Security Issues of Privacy Preserving in Data Mining

[1.]S. A. MD. Noorulla Baig,
Dept. of Computer Science, RIIMS,
Tirupati, India.

[2.]P. Madhura,
Dept. of Computer Science, RIIMS,
Tirupati, India.

[3.]P. V. Ramesh
Dept. of Computer Science, RIIMS,
Tirupati, India.

***Abstract*— Privacy is one of the most important properties of an information system must satisfy, in which systems the need to share information among different, not trusted entities, the protection of sensible information has a relevant role. Thus privacy is becoming an increasingly important issue in many data mining applications. For that privacy secure distributed computation, which was done as part of a larger body of research in the theory of cryptography, has achieved remarkable results. These results were shown using generic constructions that can be applied to any function that has an efficient representation as a circuit. A relatively new trend shows that classical access control techniques are not sufficient to guarantee privacy when data mining techniques are used in a malicious way. Privacy preserving data mining algorithms have been recently introduced with the aim of preventing the discovery of sensible information. In this paper we will describe the implementation of cryptography in that data mining for privacy preserving.**

Keywords*— *Privacy preserving, Cryptography, Data Mining, Security.

I. INTRODUCTION

Privacy preserving data mining is an important property that any data mining system must satisfy. So far, if we assumed that the information in each database found in mining can be freely shared. Consider a scenario in which two or more parties owning confidential databases wish to run a data mining algorithm on the union of their databases without revealing any unnecessary information. For example, consider the police(CBI) force that wish to conduct a joint research while preserving the privacy of their criminals. In this scenario it is required to protect privileged information, but it is also required to enable its use for research or for other purposes. In particular, although the parties realize that combining their data has some mutual benefit, none of them is willing to reveal its database to any other party.

The common definition of privacy in the cryptographic community limits the information that is leaked by the distributed computation to be the information that can be learned from the designated output of the computation. Although there are several variants of the definition of privacy, for the purpose of this discussion we use the definition that compares the result of the actual computation to that of an "ideal" computation: Consider first a party that is involved in the actual computation of a function (e.g. a data mining algorithm). Consider also an "ideal scenario", where in addition to the original parties there is also a "trusted party" who does not deviate from the behavior that we prescribe for him, and does not attempt to cheat. In the ideal scenario all parties send their inputs to the trusted party, who then computes the function and sends the appropriate results to the other parties. Loosely speaking, a protocol is secure if anything that an adversary can learn in the actual world it can also learn in the ideal world, namely from its own input and from the output it receives from the trusted party. In essence, this means that the protocol that is run in order to compute the function does not leak any "unnecessary" information.

II. PRIVACY PRESERVING

Explosive progress in networking, storage and processor technologies has led to the creation of ultra large database that record unprecedented amount of transactional information. Privacy preserving protocols are designed in order to preserve privacy even in the presence of adversarial participants that attempt to gather information about the inputs of their peers. There are, however, different levels of adversarial behavior. Cryptographic research typically considers two types of adversaries: A semi-honest adversary (also known as a passive, or honest but curious adversary) is a party that correctly follows the protocol specification, yet attempts to learn additional information by analyzing the messages received during the protocol execution. On the other hand, a malicious adversary may arbitrarily deviate from the protocol specification. (For example, consider a step in the protocol where one of the parties is required to choose a random number and broadcast it. If the party is semi-honest then we can assume that this number is indeed random. On the other hand, if the party is malicious, then he might choose the number in a sophisticated way that enables him to gain additional information.) It is of course easier to design a solution that is secure against semi-honest adversaries, than it is to design a solution for malicious adversaries.

A common approach is therefore to first design a secure protocol for the semi-honest case, and then transform it into a protocol that is secure against malicious adversaries. This transformation can be done by requiring each party to use zero-knowledge proofs to prove that each step that it is taking follows the specification of the protocol. More efficient transformations are often required, since this generic approach might be rather inefficient and add considerable overhead to each step of the protocol. We remark that the semi-honest adversarial model is often a realistic one. This is because deviating from a specified program which may be buried in a complex application is a non-trivial task, and because a semi-honest adversarial behavior can model a scenario in which the parties that participate in the protocol are honest, but following the protocol execution an adversary may obtain a transcript of the protocol execution by breaking into a machine used by one of the participants.

III.PRIVACY PRESERVING COMPUTATION

In this section we will describe the various computation techniques which we are using for data.

3.1 Classification

Definition: Given a database D = {t1 , t2 , ... , tn } of tuples (items, records) and a set of classes C = { C 1, ... , Cm }, the classification problem is to define a mapping f: D→C where each ti is assigned to one class. A class, Cj , contains precisely those tuples mapped to it; that is, Cj = {ti | f(ti) = Cj , 1 ≤ i ≤ n and ti ∈D}.

Our definition views classification as a mapping from the database to the set of classes. Note that the classes are predefined, are nonoverlapping, and partition the entire database. Each tuple in the database is assigned to exactly one class.

X has a private database D1 and Y has private database D2. How can X and Y build a decision tree based on D1 x D2 without disclosing the contents of their private database to each other? Several algorithms like ID3, Gain Ratio, Gini Index and many other can be used for Decision Tree.

3.2 Data Clustering

Definition: Given a database D = {t1 , t2 , . . . , tn } of tuples and an integer value k, the clustering problem is to define a mapping f : D →{ 1 , . . . , k} where each ti is assigned to one cluster Kj , 1 ≤j ≤ k. A cluster, Kj . contains precisely those tuples mapped to it; that is, Kj = {ti | f (ti) = Kj , 1 ≤i ≤n, and t∈D}.

X has a private database D1 and Y has private database D2. X and Y want to jointly perform data clustering on D1 x D2. This is primarily based on data clustering principle that tries to increase intra class similarity and minimize interclass similarity.

3.3 Mining Association Rules

Definition: Given a set of items I = {I_1 , I_2, . . . , I_m } and a database of transactions D = {t_1 , t_2 , . . . , t_n } where t_i = {I_{i1} , I_{i2}, . . . , I_{ik} } and I_{ij}∈I , an association rule is an implication of the form A ➔ B where A, B ⊂ I are sets of items called itemsets and X ∩ Y = θ.

Definition for Support : The support (s) for an association rule A ➔ B is the percentage of transactions in the database that contain A ∪ B.

Definition for Confidence: The confidence o r strength (a) for an association rule A➔B is the ratio of the number of transactions that contain X U Y to the number of transactions that contain X.

Let X has a private database D1 and Y has private database D2. If X and Y wish to jointly find the association rules from D1x D2 without revealing the information from individual databases.

3.4 Data Generalization, Summarization and Characterization

Let X has a private database D1 and Y has private database D2. If they wish to jointly perform data generalization, summarization or characterization on their combined database D1xD2, then this problem becomes an Secure Multiparty Communication problem.

3.5 Profile Matching

X has a database of hacker's profile. Y has recently traced a behavior of a person, whom he suspects a hacker. Now, if Y wants to check whether his doubt is correct, he needs to check X's database. X's database needs to be protected because it contains hacker's related sensitive information. Therefore, when Y enters the hacker's behavior and searches the X's database, he can't view his whole database, but instead, only gets the comparison results of the matching behavior.

IV.SECURE COMPUTATION AND PRIVACY PRESERVING IN DATA MINING

There are two distinct problems that arise in the setting of privacy-preserving data mining. The first is to decide which functions can be safely computed, where safety means that the privacy of individuals is preserved. For example, is it safe to compute a decision tree on confidential data in an organization and publicize the resulting tree? For the most part, we will assume that the result of the data mining algorithm is either safe or deemed essential. Thus, the question becomes how to compute the results while minimizing the damage to privacy. For example, it is always possible to pool all of the data in one place and run the data mining algorithm on the pooled data. However, this is exactly what we don't want to. Thus, the question we address is how to compute the results without pooling the data, and in a way that reveals nothing but the final results of the data mining computation. This question of privacy-preserving data mining is actually a special case of a long-studied problem in cryptography called secure multiparty computation. This problem deals with a setting where a set of parties with private inputs wish to jointly compute some function of their inputs. Loosely speaking, this joint computation should have the property that the parties learn the correct output and nothing else, even if some of the parties maliciously collude to obtain more information. Clearly, a

protocol that provides this guarantee can be used to solve privacy-preserving data mining problems of the type discussed above.

V.PROTOCOLS FOR SECURE MULTI PARTY COMPUTATION

Two or more parties would like to compute some function collaboratively without revealing their input to other parties, only the final result of the computation will be known to the parties.

A. Yao's Millinaire Problem

Secure multi-party computation is initiated by Yao's Millionaires problem. In this two millionaires wish to know who is richer, with neither revealing their net worth to each other. The cryptographic solution by Yao has communication complexity that is exponential in the number of bits of the numbers involved, using an untrusted third party. Cachin projected a solution based on Φ-hiding assumption. His protocol uses an untrusted third party that can misbehave on its own. The communication complexity of Cachin's scheme is O (l), where l is the number of bits of each input number.

B. 1-out-of-N Oblivious Transfer Protocol

Goldreich's circuit evaluation protocol uses the 1-out-of-N Oblivious Transfer. An 1-out-of–N Oblivious refers to a protocol where at the beginning of the protocol one party, A has N inputs X_1, X_2…, X_N and at the end of the protocol the other party, B, learns one of the inputs Xi for some $1 \leq i \leq N$ of his choice, without learning anything about the others inputs.An efficient 1-out-of-N Oblivious Transfer Protocol was proposed by Naor and Pinkas. By joining this protocol with the scheme by Cachin protocol, the 1-out-of–N Oblivious Transfer protocol could be achieved with polylogarithmic communication complexity.

C. Homomorphic Encryption Scheme

Public key cryptosystems are required with a homomorphic property for some of secure multi party computation protocols. A homomorphic encryption permits certain algebraic operations to be performed on the encrypted data by utilizing an efficient operation to the corresponding decrypted data. Secure public key cryptosystems are called as homomorphic if it satisfies the following homomorphic property:

1. $E_k(x) * E_k(y) = E_k(x+y)$
2. $E_k(x)^y = E_k(xy)$

In the above equation k is a key, x and y is the data to be encrypted. E() denotes encryption.

A beneficent property of homomorphic scheme is addition operation. It can be performed based on encrypted plaintext without decrypting them.

D. Scalar product Protocol

Scalar product protocol is an important cryptographic protocol in the process of designing several secure multi-party computation protocols. Most of the problems can be reduced to computing scalar product. Weijiang xu et al [20] proposed privacy preserving add and multiply exchanging technology. It contains two protocols, in which Privacy Preserving Multiply to Add protocol (PPMtAP) is one-dimensional scalar Product protocol. Privacy Preserving Add to Multiply protocol (PPAtMP) is reverse of PPMtAP. In[10], three different approaches to PPAtMP was discussed with correctness and security. They are PPAtMP based on homomorphic encryption system (PPAtMP_HES), based on oblivious transfer protocol and based on semi-honest third party (PPAtMP_STP). In these three protocols, PPAtMP_HES and PPAtMP_STP have less communication expenses and reveals nothing of privacy unless colluding. PPAtMP_OTP has higher communication & computation complexity. They also extended the PPAtMP protocol to Privacy Preserving Adding Scalar Product Protocol (PPAtSPP). It has better security and more powerful in higher security situation.

E. Privacy Preserving Set Intersection Protocol(PPSI)

In PPSI, there are N parties, each party has a set (multiset) T_i and $|T_i|=S$, all parties wish to know the intersection $TI=T_1 \cap T_1 \cap \ldots\ldots \cap T_N$, without discovering any data other than the computed output. Y. Sang et al[15] proposed an efficient PPSI protocol for the semi-honest model and solved PPSI by efficiently constructing & evaluating polynomials whose roots are elements of the set intersection. They are also extended protocol of [18] to the malicious model. The correctness with probability of this protocol is $((N- 1)/N)^{N-1}$ and computation cost is $O(c^2s^2lgN)$.This protocol has more correctness and less computation cost compared to already existing PPSI protocol in the malicious model.

F. Virtual Party Protocol

Rohit Parthak et al [9] proposed virtual party protocol to ensure the privacy of individuals and preserving the data of the organization without revealing their private data. The four layers of virtual party protocols are party layer, virtual party layer, anonymizer layer and computation layer. In this method, fake data and virtual parties are generated. The data can be sent with modified tokens to carry out computation on encrypted data. Anonymization layer is used to conceal the identity of the parties. Virtual party protocol is extremely scalable and optimized for computation of banking, business etc. It can also grant us to reach zero hacking security for a several kind of applications.

TABLE 1. COMPARATIVE ANALYSIS OF SMC PROTOCOLS

S.No	Available Protocol	Communication Complexity	Privacy
1.	Yao's Millionaire Problem	Exponential in the number of bits of the numbers involved.	Low
2.	1-out-of-N Oblivious Transfer	O(m) where m is security parameter	Medium
3.	Scalar Product Protocol	Communication cost is high.	Medium
4.	Privacy Preserving Set Intersection Protocol	Total communication cost of all parties is $O(cN^2S^2lgN)$	High
5.	Virtual Party Protocol	Virtual Party Protocol has high communication cost	High

VI.PROTOCOLS FOR DATA PERTURBATION

Many cryptographic protocols are developed for multiparty collaborative mining using geometric data perturbation. They are all limited to a small number of parties [14]. The multiparty collaboration is scale up by service oriented framework. The quality of unified perturbation is impressed through three important factors: privacy guarantee, utility of collective data and the efficiency of perturbation protocol. These factors are considered in designing of the simple, negotiation and space adaptation protocols. These three protocols have been developed [6] for perturbation unification. In all these protocols, the data provider can get a public key from the service provider to encrypt the data. So the service provider can only decrypt the data. The following common steps are used in the three protocols:

1. Data mining process can be performed on the gathered data at the server side.
2. The data provider can apply the mined model to new data.

This section will describe the concept of these three protocols with their cost and privacy guarantee. Analysis of Geometric perturbation protocols are described in Table 1.

A. Simple Protocol

In simple protocol, the original data is perturbed with same randomly generated perturbation by the data providers. The group-key based random perturbation generation can be used to preventing curious service provider knowing the unified perturbation. The same random group key is utilized by all the data providers to generate the same perturbation locally. The perturbed data will not be transmitted to the service provider directly for security purpose. The public key of service provider is known by all the data providers. The data provider encrypts the perturbed data with service provider's public key and transmits encrypted perturbed data to the service provider. The service provider decrypts the received data by using their own private key and collects the data together to mine a unified model. This unified model will be sent to the data provider.

The simple protocol will not achieve same privacy guarantee for all the data providers due to random perturbation and also encryption makes the perturbed data used in the current collaboration cannot be reusable in other collaborations. The metrics for the simple protocol is listed in [6]. It takes O(knd) encryption cost, where k represents number of data providers with n number of records and each record has d dimensions.

B. Negotiation Protocol

The main goal of negotiation protocol is to enhance the overall privacy guarantee for all the data providers. In this protocol all the data providers can review the candidate perturbation and vote for the candidate or against the candidate. A data provider may prepare a different locally optimal perturbation due to different data distribution of the locally owned dataset. The data providers may also need to accept some suboptimal perturbation finally. The satisfaction level of a unified perturbation for the data provider Pi is defined by

$Si = pi/pi°$

In this equation $pi°$ is the privacy guarantee for a data provider Pi is given by the locally optimized perturbation Gi and pi is given by the unified perturbation Gt. Each data provider Pi sets their own minimum satisfaction level si^{min} is the lower bound to accept the global perturbation. In the negotiation process local minimum satisfaction level is set, which leads a trade-off between the level of privacy guarantee and the efficiency of negotiation. In[6], negotiation protocol takes $O(rkd^2+knd)$ encryption cost and local optimization cost $rk\pi$ where r represents average number of negotiation rounds and π is the data set size. The perturbed data cannot be reusable in negotiation protocol.

C. Space Adaptation Protocol

Space adaptation protocol achieves the concept of space adaptation for reducing identifiability of data sources by using secure random exchange of perturbed datasets between data providers. The space adaptation approach is based on geometric perturbations conversion. If Gt is the target perturbation, the transformation of perturbation is defined from G_i to G_t as $G_i \rightarrow t$, "the space adaptator", G_t can be represented as the composition of G_i and $G_i \rightarrow t$: $G_t = G_i \circ G_i \rightarrow t$.

$G(X) = (G_i \circ G_i \rightarrow t)(X) = G_i \rightarrow t\,(G_i(X))$

In this equation, X is a data set. The data provider can just distribute $G_i(X)$ and the particular collaboration. So that $G_i(X)$ can be reused by future collaboration. In [6,10], discussed communication cost, optimization cost and maintenance cost. It takes $O(kd^2)$ encryption cost and $k\pi$ local optimization cost, so that it reduces the cost of encryption as well as maintenance cost.

In general, the overall satisfaction level of the space adaptation protocol can be improved with negotiation protocol and it also gives a better balance between flexibility and scalability of data distribution.

TABLE 2. COMPARATIVE ANALYSIS OF GEOMETRIC PERTURBATION PROTOCOLS

S.No	Available Protocol	Communication Complexity	Privacy	
			Curious Data Provider	Curious Service provider
1.	Simple Protocol	O(k(1+nd)) where k represents number of data providers with n number of records and each record has d dimensions.	Medium	Low
2.	Negotiation Protocol	It takes $O(rk^2d^2+knd)$ cost. Where r is average number of rounds.	Medium	Medium
3.	Space adaptation Protocol	O(k(1+nd)) same as simple protocol	High	Low

VII. CONCLUSIONS AND FUTURE SCOPE

In this paper, available protocols for multi-party computation in privacy preserving data mining have been studied extensively along with computation and communication complexity. Several Secure Multiparty Computation problems are existing in the real world such as database queries, intrusion detection, geometric computation, and scientific computation. These problems can be solved using available protocols like set intersection, which is also discussed in this paper. Secure Multiparty computation can provide better balance between privacy and accuracy. But it cannot be scalable. Still Researchers are having a lot of interest and attention to get efficient solutions to all secure Multiparty computation problems with minimum communication and computation complexity. Also this paper provides basic idea on simple, negotiation and space adaptation protocols for geometric perturbation unification. Space adaptation protocol has better scalability, flexibility of data distribution and overall satisfaction level of privacy guarantee compared to the other two protocols. Currently available protocol assumes that service provider and data provider do not collude with each other. The other concerns to be addressed are investigating challenging situation where this assumption is relaxed and examining anonymization factor in the protocol to further enhance privacy preservation. Finally Privacy preserving multiparty collaborative data mining is an ongoing research area and there is a lot of issues that needs to be addressed because of the complexity of the privacy problem.

REFERENCES

[1] Feng He,Ting Wang ,"Research and Application of Secure Multi-party Computation in Several Computational Geometry Problems", In American Journal of Engineering & Tech. Research,Vol.11,No.9, pp.2514-2519,2011.

[2] Zulfa Shaikh,Poonm Garg,"A Comparative Study of Available Protocols during Privacy Preservation in Secure Multi-party Computation", In Journal of Emerging trends in Computing & Information Science,Vol.2,No.5,pp.219-221,May 2011.

[3] D. Beaver, S. Micali and P. Rogaway, The round complexity of secure protocols, Proc. of 22nd ACM Symposium on Theory of Computing (STOC), pp. 503-513, 1990.

[4] M. Bellare and S. Micali, Non-Interactive Oblivious Transfer and Applications, Advances in Cryptology - CRYPTO '89. Lecture Notes in Computer Science, Vol. 435, Springer-Verlag, 1997, pp. 547-557.

[5] M. Ben-Or, S. Goldwasser and A. Wigderson, Completeness theorems for non cryptographic fault tolerant distributed computation, Proceedings of the 20th Annual Symposium on the Theory of Computing (STOC), ACM, 1988, pp. 1–9.

[6] D. Chaum, C. Crepeau and I. Damgard, Multiparty unconditionally secure protocols, Proceedings of the 20th Annual Symposium on the Theory of Computing (STOC), ACM, 1988, pp. 11–19.

[7] S. Even, O. Goldreich and A. Lempel. A Randomized Protocol for Signing Contracts. Communications of the ACM, 28(6):637-647, 1985.

[8] O. Goldreich. Foundations of Cryptography: Volume 2 { Basic Applications. Cambridge University Press, 2004