Indian Journal of Science and Technology, Vol 9(14), DOI: 10.17485/ijst/2016/v9i14/74588, April 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Preservation of Private Information using Secure Multi-Party Computation

S. Bhanumathi[1*] and P. Sakthivel[2]

[1]Department of Computer Science and Engineering, Sathyabama University, Rajiv Gandhi Road, Jeppiaar Nagar, Chennai - 600119, Tamil Nadu, India; banujun8@gmail.com
[2]Department of Electronics and Communication Engineering, Anna University, Guindy, Chennai - 600025, Tamil Nadu, India; psv@annauniv.edu

Abstract

Background/Objectives: Secure Multi-party Computation (SMC) method is used to secure individual's sensitive data during privacy preserving data mining and data publishing. This paper proposes a new protocol using real and ideal models of SMC to compute sum of multiple parties' private data without revealing their data to each other. **Methods/Statistical Analysis**: Many approaches have been utilized for preserving privacy of sensitive data such as anonymization, data perturbation and SMC. In SMC, several protocols have been used for this purpose. Secure sum protocol is one of the important protocols, which is used to calculate the sum of private data secretly. Multiple parties perform the addition and subtraction operation based on the secure sum protocol and they transfer the intermediate sum among them through a trusted third party. Finally, trusted third party transforms sum to all the parties. **Findings:** The computation and communication cost is calculated in each round of computation and compared with the existing protocol. The empirical result shows that the proposed protocol out performed than the existing protocol in terms of computation and communication complexity. **Applications/ Improvements:** This protocol can be applied in various fields where the privacy preservation of sensitive data is needed such as insurance companies, banking system, government survey, hospitals, etc. The complexity of the protocol can be further reduced with high security in future.

Keywords: Computation and Communication Complexity, Privacy, Privacy Preservation, Secure Multi-Party Computation, Secure Sum Protocol

1. Introduction

In recent years, various methods have been employed to protect sensitive data, for example, perturbation[1,2], anonymization[3,4], encryption[5]. All these methods have their own advantages and disadvantages. On the other hand, the Secure Multi-party Computation (SMC) method is mainly applied on data when multiple parties want to find the sum of private data securely. The major purpose of studying SMC is to design systems that increase information utility without compromising the privacy of secret data. SMC issue is not a problem of the single party, it is the problem of multiple parties i.e. n parties[6,7]. There is the possibility of some party maliciously act and try to know information about other parties during the secure sum computation.

At present, there are two models: real model and ideal model[8]. In real model, whole computation is done by a Trusted Third Party (TTP), parties send their data in a secure manner to TTP. Numbers of protocols are proposed by researchers for this model. Other than this real model, the ideal model does not use TTP for computation. Computations are done by parties itself and share their data with each other in a secure manner. To preserve privacy, party can encrypt or segment their private data.

*Author for correspondence

Several techniques are proposed by many researchers for sharing data with each other or with TTP in both models. There are so many practical examples where privacy of data is the main concern[9]. For example, in insurance companies, they wish to calculate how many persons are insured and at the same time they do not want to reveal their number of customers. Using SMC, total number of persons insured is calculated. Another example is if two banks cooperatively want to know the details about some customers, but no bank is willing to disclose the private data of the customer to other bank due to the privacy concern of the customer and policy of the bank.

A protocol is presented for SMC using real model. In this, party's data are segmented and distributed among them for computation. This protocol does not perform well for three parties, if any party behaves like malicious party[10]. In 2010, the modified ck-secure sum protocol is developed which gives more security compare to the previous protocol. The problem of this protocol is change of computation network topology in every round[11,12]. After this Mishra et al. proposed several protocols for SMC using ideal model. They have first implemented two layered framework and then it is extended by adding a third layer that is an anonymizer layer to hide the identity of parties. Moreover, they added a fourth layer which is a packet layer to protect data from malicious parties[13].

A hybrid protocol is proposed to compute sum of multiple parties' securely. The parties' data is segmented and random number is used to protect their data. In this protocol, communication and computational complexity is high[14]. Zhang and Cai[15] suggested collusion free rational secure sum protocol in which they combined SMC with game theory to provide more security. Jung et al[16]. proposed collusion tolerable method for secure sum and product calculation. The major issue of SMC in the existing protocol is complexity. In this paper, a new protocol is proposed to reduce communication and computation cost needed for finding secure sum.

2. Proposed System Framework

The entire system framework is shown in Figure 1. Initially the party (user) begins with registering the details and the party's individual data is segmented. The segmentation of the inputs is decided by the party depending on the party's individual's inputs. Once the data is segmented

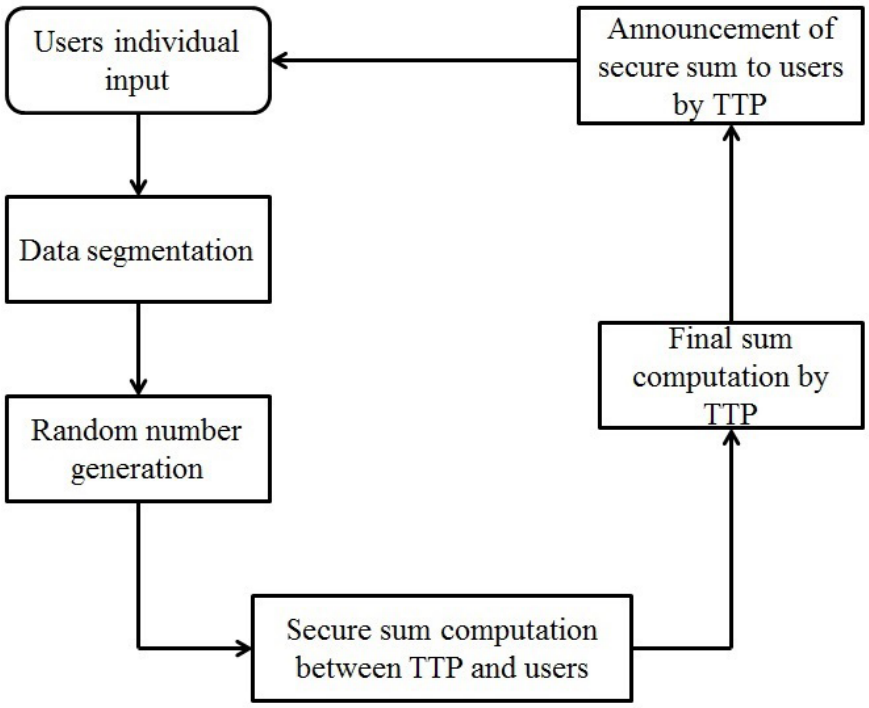

Figure 1. Proposed system framework.

each party generated the random numbers and combined each segmented data with separate random number and sent to TTP during the first round. The segmented data is given to TTP, the TTP applied a secure sum over the segmented data with the party's and finally TTP announces the combined result to all the parties. If any party turns to be malicious to trace other party data in such case only combined result will be retrieved by the malicious party without revealing the private inputs of the particular party.

2.1 Secure Sum Protocol Algorithm

(i) Each party divided their private data into n segments and generated random number for each segment.
(ii) Each party sends sum of their first segment with random number to TTP and TTP sends new sum to P_1.
(iii) Party P_1 performs subtraction and addition operation to find new sum and sends it to P_2. This process will be repeated until P_n.
(iv) Party P_n performs same operation as step3 in reverse order and finally party P_1 sends new sum to TTP and then it sends the same to P_n.
(v) Party P_n to P_1 performs subtraction and addition operation to find new sum and sends it to TTP. TTP announces this as the final sum to all parties.

3. Experiments

The proposed framework used four different PC in which three considered for parties and one considered for TTP. Java is used for implementation of proposed protocol to calculate secure sum. The computational and communicational cost is compared with the hybrid protocol[14].

3.1 Data Segmentation and Random Number Generation

The sensitive data of the party are divided into segments by the party. The party decides on the decision of segmenting the data. The main purpose of segmenting the data is to provide more privacy to the party's private inputs. Moreover, the TTP also cannot obtain the individual data of the party. Once the data is segmented, the segmented data are combined with a random number to increase the degree of privacy. Random number generator is used to generate random numbers for each segment. Generally three random numbers are generated for all parties segmented data. Random numbers are used to protect party's private data with more secrecy. Due to the addition of random numbers even the TTP cannot hack or trace the individual data of any party. The segmented data are sent in secured manner to TTP.

3.2 First Round Computation of Secure Sum

During the first round of computation all the first segments of all parties are added with a random number and sent to TTP for combined computation. The TTP combined the data along with random numbers and sends the result to P_1. Three parties and a TTP are considered in the experiment for simplicity. For example, let the individual data of P_1 is 600, P_2 is 700, and P_3 is1000. These data are segmented and random numbers are produced for each segment. Now, the entire first segment is added with the corresponding random number and sent to TTP. Then, TTP added all the segments and sent sum to P_1.

3.3 Second and Third Round Computation of Secure Sum

In the second round, the party P_1 subtracted the first random number from the sum sent by TTP and added the second segment and a second random number to sum and sent the sum to P_2. The party P_2 also performed the same task as P_1. This process is repeated until the nth party to complete the second round. During the third round, the computation is done in the reverse order i.e. the nth party sent the sum to P_2, the second random number is subtracted by P_2 and combines the third segment and random number to the sum. Then, the sum is sent to P_1 where the P_1 also performed the same task as the P_2 in reverse order.

3.4 Final Round Computation of Secure Sum

In the final round, the sum is received by P_3. As soon as P_3 received the sum, the second random number is subtracted from the sum and added the third segment with random number and the sum is sent to P_2 where it subtracted the third random number and sent the sum to P_1. The third random number is subtracted by P_1 and the correct sum is sent to TTP. Finally, the TTP announced the final sum to all the parties in the system. From the combined result, none of the parties can predict the amount of individual data of any party in the system. Figure 2 shows private data segments of user 1 (party 1) and secure sum which is announced by the TTP.

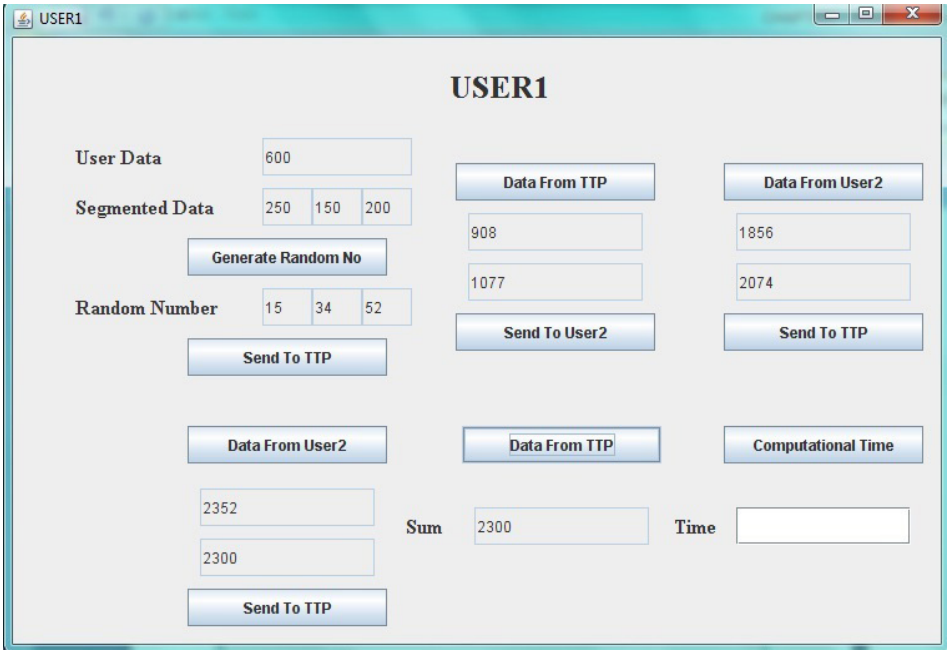

Figure 2. TTP transfers secure sum to user 1.

4. Performance Analysis

During the execution of the new protocol, if any one party and TTP become malicious means the parties will know only its own data and TTP will know the segments of all parties involved in computation, but the segments of each party will be wrapped with any random number generated by each party. So in this case there is no chance for any party or either TTP to obtain the individual input. There is also another case in which two parties can become malicious, if it happens each party cannot obtain data of another party because the individual data of each party is divided into segments and each segment is transferred in a secure manner. With each segment, the random number is added and subtracted by the respective parties itself in each round. If every party follows the new protocol honestly, it runs in a smooth fashion for obtaining the sum. From the entire implementation of our protocol, it is found that the protocol provides more security and privacy of individual inputs with zero data leakage at all rounds. The computational time and communicational time are also calculated.

4.1 Computational Complexity

The computational complexity is represented in the form of a computational time, which is defined as the time is taken by the party or TTP for calculation at the end of each round. During the end of the first round the computational time is estimated by the TTP. The computational time is approximately calculated to be less than 1. During the second round of computation, the computational time for all parties is calculated in a clockwise direction, i.e., from P_1 to P_n. In the third round of computation, the computational time among the party is calculated in reverse order from $P_{(n-1)}$ to P_1. In the fourth round, the protocol works in an anticlockwise direction and computational time is calculated from P_n to P_1. The computational time predicted in each round and numbers of rounds are represented in x-axis and y-axis, respectively in which x-axis contained 4 rounds and y-axis maintained the computational time, which appeared within the range of 3n.

The computational complexity of our secure sum protocol is represented in Figure 3. The computational time of the system changes dynamically during the execution of each round. In the first round, each and every user in the system sent the first segment to TTP and TTP combined the segments. The computational time is calculated to be 0.13mins. For the period of second round, the computational time is calculated from P_1 to P_n is 2.08mins. During the third round of the secure sum computation the time is calculated from P_2 to P_1 which is calculated as 1.99mins. The computational time in the final round is estimated as 2.08mins. If the parties involved in the system are 3, the computational complexity of the new secure sum protocol

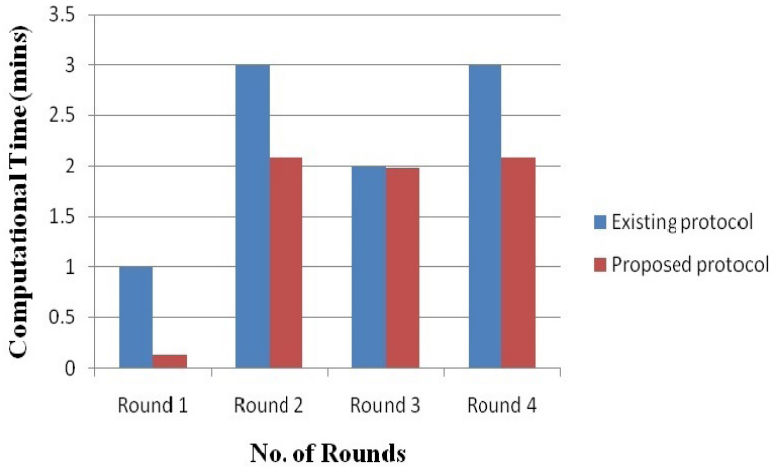

Figure 3. Computational complexity for each round.

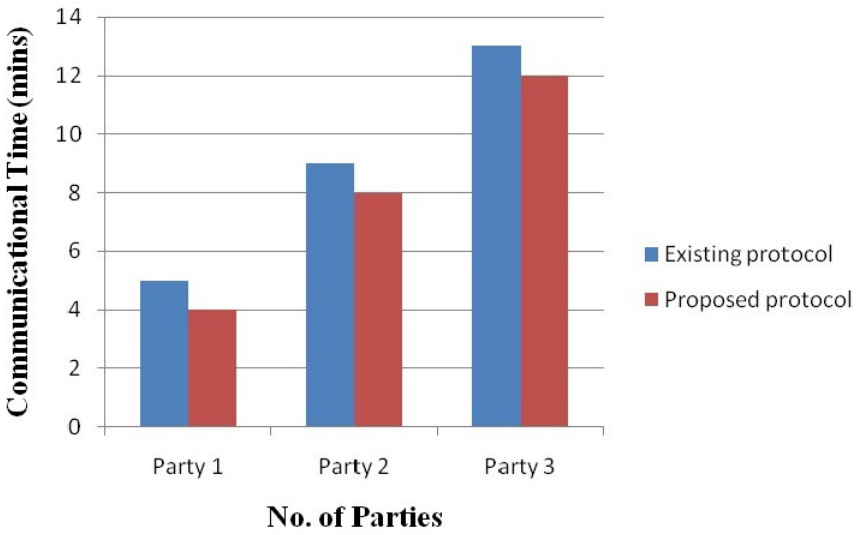

Figure 4. Communicational complexity for each party.

is computed as 7. Generally, (3n-2) is the computational complexity of our protocol. Hence, our protocol reduced the computational complexity.

4.2 Communicational Complexity

The communicational complexity is represented in terms of communication cost. The communication cost is computed for entire computation at the completion of entire rounds. It defines the number of communications between the parties and TTP. The communication cost is calculated to be m*n where m represents the number of rounds and n denotes the number of parties. If the number of parties increased, the communicational cost also can be increased. The communicational complexity is 4n where number of round is 4. The communicational time complexity for existing and proposed system is shown in Figure 4.

5. Conclusion

SMC is the important tool to protect the data when multiple parties want to compute the sum. For doing this task, protocols are required. In this paper, a new protocol is suggested using real and ideal model for computing secure sum. It protected the private data with zero leakage. There is no chance to know about the private data of

the party when some party acts as a malicious party. Also, it reduced computational and communicational complexity compared to the existing protocol. The complexity of the protocol can be further reduced with high security in future.

6. References

1. Selvan PT, Veni S. Social Ant based Sensitive Item Hiding with Optimal Side Effects for Data Publishing. Indian Journal of Science and Technology. 2016 Feb; 9(2):1–9.
2. Priyadarsini RP, Valarmathi ML, Sivakumari S. Attribute Segregation based on Feature Ranking Framework for Privacy Preserving Data Mining. Indian Journal of Science and Technology. 2015 Aug; 8(17):1–9.
3. Irudayasamy A, Arockiam L. Parallel Bottom-up Generalization Approach for Data Anonymization using Map Reduce for Security of Data in Public Cloud. Indian Journal of Science and Technology. 2015 Sep; 8(22):1–9.
4. Rajalakshmi V, Mala GA. Anonymization by data relocation using sub-clustering for privacy preserving data mining. Indian Journal of Science and Technology. 2014 Jul; 7(7):975–80.
5. Mohan M, Devi MK, Prakash VJ. Security Analysis and Modification of Classical Encryption Scheme. Indian Journal of Science and Technology. 2015 Jul; 8(14):542–48.
6. Yao AC, Protocol for Secure Computations. Proceedings of the 23rd Annual IEEE Symposium on Foundation of Computer Science. 1982; p. 160–64.
7. Aggarwal CC, Yu PS. Springer: US: Privacy-Preserving Data Mining Models and Algorithms. 2008; p. 11–52.
8. Prabhakaran MM. Netherlands: IOS Press: Secure Multi-party Computation. 2013.
9. Du W, Atallah MJ. Secure Multi-party Computation and Applications. Cloudcroft, New Mexico, USA: Proceeding of New Security Paradigm Workshop. 2001; p. 13–22.
10. Sheikh R, Kumar B, Mishra DK. A Distributed k-Secure Sum Protocol for Secure Multi-party Computation. International Journal of Computer Science and Information Security. 2009; 5(2):184–88.
11. Sheikh R, Kumar B, Mishra DK. Changing Neighbors k-Secure Sum Protocol for Secure Multi-party Computation. International Journal of Computer Science and Information Security. 2010; 7(1):239–43.
12. Sheikh, R, Kumar B, Mishra DK. A Modified ck-Secure Sum Protocol for Multi-Party Computation. International Journal of Computer Science and Information Technology. 2010; 2(2):62–66.
13. Mishra DK, Naha K, Nikhil K, Ravish B. A secure multi-Party computation protocol for malicious computation prevention for preserving privacy during data mining. International Journal of Computer Science and Information Security. 2009; 3(1):79–85.
14. Priyanka J, Gajendra SC, Mishra DK. Hybrid Technique of Secure Sum Protocol. World of Computer Science and Information Technology Journal (WCSIT). 2006; 1(5):198–201.
15. Zhang E, Cai Y. Collusion-Free Rational Secure Sum Protocol. Chinese Journal of Electronics. 2013; 22(3):563–66.
16. Jung T, Li XY, Wan M. Collusion-Tolerable Privacy-Preserving Sum and Product Calculation without Secure Channel. IEEE Transactions on Dependable and Secure Computing. 2014; 12(1):45–57.