ISSN:0975-9646

A Method for Privacy Preserving Data Mining in Secure Multiparty Computation using Hadamard Matrix

Ms. Neha Pathak ,Prof. Anand Rajavat

Computer Science Department ,
SVITS,
Indore, India

Abstract- **Secure multiparty computation allows multiple parties to participate in a computation. SMC (secure multiparty computation) assumes n parties where n>1. All the parties jointly compute a function. Privacy preserving data mining has become an emerging field in the secure multiparty computation. Privacy preserving data mining preserves the privacy of individual's data. Privacy preserving data mining outputs have the property that the only information learned by the different parties is only the output of the algorithm. In this paper, we use a mathematical function hadamard matrix. All the computation multiplied by the hadamard matrix. Using this, security and privacy of the individual's data increased. Thus, we can say that this protocol fulfill the requirement of privacy and security.**

Keywords: Privacy preservation, secure multiparty computation (SMC), secures sum protocol, hadamard matrix.

I. INTRODUCTION

A data mining process extracts useful information from the huge amount of information available. Privacy preserving data mining is one of the most recent research areas of the data mining research. Privacy preserving data mining allows the sharing of the data between parties, but at the same time also preserves the privacy of the data. Privacy preserving data mining used to extracts the knowledge without divulging any information.

The purpose of secure multiparty computation is to give security and privacy. This achieved when at the end of calculation no party knows anything else other than the result. One way is to use trusted third-party. Each party shares its data with trusted party and all the calculations done with this party. The other way is to distribute the segments between the parties.

Suppose, there is n number of parties and each party holds input x. Before sending, each party divides its input x into $x_1, x_2, x_3, \ldots\ldots, x_n$.

Thus, the concept is that parties hold their own data, but aid to get the main outcome.

The main idea is to compute the result securely. The party does not know any information of the other parties.

The first protocol is a secure sum protocol, in which all the parties arranged in a circle. Each party breaks its data block into the fixed number of fragments. The results run only in the forward direction. Among all the parties, one party starts the computation. This party is a protocol initiator. This protocol initiator announces the result and passes first data segment to the next party.

A trusted party is third-party which performs computation. If, two parties want to compute the sum of their data then this parties send their facts to trusted third-party (TTP). Trusted third-party performs function on that data and sends output of the computation to all the party. The assumption is that, the third-party trusted one.

There are two models used for secure multiparty computation, real model and idyllic model. In the real model, there is no trusted party. All the parties run a real protocol with no trusted help. In the idyllic (ideal) model, there is a trusted third-party. All the parties send inputs to a trusted party, who computes the function for them. Both the models provide privacy and correctness. An opponent cannot learn more about the honest party's input other than function output and the function always computed correctly.

A hadamard matrix is a square matrix whose entries are either +1 or -1 and whose rows are mutually orthogonal.

Let, H be a hadamard matrix of order n. the transpose of H is closely related to its inverse.

$$HH^T=nI_n$$

II. BACKGROUND

The history of the secure multi party computation problem is wide spread. This problem firstly introduced by Yao in 1982. Yao defines two persons want to compute the function without knowing any information about each other. In [1], the secure multiparty computation problems studied and defined the secure multiparty computation. They give solutions using the cryptographic method and make use of oblivious transfer protocol. An efficient protocol given in[2], which give a secure protocol for Yao's millionaire's problem. Also, computes the function without disclosing any information. The basic structure of the secure multiparty computation and methods, like cryptography for privacy preservation presented in [3]. In k secure sum protocol, parties compute the sum and keep their data secure. The data of the parties broken into the fixed number of segments [4] and each segment holds random number.

Randomization, cryptography, perturbation techniques used for privacy preserving data mining, introduced in [5]. Merits and demerits also presented by them. In [6], a hybrid protocol presented, which assigns random number to each segment. This protocol provides security, but used n-1 random number for last party. In [7], an approach presented, which uses k-anonymity technique for the secure multiparty computation. In [10], several secure building blocks such as fast secure matrix multiplication, secure scalar product and the secure inverse of the matrix sum presented and compared it. A survey of approaches and the pros and cons of all the approaches presented in [11]. In [12], many tools provided for the privacy preserving data mining. Such as, secure sum protocol, secure union, and secure intersection. There are many applications of it. In [13], a walsh-hadamard transformation used for secures multiparty computation.

III. Proposed Protocol

The proposed algorithm is the concept of secure sum protocol. This algorithm takes into account the both idyllic and real mode.

The number of parties assumed in a circle. Each party divides data block into the fixed number of segments which is odd. Segments are distributed between multiple parties. A hadamard matrix used in this algorithm. Thus, all the parties multiplied its data with hadamard matrix.

Initially, all the parties send the sum of its first segment and random number to trusted third-party. Trusted third-party computes the sum of all the data [(segment + random number)*H]. Now, trusted third-party sends this output to P_1. This process provides ideal model of the security. In this, parties are exchanging their data to trusted third party and computation done by this party. This party sends result to protocol initiator party. In our case, P_1 is protocol initiator. Now, next process provides real model of the security for computing the function. All the parties used a real protocol for computation.

A. Description of protocol

Assume there are P_1, P_2, P_3, $P_{4,}$ …, P_n. The numbers of parties involved in computations are n, where each party breaks its own data block into the fixed number of segments.

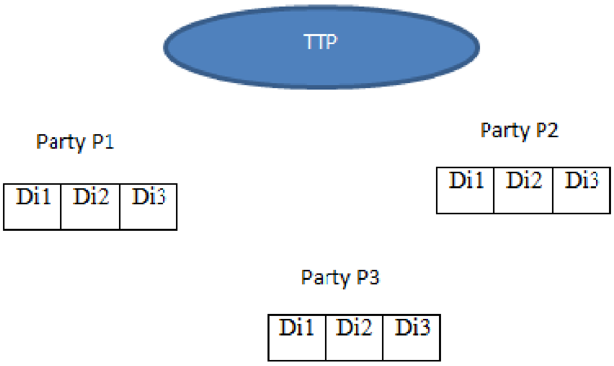


Fig. 1 Arrangement of Parties

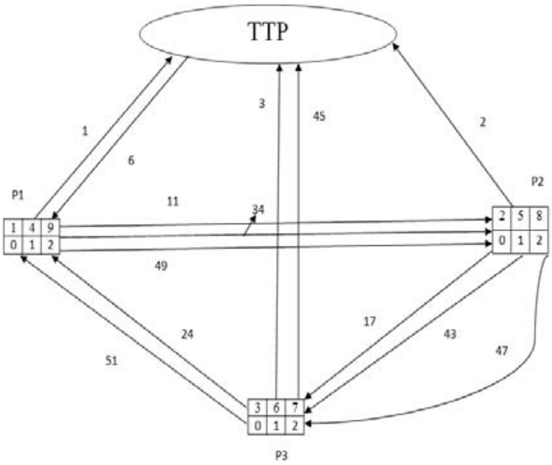


Fig. 2 Computation process

The number of segments is odd in secure multiparty computation. Segmentation of data block done on the basis that the sum of all the segments is equal to the value of the data block of that party. There are n numbers of parties. Each party used distribution process. Thus, each party distributes it segment to other party. In the first round, all the parties send the sum of its first segment and random number to trusted third-party. This value multiplied by hadamard matrix. The order of a hadamard matrix is 1, 2 or 4n, where n is an integer. Trusted third-party computes the sum of all the data (segment + random number). Now, trusted third-party sends this output to P_1.

In the second round, P_1 subtracts its first random number from this value and adds second segment and random number. P_1 sends this value to P_2. Now, P_2 subtracts its first random number from this value and adds its second segment and second random number to this and send to P_3. P_3 will use the same process and sends value to P_1. In the third round, P_1 now subtracts second random number and adds third segment and third random number to it. P_1 forwards this to P_2. P_2 and P_3 will do same. P_3 sends output to P_1. In the next round, only the last random number subtracts to this value by P_1. P_2 and P3 also subtract last random number. Now, after all segments complete P_3 sends the result to trusted party. Trusted party announces the result and forward to all other parties.

A hadamard matrix plays an important role, because it is a robust parameter designs for investigating noise factor impacts on response. Now, even if the two parties maliciously aid to know the data of the other party, they will not be able to know the actual data of any party. The sum of the segments is a garbage value and thus, worthless for hacker party. For example, if trusted party is dishonest then it's also cannot know the data of the other parties, because only final result and the first segment from each party accessed by the trusted party. Thus, chance of data leakage by any party is zero.

B. Proposed algorithm

(1) Define parties P_1, P_2, P_3, P_4, P_n where n>1. Each party involved in computation.

(2) Suppose, each party has input data blocks X_1, X_2, X_3, X_4… Xn.

(3) Each party breaks it data blocks into the k number of segments.

(4) Each party used distribution function.

(5) Arrange parties in a ring as P_1, P_2 , …, Pn and selects P_1 as the protocol initiator.

(6) Each party decides random numbers for each segment $r_{i1}, r_{i2} \ldots\ldots r_{ik}$.

(7) For i=1 to n

$$SUM= \sum_{i=1}^{n} (D_{i1}+r_{i1})*H$$

This SUM computed by trusted party.

(8) Trusted party sends SUM to party P_1.

(9) For i=1 to n

$$SUM= [(SUM-r_{i1})*H + (D_{i2}+r_{i2})*H]$$

(10)The n^{th} party sends SUM to first party, because all the parties arranged in a ring.

(11) Again, first party computes this process

$$SUM= [(SUM-r_{i2})*H + (D_{i3}+r_{i3})*H]$$

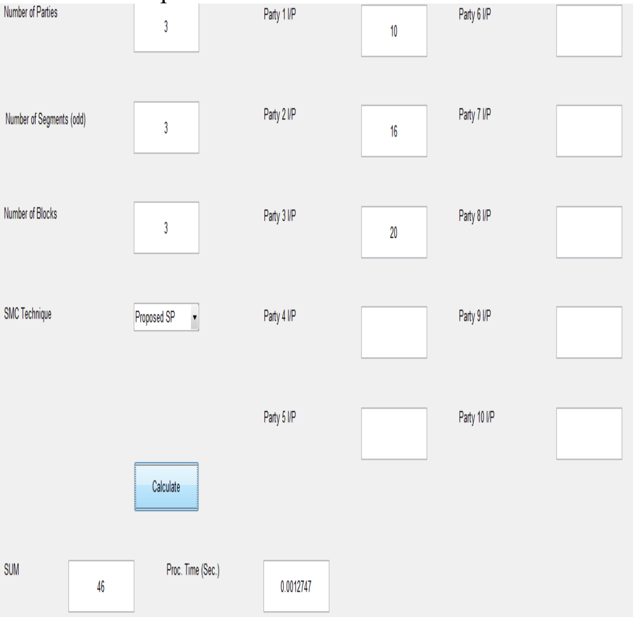
(12) The n^{th} party sends SUM to first party.

(13) For i=1 to n

$$SUM= (SUM-r_{i3})H$$

(14) Now, the n^{th} party sends final sum to trusted party.

(15) The trusted party announces the result and broadcasts to all the parties.



Number of Parties 3

Number of Segments (odd) 3

Number of Blocks 3

SMC Technique Proposed SP

Party 1 I/P 10

Party 2 I/P 16

Party 3 I/P 20

Party 4 I/P

Party 5 I/P

Party 6 I/P

Party 7 I/P

Party 8 I/P

Party 9 I/P

Party 10 I/P

SUM 46

Proc. Time (Sec.) 0.0012747

Fig. 3 Snapshot of the Algorithm

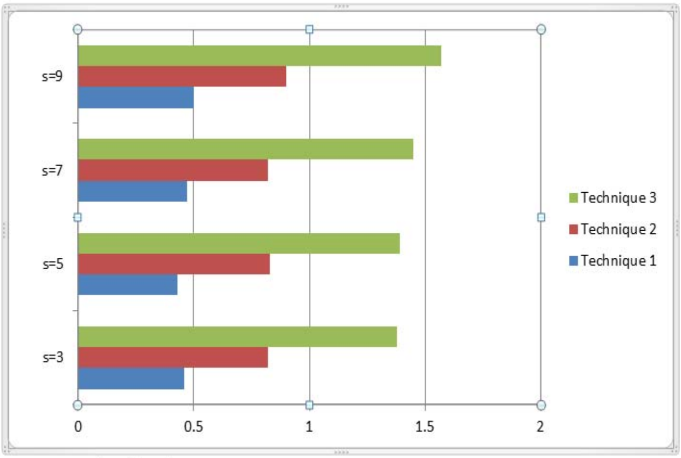
C. *Protocol analysis*

In this protocol, distribution of segments is used. A hadamard matrix makes the value of segments garbage, if any two party want to collude. This is the secret of the parties. In our protocol, there is zero probability of the data leakage because no two parties know the data of the other parties. The semi honest parties cannot learn more information than the result. This is an enhancement over the previous protocol given in [6].

Thus, each party divides its data into segments and each segment secure with random number and all the segments value and random number multiplied by hadamard matrix. So that, no two parties collide.

In this paper, we compare three techniques. The first technique given in [6] and second given in [15]. The last technique given in this paper.

The main task of our algorithm is to provide security at any cost. We are showing here, the experimental graph and the result.



Assume number of parties=4

Fig. 4 Time Performance Graph

IV. CONCLUSION

This paper presents a protocol which uses both real and idyllic model. This protocol is different from the existing hybrid secure sum protocol [6] because of redistribution of data blocks and hadamard transformation. Thus, there is no chance of attack because of this. We used hadamard matrix, in this if any row and column is multiplied by –1, the hadamard property is retained. Hence it is always possible to have the first row and first column of a hadamard matrix contain only +1 entries. This protocol provides a zero leakage chance. In the future, we can work on reducing the complexity of the protocol.

REFERENCES

[1] Wanliang Du, Mikhail J. Atallah "Secure Multi-Party Computation Problems and Their Applications; A Review and Open Problems*"*. Proeedings of new security paradigms workshop*, September 2001.*

[2] Ioannis Ioannidis, Ananth Grama "An Efficient Protocol for Yao's Millionaires' Problem*"*. Proceedings of the 36th Hawaii International Conference on System Sciences *IEEE 2002*.

[3] Yehuda Lindell and Benny Pinkas, "Secure Multipart Computation for privacy-preserving data mining*"*. The Journal of Privacy and Confidentiality, *Vol. 1, 2009*, pp. 59-98.

[4] Durgesh Kumar Mishra, Rashid Sheikh, Beerendra Kumar, "Privacy-Preserving k-Secure Sum Protocol*"*. (IJCSIS) International Journal of Computer Science and Information Security*, Vol. 6, 2009.*

[5] Gayatri Nayak and Swagatika Devi, "A servey on privacy preserving data mining: approaches and techniques". International Journal of Engineering Science and Tchnology (IJEST), *Vol. 3, march 2011.*

[6] Durgesh Kumar Mishra, Priyanka Jangde, Gajendra Singh Chandel, "Hybrid Technique for secure sum protocol". (WCSIT) World of Computer Science and Information Technology Journal, *vol. 1, 2011*.

[7] Mehmet Ercan Nergiz, Abdullah Ercument Cicek, Thomas B. Pedersen, and Yucel Saygin, "A Look-Ahead Approach to Secure Multiparty Protocols*"*. IEEE Transactions on Knowledge and Data Engineering, *Vol. 24, July 2012*.

[8] Durgesh Kumar Mishra, Rashid Sheikh, Beerendra Kumar, "A Distributed k-Secure Sum Protocol for Secure Multi-Party Computations". Journal of Computing, *Vol 2*, Issue 3, *March 2010*.

[9] Durgesh Kumar Mishra, Rashid Sheikh, Beerendra Kumar, "Changing Neighbors k-Secure Sum Protocol for Secure Multi- Party Computation". (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, 2010.

[10] Sin G Teo, Vincent Lee, Shuguo Han, "A study of Efficiency and Accuracy of Secure Multiparty Protocol in Privacy-Preserving Data Mining". 26th International Conference on Advanced Information Networking and Applications Workshops, *2012 IEEE*.

[11] Alexandre Evfimievski, Tyrone Grandison, "Privacy Preserving Data Mining". IBM Almaden Research Center.

[12] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, Michael Y. Zhu "Tools for Privacy Preserving Distributed Data MIning"*. Vol. 4.*

[13] Hanumantha Rao Jalla, P N Girija "Distance Based Transformation for Privacy Preserving Data Mining Using Hybrid Transformation". CS & IT-CSCP 2014.

[14] Neha Pathak,Shweta Pandey, "Distributed Changing Neighbors k-secure sum Protocol for Secure Multiparty Computation". *IEEE* Nirma University International Conference on Engineering (NUiCONE) *2013*.

[15] Neha Pathak,Shweta Pandey, "An Efficient Method for Privacy Preserving Data Mining in Secure Multiparty Computation". *IEEE* Nirma University International Conference on Engineering (NUiCONE) *2013*.