www.ijecs.in
*International Journal Of Engineering And Computer Science ISSN:2319-7242*
*Volume 4 Issue 4 April 2015, Page No. 11698-11703*

# Security in Privacy Preserving Data Mining

*Neha Kashyap*[1], *Dr. Vandana Bhattacharjee*[2]

[1]Birla Institute of Technology, Department of Computer Science & Engineering,
Extension Centre, Lalpur, Ranchi-834001, Jharkhand, India
*Kashyap9.neha@gmail.com*

[2]Birla Institute of Technology, Department of Computer Science & Engineering,
Extension Centre, Lalpur, Ranchi-834001, Jharkhand, India
*vbhattacharya@bitmesra.ac.in*

***Abstract: Data mining has attracted a great deal of information in recent years, due to the wide availability of huge amount of data and the imminent need for such data into useful information and knowledge, which can be used for applications ranging from market analysis, fraud detection and customer retention, to production control and science exploration. The real privacy concerns are with unconstrained access of individual records, like credit card, banking applications, customer ID, which must access privacy sensitive information. Due to privacy infringement while performing the data mining operations this is often not possible to utilize large databases for scientific or financial research. To address this problem, several privacy-preserving data mining techniques are used. The aim of privacy preserving data mining (PPDM) is to extract relevant knowledge from large amounts of data while protecting at the same time sensitive information.***

**Keywords:** Privacy Preserving Data Mining, Trust Third Party Model, Secure Multiparty Computation Technique, Homomorphic Encryption, Threshold Decryption.

## 1. Introduction

Data mining, otherwise known as *knowledge discovery*, can extracted “meaningful information” or “knowledge” from the large amounts of data, so supports people’s decision-making [2]. However, traditional data mining techniques and algorithms directly operated on the original dataset, which will cause the leakage of privacy data. At the same time, a large amount of data implicates the sensitive knowledge that their disclosure cannot be ignored to the competitiveness of enterprise. These problems challenge the traditional data mining, so privacy-preserving data mining (PPDM) has become one of the newest trends in privacy and security and data mining research. In privacy-preserving data mining (PPDM), data mining algorithms are analyzed for the side-effects they incur in data privacy, and the main objective in privacy preserving data mining is to develop algorithms for modifying the original data in some way, so that the private data and private knowledge remain private even after the mining process [3]. A number of techniques such as Trust Third Party, Data perturbation technique, Secure Multiparty Computation and game theoretic approach, have been suggested in recent years in order to perform privacy preserving data mining. However, most of these privacy preserving data mining algorithms such as the Secure Multiparty Computation technique, were based on the assumption of a semi-honest environment, where the participating parties always follow the protocol and never try to collude. As mentioned in previous works on privacy-preserving distributed mining [5], it is rational for distributed data mining that the participants are assumed to be semi-honest, but the collusion of parties for gain additional benefits cannot be avoided. So there has been a tendency for privacy preserving data mining to devise the collusion resistant protocols or algorithms, recent research have addressed this issue, and protocols or algorithms based on penalty function mechanism, the Secret Sharing Technique, and the Homomorphic Threshold Cryptography are given [4], [8]. This paper is organized as follows. In Section 2, we introduce the related concepts of the PPDM problem. In Section 3, we describe privacy preserving data mining statement. In Section 4, we discuss technique for privacy preserving data mining. In section 5, the conclusion in Privacy Preserving Data Mining

## 2. The related concepts of PPDM

The concept of privacy is often more complex, In particular, in data mining, the definition of privacy preservation is referred to “getting valid data mining results without learning the underlying data values.”[6], [7] also indicated PPDM encompasses the dual goal of meeting privacy requirements and providing valid data mining results, so the definition emphasizes the dilemma of balancing privacy preservation and knowledge disclosure.

### 2.1 Defining privacy preservation in data mining

Privacy-preserving data mining considers the problem of running data mining algorithms on confidential data that is not supposed to be revealed even to the party running the algorithm. The main consideration of PPDM is twofold [12]. First, sensitive raw data like identifiers, names, addresses and so on, should be modified or trimmed out from the original database, in order for the recipient of the data not to be able to compromise another person's privacy. Second, sensitive knowledge which can be mined from a database by using data mining algorithms should also be excluded, because such knowledge can equally well compromise data privacy. So, privacy preservation occurs in two major dimensions: users' personal information and information concerning their collective activity. The former is referred to individual privacy preservation and the latter is referred to collective privacy preservation [15].

• **Individual privacy preservation:** The primary goal of data privacy is the protection of personally identifiable information. In general, information is considered personally identifiable if it can be linked, directly or indirectly, to an individual person. Thus, when personal data are subjected to mining, the attribute values associated with individuals are private and must be protected from disclosure. Miners are then able to learn from global models rather than from the characteristics of a particular individual.

• **Collective privacy preservation:** Protecting personal data may not be enough. Sometimes, we may need to protect against learning sensitive knowledge representing the activities of a group. We refer to the protection of sensitive knowledge as collective privacy preservation. The goal here is quite similar to that one for statistical databases, in which security control mechanisms provide aggregate information about groups and, at the same time, should prevent disclosure of confidential information about individuals. However, unlike as is the case for statistical databases, another objective of collective privacy preservation is to preserve strategic pattern that are paramount for strategic decisions, rather than minimizing the distortion of all statistics. In other words, the goal here is not only to protect personally identifiable information but also some patterns and trends that are not supposed to be discovered. Privacy Preservation in Data Mining has some limitations: Privacy Preservation Data Mining techniques do not mean perfect privacy, for example, The SMC computation won't reveal the sensitive data, but the data mining result will enable all parties to estimate the value of the sensitive data. It isn't that the SMC was "broken", but that the result itself violates privacy.

## 2. Models of PPDM

In the study of privacy-preserving data mining (PPDM), there are mainly four models as follows:

### 2.1. Trust Third Party Model:

The goal standard for security is the assumption that we have a trusted third party to whom we can give all data. The third party performs the computation and delivers only the results except for the third party, it is clear that nobody learns anything not inferable from its own input and the results. The goal of secure protocols is to reach this same level of privacy preservation, without the problem of finding a third party that everyone trusts. Preservation, without the problem of finding a third party that everyone trusts. Except for the third party, it is clear that nobody learns anything not inferable from its own input and the results. The goal of secure protocols is to reach this same level of privacy preservation, without the problem of finding a third party that everyone trusts. Preservation, without the problem of finding a third party that everyone trusts.

### 2.2. Semi-honest Model

In the semi-honest model, every party follows the rules of the protocol using its correct input, but after the protocol is free to use whatever it sees during execution of the protocol to compromise security.

### 2.3. Malicious Model

In the malicious model, no restrictions are placed on any of the participants. Thus any party is completely free to indulge in whatever actions it pleases. In general, it is quite difficult to develop efficient protocols that are still valid under the malicious model. However, the semi-honest model does not provide sufficient protection for many applications.

### 2.4. Other Models - Incentive Compatibility

While the semi-honest and malicious models have been well researched in the cryptographic community, other models outside the purview of cryptography are possible. One example is the interesting economic notion of incentive compatibility. A protocol is incentive compatible if it can be shown that a cheating party is either caught or else suffers an economic loss. Under the rational model of economics, this would serve to ensure that parties do not have any advantage by cheating. Of course, in an irrational model, this would not work. We remark, in the "real world", there is no external party that can be trusted by all parties, so the Trust Third Party Model is an ideal model.

## 3. Privacy Preserving Data Mining Statement

Privacy Preserving Data mining Analysis is an amalgamation of the data of heterogeneous users without disclosing the private and susceptible details of the users.

### 3.1. Problem Statement

Stipulation of a comprehensible but prescribed approach for early privacy preserving analysis in the milieu of component based software development, in order to evaluate and compare with apiece and all the Techniques in a universal platform and to devise, build up and execute functionalities like a User friendly framework, portability etc.

### 3.2. Classification of Privacy Preserving Techniques

There are many approaches which have been adopted for privacy preserving data mining. We can classify them based on the following dimensions:

- Data distribution
- Data modification
- Data mining algorithm
- Data or rule hiding
- Privacy preservation

The first dimension refers to the distribution of data. Some of the approaches have been developed for centralized data. Distributed data scenarios can also be classified as horizontal data distribution and vertical data distribution. The second dimension refers to the data modification In general; data modification is used in order to modify the original values of a database that needs to be released to the public and in this way to ensure high privacy protection

- Perturbation, which is accomplished by the alteration of an attribute value by a new value (i.e., changing a 1-value to a 0-value, or adding noise),
- Blocking, which is the replacement of an existing attribute value with a "?",
- Aggregation or merging which is the combination of several values into a coarser category.
- Swapping that refers to interchanging values of individual records.
- Sampling, which refers to releasing data for only a sample of a population?

The third dimension refers to the data mining algorithm, for which the data modification is taking place. This is actually something that is not known beforehand, but it facilitates the analysis and design of the data hiding algorithm. The fourth dimension refers to whether raw data or aggregated data should be hidden. The complexity for hiding aggregated data in the form of rules is of course higher, and for this reason, mostly heuristics have been developed. The last dimension, which is the most important, refers to the privacy preservation technique used for the selective modification of the data. Selective modification is required in order to achieve higher utility for the modified data given that the privacy is not jeopardized. The techniques that have been applied For this reason are:

- Heuristic-based techniques like adaptive modification that modifies only selected values that minimize the utility loss rather than all available values.
- Cryptography- based techniques like secure multiparty computation where a computation is secure if at the end of the computation, no party knows anything except its own input and the results.
- Reconstruction-based techniques where the original distribution of the data is reconstructed from randomized data.

## 4. Secure multiparty computation technique

### 4.1 Background

In privacy preserving distributed data mining, 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 separate medical institutions that wish to conduct a joint research while preserving the privacy of their patients. One way to view this is to imagine a trusted third party-- everyone gives their input to the trusted party, who performs the computation and sends the results to the participants. However, this is exactly what we don't want to do, for example, hospitals are not allowed to hand their raw data out, security agencies cannot afford the risk, and governments risk citizen outcry if they do. Thus, the question is how to compute the results without having a trusted party, and in a way that reveals nothing but the final results of the data mining computation. Secure Multiparty Computation enables this *without* the trusted third party. The concept ofSecure Multiparty Computation was introduced in [11] and has been proved that there is a secure multi-party computation solution for any polynomial function [10]. The basic idea of Secure Multiparty Computation is that a computation is secure if at the end of the computation, no party knows anything except its own input and the results. This approach was first introduced to the data mining community by Lindell and Pinkas [13], with a method that enabled two parties to build a decision tree without either party learning anything about the other party's data, except what might be revealed through the final decision tree. Now these techniques have been developed for association rules, clustering, k-nearest neighbor classification, and are working on others.

**Allowed adversarial behavior :** there are two main types of adversaries. [13]

a. **Semi-honest adversaries:** In semi-honest adversarial model, it correctly follows the protocol specification, yet attempts to learn additional information by analyzing the transcript of messages received during the execution. This is a rather weak adversarial model. However, there are some settings where it can realistically model the threats to the system. Semi-honest adversaries are also called "honest-but-curious "and "passive".

b. **Malicious adversaries:** In malicious adversarial model , a party may arbitrarily deviate from the protocol specification .In general, providing security in the presence of malicious adversaries is preferred, as it ensures that no adversarial attack can succeed. Malicious adversaries are also called "active". We remark that although the semi-honest adversarial model is far weaker than the malicious model, it 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.

### 4.2 Techniques for building secure multiparty computation protocols

In this section, we describe here some simple protocols that are often used as basic building blocks, or primitives, of secure computation protocols.

**Oblivious Transfer:** Oblivious transfer is a simple functionality involving two parties. It is a basic building block of many cryptographic protocols for secure computation. The notion of 1-out-2 oblivious transfer was suggested by [14] (as a variant of a different but equivalent type of oblivious transfer that has been suggested by [16]. The protocol involves two parties, the sender and the receiver. and its functionality is defined as follows:

• **Input:** The sender's input is a pair of strings $(x_0, x_1)$ and the receiver's input is a bit $\sigma \in \{0,1\}$ .

• **Output:** The receiver's output is $x_\sigma$(and nothing else), while the sender has no output.

In other words, 1-out-of-2 oblivious transfer implements the function $((x_0, x1), \sigma) \rightarrow (\lambda, x_\sigma)$, where λ denotes the empty string (i.e., no output). Oblivious transfer protocols have been designed based on virtually all known assumptions which are used to construct specific trapdoor functions (i.e. public key cryptosystems), and also based on generic assumptions such as the existence of enhanced trapdoor permutations. There are simple and efficient protocols for oblivious transfer which is secure only against semi-honest adversaries (Even et al., 1985).

Oblivious Polynomial Evaluation**:** The problem of “oblivious polynomial evaluation” (OPE) involves a sender and a receiver. The sender’s input is a polynomial $Q$ of degree $k$ over some finite field $F$, namely a polynomial $Q\ z = \sum_{i=0}^{k} aizi$ (the degree $k$ of the polynomial public). The receiver’s input is an element . The protocol is such that the receiver obtains $Q(z)$ without learning anything else about the polynomial $Q$ , and the sender learns nothing. That is, the problem considered is the private computation of the function $(Q, z) \rightarrow (\lambda\ , Q(z))$ .where $\lambda$ is the empty output. The major motivation for oblivious polynomial evaluation is the fact that the output of a *k degree* random polynomial is $k + 1$wise independent; this is very useful in the construction of cryptographic protocols. Another motivation is that polynomials can be used for approximating functions that are defined over the Real numbers.

**Homomorphic Encryption:** A homomorphic encryption scheme is an encryption scheme which allows certain algebraic operations to be carried out on the encrypted plaintext, by applying an efficient operation to the corresponding cipher text. In particular, we will be interested in additively homomorphic encryption schemes (Paillier ,1999) that is comparable with the encryption process of RSA in terms of the computation cost, while the decryption process of the additive homomorphism is faster than the decryption process of RSA. An additively homomorphic cryptosystem has the nice property that for two plain text message $m1$ and $m2$ , it holds $e(m1\ ) \times e(m2\ ) = e(m1 + m2\ )$ ,where × denotes multiplication. This essentially means that we can have the sum of two numbers without knowing what those numbers are. Moreover, because of the property of associativity, $e(m_1 + m_2 + \ldots + m_s\ ) = e(m1\ ) \times e(m2\ ) \times \ldots \times e(m_s\ )$ , where $e(m_i\ ) \neq 0$ . And we can easily have the following corollary: $e(m1\ )^{m2} \times e(m2\ )^{m1} = e(m_1 + m_{2)}$ An efficient implementation of an additive homomorphic encryption scheme with semantic security was given by Paillier [21].

**Threshold decryption:** Threshold decryption is an example of a multiparty functionality. The setting includes $m$ parties and an encryption scheme. It is required that any $m' < m$ of the parties are able to decrypt messages, while any coalition of strictly less than $m'$ parties learns nothing about encrypted messages. This functionality can, of course, be implemented using generic constructions, but there are specific constructions implementing it for almost any encryption scheme, and these are far more efficient than applying the generic constructions to compute this function ality. Interestingly, threshold decryption of homomorphic encryption can be used as a primitive for constructing a very efficient generic protocol for secure multiparty computation, with a communication overhead of only $O(mk/c/\ )$ bits (Franklin & Haber (1996) for a construction secure against semi-honest adversaries, and Cramer et al.(2001) for a construction secure against malicious adversaries).

**Other Cryptographic Tools:** Many basic security operations now have been applied to Secure protocols of privacy preserving data mining, such as Secure Sum, Secure Set, Secure Size of Set Intersection Union, Scalar Product [17].

### 4.3 Application of the secure multiparty computation technique

Secure Multi-party Computation (SMC) technique is a common approach for distributed privacy preserving data mining, and now has been extended to a variety of data mining problems. For example, [13] introduced a secure multi-party computation technique for classification using the ID3 algorithm, over horizontally partitioned data. Specifically, they consider a scenario in which two parties owning confidential databases wish to run a data mining algorithm on the union of their databases, without revealing any unnecessary information. Du & Zhan (2002) [18] proposed a protocol for making the ID3 algorithm privacy-preserving over vertically partitioned data. Vaidya & Clifton (2002) presented the component scalar product protocol for privacy-preserving association rule mining over vertically partitioned data in the case of two parties; Wright & Yang (2004) [20] applied homomorphic encryption to the Bayesian networks induction for the case of *two* parties. Zhan et al., (2007) [19] proposed a cryptographic approach to tackle collaborative association rule mining among multiple parties.

### 4.4 Common errors of the secure multiparty computation

There are common errors which often occur when designing secure protocols, here we would like to use this section to introduce some of these errors briefly, interested reader can refer to [13].

• **Semi-honest Behavior does not Preclude Collusions:** Assuming that adversaries are semi-honest does not ensure that no two parties collude. The “semi-honest adversary” assumption merely ensures that an adversary follows the protocol, and only tries to learn information from messages it received during protocol execution. It is still possible, however, that the adversary controls more than a single party and might use the information it learns from all the parties it controls.

• **Deterministic Encryption Reveals Information:** A common misconception is that encrypting data, or hashing it, using any encryption system or hash function, keeps the data private. The root of the problem is the use of a deterministic function (be it a hash function or a deterministic encrypting scheme such as textbook RSA). One should therefore never apply a deterministic function to an item and publish the result. Instead, a semantically secure encryption scheme must be used. Unfortunately, this rules out a number of “simple and efficient” protocols that appear in the literature (indeed, these protocols are not and cannot be proven secure).

• **Input Dependent Flow:** the flow of the protocol (namely, the decision which parts of it to execute), must not depend on the private input of the parties. Otherwise, the protocol is not secure

• **Security Proofs:** It is tempting to prove security by stating what constitutes a “bad behavior” or an “illegitimate gain” by the adversary, and then proving that this behavior is impossible. Any other behavior or gain is considered benign and one need not bother with it. This approach is often easier than the use of simulation based proofs. However , it is hard to predict what type of corrupt behavior an adversary might take and thus dangerous to disregard any other behavior that we have not thought of as useless for the adversary. Indeed, real world attackers often act in ways which were not predicted by the designers of the system they attack. It is also hard to define what constitutes a legitimate gain by the adversary, and allow it while preventing illegitimate or harmful gains. The notion of “harmful” might depend on a specific application or a specific scenario, and even then it might be very hard to define. So the protocol designers must prove security according to the simulation based proof [13] which prevent any attack which is not possible in an idealized scenario.

### 4.5 Evaluation of the secure multiparty computation technique

Secure Multiparty Computation enables distributed privacy preserving data mining without the trusted third party, Moreover, the secure multiparty computation technique make the result of data mining correct without information loss. The shortcoming of the technique is the computation and communication overhead of protocol is very high, especially for the large database, which hinder its application in practice. So secure multiparty computation, due to its high computational requirement, is most suitable for situations where the number of distributed sources is relatively small and the global analysis to be supported can be derived by the given set of primitives.

## 5. Conclusion

Due to the right to privacy in the information ear, privacy-preserving data mining (PPDM) has become one of the newest trends in privacy and security and data mining research. In this paper, we introduced the related concepts of privacy-preserving data mining and privacy preserving technique, explain the techniques for building secure multiparty computation protocols & Evaluation of Secure Multiparty Computation technique.

.

## References

[1] A. C. (1986). Yao How to Generate and Exchange Secrets. *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pp. 162-167, ISSN 0-8186-0740-8.

[2] Han J.& Kamber M.(2006). *Data Mining: Concepts and Techniques*. 2nd edition, San Francisco: Morgan Kaufmann Publishers

[3] Verykios V. S., Elmagarmid A. K., Bertino E., Saygin Y. & Dasseni E.(2004b) . Association Rule Hiding. *IEEE Transactions on Knowledge and Data Engineering*, Vol 16, Issue 4, pp.434–447, ISSN 1041-4347.

[4] Kargupta H., Das K.& Liu d K.(2007). Multi-party, privacy-preserving distributed data mining using a game theoretic framework. *PKDD*, Vol.4702, pp.523–531,Springer.

[5] Lindell Y. & Pinkas B.(2009). Secure Multiparty Computation for Privacy-Preserving Data Mining. *Journal of Privacy and Confidentiality*, Vol 1, No 1, pp.59-98.

[6] Clifton C., Kantarcioglu M. &Vaidya J.(2002a). Defining privacy for data mining. *Proceeding of the National Science Foundation Workshop on Next Generation Data Mining*, pp.126-133, Baltimore, MD, USA.

[7] Stanley R. M. Oliveira and Osmar R. Zaïane. (2004). Toward standardization in privacy preserving data mining, *ACM SIGKDD 3rd Workshop on Data Mining Standards*, pp. 7–17, Seattle, WA, USA.

[8] [8] Jiang W., Clifton C. & Kantarcioglu M. (2008). Transforming semi-honest protocols to ensure accountability. *Data and Knowledge Engineering*, Vol.65, pp.57–74, ISSN 0169-023X

[9] Rabin M. O. (1981). How to Exchange Secrets by Oblivious Transfer, *Technical Report* TR-81, Aiken Computation Laboratory.

[10] Goldreich O. (1998). Secure Multi-party Computation, http://www.wisdom weizmann.ac.il/.

[11] Yao A. C. (1986). How to Generate and Exchange Secrets. *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pp. 162-167, ISSN 0-8186-0740-8.

[12] Verykios V. S., Elmagarmid A. K., Bertino E., Saygin Y. & Dasseni E.(2004b) . AssociationRule Hiding. *IEEE Transactions on Knowledge and Data Engineering*, Vol 16, Issue 4, pp.434–447, ISSN 1041-4347

[13] Lindell Y. & Pinkas B.(2009). Secure Multiparty Computation for Privacy-Preserving Data Mining. *Journal of Privacy and Confidentiality*, Vol 1, No 1, pp.59-98.

[14] ] Even S., Goldreich O. & Lempel A.(1985). A Randomized Protocol for Signing Contracts, *Communications of the ACM*, vol. 28, Issue 6, pp. 637–647, ISSN 0001-0782.

[15] Stanley R. M. Oliveira and Osmar R. Zaïane. (2004). Toward standardization in privacy preserving data mining, *ACM SIGKDD 3rd Workshop on Data Mining Standards*, pp.7–17, Seattle, WA, USA

[16] Rabin M. O. (1981). How to Exchange Secrets by Oblivious Transfer, *Technical Report* TR-81, Aiken Computation Laboratory.

[17] Clifton C., Kantarcioglu M., Vaidya J., Lin X. & Zhu M.Y. (2002b). Tools for Privacy Preserving Distributed Data Mining. *ACM SIGKDD Explorations*, Vol 4, No 2, pp.28-34, ISSN 1931-0145.

[18] ] Du W. & Zhan Z. (2002). Building decision tree classifier on private data. *Proceedings of the IEEE international conference on Privacy, security and data mining*, pp. 1-8, ISSN 0-909- 92592-5, Maebashi City, Japan.

[19] Zhu Y.& Liu L. (2004).Optimal Randomization for Privacy-Preserving Data Mining. *ACM KDD Conference*, pp.761-766, ISBN1-58113-888-1, Seattle, WA, USA.

[20] Wright R. & Yang Z. (2004). Privacy-preserving bayesian network structure computation on distributed heterogeneous data. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 713-718, ISBN:1-58113-888-1. ACM, New York, NY, USA.

[21] Paillier P. (1999). Public-key Cryptosystems based on Composite Degree Residuosity Classes. *Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, pp. 223–238, ISSN 3-540-65889-0, Prague, Czech Republic

.

## Author Profile

1. **Neha Kashyap** received the B.Sc. in Computer Application from Ranchi Women's

College, Ranchi during 2001-2004 and MCA from IGNOU during 2006-2009. She is pursuing M.Tech. in Computer Science from Birla Institute of Technology, Extension Centre, Lalpur. She has 3 years of teaching experience as a Lecturer of Computer Application. She has served as a programmer in Rural Development Department of Jharkhand.

**2.** **Dr. Vandana Bhattacharjee** P.hd. in Computer Science, M.Tech. in Computer Science & B.E. in Computer Science. She is Professor in Birla Institute of Technology, Department of Computer Science & Engineering, Extension Centre, Lalpur (Ranchi City).