IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 21, Issue 2, Ser. II (Mar - Apr 2019), PP 71-78
www.iosrjournals.org

Implementing Token Slicing Scheme for Secure Multiparty Communication

Amit Pandey[1], Abdella K. Mohammed[1]
[1]Faculty of Informatics, Hawassa University, Ethiopia
Corresponding Author: Amit Pandey

Abstract*: In today's competitive world of industries and companies number of times we need to draw comparisons and analyses on the data that the owner is unwilling to share commonly. To overcome this problem of secure data communication within multiparty over the network there are many algorithms and techniques. Some of them use anonymous layers as trusted third party and some use complex functions for data manipulation and have time complexity in exponential order. In this paper a new token slicing scheme is introduced which serves the purpose of secure data communication within multiparty over the network and has runtime complexity of order $O(n^2)$. Also one of the most striking things about this privacy preserving data mining algorithm is its durability. It can withstand the trade fraud of up to (N-2) participants out of N participating parties in the transaction, that is for N number of participants the data transmitted on the network between the participants is safe even all N-2 participants will try to group together to reveal the data for any particular participant, considered that N should be greater than two.*

Keywords: *SMC, Secure Multiparty Computation, privacy preserving data mining, Token Slicing, Message splitting, Message partitioning*

Date of Submission: 27-03-2019 Date of acceptance: 11-04-2019

I. Introduction

Consider a scenario in which there is a group of companies that produces two wheelers. Suppose they wanted to calculate the total consumption of two wheelers in any current financial year, but no one is willing to disclose his individual annual sales. This can be considered as the typical example of secure multiparty communication problem where a secure way of data communication is needed between the participants to calculate the total sum but individual data of any participating entity should not be revealed to any other participants. So we have to communicate the data securely between all the participants on network without revealing one's data to any other participant.

In this paper we are proposing a data mining algorithm for secure data communication between the participating entities on a network using which they can get the sum of collective data in end without disclosing their particular values to any other participants and solving the problem of secure multiparty communication. Our algorithm works for any N number of participants greater than two because in calculating the total sum of individual data between two participants the second participant's data will automatically be revealed to the first participant, no matter in which manner the sum is calculated.

In this algorithm we implement the secure multiparty communication by partitioning the message token and then distributing each partition within participants. Then we sum all the message partitions at one central location, the participant who is elected as leader to get the sum of collective data without leak of any individual data. Here the process of leader election [1] has the communication and time complexity both of order O(n) and also the message partition's distribution has the communication complexity of order of O(n2), making overall complexity for the algorithm of the order O(n2).

II. Related Work

Yao in 1982 [2] pioneered the idea of secure communication between the parties by providing the solution for the millionaires' problem, that is considering there are two parties Alice and Bob with their respective values A and B and both of them wanted to know that whose value is greater without disclosing their value to other party. Yao suggested use of some encryption functions by both the parties and then their submission to the algorithm for calculating the result. But in achieving that solution the algorithm acquires an exponential order time complexity. This was improved by Ioannidis et al. [3] by providing more efficient algorithm for millionaires' problem with the time complexity of the order $O(d^2)$. Here d is the number of input bits.

Later Beaver et al. [4] gave a cryptography based algorithm for secure multiparty computation with constant rounds. Feige et al. [5] suggested a model for secure multiparty computation involving third party. In this model the result was calculated by third party considering it as trusted party which will not learn anything by itself and never leak the data to any other party. In the same referred model, Cachin and Camenisch [6,7] induced new concept that third party will only interfere when any party will misbehave. Further Mishra et al. [8] introduced the concept of anonymous layer as trusted third party in secure multiparty computation.

Further, Rabin Proposed an EOS protocol for exchanging secrets with oblivious transfer [9]. This transfer was based on the use of private public key cryptography. That is the participating entities provide the authentication and encryption using their private and public keys.

In 1994 Benaloh [10] and also in 1999 Paillier [11] has used this harmonic encryption in their work. Some recent work on this topic also includes harmonic encryption [12] which permits performance of a specific algebraic operation on a plain text by performing an algebraic operation on the cipher text.

III. Proposed Algorithm

Consider a scenario where there are N participants as P_0, P_1, P_2, - - - P_{N-2}, P_{N-1} with their respective private data values as D_0, D_1, D_2, - - - D_{N-2}, D_{N-1}. Here N is greater than two. Now for calculating the total sum of individual data values without disclosing private value of any participant to any other participant we should follow the steps below.

1) Each participant Pi for i = 0, 1, - - - (N-1), partitions its data token Di into N unequal parts D_{i0}, D_{i1}, - - - $D_{i(N-1).}$ Such that, for each data token,

Di $= \sum_{j=0}^{(N-1)}$ Dij

2) Now each participant sends one partition of its data token to all other remaining (N-1) participants keeping one for it.
3) Now each participant will sum up all the N partitions of data token it is having. As,

Sumi = $\sum_{i=0}^{(N-1)} \mathbf{D}$ij

4) Select a leader among the participants, using a leader suitable election algorithm.
5) Now each participant will send its Sumi value to the elected leader.
6) The elected leader will receive Sumi values from all the other (N-1) participants and add it together to get the collective sum of all data tokens. As,

Collective Sum = $\sum_{i=0}^{(N-1)} \mathbf{Sum}_i$

1. Validity of Algorithm

All Here for given N = 4 participants as P0, P1, P2, P3 with their respective private data values as D0, D1, D2, D3. We can process this algorithm in following manner,

STEP 1: Partition each data token Di into N unequal partitions. Here N = 4 (see table 1).

Table 1: Partitioning each participant's private data.

	D_{i0}	D_{i1}	D_{i2}	D_{i3}
P_0	D_{00}	D_{01}	D_{02}	D_{03}
P_1	D_{10}	D_{11}	D_{12}	D_{13}
P_2	D_{20}	D_{21}	D_{22}	D_{23}
P_3	D_{30}	D_{31}	D_{32}	D_{33}

STEP 2: now after distributing these partitions of data tokens one to each participant, Sumi is calculated (see Table 2).

Table 2: Distributing partitioned tokens to all participants and Calculating their sum.

	P_0	P_1	P_2	P_3
Sum_i	D_{00}+ D_{12}+ D_{22}+ D_{32}	D_{02}+D_{11}+ D_{20}+ D_{33}	D_{01}+ D_{10}+ D_{21}+ D_{31}	D_{03}+ D_{13}+ D_{23}+ D_{30}

STEP 3: Let P1 is elected as leader. Then all other participants will send their respective Sumi values to the elected leader and leader will sum them all to get the collective sum. Here in calculation of collective sum of data the privacy of any participant's data was never compromised (see table 3).

Table 3: Calculation of collective sum by leader.

	COLLECTIVE SUM
P_1	$(D_{00}+ D_{12}+ D_{22}\ D_{32}) + (D_{02}+D_{11}+ D_{20}\ D_{33}) +$ $(D_{01}+ D_{10}+ D_{21}\ D_{31}) + (D_{03}+ D_{13}+ D_{23}+ D_{30})$ $= (D_{00}+ D_{02}+ D_{01}+ D_{03}) + (D_{10}+ D_{11}+ D_{12}+ D_{13}) +$ $(D_{20}+ D_{21}+ D_{22}+ D_{23}) + (D_{30} + D_{31}+ D_{32} + D_{33})$ $= D_0 + D_1 + D_2$

Suppose even if P1 and P2 get involved in trade fraud and share their data to reveal the private value of P0, then also it is not possible for them to get it because they have only two of the four tokens from P0. This algorithm can withstand a trade fraud of up to (N – 2) participants.

2. Working Example

To understand how this algorithm works we can take a case of four participants who wanted to calculate the total sum without sharing their private values, Let their respective private values are P_0 = 35, P_1 = 43, P_2 = 28 and P_3 = 34 (see fig 1).

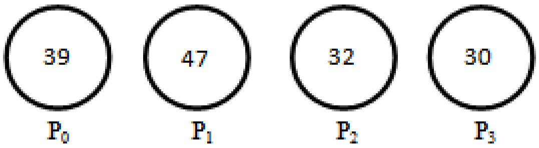


Fig 1. Four participants with their private values.

Now each participant will split its data into N unequal parts as shown in the figure (see fig 2), so that it is impossible to guess the value of any one part on the basis of any other partition,

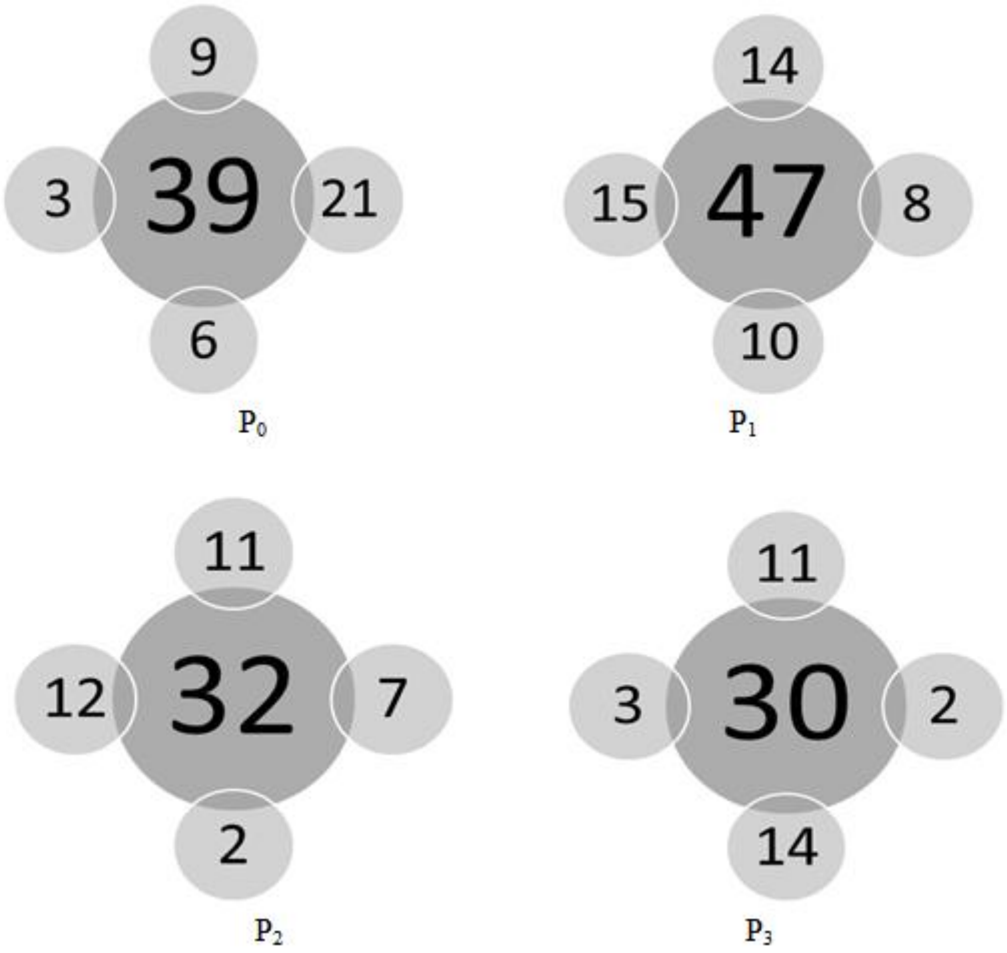


Fig 2. Each Participant splits its data in to N unequal parts.

Now each participant will send one partitioned token of its private value to all other remaining (N-1) participants keeping one for it. Below is the distribution for P_0 (see fig 3).

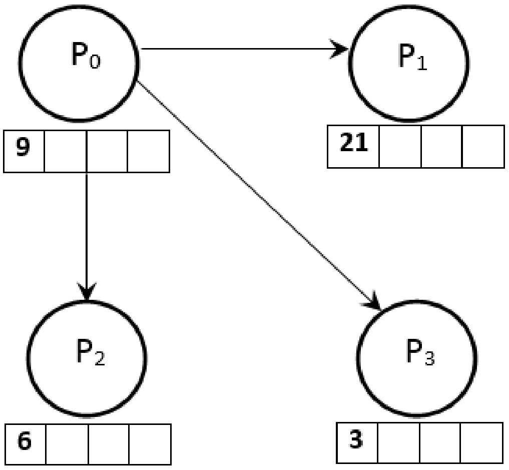


Fig 3. P_0 sending its partitioned tokens to other (N-1) participants.

Below is the distribution for P_1 (see fig 4).

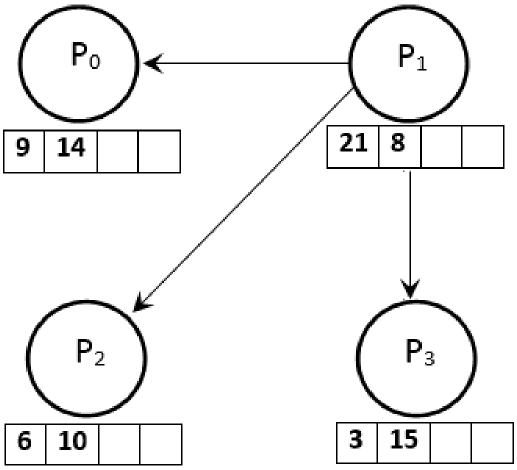


Fig 4. P_1 sending its partitioned tokens to other (N-1) participants.

Below is the distribution for P_2 (see fig 5).

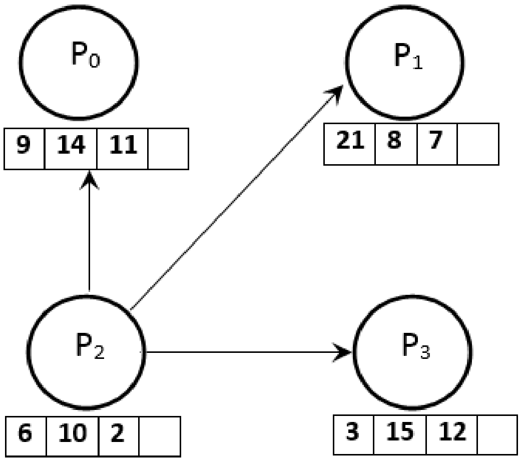


Fig 5. P_2 sending its partitioned tokens to other (N-1) participants.

Below is the distribution for P_3 (see fig 6).

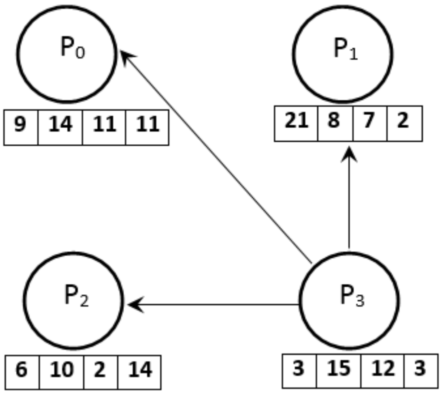


Fig 6. P_3 sending its partitioned tokens to other (N-1) participants.

Now each participant will sum up all the data values received by them, as shown in the figure below (see fig 7).

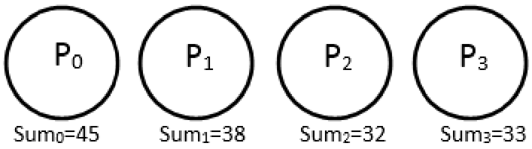


Fig 7. Calculation of individual sum by each participant.

Let P_1 is selected as leader now and all individual sums are converged to the leader to calculate the required total sum (see fig 8).

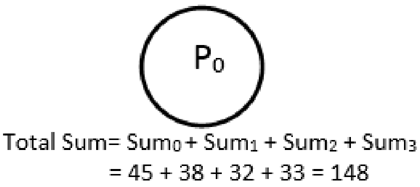


Fig 8. Calculating the required total sum at the leader.

Finally using this algorithm the required total sum is calculated at the elected leader's location without revealing the private value of any participant.

3. Illustration

Consider a general case where number of participants N is 4 such that N >2. These four participants P_0, P_1, P_2 and P_3 have their yearly sales data for a particular product type as shown below (see table 4).

Table 4: Participants with their private data values.

	Sales Data (D)
P_0	39
P_1	47
P_2	32
P_3	30

Now the data from each participant is partitioned in to four unequal parts or tokens (see table 5).

Table 5: Partitioning each participant's private data.

	D_{i0}	D_{i1}	D_{i2}	D_{i3}
P_0	*9*	*21*	*6*	*3*
P_1	*14*	*8*	*10*	*15*
P_2	*11*	*7*	*2*	*12*
P_3	*11*	*2*	*14*	*3*

Now from each participant one data token is sent to all other participants (see table 6).

Table 6: Distributing partitioned tokens to all participants.

	From P_0	From P_1	From P_2	From P_3
P_0	*9*	*14*	*11*	*11*
P_1	*21*	*8*	*7*	*2*
P_2	*6*	*10*	*2*	*14*
P_3	*3*	*15*	*12*	*3*

Now each participant will sum all the data received by it (see table 7).

Table 7: Calculating sum of tokens at each participant.

	P_0	P_1	P_2	P_3
Sum_i	9 + 14 + 11 + 11 = 45	21 + 8 + 7 + 2 = 38	6 + 10 + 2+ 14 = 32	3 + 15 + 12 + 3 = 33

Now each participant has sum of data tokens received from other participants. But by using this information none of the participant can reveal the data for annual sales of any other participant. Now a leader is elected among the participants, suppose P_1 is elected as leader. Then each of the participants will send the sum of its received data tokens to the elected leader to calculate the total sum (see table 8).

Table 8: Calculation of collective sum by leader.

	COLLECTIVE SUM
P_1	45 + 38 + 32 + 33 = 148

Finally the total sum of annual sales is calculated without revealing the data of any individual participants. As we can see that,

$$(45 + 38 + 32 + 33) = (39 + 47 + 32 + 30) = 148$$

4. Limitations

In this approach there lies some limitations, which are needed to be treated carefully for protection of the data.

7.1 When number of participants N = 2

This approach is vulnerable when number of participants is two. Since being only two participants when the elected leader will get the collective sum of the data it can subtract its own data value from collective sum to obtain the other remaining data value which is the private data of the second remaining participant.

7.2 When all (N – 1) participants are involved in trade fraud

If all (N – 1) participants are ready to exchange their data with each other. Then in this case the private data of the remaining participant can be revealed. As all (N – 1) participants are behaving as one group and remaining participant is the second group. So it will be the same case as of two participants.

5. Complexity

In this algorithm first all N participants partition their data token and then distribute them in to (N-1) other participants. During this process they are doing N x (N-1) communications, making the communication complexity of order of $O(n^2)$. Then each participant sum up the received partitions of data tokens and further send their respective sum values to the elected leader in (N-1) communications. The communication complexity for the process of leader election [1] is also of the order O(n). Thus the overall complexity for the token slicing algorithm turns out to be of the order of $O(n^2)$.

6. Implementation and Results

In this part we have shown the practical implementation of the current approach. Below are the screenshots from the implemented application. Fig. 9 shows the interface where each user enters his individual value.

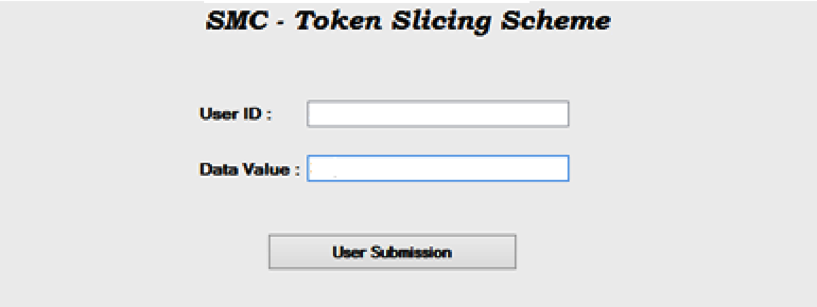
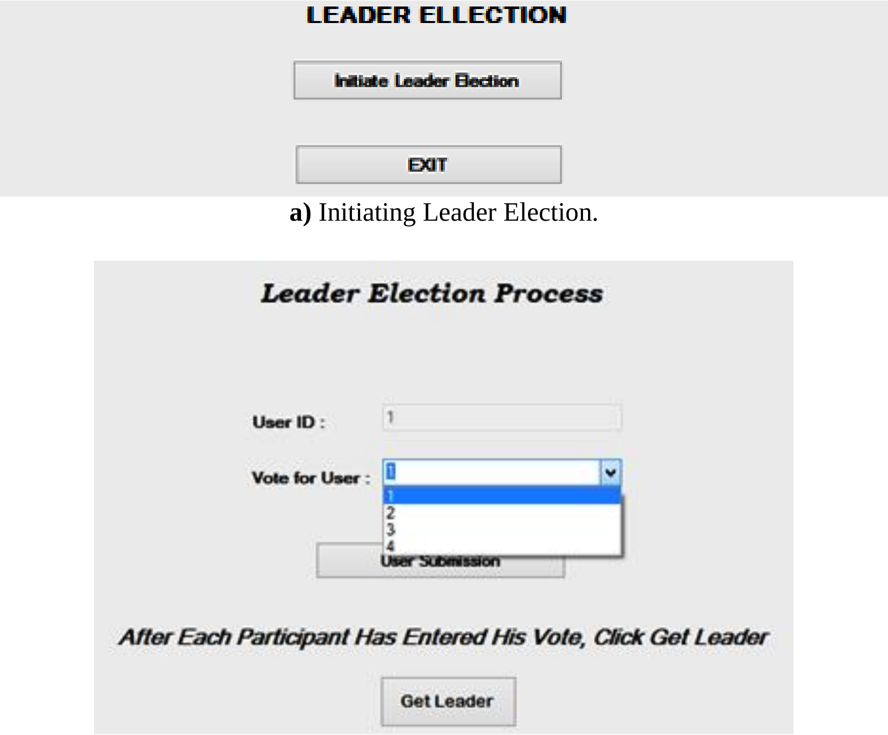


Fig 9. Interface for entering participant's value.

Fig. 10 (a) shows the interface for initiating the leader election process. Further, Fig. 10 (b) shows the voting process for leader election.



User ID : 1

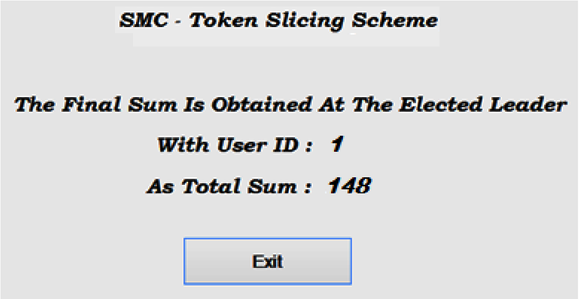
After Each Participant Has Entered His Vote, Click Get Leader

a) Initiating Leader Election.

b) Voting for Leader Election.

Fig 10. Leader Election Process

Finally, Fig. 11 shows the final value calculated at the elected leader. It can be easily verified that this value is same as the total sum of individual values of each participating user.



The Final Sum Is Obtained At The Elected Leader

With User ID : 1

As Total Sum : 148

Fig 11. Calculating the required total sum at the leader.

Fig. 12 shows the graphical representation of values accumulated after implementation, confirming the O(n^2) nature of its running time complexity.

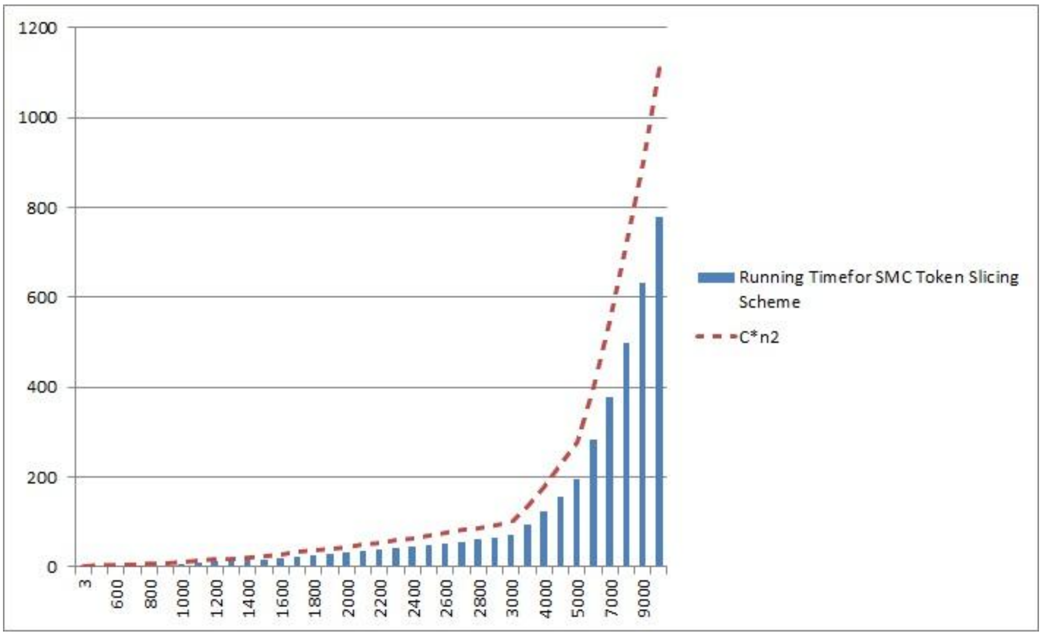


Fig 12. Graph verifying O(n^2) complexity of the algorithm.

IV. Conclusion

In this paper we have presented the token slicing algorithm for secure multiparty communication, with communication complexity of the order O(n^2).

This token slicing algorithm can work on any number of N participants, where N is greater than two and can withstand the trade fraud of up to (N-2) participants.

Also there is no need of trusted third party for calculating the collective sum in this algorithm, making it more reliable and trustworthy for practical implementations.

References

[1]. J. Villadangos, A. C´ordoba, F. Fari˜na and M. Prieto, Efficient leader election in complete networks, *in proceedings of Euromicro-PDP*, 2005, 136 – 143.

[2]. A. C. Yao, Algorithms for Secure Computations, *in proceedings of Symposium on Foundations of Computer Science* , 1982, 160 – 164.

[3]. I. Ioannidis and A. Grama, An Efficient Algorithm for Yao's Millionaires' Problem, *in Proc. Hawaii International Conference on System Sciences*, 2003.

[4]. D. Beaver, S. Micali, and P. Rogaway, The round complexity of secure algorithms, *in Proc. 22nd Annual ACM Symposium on Theory of Computing (STOC)*, 1990, 503- 513.

[5]. U. Feige, J. Kilian, and M. Naor, A minimal model for secure computation, *in Proc. 26th Annual ACM Symposium on Theory of Computing (STOC)*, 1994, 554-563.

[6]. C. Cachin, Efficient Private Bidding and Auctions with an Oblivious Third Party, *in Proc. 6th ACM Conference on Computer and Communications Security (CCS)*, 1999, 120 – 126

[7]. C. Cachin and J. Camenisch, *Optimistic Fair Secure Computation*, In Mihir Bellare, editor, Advances in Cryptology: CRYPTO 2000, Vol. 1880, Lecture Notes in Computer Science, 2000, 94-112.

[8]. D.K. Mishra and M. Chandwani, Anonymity Enabled Secure Multiparty Computation for Indian BPO, *in Proceeding of the IEEE Tencon 2007: International conference on Intelligent Information Communication Technologies for Better Human Life*, Taipei, Taiwan, 2007, 52-56.

[9]. M. O. Rabin, *How To Exchange Secrets with Oblivious Transfer*. IACR Cryptology ePrint Archive 2005, 187.

[10]. J. Benaloh, Dense probabilistic encryption. *In proceedings of the workshop on selected areas of cryptography*. 1994.

[11]. P. Paillier, *Public-key cryptosystems based on composite degree residuosity classes*. Advances in cryptology—EUROCRYPT'99. Springer Berlin Heidelberg, 1999.

[12]. V. A. Oleshchuk and V. Zadorozhny. Secure multi-party computations and privacy preservation: Results and open problems. *In Proceedings of TELEKTRONIKK*, 2007, 20.

Amit Pandey. "Implementing Token Slicing Scheme for Secure Multiparty Communication" IOSR Journal of Computer Engineering (IOSR-JCE) 21.2 (2019): 71-78.