Fast Secure Scalar Product Protocol with (almost) Optimal Efficiency

Youwen Zhu(✉), Zhikuan Wang, Bilal Hassan, Yue Zhang, Jian Wang, and Cheng Qian

College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
zhuyw@nuaa.edu.cn

Abstract. Secure scalar product protocol has wide applications for privacy-preservation in collaborative computation. In this paper, we propose a new secure scalar product protocol, which does not employ any public-key encryption and third party. Compared to scalar product computation without privacy-preservation, our proposed scheme introduces no extra communication overheads and little extra computation cost. That is, the new scheme can achieve almost optimal running efficiency, and thus is much applicable to privacy-preservation for large-scale data in collaborative computation. Theoretical analysis and evaluation indicate the security and efficiency of our scheme.

Keywords: Privacy preserving · Collaborative computation · Security · Scalar product protocol

1 Introduction

With the rapid development of computing devices and transmission mediums, distributed collaborative computation has become more and more popular, in which independent individuals/organizations could collaborate with each other to perform various computations on the union of data they hold such that they can achieve a comprehensive computation result.

Nevertheless, this collaborative computation paradigm also introduces several challenges, especially the data security and privacy concerns. For example, a company would like to evaluate the prospect of a project. To obtain an accurate result, the company might need the data of other institutions. Nevertheless, the other institutions may not want to disclose their, because their data might contain valuable business information and sensitive personal information, the disclosure of which will result in big losses or even violate some relevant law and regulation [1,2]. To respond this embarrassing situation, various privacy-preserving approaches have been put forward. Since being introduced to privacy preserving collaborative data mining by Lindell and Pinkas [3], secure multi-party computation (SMC) [4,5] is shown to be a useful instrument for preserving data privacy

S. Guo et al. (Eds.): CollaborateCom 2015, LNICST 163, pp. 234–242, 2016.
DOI: 10.1007/978-3-319-28910-6_21

in collaborative computation. SMC enables two or more participants to implement the collaborative computation on their dataset without revealing the data of a participant to anybody else, including other participants. That is, SMC can currently support collaborative computation and privacy-preservation.

For two n-dimensional vectors $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ and $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)$, the scalar product of them is the sum $\sum_{i=1}^{n} x_i y_i$, which is also called dot product. The scalar product computation is a common step in many applications, such as computing Euclidean distance [6], item similarity [7], or trust value [8]. While the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are holden by two different parties, it is a challenging problem to compute the scalar product of them without violating the privacy of any data. As one of most significant SMC protocols, scalar product protocol (SPP) aims at completing the challenging privacy preserving scalar product computation, i.e., computing the dot product and currently keeping each input vector private to its owner throughout the computation. SPP has been widely used in various privacy-preserving collaborative computation [6,9–13]. As being a fundamental role, an efficient SPP can boost the practical process of privacy preserving distributed collaborative computation.

Up to now, many schemes [14–20] have been proposed to perform the privacy preserving scalar product computation. Du and Zhan presented two schemes in [14]: dot product protocol employing a commodity server and scalar product protocol using random invertible matrix. Nevertheless, the former one requiring a third party, i.e., the commodity server, which will bring about fully privacy disclosure once the third party colludes with any participant. The latter does not need the third party, but takes $O(n^2)$ computation time which is not suitable for large-scale computation. Through algebraic transformation, another scalar product protocol was introduced in [15]. As yet, the scheme in [15] also needs $O(n^2)$ time. In [17,18], Zhu et al. discussed the relation of secure scalar product protocol and privacy preserving add to multiply protocol, but did not provide efficient solution for them. Based on the additively homomorphic encryption system, three solutions for securely computing dot product of private vectors are given in [16,19,21], respectively. As well known, homomorphic encryption system is quite expensive in real-world applications. Recently, a secret sharing-based scalar product protocol was presented by Shaneck and Kim [20]. Unfortunately, the solution also employs a third party, and while the third party colludes with a participant it will reveal the private data of the other participant. Lately, Zhu et al. [22,23] proposed an efficient approach to securely compute the scalar product while the dimension n is even. The state-of-the-art scheme can achieve $O(n)$ complexity without requiring any third party and public key encryption system.

In this paper, we investigate the fundamental and widely-used SPP. We observe that both computation cost and communication overheads of Zhu et al.'s scheme [22,23] can be dramatically reduced while sacrificing no security. Then, we proposed a new solution to SPP. Comparing with the state-of-the-art SPP scheme (which is also the fastest existing one) in [22,23], our proposed scheme requires less than half cost in computation and communication both, but keeps the same security. Generally, our main contributions in this paper are as follows.

- We present a new approach to preform scalar product computation on two private vectors of independent participants and currently preserve the data privacy of each party. We can dramatically reduce the computation and communication cost by more than 50 % while achieving the same security, compared to the sate-of-the-art one in [22,23].
- We take no extra communication overheads and little extra computation cost, comparing with computing the scalar product without any privacy-preservation. That is, we can attain almost optimal efficiency.
- Through theoretical analysis and evaluation, we indicate the security, correctness, and efficiency of our proposed scheme.

The rest of the paper is organized as follows. Section 2 introduces our system model, and discusses the state-of-the-art scheme. Then, Sect. 3 proposes our new scheme. Section 4 evaluates our proposed scheme through theoretical analysis and simulation experiments. At last, Sect. 5 concludes the paper.

2 System Model and Preliminaries

2.1 System Model

We consider a distributed collaborative computation model consisting of two participants: Alice and Bob. Here, Alice privately holds a vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ and Bob has the other private vector $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)$. In this paper, we assume n to be an even integer, i.e., we focus on the even-dimension SPP. Without loss of generality, suppose $n = 2 * k$ where k is a positive integer. It should be pointed out that any even-dimension SPP can be transformed into a general SPP, through the hybrid method in [23].

The object of SPP is that Alice attains a private number u and Bob receives a confidential output v while the private vectors are not disclosed to the other participant or anybody else. Besides, the output numbers u and v should satisfy the following Eq. (1).

$$\boldsymbol{x} \cdot \boldsymbol{y} = u + v \tag{1}$$

That is, $u + v = \sum_{i=1}^{n} x_i y_i$.

2.2 Threat Model

Generally speaking, SMC has two assumptions for the participant behaviors: semi-honest model, and malicious model. A semi-honest participant is also called to be honest-but-curious. Under the semi-honest model, each participant is assumed to correctly follow the steps of SMC protocol, but may keep a record about what he legally received to find out as much other participants' confidential information as possible. In contrast, a malicious participant might do anything in the collaborative computation. The work [4] has proved that any SMC protocol in semi-honest model can be transformed into a secure computation protocol in malicious model.

In this paper, we assume the participants to be semi-honest, i.e., they will exactly implement the protocol according to the specified steps. We also suppose the communication channels between the participants are secure and authenticated, which can be realized by conventional cryptography.

2.3 Discussion of the Sate-of-the-ART Scheme

Latley, Zhu et al. [22,23] put forward an efficient SPP (called EDSPP) for even-dimension private vectors, the detailed steps of which are shown in Protocol 1. In Step 1 of the protocol, for each $j = 1$ to k, the participants will totally generate 4 random numbers, complete 12 additions (including subtractions) and 6 multiplications, and send 6 numbers. Step 2 of the protocol contains $2k$ additions. Therefore, EDSPP will generate $4k$ random numbers, require $14k$ additions and $6k$ multiplications, and send $6k$ numbers.

The work [22,23] has shown that EDSPP will disclose $(x_{2j-1} + x_{2j})$ to Bob, and reveal $(y_{2j-1} - y_{2j})$ to Alice. Though the disclosed summation might reveal partial information about private input, the security still can be acceptable in some real-world applications shown in [22,23]. In this paper, we will propose a new even-dimension SPP with much higher efficiency and the same security.

Protocol 1. Even-Dimension Scalar Product Protocol (EDSPP) in [22]

Input: Alice has a private $2k$-dimension vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_{2k})$ and Bob holds another confidential $2k$-dimension vector $\boldsymbol{y} = (y_1, y_2, \cdots, y_{2k})$. $(k \in \mathbb{Z}^+,\ x_i, y_i \in \mathbb{R}, i = 1, 2, \cdots, 2k)$

Output: Alice obtains private output u and Bob securely gets v which meet $u + v = \boldsymbol{x} \cdot \boldsymbol{y} = \sum_{i=1}^{2k} x_i y_i$.

1: **Step 1**:
2: **for** $j = 1$ to k **do**
3: **Step 1.1**: Alice locally generates two random real numbers a_j and c_j such that $a_j + c_j \neq 0$. Then, she computes $p_j = a_j + c_j$, $x'_{2j-1} = x_{2j-1} + a_j$ and $x'_{2j} = x_{2j} + c_j$, and sends $\{p_j, x'_{2j-1}, x'_{2j}\}$ to Bob by a secure channel. Bob randomly generates two real numbers b_j and d_j which meet $b_j - d_j \neq 0$, and computes $q_j = b_j - d_j$, $y'_{2j-1} = b_j - y_{2j-1}$ and $y'_{2j} = d_j - y_{2j}$. Then, he securely sends $\{q_j, y'_{2j-1}, y'_{2j}\}$ to Alice.
4: **Step 1.2**: Alice locally calculates

$$u_j = y'_{2j-1}(x_{2j-1} + 2a_j) + y'_{2j}(x_{2j} + 2c_j) + q_j(a_j + 2c_j)$$

and Bob, by himself, computes

$$v_j = x'_{2j-1}(2y_{2j-1} - b_j) + x'_{2j}(2y_{2j} - d_j) + p_j(d_j - 2b_j).$$

5: **end for**
6: **Step 2**: Alice obtains $u = \sum_{j=1}^{k} u_j$ and Bob gets $v = \sum_{j=1}^{k} v_j$.

---

3 Our New Scheme

In this paper, we focus on securely computing the scalar product of two private even-dimension vectors. For simplicity of presentation, we will introduce our scheme by using two 2-dimensional vectors (i.e., $n = 2$). Our complete solution for any $2k$-dimensional vectors ($n = 2k,\ k > 0$) will be presented in the last part of this section.

While $n = 2$, Alice holds $\boldsymbol{x} = (x_1, x_2)$ and Bob has $\boldsymbol{y} = (y_1, y_2)$. To compute u and v which meets $u + v = \boldsymbol{x} \cdot \boldsymbol{y} = x_1y_1 + x_2 + y_2$, Alice and Bob can exchange one dimension with each other. Concretely, Alice sends x_2 to Bob, and Bob shares y_1 with Alice, then Alice can set $u = x_1y_1$ and Bob can attain $v = x_2y_2$. However, the simple interchange will violate the privacy of x_2 and y_1. Our secure scheme is achieved by improving the above simple approach.

We first transform the problem as follows. Let X be the 1×2 matrix (x_1, x_2), and Y be the 2×1 matrix $(y_1, y_2)^T$. Then, $\boldsymbol{x} \cdot \boldsymbol{y} = XY$.

Further, while M is a 2×2 invertible matrix, we have $\boldsymbol{x} \cdot \boldsymbol{y} = XY = (XM)(M^{-1}Y)$. Let $X' = XM$ and $Y' = M^{-1}Y$. If Alice and Bob shares the first dimension of X' and the second dimension of Y' with each other respectively, they can attain u and v. Through selecting appropriate matrix M, we can also preserve the privacy of both participants.

Here, we set the 2×2 matrix

$$M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix},$$

and correspondingly

$$M^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}.$$

Hence, $X' = XM = (x_1 + x_2,\ x_1)$ and $Y' = M^{-1}Y = (y_2,\ y_1 - y_2)^T$. Let Alice share $(x_1 + x_2)$ with Bob, and Bob give $(y_1 - y_2)$ to Alice. After that, Alice and Bob computes $u = x_1(y_1 - y_2)$ and $v = (x_1 + x_2)y_2$, respectively. Then, we have $u + v = (XM)(M^{-1}Y) = \boldsymbol{x} \cdot \boldsymbol{y}$. That is, we can complete the scalar product computation with merely disclosing $(x_1 + x_2)$ and $(y_1 - y_2)$, which achieves the same security with the work in [22,23]. More importantly, we require much less computation and communication cost.

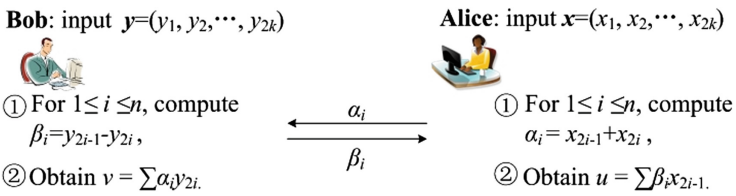

Fig. 1. Our Efficient Even-Dimension Scalar Product Protocol (ESPP)

Our complete scheme, called Efficient Even-Dimension Scalar Product Protocol (ESPP), is formally described in Protocol 2 . To vividly show our method, we also present our scheme in Fig. 1.

Protocol 2. Efficient Even-Dimension Scalar Product Protocol (ESPP)

Input: Alice's private $2k$-dimension vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_{2k})$,
Bob's confidential $2k$-dimension vector $\boldsymbol{y} = (y_1, y_2, \cdots, y_{2k})$.
($k \in \mathbb{Z}^+$, for all $j \in [1, 2k]$, $x_j, y_j \in \mathbb{R}$)

Output: Alice obtains private output $u \in \mathbb{R}$ and Bob securely gets $v \in \mathbb{R}$ which meet $u + v = \boldsymbol{x} \cdot \boldsymbol{y}$, i.e., $u + v = \sum_{j=1}^{2k} x_j y_j$.

1: **Steps:**
2: Alice and Bob set the initial values $u = 0$ and $v = 0$.
3: **for** $i = 1$ to k **do**
4: Alice computes $\alpha_i = x_{2i-1} + x_{2i}$, and Bob simultaneously sets $\beta_i = y_{2i-1} - y_{2i}$.
5: Then, Alice and Bob send α_i and β_i to each other.
6: At last, Alice locally calculates $u = u + x_{2i-1}\beta_i$, and Bob computes $v = v + \alpha_i y_{2i}$.
7: **end for**

4 Evaluation

4.1 Correctness

We consider the correctness of our scheme as follows.

For each $i = 1$ to k in Protocol 2 , we always have

$$x_{2i-1}\beta_i + \alpha_i y_{2i} = x_{2i-1}(y_{2i-1} - y_{2i}) + (x_{2i-1} + x_{2i})y_{2i}$$

$$= x_{2i-1}y_{2i-1} + x_{2i}y_{2i}.$$

Thus, $u + v = \sum_{i=1}^{k}(x_{2i-1}y_{2i-1} + x_{2i}y_{2i})$ in our Protocol 2 .

That is,

$$u + v = \sum_{j=1}^{2k} x_j y_j = \boldsymbol{x} \cdot \boldsymbol{y},$$

which completes our proof.

4.2 Security

It is easy to see that our scheme discloses nothing but $(x_{2i-1} + x_{2i})$ and $(y_{2i-1} - y_{2i})$. Thus, our scheme can achieve the same security with the existing work in [22,23]. The security has been analyzed by [22,23] in detail, and therefore we do not provide more detail about the security here.

Table 1. Comparison of cost

	EDSPP [22,23]	Non Privacy-preservation	Our Scheme
Addition	$14k$	$2k$	$4k$
Multiplication	$6k$	$2k$	$2k$
Communication	$6k\mathcal{B}$	$2k\mathcal{B}$	$2k\mathcal{B}$

4.3 Efficiency

Our protocol requires 4 additions and 2 multiplications, and sends 2 numbers. Thus, we need $4k$ additions and $2k$ multiplications, and sends $2k$ numbers in total.

Assume the bit length of each number is $\mathcal{B}$. In Table 1, we compare the cost of EDSPP in [22,23], our scheme, and the scalar product computation without privacy-preservation. It shows that our scheme is much more efficient than EDSPP [22,23], in both computation cost and communication overheads. Comparing with scalar product computation without privacy-preservation, our scheme introduces no extra cost apart from a few additions. Therefore, our scheme can achieve almost optimal efficiency for privacy-preserving distributed collaborative scalar product computation.

5 Conclusion

In this paper, we proposed a new even-dimension scalar product protocol, ESPP. Our proposed scheme can attain the same security with the state-of-the-art solution, while dramatically reducing the computation cost and communication overheads. Additionally, our scheme introduces no extra cost apart from a few additions, comparing with scalar product computation without privacy-preservation. It indicates that our scheme can achieve almost optimal efficiency for privacy-preserving distributed collaborative scalar product computation.

For the future work, we will devote to the formally secure SPP with high efficiency.

Acknowledgement. This work is partly supported by the Fundamental Research Funds for the Central Universities (No. NZ2015108), the Natural Science Foundation of Jiangsu Province of China (No. BK20150760), the China Postdoctoral Science Foundation funded project (No. 2015M571752), and the Jiangsu Planned Projects for Postdoctoral Research Funds (No. 1402033C). We want to thank Prof. Wei Yang for his helpful discussion with us.

References

1. HIPAA: The health insurance portability and accountability act of 1996, October 1998. http://www.ocius.biz/hipaa.html
2. Cios, K.J., Moore, G.W.: Uniqueness of medical data mining. Artif. Intell. Med. **26**(1–2), 1–24 (2002)

3. Lindell, Y., Pinkas, B.: Privacy preserving data mining. J. Cryptology **15**(3), 177–206 (2002)
4. Goldreich, O.: Foundations of Cryptography: Volume II, Basic Applications. Cambridge University Press, Cambridge (2004)
5. Lindell, Y., Pinkas, B.: Secure multiparty computation for privacy-preserving data mining. J. Priv. Confidentiality **1**(1), 59–98 (2009)
6. Xiao, M., Huang, L., Xu, H., Wang, Y., Pei, Z.: Privacy preserving hop-distance computation in wireless sensor networks. Chin. J. Electron. **19**(1), 191–194 (2010)
7. Agrawal, R., Imieliski, T., Swami, T.: Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 207–216 (1993)
8. Yuan, X., Huang, L., Yang, W.: Privacy preserving computation of trust-value in wireless sensor networks. In: IEEE 3rd International Conference on Communication Software and Networks, pp. 573–576 (2011)
9. Murugesan, M., Jiang, W., Clifton, C., Si, L., Vaidya, J.: Efficient privacy-preserving similar document detection. VLDB J. **19**(4), 457–475 (2010)
10. Chen, T., Zhong, S.: Privacy-preserving back-propagation neural network learning. IEEE Trans. Neural Netw. **20**(10), 1554–1564 (2009)
11. Bansal, A., Chen, T., Zhong, S.: Privacy preserving back-propagation neural network learning over arbitrarily partitioned data. Neural Comput. Appl. **20**(1), 143–150 (2011)
12. Zhu, Y., Huang, L., Dong, L., Yang, W.: Privacy-preserving text information hiding detecting algorithm. J. Electron. Inf. Technol. **33**(2), 278–283 (2011)
13. Smaragdis, P., Shashanka, M.: A framework for secure speech recognition. IEEE Trans. Audio Speech Lang. Process. **15**(4), 1404–1413 (2007)
14. Du, W., Zhan, Z.: A practical approach to solve secure multi-party computation problems. In: 2002 Workshop on New Security Paradigms, pp. 127–135. ACM, New York (2002)
15. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 639–644. ACM, New York (2002)
16. Goethals, B., Laur, S., Lipmaa, H., Mielikainen, T.: On private scalar product computation for privacy-preserving data mining. In: 7th Annual International Conference on Information Security and Cryptology, pp. 104–120 (2004)
17. Zhu, Y., Huang, L., Yang, W.: Relation of PPAtMP and scalar product protocol and their applications. In: IEEE symposium on Computers and Communications, pp. 184–189 (2010)
18. Zhu, Y., Huang, L., Yang, W., Li, D., Luo, Y., Dong, F.: Three new approaches to privacy-preserving add to multiply protocol and its application. In: Second International Workshop on Knowledge Discovery and Data Mining, pp. 554–558 (2009)
19. Amirbekyan, A., Estivill-Castro, V.: A new efficient privacy-preserving scalar product protocol. In: Sixth Australasian Conference on Data Mining and Analytics, vol. 70, pp. 209–214. Australian Computer Society (2007)
20. Shaneck, M., Kim, Y.: Efficient cryptographic primitives for private data mining. In: 2010 43rd Hawaii International Conference on System Sciences, pp. 1–9. IEEE Computer Society (2010)
21. Dong, C., Chen, L.: A fast secure dot product protocol with application to privacy preserving association rule mining. In: Tseng, V.S., Ho, T.B., Zhou, Z.-H., Chen, A.L.P., Kao, H.-Y. (eds.) PAKDD 2014, Part I. LNCS, vol. 8443, pp. 606–617. Springer, Heidelberg (2014)

22. Zhu, Y., Takagi, T., Huang, L.: Efficient secure primitive for privacy preserving distributed computations. In: Hanaoka, G., Yamauchi, T. (eds.) IWSEC 2012. LNCS, vol. 7631, pp. 233–243. Springer, Heidelberg (2012)
23. Zhu, Y., Takagi, T.: Efficient scalar product protocol and its privacy-preserving application. Int. J. Electron. Secur. Digit. Forensics **7**(1), 1–19 (2015)