MJS, Vol. 8, No. 1, Jan - June 2009, pp. 35-44
ISSN 0975-3311
https: / / doi.org / 10.12725 / mjs.14.5

PERFORMANCE ANALYSIS OF SECURE MULTIPARTY COMPUTATION PROTOCOL

Samiksha Shukla*, D.K. Mishra**

ABSTRACT

In this paper we address the issue related to privacy, security, complexity and Implementation, various adversaries exist which hamper the secure multiparty computation. In the secure multiparty computation, a set of parties wishes to jointly compute some function of their inputs. Such a computation must preserve certain security properties, like privacy and correctness, even if some of the participating parties or an external adversary colludes to attack the honest parties.

Introduction

A main issue that arises during data mining is of data privacy. Privacy may be needed sometimes due to law or sometime due to business needs. Data sharing is also privileged for mutual benefits while keeping the privacy and security in mind. Large amount of data is represented as sets or multi-sets, including many databases. For example, secure multi-party computation (SMC) techniques which utilize a layout of multi-set operations leads to the design of efficient and secure protocol. In order to achieve more efficiency and flexibility, we can define two types of security, owner and data. Protocol that achieves both types of security can secure personal

* Department of Computer Science, Christ University, Bangalore, Karnataka, India.

** Acropolis Institute of Technology and Research, Indore, Madhya Pradesh, India.

as well as compound data. For achieving this, various protocols have been designed and developed which have mechanisms like cryptography, anonymizer and trusted third party (TTP) so that we need not compromise on security. We are also going to propose a protocol for the same in this paper.

Secure Multi-Party Computation

In SMC, there are *n* parties who want to compute their private data jointly and get the result. The main parameter in SMC is to maintain the privacy of individuals and the correctness of result.

Various works have been done in secure multi-party computations based on various practical problem-solving strategies. Mikhail J et al. investigated how different computational geometry problems can be solved in a co-operative environment. Atallah solved two parties problem based on their individuals data but without disclosing either parties' private data [Atallah, 2001].

Yao initially proposed this problem [Yao, 1982], his first solution uses a centralized TTP which is selected by majority of honest parties, and shows synchronous system with cryptography [Goldreich, 1987]. Yao also presented a secure protocol for Yao's millinaries problem in which each of the two participating parties have a number and the objective is to determine whose number is larger without disclosing any information about the numbers. Yao had demonstrated analytically as well as experimentally the performance characteristics and security and proved that for his range of numbers, his protocol is superior [Ioatemnnies, 2003] .

Nabil R. Adam and Shoshani A described a security control method and characteristics problem for statistical data base [Nabil, 1989; Shoshani, 1982]. Agrawal R. et al. has worked extensively in the field of multi-party computations which have been propelled by the hope to be able to provide statistical information without compromising private information of individual [Agrawal, 2000].

There are various techniques, which have been proposed earlier such as data perturbation, clustering, and query restriction. A query restriction was proposed by restricting the size of query result and controlling the overlap amongst successive queries [Dobkin, 1979; Fellegi, 1972; Denning, 1979]. Yu Ct et al. proposed clustering technique by clustering the entity into mutually exclusive atomic population [Yu, 1977]. The perturbation technique includes swapping between records [Denning, 1982]. An exact disclosure occurs if by issuing one or more queries; user can determine the exact value of confidential attribute of an individual. A partial disclosure occurs if a user is able to obtain an estimator whose variance is below a given threshold. Two methods are adopted for modifying value field from statistics literatures; they are value class membership and value distortion membership [Conway, 1976].

Chris Clifton et al. presented the various type of privacy, which need to be studied when going for secure multi-party computation [Clifton, 2002]. Yucel Sayqin et al. presented different methods namely secure sum, secure set union and secure set intersection. They were used for privacy preservation [Sayqin, 2002]. Malin B et al. worked on concept of scalar product protocol for SMC [Malin, 2005].

The SMC Problem is one of the most fundamental problems in security issues. Suppose a set of n players that wants to compute the result of a common function ¦. Then the security means correctness of output as well as privacy of players even in presence of malicious players. If we assume that there exist a TTP, then the problem become easier but the security is not up to the point. Lindell presents the concept of general composition and universal composability [Lindell, 2003]. Trevathan et al. describe the privacy preserving association rules and uses the concept of support and count for maintaining privacy during computation. Main drawbacks of these techniques are efficiency and communication complexity [Trevathan, 2005].

The SMC on anonymity is implemented in secure broadcast. One can apply SMC protocols [Trabin, 1989; Cramer, 1999] and obtain general t-secure MPC with $t < N/2$. Yuval Ishai et al. define a statistically-secure reduction from the private anonymity functionality [Yuval, 2006]. It therefore suffices to reduce the broadcast functionality to private anonymity. It turns out that this type of reduction is implicit in the work of Ptzmann and Waidner on obtaining fully resilient broadcast using preprocessing [Ptzmann, 1996].

Proposed Protocol

The entire literature uses TTP to solve SMC problems but now days, a party cannot trust TTP so the TTP can do the computation and the privacy is performed by the parties itself. Our proposal is to use anonymizer so that the party can hide its own the identity. We make the following assumptions:

1. TTP computes the result of the function $y=f(x_1, x_2, \ldots, x_n)$ correctly.
2. Each party with their input can communicate with a trusted anonymizer. A trusted anonymizer (A_i) is a system that acts as an intermediary between the party having the input and the TTP which will carry out the computation. Thus, A_i hides the identity of P_i (Party) from the TTP.

3. TTP has the capability to announce the result of the computation publicly.

4. The communication channels used by the input providing parties to communicate with the anonymizers are secure. That is, no body can catch the data transferred between them.

5. The anonymizer in any condition will not disclose the identity of the data source, from which it is forwarding the data to the third party. Anonymizer never stores data.

Formal description of Proposed Protocol

Our protocol is based on three layers namely computation layer, security layer and input layer. In Figure 3.1, there are *n* parties $P_{1,} P_{2}\ldots P_{n}$, each having input x_i that will be used in computing $y=f(x_1, x_2,\ldots,x_n)$. To calculate the value, each input providing party P_i takes the following steps in the form of an algorithm [Mishra, 2007b].

Algorithm

```
1. $P_{1,} P_{2,} \ldots, P_{n.}$ are parties;
2. $A_1, A_2, \ldots, A_n$ are anonymizers;
3. Generate a random key, $R_{i,}$ as an identifier for each $P_i$;
4. Compute $d_i=x_i||R_i$;
5. $P_i. \rightarrow \quad _{Ai} (A_1, A_2, \ldots, A_n)$;
   /* $P_i$ randomly selects an anonymizer $A_i$ through which Pi will communicate
   with the UTP. is the function which select $A_i$ from the $A_1, A_2, \ldots, A_n$ */
6. Send $d_i$ to $A_i$;
7. $A_i$ Send $d_i$ to UTP;
   /*The UTP takes each data unit $d_i$ and extract xi and $R_i$ from it.*/
End
```

Performance and Security Achieved by Protocol

Above protocol performs satisfactorily as per need. As R_i was generated by the P_i randomly, and A_i has hidden the identity of P_i from UTP, UTP does not have any way to find out from which P_i the current data is coming means P_i's identity is hidden. UTP then calculates the value of the function $y=f(x_1, x_2 ... x_n)$ and announces the results publicly so that each P_i can see the results. In case the participating parties do not want the results of the computation to be publicly accessible, UTP can return the results of the computation to A_i's. It is now the job of the A_i to return the result to the P_i that has submitted the data through it.

In the worst case scenario, the third party can, at most, publicly announce the data it got from the anonymizers. Even though this seems to be a serious flaw, but as the only identity for the actual source of data is the randomly generated key R_i for each data source P_i, third party can post only triple (x_i, R_i, A_i) pairs. As there is no direct relation amongst x_i, R_i and A_i, it is difficult to map each triple (x_i, R_i, A_i) to a unique P_i.

Now let us take the case that out of the n data providing parties, k parties cooperate to uncover the data of the remaining *n-k* parties. Here, the UTP has posted a set of triples M=. The *k* adversary parties can remove their *k* triples from *M*, leaving M¢ containing records for remaining n-k parties. Now, in order to assign each of the remaining *n-k* records to respective Pi, the total number of permutations that they will have to try is (*n-k*)!, which cannot be done in polynomial time. This non-polynomial complexity of problem is desirable from the view point of the security to be achieved.

Performance of Protocol in Various Conditions

Case-1: Let there are *n* parties working with fixed set of anonymizer respectively. The probability of one anonymizer become malicious will be 1/n. Thus the probability of *k* anonymizer become malicious will be *k/n* (where k < n).

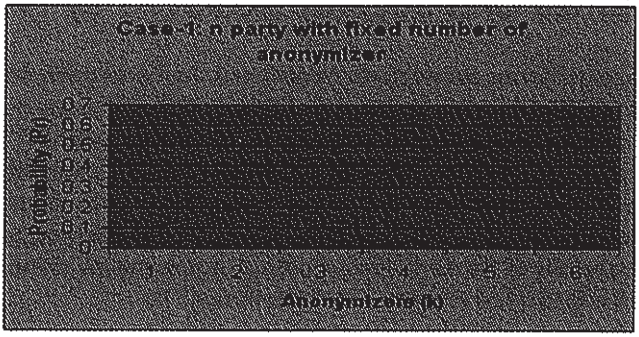

Figure 1 Probability of security breach for *n* parties with fixed number of anonymizer against malicious anonymizer

Figure 1 depicts that as malicious anonymizer increases the probability of failure increases linearly.

Case-2: Let there are *n* parties and same number of anonymizer. Here anonymizers are selected randomly. The probability of any anonymizer become malicious is 1/n. To choose this anonymizer the probability is 1/n x1/n. i.e. $1/n^2$.

$Pr = 1/n^2$

Therefore probability of *k* anonymizer will become malicious is k/n^2.

Out of *n* anonymizers the probability of selection of one such type of anonymizer will be

$Pr = 1/n^2 \text{ x } 1/n$

$Pr = 1/n^3$

Out of *n* anonymizers the probability of *k* anonymizers will become malicious will be

$Pr = k/n^3$

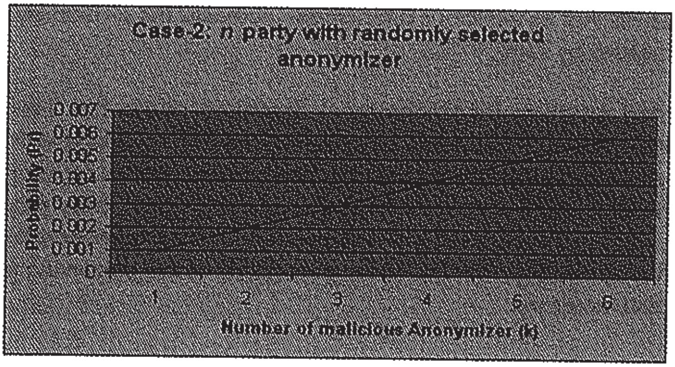

Figure 2 Probability of security breach for n parties with randomly selected anonymizer against malicious anonymizer

Figure 2 depicts that as number of anonymizer increases and selected as randomly the probability of failure will also increase linearly.

Case-3: Suppose there are *n* parties and *m* anonymizer and any party can select any anonymizer randomly then the probability of selection of one anonymizer will be 1/m. Similarly the probability of any anonymizer to be malicious will be 1/m. Therefore the probability of any anonymizer to be malicious is $1/m^2$. Hence the probability of *k* anonymizer become malicious is k/m^2 and probability of selection for these anonymizer will be k/nm^2. In this case some times it may happen that some of the anonymizer will sit idle and some will be over loaded. Therefore there is no proper distribution of packets. If a party finds that an anonymizer is behaving maliciously then at the time of selection it can be ignored.

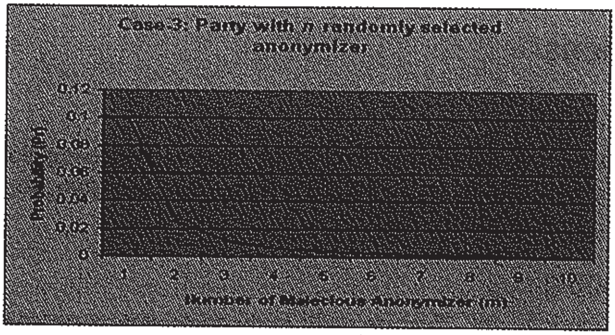

Figure 3 Probability of security breach for *n* parties with *m* randomly selected anonymizer against malicious anonymizer.

Figure 3 depicts that as malicious anonymizer increases the probability of failure will also increases by $1/m^2$.

The simulated program for the protocol described above has been run for approximately 10000 times each, by keeping one TTP's working with fix set of anonymizer, randomly selected anonymizer and n parties and m anonymizers. The results of the simulation are shown through various graphs. These graphs show that in case-3 as in this case security is dependent on total number anonymizers, as the number of anonymizer increases probability of the preserving privacy of individual also increases.

Conclusion

The best solution for the entire problems stated above is to acquaint a third party, who has the assurance and belief of all participants is called as trusted third party (abbreviated as TTP). The TTP will execute all the calculation after receiving the complete data from all the participants. The result is good enough until TTP is honest. As soon it losses trust of any participants the system will fail.

Finally, it is accomplished that, at the initial stage of our research work the main stress was given to the privacy and security of data and parties. For this reason we introduced the concept of anonymizer for maintaining the privacy. During the time span of research work, we come to know that correctness is also equally essential feature of SFE. So later on it was specially taken in mind. During the analysis of the correctness in SFE, the effect of adversaries came into the scenario. Therefore the unique concerns were taken for minimizing the effect of adversaries.

References

[Agarwal, 2000]

Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In Proceedings of the 2000 ACM SIGMOD on Management of data, pages 439{450, Dallas, TX USA, May 15 - 18 2000.

[Atallah, 2001]

Atallah M.J. and Du. W., "Secure multi-party computation geometry," *Seventh international workshop on algorithms and data structures*, Providence, Rhode Island, USA, 8 –10 Aug.2001, pp.136-152

[Conway, 1986]

Conway R., Strip D., "Selective partial access to a data base," *In Proceeding of ACM Annual Conference,* 20-22 Oct.1976, pp.85-89.

[Dobkin, 1979]

Dobkin D., Jones A.K., Lipton R.J., "Secure database: Protection against user influence," In *Proceeding of ACM Transaction on database system, ACM-TODS,* Vol.4, No.1, Mar.1979, pp.97-106.

[Goldreich, 1987]

Goldreich O., Micali S., Wigderson A., "How to Play Any Mental Game – A Completeness Theorem for Protocols with Honest Majority," Nineteenth *ACM Symposium on the Theory of Computation,* 1987, pp. 218-229.

[loatemnnies, 2003]

loatemnnies I., Ananth G., "An efficient protocol for Yao's millionaires problem," *In Proceeding of thirty sixth hawaii international conference on system sciences, HICSS-36 2003,* Big Island, HI, USA, 6-9 Jan.2003, IEEE Press, pp.6-11.

[Malin, 2005]

Malin B., Sweeney L., "A secure protocol to distribute unlikable health data," In *Proceeding of the american medical informatics association annual symposium,* Washington D.C., 22-26 Oct.2005.

[Mishra, 2007a]

Mishra D.K., Chandwani M., "Extended protocol for secure multi-party computation using ambiguous identity," *WSEAS Transactions on Computer Research,* Greece, Vol.2, No.2, Feb.2007, pp.227-233.

[Mishra, 2007b]

Mishra D.K., Chandwani M., "Anonymity enabled secure multi-party computation for Indian BPO," Accepted for presentation in *IEEE Tencon 2007*: International conference on Intelligent Information Communication Technologies for Better Human Life, Taipei, Taiwan on 28 Oct – 02 Nov 2007, pp. 52-56.

[Ptzmann, 2000]

Ptzmann B., Waidner M., "Information-theoretic pseudo signatures and Byzantine agreement for t n/3," *IBM research report RZ 2882 (#90830), IBM Research Division, Zurich,* 18 Nov.1996.

[Sayqin, 2002]

Sayqin Y., Verykios S., Ahned E., "Privacy preserving association rule mining," In *Proceeding of twelfth international workshop on research issue in data engineering, RIDE-2EC,* San Jose, CA, USA, 25-26 Feb.2002, pp.151-158

[Trabin, 1989]

Trabin, Ben-Or M., "Verifiable secrete sharing and multiparty protocols with honest majority," In *Proceeding of twenty first annual symposium on theory of computing,* Seattle, Washington, USA, 14-17 May 1989, pp.73-85.

[Yao, 1982]

Yao A.C. "Protocols for secure computations," In Proceeding of *IEEE FOCS 82*: Twenty-third IEEE Symposium on the Foundation of Computer Science, 3-5 Nov, 1982, pp. 160-164.