem can be solved by a circuit protocol, but the size of the corresponding circuit is usually too large to realize [15]. So researchers choose to design special protocol for special use in a viable way.

2.2 Secure Sum Protocol

Suppose there are $m \geq 3$ parties P_1, P_2,$\cdots$,P_m who join in the computation. Each participant $P_j(j = 1, 2, ..., m)$ has his private information d_j. They want to calculate the function $\sum_{j=1}^{m} d_j$ together, but no one will leak his secret to others. Secure sum protocol solve the problem with the help of data disrupt technique in [15], and Figure 1 shows the meaning of this technique.

1) First of all, $P_j(j = 1, 2, ..., m)$ generates m random shares $x_{j,k}$ for $k = 1, 2, \cdots, m$, such that $d_j = \sum_{k=1}^{m} x_{j,k}$;

2) Then, $P_j(j = 1, 2, ..., m)$ sends P_k with $x_{j,k}$, for $k = 1, 2, \cdots, m$ and $k \neq j$;

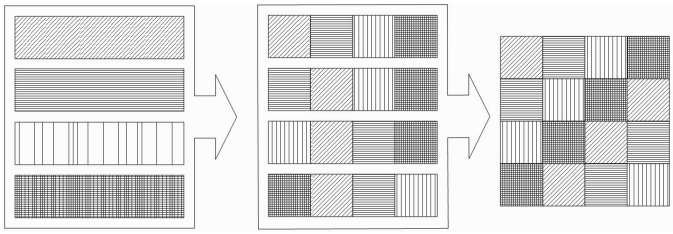
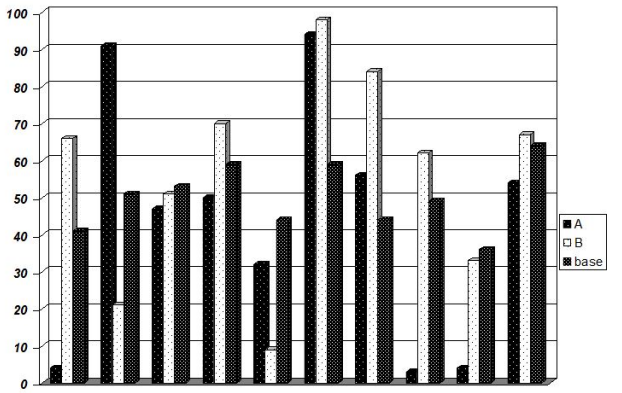


Figure 1: Schematic diagram of secure sum protocol



	1	2	3	4	5	6	7	8	9	10
A	4	91	47	50	32	94	56	3	4	54
B	66	21	51	70	9	98	84	62	33	67
base	41	51	53	59	44	59	44	49	36	64

Figure 2: Case of similarity comparison

3) After $P_j(j = 1, 2, ..., m)$ gets all the $x_{k,j}$ from P_k that $k = 1, 2, \cdots, m$ and $k \neq j$, he computes $\hat{x}_j = \sum_{k=1}^{m} x_{k,j}$ and broadcasts it;

4) At last, $P_j(j = 1, 2, ..., m)$ computes $X = \sum_{j=1}^{m} \hat{x}_j$.

2.3 Similarity Comparison Algorithm

In this paper, similarity comparison algorithm is chosen for dimensionality reduction. For the n dimension system, A and B want to know who is more similar as the baseline $Base$. Suppose $D_A = (a_1, a_2, \cdots, a_n)$, $D_B = (b_1, b_2, \cdots, b_n)$ and $D_{Base} = (e_1, e_2, \cdots, e_n)$, similarity comparison algorithm compare $c_A = \sum_{k=1}^{n} a_k \times e_k$ and $c_B = \sum_{k=1}^{n} b_k \times e_k$.

Take numbers in Figure 2 as example, the similarity comparison algorithm says A is closer to $Base$ than B for $c_A = 23411$ and $c_B = 28998$.

2.4 Secret Comparison Protocol

In 1982, A.C. Yao brought forward the famous millionaires problem in [15]: two millionaires, Alice and Bob, want to know which is richer, without revealing their respective wealth. In 2004, Qin brings forward a method for two parties comparing if $a = b$ privately. The method validates its security by computational indistinguishabil-

ity through homomorphism encryption and Φ-hiding assumption. It returns which one is greater or equal to the other, while Yao's method couldn't return the equality message. This protocol is complex in computation and safe to resist decoding, it reduces the communication of random data perturbation techniques in Yao's method.

In 2009, Doctor Luo proposed a three-round protocol for solving the secure comparison problem in the semi-honest setting based on the property of the cross products, who improved cross products protocol by using the Paillier's additive homomorphic encryption firstly. His method can compare two real numbers in addition to integers, and determine whether $a > b$, $a > b$ or $a = b$ for two partners.

Secret comparison protocol is one of the most important protocols in SMC, it was widely used in privacy-preserving computation geometry [16], social networking services [17] and so on.

Equality-testing is a special kind of secure comparison protocol. It helps two parties know whether their private data are equal or not, moreover, nobody can seek the other's information from the result if they are not the same.

2.5 Secure Scalar Product Protocol

Secure scalar product protocol is a basic tool in SMC, and it has been applied to the privacy-preserving cooperative calculation widely. It can help two partners compute the scalar product of their private vectors correctly and securely. Suppose Alice has a private vector $X = (x_1, x_2, \cdots, x_n)$ and Bob has his own $Y = (y_1, y_2, \cdots, y_n)$. After the protocol, Alice get $u = X \cdot Y + v = \sum_{i=1}^{n} x_i \times y_i + v$ where v is a random number selected by Bob. Meanwhile Alice cannot get any information about $X \cdot Y$ or y_i from u, Bob can get nothing about u or x_i from v either.

3 Privacy-preserving Similarity Sorting Protocol

3.1 Computational Model

Generally speaking, there are potential malicious attacks against any multi-party protocol. In this paper, we study the problem under a semi-honest model, in which each party follows the protocol without trying to intermit or disturb with dummy data, even they will keep a record of all its intermediate computation [15]. This model is practical and useful, because everybody in the cooperation expects the right result rather than others' private information.

3.2 Security Model

The classical definition of security is stated in [4]. Let f be a function that $P_j(j = 1, 2, \cdots, m)$ will compute cooperatively. If there is a protocol Π, for each P_j it can generate a simulator which can get all messages though the process only with its view and output, then it is secure. It is to say: The protocol Π to compute function f is secure when it satisfies the conditions as follows: There exists a probabilistic polynomial-time simulator $S_j(j = 1, 2, \cdots, m)$, it holds that

$$\{(S_j(x_j, t_j), t_1, t_2, \cdots, t_{j-1}, t_{j+1}, \cdots, t_m)\} \quad (1)$$
$$\equiv \{view_j^{\Pi}(x_1, \cdots, x_m), v_1, v_2, \cdots, v_{j-1}, v_{j+1}, \cdots, v_m\}$$

where

$$t_j = f_j(x_1, x_2, \cdots, x_m)$$

and

$$v_j = output_j^{\Pi}(x_1, x_2, \cdots, x_m)$$

While the party's view consists of its initial input, an auxiliary initial input (which is relevant only for modeling adversarial strategies), its random-tape, and the sequence of message it has received so far.

3.3 Privacy-preserving Similarity Sorting Protocol

Input: There are m members in this group, each one has his private data in the form of $D_j = (d_{1,j}, d_{2,j}, \cdots, d_{n,j})^T$ which $j = 1, 2, \cdots, m$.
Output: Reasonable sorting consequence for the group.

Algorithm 1 Privacy-preserving similarity sorting protocol

1: Begin
2: Set the quantitative standard.
Everybody agrees to take part in the n-dimensional coordinate system after the preprocess stage.
3: all the participants join in the secure sum protocol for n times to get the sum s_i for each dimension, that
$$s_i = d_{i,1} + d_{i,2} + \cdots + d_{i,m}$$
Then each one gets the average
$$\bar{E} = (\frac{s_1}{m}, \frac{s_2}{m}, \cdots, \frac{s_n}{m})^T = (e_1, e_2, \cdots, e_n)^T.$$
4: each participant $P_j(j = 1, 2, \cdots, m)$ calculates the cross product $c_j = \sum_{k=1}^{n} d_{k,j} \times e_k$ with his own D_j and the public average $\bar{E}$.
5: P_j broadcasts his c_j and gains the order.
6: End

4 Analysis

In this section, we analyze the complexity, security and applicability for this protocol.

4.1 Complexity Analysis

Conclusion 1: The time complexity of privacy-preserving similarity sorting protocol is $O(mn + n +$

$m \log m$), while there are m partners in n dimensions system.

In a distributed algorithm without leader, time complexity means the time each party spending on the work locally.

In Step 3, secure sum protocol costs each member $m-1$ times random number generation and $2m-2$ times addition. In total, it is $O(mn)$ times addition. In Step 4, each party needs n times multiplication and $n-1$ times addition. In Step 5, there is a sorting algorithm finished in $O(m \log m)$.

In addition, the time complexity turns to be $O(m \log m)$ when $m \gg n$, and be $O(mn)$ when $m \ll n$.

Conclusion 2: The communication complexity of privacy-preserving similarity sorting protocol is $O(m)$ that m is partner number in the group.

There is no interactive communication in Step 4 and Step 5. Only Step 3 sends and receives one piece of message towards each other. In total, the protocol spends $2(m-1)$ messages which is $O(m)$ in short.

4.2 Security Analysis

Conclusion 3: Privacy-preserving similarity sorting protocol can execute securely without leaking privacy.

Now, we analyze the message leaked at each step in this protocol.

- *Secure sum protocol (Step 3)*: Each member $P_j (j = 1, 2, ..., m)$ taking part in the secure sum protocol will get a view as below:

$$\begin{aligned} view_j \quad = \quad & \{d_{i,j}, piece_{i,j,k}, piece_{i,k,j}, s_i, e_i, m\} \\ & i = 1, 2, \cdots, n; j = 1, 2, \cdots, m; j \neq i; \\ & k = 1, 2, \cdots, m \end{aligned}$$

  where $(d_{1,j}, d_{2,j}, \cdots, d_{n,j})^T$ is P_j's private data. $picec_{i,A,B}$ is the piece of data sent from member A to B on the ith dimension, satisfying $\sum_{j=1,k=1}^{m,m} piece_{i,j,k} = d_{i,j}$ and $\sum_{j=1,k=1}^{m,m} piece_{i,k,j} = s_i$. s_i is the result of secure sum protocol on the ith dimension, and e_i is the average that $e_i = \frac{s_i}{m}$. Meanwhile, m is the number of people in the calculation group.

  From the view, no one can analyze other's private data at all. Even $d_{i,j} = \sum_{k=1}^{m} piece_{i,j,k}$, $\sum_{k=1}^{m} piece_{i,k,j}$ means nothing. At the end of this step, the sum and average on each dimension is known by everybody, while nothing sensitive is leaking.

- *Cross Product (Step 4)*: Everybody computes the cross product locally and no information can be got by others.

- *Broadcast and Sorting Phase (Step 5)*: Even P_j knows $c_l = \sum_{k=1}^{n} d_{k,l} \times e_k$ where $l = 1, 2, \cdots, m$, he can't get anything more about $d_{k,l} (k = 1, 2, \cdots, n)$. Because on each dimension, P_j knows only one equation against n unknown numbers, the mathematical analysis helps keeping secret. After knowing c_l, P_j calls sorting algorithm locally for his order at last.

Because the three steps are independence and there is no rule between them, neither can analyze to know the others' information through the privacy-preserving protocol. This does preserve the parties' privacy.

Furthermore, if there are some people dishonest joining hands for other's privacy, privacy-preserving similarity sorting protocol can hold on security at a certain extent.

$$\mathbf{D} = (D_1, D_2, \cdots, D_m) = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,m} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,m} \\ \cdots & \cdots & \cdots & \cdots \\ d_{n,1} & d_{n,2} & \cdots & d_{n,m} \end{bmatrix}$$

For example, participants $\tilde{P}_j = \{P_k | k = 1, \cdots, m; k \neq j\}$ join together to expose P_j. What they want to know is $D_j = (d_{1,j}, d_{2,j}, \cdots, d_{n,j})^T$ and what they know is $\bar{E} = (\frac{s_1}{m}, \frac{s_2}{m}, \cdots, \frac{s_n}{m})^T = (e_1, e_2, \cdots, e_n)^T$ that $s_i = d_{i,1} + d_{i,2} + \cdots + d_{i,m}$ and $c_j = \sum_{k=1}^{n} d_{k,j} \times e_k$. They can only reveal n unknown numbers from one certain equation, and when they want to use $s_i = d_{i,1} + d_{i,2} + \cdots + d_{i,m}$ there will be more unknown numbers appear in their equations. Even more, it turns to be more and more unprocurable when n turns bigger.

Another risk is the secure sum protocol in Step 3, the protocol needs $m \geq 3$ people, while it is secure when there is no more than $m-2$ dishonest co-conspirators.

4.3 Applicability Analysis

- Case study:Take $n = 10$ and $m = 10$ for example, 100 random numbers from 1 to 100 are showing in Table 1.

Table 1: Case study for $n = 10$ and $m = 10$

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
4	66	61	4	47	8	90	85	77	45
91	21	34	97	81	31	61	2	44	57
47	51	46	73	17	27	28	93	76	23
50	70	54	60	51	63	58	35	66	83
32	9	61	37	35	14	68	6	69	99
94	98	2	97	52	86	27	45	63	52
56	84	13	71	38	48	13	64	46	25
3	62	20	69	82	46	37	63	9	80
4	33	92	52	83	15	45	54	20	8
54	67	63	35	39	69	69	88	32	69
3^{rd}	9^{th}	1^{st}	10^{th}	6^{th}	2^{nd}	4^{th}	7^{th}	5^{th}	8^{th}

The average $\bar{E}$ calculated in Step 3 shows at the last column. Then, Step 4 public each one's cross product value as (23405, 29451, 21925, 30595, 26375, 22007,

25199, 27157, 25723, 27960). At last, they get the sort sequence shows in Figure 3.

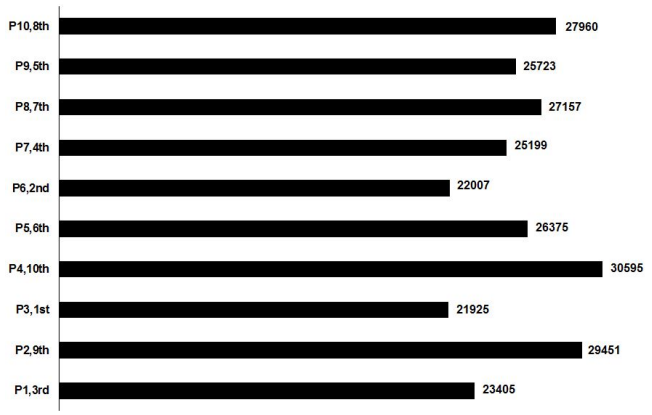


Figure 3: Sequence after sorting

- Advanced by weight modified: This protocol can be modified by adding weight w_k to the k^{th} dimension for some reason. In this case, Step 4 calculates $c_j = \sum_{k=1}^{n} d_{k,j} \times e_k \times w_k$ for a new rank rule. It helps the group keep the flexibility of adjusting weight for a new sort sequence. It is important to note that proposing weight modified protocol many times in the same group must be avoided. Or else, it turns insecure for their privacy.

5 Improved Protocol

In this session, privacy-preserving similarity sorting protocol is improved in security.

5.1 Advanced by Secure Comparison Protocol

In Step 5, P_j broadcasts his c_j and gains the order. If the group doesn't want the sort sequence but only two people's relationship, they can use secure comparison protocol instead. In this way, the two people get the right result without leaking information about c_j. It is a secure method at a higher level taking complexity in exchange.

The improved protocol shows in Algorithm 2:

Input: There are m members in this group, each one has his private data in the form of $D_j = (d_{1,j}, d_{2,j}, \cdots, d_{n,j})^T$ which $j = 1, 2, \cdots, m$.
Output: The relationship of P_a and P_b.

If there are l people wants the sort sequence while $2 < l < m$, the number of comparison time will be $\log_2 l$ according to the dichotomy.

5.2 Advanced by Data Compression

Data compression can not only strengthen security, but also reduce complexity. After the analysis of test numbers for many times, there are two methods adding data compression technology to the privacy-preserving similarity sorting protocol.

Algorithm 2 Advanced Protocol

1: Begin
2: Set the quantitative standard.
3: All the participants join in the secure sum protocol for n times to get the sum
$$s_i = d_{i,1} + d_{i,2} + \cdots + d_{i,m}$$
for each dimension.
Then each one gets the average
$$\bar{E} = (\frac{s_1}{m}, \frac{s_2}{m}, \cdots, \frac{s_n}{m})^T = (e_1, e_2, \cdots, e_n)^T.$$
4: P_a calculates $c_a = \sum_{k=1}^{n} d_{k,a} \times e_k$ and P_b calculates $c_b = \sum_{k=1}^{n} d_{k,b} \times e_k$.
5: Secure comparison protocol helps P_a and P_b get their relationship.
6: End

- **Compression before calculation**: In this case, Step 2 of Algorithm 1 should be modified below:

  2: In each dimension, the group agrees on the standard that divided in 100 degrees. Each partner's private data can be map in the standard and $d_{i,j}$ should be an integer from 1 to 100.

  In this method, privacy is hided by compression before calculation, because $d_{i,j}$ is closely related to partner's private data but not the exact value. By the way, the compression maps values into integers, it simplifies the calculation at the next steps.

  This kind of compression also imports some more tasks. If the group wants the preprocess's standard, they must spend time on getting boundaries. Maximum and minimum value must be detected before the comparison stage. Secure protocol choosing max/min number in secure multi-party computation maybe in use.

- **Compression before broadcasting**: In this case, Step 5 of Algorithm 1 should be modified as below:

  5: P_j selects the first $\lceil \log_2 m + 1 \rceil$ numbers of c_j as $\bar{c_j}$. He broadcasts $\bar{c_j}$ and gains the order.

  Comparing with compression before calculation, this method is more convenient and efficient. It doesn't increases the complexity but improve its security. Because data slot leaks less information than the whole data. Take case study in Table 1 as example,

  $$m = 10, \text{ and } \lceil log_2 m \rceil = 4.$$

  Step 5 publish each one's cross product value as (2340, 2945, 2192, 3059, 2637, 2200, 2519, 2715, 2572, 2796) instead of (23405, 29451, 21925, 30595, 26375, 22007, 25199, 27157, 25723, 27960), and they will get the same sequence as before compression.

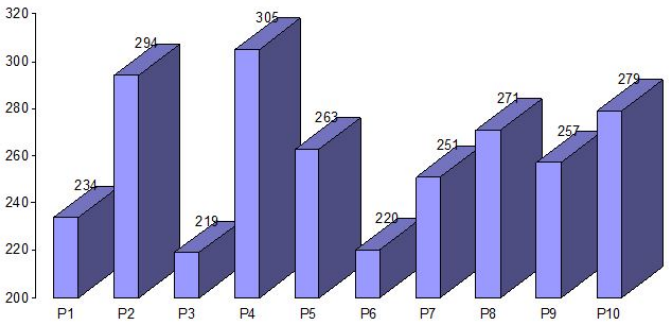


Figure 4: Case of compression before broadcasting

Sometimes, broadcasting only $\lfloor log_2 m \rfloor$ numbers of c_j will work. For (234, 294, 219, 305, 263, 220, 251, 271, 257, 279) also gets the same sequence in Figure 4.

In some special case, compress method should modified to take the middle number for sorting, like (34, 94, 19, 05, 63, 20, 51, 71, 57, 79) for the case study in Table 1. So, the way of compression before broadcasting can be changed in special environment.

6 Arbitration Procedure

Partner may cheat for sake, he might bring out a false number in the broadcast step. In this case, an arbitration procedure will be in need.

6.1 Third-party Arbitration

If there is a trusted third party who calculates the suspect number honestly, fraud case can be detected easily.

What TTP must verify are two aspects.

- If the calculation result matches the number broadcasted in Step 5.
- If the calculation factors match the candidates' private information.

Although the third-party arbitration is not complex, it is unsafe for partner's sensitive information. A method without TTP is more useful in the social network.

6.2 Privacy-preserving Arbitration

By the help of secure equality-testing and scalar product protocol, the arbitration procedure can be carried out without leaking privacy.

Suppose that P_A suspects that P_B broadcasted a false $\bar{c_B} \neq c_B$ in Step 5, the privacy-preserving arbitration runs as below.

- P_A selects a random number r satisfying $0 < r < 10$.
- P_A holds the $\check{E} = (e_1 + r, e_2 + r, \cdots, e_n + r)^T$ and P_B holds his privacy D_B. After the secure scalar product protocol, P_B holds $u = \check{E} \cdot D_B + v = \sum_{k=1}^{n} (e_k + r) \times d_{k,B} + v = \sum_{k=1}^{n} e_k \times d_{k,B} + r \times \sum_{k=1}^{n} d_{k,B} + v$ while v is another random number selected by P_A.
- P_A demands P_B to publish his sum $sum_B = \sum_{k=1}^{n} d_{k,B}$.
- Secure equality-testing protocol tells P_A if $\bar{c_B} + r \times sum_B + v$ is equal to u, it means if P_B broadcasted a false number for cheat.

7 Conclusions and Further Work

Privacy preserving similarity sorting protocol offers a secure solution for problems among m partners in n dimensional systems, which uses secure sum protocol for security and a rational dimensionality reduction for efficiency. After proposing the protocol, we present complexity and security analysis detailed in steps. A case showing computation process is posed in this paper with discussions of some improvements for certain settings and arbitration award for fraud. A further research on secure self-adaption sorting will be considered, while a more reasonable dimensionality reduction method will be brought in for different conditions. In addition, customized version for special use is also considered in the future.

Acknowledgments

The author wishes to thank the fellows in National High Performance Computing Center for helpful comments. And we are very grateful to Professor W. Yang, who is at Suzhou Institute of Advance Study, University of Science and Technology of China, for useful suggestion and some corrections. This work is supported by Funding Project for high tech and industry from Jilin Development and Reform Commission under Grant No. [2014]817.

References

[1] I. Casas, J. Hurtado, and X. Zhu, "Social network privacy: Issues and measurement," in *Web Information Systems Engineering (WISE'15)*, pp. 488–502, Miami, USA, Nov. 2015.

[2] B. Dan, S. Laur, and R. Talviste, *Oblivious Sorting of Secret-Shared Data*, Tallinn, Estonia: Institute of Information Security, 2013.

[3] B. Dan, S. Laur, and R. Talviste, "A practical analysis of oblivious sorting algorithms for secure multiparty computation," in *Secure IT Systems*, pp. 59–74, Tromso, Norway, Oct. 2014.

[4] W. Du and Z. Zhan, "A practical approach to solve secure multi-party computation problems," in *Proceedings of the 2002 workshop on New security paradigms (NSPW'02)*, pp. 127–135, Virginia Beach, USA, Sept. 2002.

[5] O. Goldreich, "Secure multiparty computation," *Chapman and Hall CRC*, vol. 2, no. 3, pp. 927–938, 2010.

[6] K. Hamada, I. Dai, K. Chida, and K. Takahashi, "Oblivious radix sort: An efficient sorting algorithm for practical secure multi-party computation," *IACR Cryptology ePrint Archive*, vol. 2014, pp. 121–150, 2014.

[7] K. Hamada, R. Kikuchi, I. Dai, K. Chida, and K. Takahashi, "Practically efficient multi-party sorting protocols from comparison sort algorithms," in *Information Security and Cryptology (ICISC'12)*, pp. 202–216, Seoul, Korea, Nov. 2012.

[8] K. V. Jónsson, G. Kreitz, and M. Uddin, "Secure multi-party sorting and applications," *IACR Cryptology ePrint Archive*, pp. 122, 2011.

[9] E. Kavakli1, C. Kalloniatis, H. Mouratidis, and S. Gritzalis, "Privacy as an integral part of the implementation of cloud solutions," *Computer Journal*, vol. 58, no. 10, pp. 2213–2224, 2014.

[10] S. Patel, D. Punjani, and D. C. Jinwala, "An efficient approach for privacy preserving distributed clustering in semi-honest model using elliptic curve cryptography," *International Journal of Network Security*, vol. 17, no. 3, pp. 328–339, 2015.

[11] S. Sarpong, C. Xu, and X. Zhang, "An authenticated privacy-preserving attribute matchmaking protocol for mobile social networks," *International Journal of Network Security*, vol. 17, no. 3, pp. 357–364, 2015.

[12] S. Sarpong, C. Xu, and X. Zhang, "Ppam: Privacy-preserving attributes matchmaking protocol for mobile social networks secure against malicious users," *International Journal of Network Security*, vol. 18, no. 4, pp. 625–632, 2016.

[13] Y. Wang, H. Zhong, Y. Xu, and J. Cui, "Ecpb: Efficient conditional privacy-preserving authentication scheme supporting batch verification for vanets," *International Journal of Network Security*, vol. 18, no. 2, pp. 374–382, 2016.

[14] C. Yang, T. Chang, and M. Hwang, "A (t,n) multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 151, no. 2, pp. 483–490, 2004.

[15] A. C. Yao, "Protocols for secure computations," in *Foundations of Computer Science Annual Symposium*, pp. 160–164, Chicago, USA, Nov. 1982.

[16] Y. Yao, S. Ning, M. Tian, and W. Yang, "Privacy-preserving judgment of the intersection for convex polygons," *Journal of Computers*, vol. 7, no. 9, pp. 2224–2231, 2012.

[17] Y. Yao, R. Zheng, and W. Shang, "Privacy preserving outlier detection in social networking services," *Journal of Computational Information Systems*, vol. 9, no. 11, pp. 4299–4307, 2013.

[18] Y. Zhao, F. Yue, S. Wu, H. Xiong, and Z. Qin, "Analysis and improvement of patient self-controllable multi-level privacy-preserving cooperative authentication scheme," *International Journal of Network Security*, vol. 17, no. 6, pp. 779–786, 2015.

Yifei Yao was born in Jilin Province, China, in 1981. She received the Ph.D. degree from the Department of Computer Science and Technology, University of Science and Technology of China in 2008. She is currently a lecturer of the College of Computer Science and Technology at Changchun Normal University. Her major research interests are information security and distributed computing.

Fanhua Yu received his B.S. degree from the College of Computer Science and Technology, Jilin University in 2004, and Ph.D degree from the School of Transportation, Jilin University in 2008. Now he is a professor and master Tutor of Changchun Normal University. His research interests include artificial intelligence, information security and big data.