Privacy Preserving Secure Communication Pattern using Cryptographic Technique via Trusted Third Party (TTP) Model in Distributed Data Mining

[1]N. K. Kamila, [2]L. D. Jena

[1]Department of Computer Science and Engineering, C. V. Raman College of Engineering, Bhubaneswar, India
[2]Gandhi Engineering College, Bhubaneswar, India

Abstract: As security plays an important role in most of the applications where small change of data can lead to major problems. Thus, to secure the information there is need of a stronger encryption, which is very hard to break. In order to achieve better results and improve security, several levels of encryption have to be applied to the information and the way of encryption should not be vulnerable to attacks. Some of the conventional encryption methods are implemented for encryption, but can be cracked easily with the high end technologies. The goal of this paper is to develop multi-level encrypting technique that can be used to encrypt top-secret data in the database. In this information era, data mining has emerged as a way for distinguishing patterns from huge quantities of information. Some database may contain private or personal data which should not be leaked out. Thus techniques of data mining without leaking the private information are needed. Also there is a demand for a stronger encryption which is very hard to crack. In this paper, we proposed a multi level of multiple encryption schemes for secure communication pattern (SCP) which enhances the security of the private data in database during data mining. In the recent past years, Privacy Preserving Data Mining (PPDM) has attracted research interest with potential for wide applications. Many techniques such as randomization, cryptography and anonymity have been experimented with privacy preserving data mining. The work considers information system based approach as not all attributes may store same level of sensitive data. Therefore, some attribute values may require higher degree of privacy preservation than some others. Here we explore the use of cryptography, namely Multilevel Encryption and Decryption (MLED) algorithm for encrypted data sharing to achieve privacy preservation. We focus on privately Secure Communication Patterns (SCP) that refers to a contributively relation of participants in computations of data mining under Trusted Third Party model. Here we analyze and assess each aspect of this issue, introducing a proposed framework based on strategies of SCP using multilevel encryption and decryption (MLED) algorithm in data mining with respect to the privacy preserving. Though our model consumes more CPU utilisation and execution time but it provides more security than the other cryptographic techniques.

Keywords: SCP, Privacy preserving, distributed data mining, cryptography

I. INTRODUCTION

In modern years, data mining has been observed as a risk to privacy because of the extensive proliferation of electronic data maintained by organizations. This has led to increased worries about the privacy of the essential data. In recent years, a number of techniques have been proposed for modifying or transforming the data in the way of preserving privacy. Data mining is used for retrieving intelligent information from huge databases. Presently these databases are distributed across the world. Distributed data must be retrieved from multiple locations in to the data warehouse. So there is a requirement for a secure transmission and maintaining confidentiality. The transmitted data may contain information which may be private to individual or corporate information which must be secured. Also it contains data perturbation technique which has different idea, that the distorted data does not reveal private information, and thus it is "safe" to use for mining.

Data are nowadays gathered in large contents by companies and national offices. These data are often analyzed either using statistical or data mining approaches. When such methods are applied within the walls of the company that has gathered them, the risk of disclosure of sensitive information might be finite. When the analysis has to be performed by third parties, privacy becomes a much more relevant subject [1]. Data Mining is now, as much as ever, a necessity in today's e-science environment [2]. Although data mining can be valuable in many usages, but the violation of privacy is feasible in the absence of enough preservation, and private information may be used for other scopes [3]. Privacy preservation in administrative, statistical and other data sets is about finding tradeoffs between the societal right to know and the individual right to private life. Thus, the passive issues of privacy is the individual citizen or, in business data sets, the individual company [4].

Privacy preserving data mining is assumed to have important role in the whole area of knowledge discovery

because of the increasing sensitivity of information that can be used for data mining process [5]. The technologies based on Privacy Preserving Data Mining (PPDM) allow us to extract relevant knowledge from a large amount of data, while hiding sensitive data from disclosure [6]. A wide variety of different approaches are recognizable in the field of privacy-preserving data mining. These techniques use different methods: in perturbation, noise is added to the database that is input to the algorithm in some occasions and to the output of queries in some other occasions to blur the values of sensitive properties. In generalization, defining properties have less specific values. In cryptography, joint computations between multiple parties are performed on encrypted data in order to hide inputs[3].Cryptography offers a well-defined approach for preserving privacy in data mining, which includes techniques for proving and quantifying it. There exists a vast toolset of cryptographic models and constructs to implement privacy-preserving data mining algorithms, on the other hand, this approach is especially difficult to scale when more than a few parties are involved. Distributed data mining considers the scenario where a number of distinct, yet connected, computing parties wish to carry out a collaborative computation.

The aim of our proposed framework is to enable parties to carry out such distributed data mining tasks in a secure manner. Thus, two important requirements on any secure communication protocol are privacy and correctness [8].

II. RELATED WORK

Recent developments in information technology have made possible the collection and analysis of millions of transactions containing personal data. These data include shopping habits, criminal records, medical histories, and credit records, among others. This progress in the storage and analysis of data has led individuals and organizations to face the challenge of turning such data into useful information and knowledge.

Data mining is a promising approach to meet this challenging requirement. The area of data mining, also called Knowledge Discovery in Databases (KDD), has received special attention since the 1990s. This new research area has emerged as a means of extracting hidden patterns or previously unknown implicit information from large repositories of data. The fascination with the promise of analysis of large volumes of data has led to an increasing number of successful applications of data mining in recent years. Undoubtedly, these applications are very useful in many areas such as marketing, business, medical analysis,

and other applications in which pattern discovery is paramount for strategic decision making.

Despite its benefits in various areas, the use of data mining techniques can also result in new threats to privacy and information security. The problem is not data mining itself, but the way data mining is done. As Vaidya & Clifton [40] state, “Data mining results rarely violate privacy, as they generally reveal high-level knowledge rather than disclosing instances of data”. However, the concern among privacy advocates is well founded, as bringing data together to support data mining projects makes misuse easier. Thus, in the absence of adequate safeguards, the use of data mining can jeopardize the privacy and autonomy of individuals.

Complex issues involved in privacy-preserving data mining (PPDM), cannot simply be addressed by restricting data collection or even by restricting the secondary use of information technology [7, 8, 10]. Moreover, there is no exact solution that resolves privacy preservation in data mining. An approximate solution could be sufficient, depending on the application since the appropriate level of privacy can be interpreted in different contexts [28, 27]. In some applications (e.g., association rules, classification, or clustering), an appropriate balance between a need for privacy and knowledge discovery should be found.

Preserving privacy when data are shared for mining is a challenging problem. The traditional methods in database security, such as access control and authentication that have been adopted to successfully manage the access to data present some limitations in the context of data mining. While access control and authentication protections can safeguard against direct disclosures, they do not address disclosures based on inferences that can be drawn from released data [5, 14]. Preventing this type of inference detection is beyond the reach of the existing methods [7, 10].

Clearly, privacy issues pose new challenges for novel uses of data mining technology. These technical challenges indicate a pressing need to rethink mechanisms to address some issues of privacy and accuracy when data are either shared or exchanged before mining. Such mechanisms can lead to new privacy control methods to convert a database into a new one that conceals private information while preserving the general patterns and trends from the original database.

The data mining results not only gives the valuable information hidden in the databases, but sometimes also reveals private information about individuals. The difficulty is that data mining process extracts or evaluates the individual data which is considered as private by means of linking different attributes. PPDM is an emerging technique in data mining where privacy and data mining can coexist. It gives the summarized results without any loss of privacy through data mining process. One of the main approaches in PPDM is cryptographic-based techniques. In cryptographic techniques the data is encrypted using encryption methods and a set of protocols are used to allow the data mining operation. The set of protocols called as secured multiparty computation (SMC) is a computation process performed by group of parties where each party performs computation with part of needed input data

within its control. In SMC the participating parties learns only the final result of the computation and no additional information is revealed during the computation. Perfect privacy in the SMC [35, 36] is achieved because any meaningful information is not released to any third party. The basic SMC PPDM techniques are secure sum, secure set union and secure size of set union [40].

Kamakshi et al. [33] presented a novel approach in which data perturbation technique used to modify the original data at the level of data owner and then used cryptographic technique to submit the result of customer query. Lakshmi et al. [34] proposed a model which adopts a hash based secure sum cryptography technique to find the global association rules by preserving the privacy constraints.

Many authors have also discussed various classes of data mining techniques and its contrasting features exist among them [37]. In [39], the authors presented ID3 classification for two parties with horizontally partitioned data by using secure protocols to achieve complete zero knowledge leakage. In [41], the authors have discussed the problem of privacy preserving data mining of association rules when the data is partitioned horizontally. They proposed algorithm which uses three basic ideas such as randomization, encryption of site results and secure computation. The state of art in the area of privacy preserving data mining techniques is presented in [42]. A framework for evaluating privacy preserving data mining algorithms and based on this frame work one can assess the different features of privacy preserving algorithms according to different evaluation criteria [43]. An enhanced scheme has been proposed by Kantarcioglu et al. [44], which is a two phase privacy preserving distributed data mining scheme. In [45], the authors discussed the problem of privacy preserving data mining in distributed data bases. They suggested a new paradigm based on two separate entities, a minor and a calculator, both are not having any parts of the data base. They also presented three algorithms based on this paradigm, one for horizontally partitioned data, one for vertically partitioned data and one for any data mining method. The authors in [46] proposed a new algorithm for mining association rules in distributed homogeneous databases based on semi honest model and negligible collision probability. In [47], the authors presented a classification, an extended description and clustering of various association rule mining algorithms. They also suggested further research directions of privacy preserving data mining algorithms by analyzing the existing work.

None of the above authors have considered the problem like use of multiple encryption and decryption algorithms in multilevel cryptographic technique under trusted model in distributed data mining. In this work, we propose a multilevel of multiple encryption schemes for secure communication pattern (SCP) which enhances the security of the private data in database during data mining. We focus on privately Secure Communication Patterns (SCP) that refers to a contributively relation of participants in computations of data mining under trusted third party model.

III. PROBLEM STATEMENT

As security plays an important role in most of the applications, any small change of data leads to major problems. Hence it needs stronger encryption which is to be hard to break. In order to achieve better results and to improve security, information has to pass through several levels of encryption process. For ensuring the security, the plain text is converted to cipher text and the process is called encryption. Although this conversion idea is old, the way of encryption should not be vulnerable to attacks. Caesar's cipher method, poly alphabetic substitution method, bit-level encryptions like substitution box; permutation box, encoding, and rotation are some of the conventional encryption methods. These methods are easy to implement but can be cracked easily with the high end technologies. The objective of this chapter is to develop multi-level encrypting technique that can be used to encrypt top-secret data of multiple parties for data mining task. It explores the use of cryptography, namely multilevel encryption and decryption (MLED) algorithm for encrypted data sharing to achieve privacy preservation. The experiments have been carried out on the size of the secret key that is proportional to the level of sensitivity of the attribute for multilevel security. This focuses on privately Secure Communication Patterns (SCP) that refers to a contributively relation of participants in computations of data mining under Trusted Third Party model.

Although many authors have worked on privacy preservation in distributed data mining, but none of the authors has used multiple encryption and decryption algorithms in multilevel cryptographic technique under trusted model in distributed data mining. Yarimi et al.(2012) have used DES and RSA algorithms to implement multilevel privacy preservation in distributed environment, but they have not considered multiple encryption and decryption algorithms for secure communication pattern.

IV. PRELIMINARIES

4.1 Privacy Preserving Data Mining (PPDM)

Privacy Preserving Data Mining (PPDM) refers to the area of data mining to protect sensitive information from disclosure. The problem with data mining output is that it also reveals some information, which is considered to be private and personal. Easy access to such personal data poses a threat to individual privacy. The actual anxiety of people is that their private information should not be misused behind the scenes without their knowledge. The real threat is that once information is unrestricted, it will be impractical to stop misuse. There has been growing concern about the chance of misusing personal information behind the scene without the knowledge of actual data owner. Privacy preserving data

mining technique gives new direction to solve this problem. PPDM gives valid data mining results without learning the underlying data values. The benefits of data mining can be enjoyed, without compromising the privacy of concerned individuals. The original data is modified or a process is used in such a way that private data and private knowledge remain private even after the mining process. The main purpose of privacy preserving data mining is to design efficient frameworks and algorithms that can extract relevant knowledge from a large amount of data without revealing of any sensitive information.

4.2 Techniques in Privacy-Preserving Data Mining

Randomization Method: The randomization method is a technique for privacy-preserving data mining in which noise is added to the data in order to mask the attribute values of records. The noise added is sufficiently large so that individual record values cannot be recovered. Therefore, techniques are designed to derive aggregate distributions from the perturbed records. Subsequently, data mining techniques can be developed in order to work with these aggregate distributions.

- **Additive Perturbation**: In this case, randomized noise is added to the data records. The overall data distributions can be recovered from the randomized records. Data mining and management algorithms need to be redesigned to work with these data distributions. In Additive Data Perturbation (ADP) method the data is changed by adding a noise term e to the attribute X resulting in Y, Y=X+e, where e is drawn from some probability distribution.

- **Multiplicative Perturbation**: In this case, the random projection or random rotation techniques are used in order to perturb the records. Privacy preserving data mining is used for secure mining from the data warehouse. Random perturbation technique is a method to convert raw data based on probability which has been discussed. Data distortion is achieved by changing the original data, in which some randomness value is added such that the original data is difficult to ascertain, while preserving global feature of a record as shown in [3]. In Multiplicative Data Perturbation (MDP) the value of e is multiplied with X to get Y the perturbed value, Y=Xe, where e has mean of 1.0 and a specified variance as shown in [3].

In randomization perturbation approach the privacy of the data can be protected by perturbing sensitive data with randomization algorithms before releasing it to the data miner. The perturbed data version is then used to mine patterns and models. In this method privacy of confidential data can be obtained by adding small noise component which is obtained from the probability distribution. In a set of data records denoted by $X = \{x_1, x_2, ..., x_N\}$. For record $x_i \epsilon X$, we add a noise component which is drawn from the probability distribution. Commonly used distributions are the uniform distribution over an interval [-a, a] and Gaussian distribution with mean $\mu = 0$ and standard deviation s. These noise components are drawn independently, and are denoted $y_1, y_2, ..., y_N$. Thus, the new set of distorted records is denoted by $x_1 + y_1, ..., x_N + y_N$. It is denoted by this new set of records $z_1, z_2 ..... z_N$, where $z_1 = x_1 + y_1$, $z_2 = x_2 + y_2$ ..... and $z_N = x_N + y_N$. In general, it is assumed that the variance of the added noise is large enough, so that the original record values cannot be easily guessed from the distorted data. One key advantage of the randomization method is that it is relatively simple, and does not require knowledge of the distribution of other records in the data.

K-Anonymity: The k-anonymity model has been developed because of the possibility of indirect identification of records from public databases. This is because combinations of record attributes can be used to exactly identify individual records. In the k-anonymity method, it reduces the granularity of data representation with the use of techniques such as generalization and suppression. This granularity is reduced sufficiently that any given record maps onto at least k other records in the data. An important method for privacy de-identification is the method of k-anonymity. The motivating factor behind the k anonymity technique is that many attributes in the data can often be considered pseudo-identifiers which can be used in conjunction with public records in order to uniquely identify the records. For example, if the identifications from the records are removed, attributes such as the birth date and zip-code can be used in order to uniquely identify the identities of the underlying records. The idea in k-anonymity is to reduce the granularity of representation of the data in such a way that a given record cannot be distinguished from at least (k - 1) other records as shown in Table 1.

Table 1: K-Anonymous Data

Age	Weight	Name
35	50	Anny
60	55	Jenny
65	50	Nikhil

(a) Original Data

Age	Weight	Name
[35,45]	[50,65]	Anny
[35,65]	[50,65]	Jenny
[55,65]	[50,65]	Nikhil

(b) K-anonymous Data

Cryptographic Techniques: In many cases, multiple parties may wish to share aggregate private data, without leaking any sensitive information at their end. For example, different superstores with sensitive sales data may wish to coordinate among themselves in knowing

aggregate trends without leaking the trends of their individual stores. This requires secure and cryptographic protocols for sharing the information across the different parties. Cryptography, the science of communication and computing in the presence of a malicious adversary extends from the traditional tasks of encryption and authentication. In an ideal situation, in addition to the original parties there is also a third party called "trusted party". All parties send their inputs to the trusted party, who then computes the function and sends the appropriate results to the other parties as shown figure 1. The protocol that is run in order to compute the function does not leak any unnecessary information. Sometimes there are limited leaks of information that are not dangerous. This process requires high level of trust.

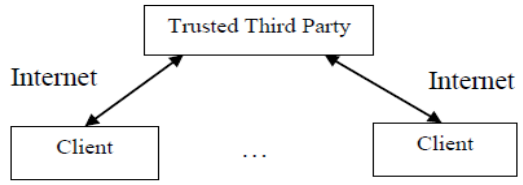

Figure 1: System using trusted third party

V. CRYPTOGRAPHY-BASED FRAMEWORK FOR CLASSIFICATION OF THE SCP APPROACHES

Analysis of techniques of Secure Communication Pattern in the process of privacy preserving distributed data mining (PPDDM) shows that these techniques can be explained from different scopes. The result of this study according to researchers and their analysis already carried out is shown in form of an analytic framework in Table 2. The Cryptography-based framework tends to be applied in resolving some challenging problems and designing and developing methods of Secure Communication Pattern in distributed data mining with respect to privacy preserving. Each of these approaches takes the concept of data mining into consideration with respect to privacy preserving from different prospective. These prospects include Specific Secure Multi-party Computation (SSMPC) model and Semi-Trusted Third Party (STTP) model.

Table 2: Cryptography-based Framework for Classification of the SCP Approaches

	Cryptography based Approaches of PPDDM							
SCP	Specific Secure Multi-Party Computation(SSMPC)				Semi-Trusted Third Party (STTP)			
	Secure Sum	Secure Set Union	Set Intersection	Scalar Product	Mixer	Miner	Commodity Server	External Unit

1.1.

1.2. Approaches of PPDDM from the Secure Communication Pattern

Distributed data mining is a process to extract globally interesting associations, classifiers, clusters, and other algorithms from distributed information [9]. Distributed data mining can be classed into two classifications [10]. The first is server-to-server where information/data are distributed across several servers. The second is client-to-server where information/data reside on each client while a server performs mining operations on the aggregate data from the clients [11]. The secure communication pattern framework refers to a contributively relation of participants in computations. Overall, the most practical options to handle this scenario is to run secure computation either in one or more numbers of participants, or in one or more third parties, assuming that either all participants are semi-honest or all third parties are semi-trusted [12]. To solve problems of privacy-preserving distributed data mining, two alternatives based on secure computation model are suggested.

5.1.1 Specific Secure Multi-Party Computation (SSMPC) Model

Secure Multi Party Computation (SMPC) is a mechanism for privacy preserving data mining which is meant for joint computations in distributed environment [13]. Many SMPC-based solutions can be used to ensure privacy preservation. Informally, this group of techniques can be described as a computational process where two or more parties compute a function based on private inputs. Privacy in this scope means that none of the parties wants to disclose its own input to any other party [14]. Multiple parties sharing the burden of creating the data aggregate. Final processing, if needed, is delegated to trusted third party (TTP). Computation is considered secure if each party only knows its input and the result of its computation.

The purpose of this model is opting for efficient and accurate solutions to tackle PPDDM issues. The semi-honest assumption holds that to cope with functions commonly taken advantage of in data mining applications rather than the general secure multi-party computation protocol, specific secure multi-party computation protocols are employed comprising the techniques of secure sum, secure set union, secure

intersection, secure scalar product, and so forth [12, 13, 15 and 16]. Being specifically attuned with the data mining tasks rather than with general functions, this kind of protocol has the advantage in design. The computing complexity is diminished, and a linear proportional cost can be obtained when the function for secure computation can be identified. In secure multi-party computation (Figure 2), a union of N sections with private inputs $x_1,\ldots, x_n$ on a network can compute a joint function of their inputs. This joint computation should have the attribute that the sections learn the accurate output $y = F(x_1,\ldots, x_n)$ and nothing else [17].

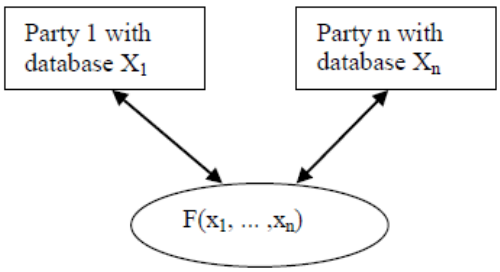

Figure 2: Secure Multi-Party Computation System

➢ ***Secure Sum****:* Secure sum algorithm is often given as a simple instance of secure multiparty computation [18]. This method can securely calculate the sum of values from different sites. Assume that each site i has some value v_i and all sites want to securely compute $S = v_1 + v_2 + \ldots + v_n$, where v_i is known to be in the range [0..n]. A site designated as master site, numbered as 1, generates a random value R, uniformly chosen from [0..n] and adds this to its v_1, then sends the sum $R + v_1$ *mod* n to next site and so on (depicted in figure 3). Site *l* receives eq. (1) and site i then computes eq. (2) [16, 18] and passes it to site *l*+1. Hence site i learns nothing about previous original values.

$$V=R+\sum_{j=1}^{l-1} vj \ \boldsymbol{mod}\ n \qquad (1)$$

$$R+\sum_{j=1}^{l} vj \ \boldsymbol{mod}\ n=(vj+ V) \ mod\ n \qquad (2)$$

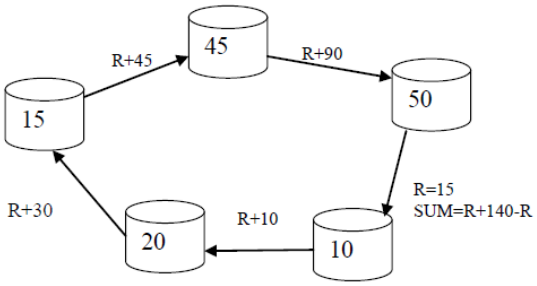

Figure 3: Secure Computation of a Sum

➢ ***Secure Set Union****:* Secure set union techniques are effective in data mining where each party needs to give rules, frequent item sets, etc., without disclosing the owner. The union of items can be evaluated using secure multi party computation methods if the domain of the items is small. Each party creates a binary vector where 1 in the i^{th} entry represents that the party has the i^{th} item. After this step, a simple circuit that or's the corresponding vectors can be built and it can be safely evaluated using universal secure multi party circuit evaluation protocols. However, in data mining the domain of the items is usually large. To overcome this problem a simple approach based on particular encryption is used, such commutative encryption that is shown in figure 4. This algorithm E which takes as input plaintext M for any non-similar encryption keys K_1, …, Kn and any permutations of i, j, the same cipher text would be the result [18].

$$E_{k_{i1}}(\ldots E_{k_{in}}(M)\ldots) = E_{k_{j1}}(\ldots E_{k_{jn}}(M)\ldots) \qquad (3)$$

$\forall M_1, M_2 \in M$ such that $M_1 \neq M_2$ and for given k $\varepsilon < 1/2^k$ P_r ($E_{k_{i1}}(\ldots E_{k_{in}}(M_1)\ldots) = E_{k_{j1}}(\ldots E_{k_{jn}}(M_2)\ldots)) < \varepsilon$ (4)

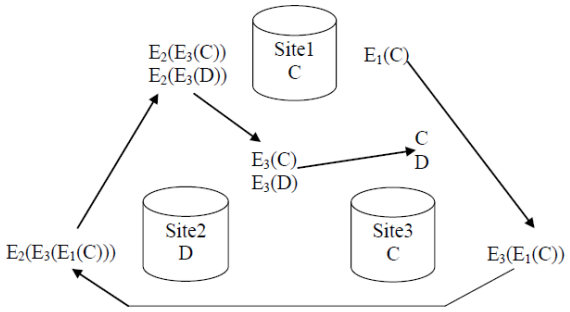

Figure 4: Determining the Union of a Set of Items

The general idea is that each site encrypts its items. Each site then encrypts the items from other sites. Since equation (3) holds, duplicates in the original items will be duplicates in the encrypted items, and can be deleted. Due to equation (4), only the duplicates will be deleted. In addition, the decryption process can occur in any order, so by permuting the encrypted items we prevent sites from tracking the source of an item [18].

➢ ***Secure Size of Set Intersection****:* Consider several parties with their own sets of items from a common domain. The problem is to securely compute the cardinality/size of the intersection of these local sets. The summary of this protocol is shown in [18]as explained below.

- Given k parties P_1 … P_k having local sets S_1 … S_k, we wish to securely calculate $|S_1 \cap \ldots \cap S_k|$.
- All k parties locally generate their public key-pair (E_i, D_i) for a commutative encryption pattern.
- Each party encrypts its items with its key and sends it along to the other parties.
- On receiving a set of items, a party encrypts each item and changes the order before sending it to the next party.

Since encryption is commutative, the results from two distinct sets will be identical if the original values were the similar. Thus, we need only count the number of values that are present in all of the encrypted item sets that this can be done by any party. None of the party is able to learn which of the items are present in the intersection set because of the encryption [18].

➢ ***Scalar Product:*** This method is a powerful component technique [18]. The problem can be defined as follows: party 1 has a n-dimensional vector X = (x_1, x_2, …, x_n), while party 2 has a n-dimensional vector Y = (y_1, y_2, …,y_n). At the end of the protocol, party 1 should get $r_a = X \cdot Y + r_b$ where r_b is a random number chosen from uniform distribution that is known only to party 2, and $X \cdot Y = x_1y_1 + x_1y_2 + \ldots + x_ny_n$. Using the scalar product protocol we can safely calculate the global support count of a data set whose items are located at different parties [16, 18]. The scalar/dot product of two vectors X and Y of cardinality n is presented as:

$$\vec{X} \cdot \vec{Y} = \sum_{i=1}^{n} \text{xi} * \text{yi} \qquad (5)$$

In [20] the problem is modelled as SMC and a solution by using oblivious transfer is presented. This, however, is not very efficient. The key vision in [21] is to use linear combinations of random values to disguise vector items and then do some computations to remove the effect of these random from the result.

5.1.2. Semi-Trusted Third Party (STTP) Model

In theory, to deal with any collaborative data mining problems, the general multi-party computation protocol can be utilized despite the fact that this kind of solution is not at all promising when the amount of database is high, and the number of participants is large, due to its intricate and complicated design. On the other hand, the trusted third party (TTP) is too naïve and straightforward, so the privacy is compromised to a larger extent at the point of the TTP [12]. Therefore, to solve the privacy issues of distributed data mining efficiently and accurately, more practical solutions have already been worked out in the past few years. Out of these solutions, two broad streams of ideas are manifesting themselves: one is to introduce a semi-trusted third party, as compared to the trusted third party (TTP) (Figure 5). In the reality, finding a semi-trusted third party is much more feasible than finding a trusted third party. The implementation of the semi-trusted third party is made feasible applying a miner [22, 23], mixer [11, 24, 25], a commodity server [26, 27], or an external unit [23, 28, 29] – all behaving in accordance with semi-trusted model.

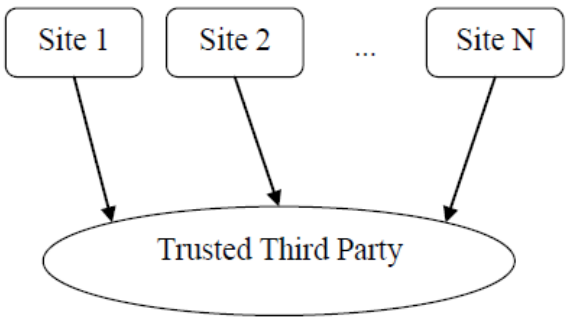

Figure 5: SCP Using Trusted Third Party

1) **Miner**: In privacy preserving distributed data mining a data miner computes results of data mining of each participant, without revealing the sensitive data. Unlike general-purpose cryptographic methods, this approach requires no interaction between participants, but each participant only needs to send a single flow of communication to the data miner [22]. Data miner can be used with external unit, for example, in [23] there is a computer called miner which manages the data mining process and reports the results to the participants. The miner has no part of the database. In the following definition [22], assume that each participant U_i has private keys x_i, y_i and public keys X_j, Y_j. A model for mining problem protects each participant's privacy against the miner and k corrupted participants in this pattern if, $\forall$ I $\subseteq$ {1, …, n} such that |I| = k, there exists a probabilistic polynomial-time algorithm M such that:

$$\{ M (d, [di, xi, yi]_{i\in I}, [Xj, Yj]_{j\notin I})\} \overset{c}{\equiv} \{ \text{view}_{\text{miner},}\{Ui\}_{i\in I} ([di, xi, yi]_{i=1}^{n}) \}. \qquad (6)$$

Where, $\overset{c}{\equiv}$ denotes computational indistinguishability, and $\{ \text{view}_{\text{miner},}\{Ui\}_{i\in I} ([di, xi, yi]_{i=1}^{n})\}$ is the joint view of the miner and the *k* corrupted participants, $\{Ui\}_{i\in I}$.

Intuitively, this definition states that a polynomial-time algorithm *M*, called a simulator, can simulate what the miner and the corrupted participants have observed in the protocol using only the final result *d*, the corrupted user's knowledge, and the public keys. Therefore, the miner and the corrupted participants jointly learn nothing beyond *d*.

2) **Mixer**: In privacy preserving distributed data mining, a semi-trusted mixer model is offered in which each party sends values to mixer. In this algorithm mixer is trusted, it mix received values and then broadcast the result. This pattern (Figure 6) is centralized and simplifies trust management. Under this approach, a protocol is developed to mine results from the union of distributed databases. In this semi-trusted mixer model, it is assumed that each party never collude with the mixer to learn the privacy of any other database. Otherwise, his/her own privacy will also be disclosed to the mixer. This protocol can protect the privacy of each party against a coalition of up to N-2 other parties (where N is the total number of participants) or even the mixer [24]. In [11], a new mixer model is proposed, where two mixers privately collect data from parties and run the protocol. The two mixers are semi-trusted that they correctly follow the protocol, but each of them may want to learn something that violates the privacy of databases. In [18], there would be a semi-trusted mixer which would receive encrypted count values from parties through its private channel. Parties communicate to the mixer through the private channel and the mixer communicates to all parties through public channel.

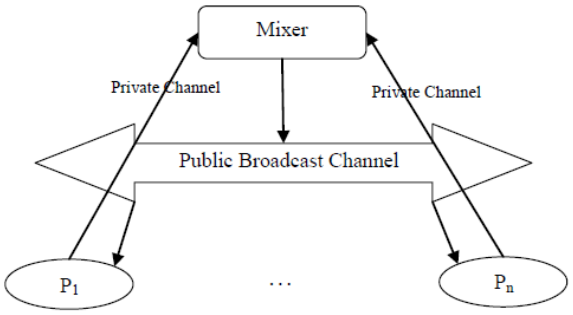

Figure 6: Semi-Trusted Mixer Model

3) **Commodity Server**: The commodity server pattern is first proposed by Beaver [30, 31], and has been used for solving private information retrieval problems [26, 27]. For performance reasons, the model introduces an extra server, the commodity server, belonging to a third party (Figure 7). Participants could send request to the commodity server and receive data (called commodities) from the server, but the commodities should be independent of participant's private data. The purpose of this pattern is to help participants to perform their desired computation [26]. Commodity server has a few appealing properties [27]. The properties are

a) It does not participate in the computation between participants, but it does prepare data for them to hide their private information.

b) The data provided by the server does not depend on the participant's private information, so the server does not want to know those private data.

As per the architecture the external unit is a calculator which computes without really knowing what item set it computes and has no part of the database [23]. It is important to note, that only the miner and the parties get the mining results while the external unit only executes ancillary computations, without knowing their meaning. This model will reduce communication in computing the protocol because firstly communication takes time longer than local mining and secondly instead of using several rounds, we use only one round for computing results of data mining.

VI. VALIDATION FOR THE SCP APPROACHES IN PPDDM

Researchers have developed some set of measuring metrics regarding privacy-preserving data mining. In [32], Elisa et al. have proposed a framework for evaluating privacy-preserving data mining. Here, the focus is on how well the algorithms perform to achieve the security goals – the efficiency parameter. The validation of SCP in PPDDM is almost complex due to various approaches, and usually it is not possible for fitting various criterions in all techniques. To assess aforementioned facets, the following criterions are taken into account, and ranking is done in three different ranks – low, medium and high shown in the table 3.

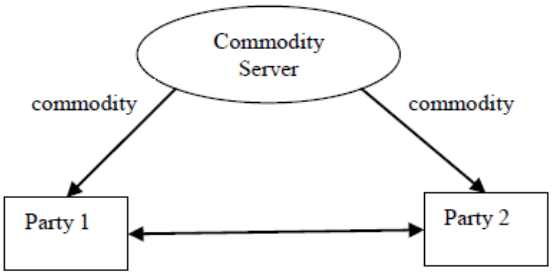

Figure 7: Commodity Server Model

4) **External Unit**: This method is a powerful component which makes use of an untrusted, non-colluding party (Figure 8), a party that is not allowed to learn anything about any of the information, but is believed not to collude with other parties to disclose data about the information. The results from all parts are combined, scrambled, and given to the untrusted, non-colluding party [29].

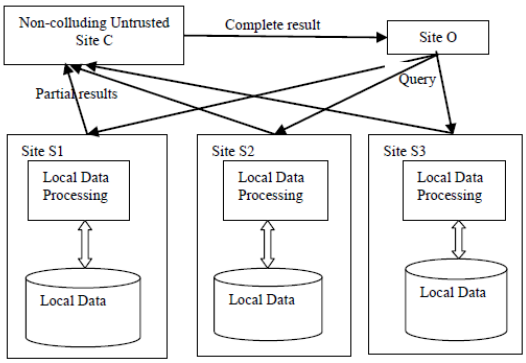

Figure 8: External Unit Model

The various comparison criteria's considered are:

- Rate of input change i.e. the degree of change in input data of participants in the process of data mining.
- Performance i.e. to employ algorithms by appropriate coding.
- Privacy preserving i.e. the degree of privacy preservation, applying methodologies to curb the disclosure of participants' exclusive information in the process of data mining.
- Accuracy of the results of data mining i.e. it is the level of validity and accuracy of the results in the end of apply data mining protocols.
- Scalability i.e. if the performance of a process won't decrease with increasing the amount of participating parties in the data mining, this process will be scalable.
- Complexity i.e. it is the total of computation and communication cost. Communication costs are dependent on the amount of sent message between different sites in the data mining process and the computation costs are calculated according to the

amount of the encryption and decryption process which are done.

Table 3: Validation for the SCP approaches in PPDDM

			COMPARISON CRITERIA					
			Rate of Input Change	***Performance***	***Privacy Preserving***	***Accuracy***	***Scalability***	***Complexity***
Secure Communication Pattern	**SSMPC**		**Low**	**Medium**	**High**	**High**	**Low**	**High**
	STTP	**Miner**	**High**	**High**	**High**	**High**	**High**	**Low**
		Mixer	**High**	**High**	**Medium**	**Medium**	**High**	**Medium**
		Commodity server	**High**	**High**	**High**	**Medium**	**High**	**Medium**
		External Unit	**High**	**High**	**Medium**	**High**	**High**	**Low**

The validation strategy is based on survey of different methods which are proposed in the fields of PPDDM. According to the survey, a thorough comparison of the performance of these two protocols (SSMPC and STTP) in terms of their communication cost and computation cost has been done by Zhuojia Xu [12]. Based on experimental results with Heart Disease Multivariate dataset consisting of 76 attributes and 293 instances, it has been identified that the STTP algorithms group have better computation and communication overhead in compare to SSMPC algorithms group. Moreover, the communication, computation cost and the communication rounds of the STTP-based algorithms are all lower than those of SSMPC-based algorithms. Therefore, STTP approaches could maintain better performance and scalability. But both groups in the proposed SCP framework provide privacy without losing so much accuracy.

VII. PROPOSED FRAMEWORK

The symmetric key algorithms and public key cryptography algorithms are always combined together in order to achieve the optimal efficiency. The use of multi level encryption & decryption algorithm and RSA algorithms for privacy preserving data mining in the distributed environment has been proposed for sharing the data with a trusted third party (TTP) for analysis and pattern extraction. The proposed approach offers a framework for each data owner to know only his/her input, the set of private keys, the public key and the outcome of knowledge extraction. The TTP would be in possession of all the sets of public keys corresponding to all the data owners to successfully decrypt the data. After executing some data mining algorithms the TTP would send the data mining results to the data owners. If the extracted patterns still disclose some sensitive information, then the patterns are encrypted using the private key set by the data owner and the TTP with the same multi level encryption & decryption algorithm and RSA algorithms. However, this is observed to be faster than data communicated as the size of knowledge is significantly less than the size of data communicated between the two parties. The working principle of the proposed model is shown in the figure 9 (schematic diagram).

7.1 The Multilevel-Privacy Preserving Procedure

The procedure for the proposed multilevel privacy preserving data mining system in a distributed environment is presented as follows.

1. A pair of keys (public and private) is generated using Rivest-Shamir-Adleman (RSA) algorithm by the trusted third party (TTP). The private one is kept a secret and the public key is published. The size of private and public key depends on the level of secrecy that we want.
2. A pair of keys (public and private) is generated using RSA algorithm by every other party.
3. The data is encrypted using the secret key in multi level encryption algorithm by all parties. The secret key is encrypted using the trusted third party's (TTP) public key. The encrypted data and the encrypted secret key are sent to the TTP.
4. Receive the encrypted data and key by TTP. TTP will use his private key to encrypt the secret key then use the secret key to decrypt the data.
5. Doing the data mining functions by TTP.
6. TTP will send the result of the data mining functions to all parties in an encrypted form or in a normal form depends on the knowledge itself. If the extracting results still disclose some sensitive information TTP will encrypt again before sending it to the other parties using multi level decryption algorithm & RSA algorithm as described in steps 3-4.

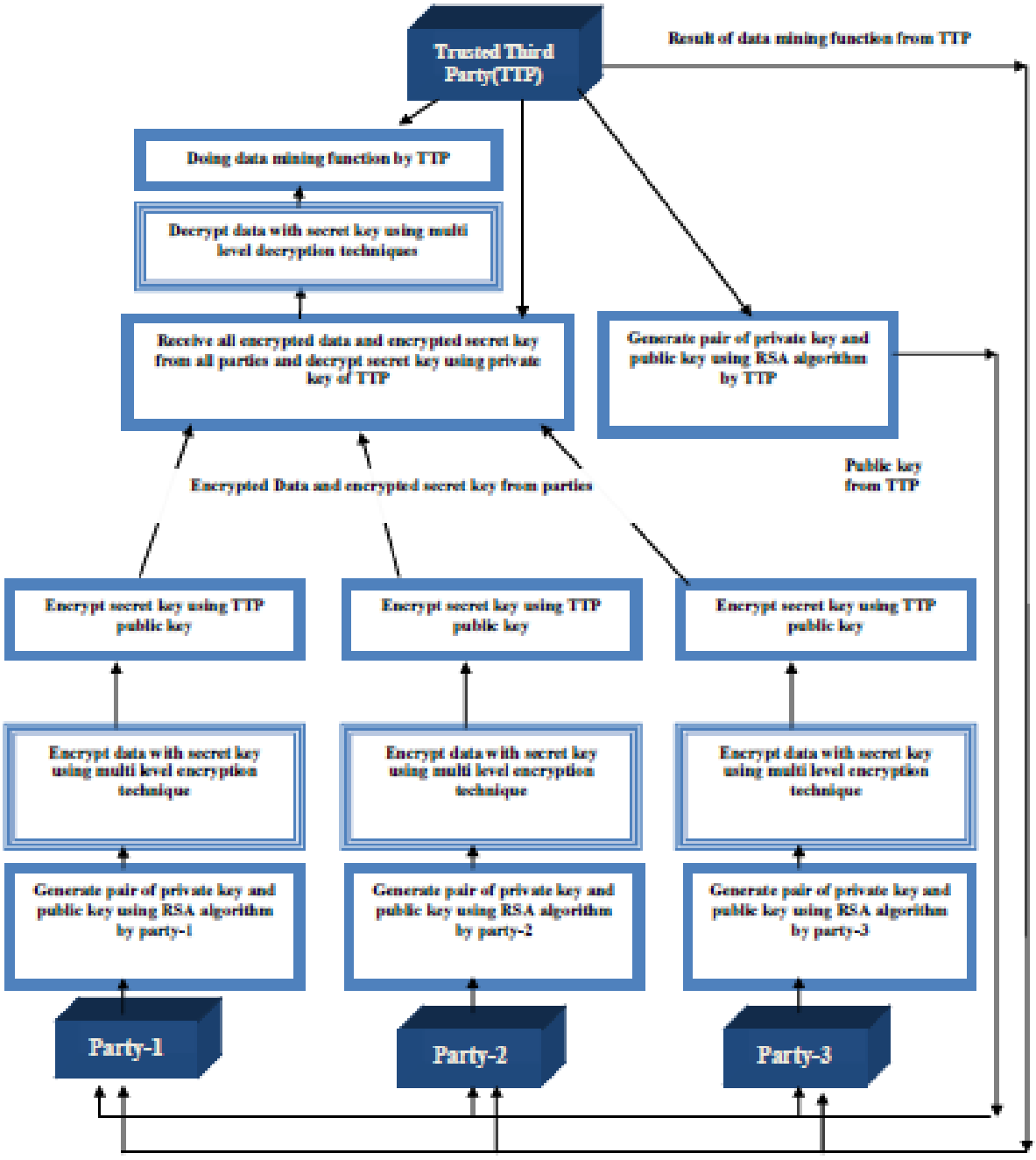

Figure 9: Privacy Preserving Data Mining (PPDM) using Multilevel Cryptographic Technique

Multilevel Encryption and Decryption (MLED) System

1. The system is developed in such a way that it supports multiple encryption algorithms; whereas the existing systems are focussed on encryption at single level.
2. A random function generator has been used to generate an n-digit random number based upon the n-number of encryption algorithms used. This generated n-digit number determines the order of selecting encryption algorithms. Since the number determining the order is completely random it is infeasible to crack the order of execution.
3. Another significant feature of this random generator is, it totally depends upon the key phrase that we provide and hence for various phrases it produces different order, which results the intruder in a worst scene.
5. Moreover the number of encryption algorithm that we use, their order of execution will always remain a secret and hence it doesn't even leave a single chance for the eavesdroppers to make a guess on our system and hence the security offered is up to the best of ever provided.
6. This proposed system is developed in order to support not only text files but also images and media files. But still many of the existing systems are developed in order to suit basic text formats.

Encryption Algorithm for MLED

1. Let *n* is the number of encryption algorithms to be used.
2. Let *F* is the inputted data file.
3. Arrange the *n* cryptographic algorithms in an order from *1* to *n* randomly.
4. Generate a key phrase *Kp* for data authentication.
5. Generate a random number of length *n*.
6. Select_Cryptography_Algorithm( ).

Select_Cryptography_Algorithm( )

1. Find the each digit *d* of the number.
2. If *d* ==0
3. then *d*=*d*+1
4. End if
5. If *d*== i , where i= 1,2, ...,*n*
6. then i^{th} cryptographic algorithm will be selected.
7. Execute_ Algorithm( ).
8. Go to Step-1.
9. End if
10. Store the resulted multi-cipher text file and send it to receiver via a communication link.

Execute_Algorithm( )

1. If *c*==0 // *c* is a counter initialized to 0, that counts the number of algorithms
   Then input the data file *F* into the i^{th} algorithm.
2. Generate the cypher text.
3. End If
4. Else
5. Input the cypher text produced by previous algorithm as input to the i^{th} algorithm.
6. Generate another multi-cypher text , which will be spplied to the next algorithm as input.
7. End Else

Decryption Algorithm for MLED

1. Receive the resulted multi-cipher text file sent via a communication link.
2. Apply the *n* cryptographic algorithms in the reverse order of their execution during the encryption process.
3. Generate the original data file.

VIII. EXPERIMENTAL SETUP

The experiments are implemented on a personal computer with an Intel Pentium IV, 2.40 GHZ CPU, 4.00 GB RAM, Microsoft Windows XP professional version 2002 operating system with JAVA. In this experiment we have considered three famous cryptographic algorithms to implement multilevel security. They are: (1) Data Encryption Standard (DES), (2) Advanced Encryption Standard (AES) and (3) Blowfish.

The input data file to our MLED system is shown in Table 4. A key phrase is then entered which is none other than a secret pass phrase or a password for data authentication. A randomizer function calculates a random number of lengths n depending on the key phrase, where n is the number of encryption algorithms that are to be used. Our system produces maximum number of combinations that can be made with the number of algorithms chosen. For example, a system which uses 3 encryption algorithms, then it generates 3! combinations namely 123,231,321,132,213,312. So in general for n bit random number it produces n! combinations. Among these possible combinations, the generated random numbers of n-bit determines the order of execution of those encryption algorithms. For example if the resulted random number is 3-1-2 for the multilevel encryption system that we used for simulation (which is specified above) then the order of execution will be Blowfish at first and then DES algorithm which is followed by AES algorithms. So the cipher text generated from the Blowfish is supplied as plaintext for DES algorithm. The resultant cipher text of DES algorithm is supplied as plaintext to AES. After executing the three algorithms in the order generated by our system, randomly we send the resulted multi-cipher text to the receiver via a communication link. At the receiver's end, the reverse of encryption order takes place. As per our case it is 2-1-3. So the cipher text is decrypted by AES algorithm which still possesses some scrambled text is supplied as plaintext for DES and it is continued for Blowfish where at last we get resulted original message that the sender wishes to communicate. The following architecture in Figure 10 shows our proposed MLED system flow design from top to down approach. The generation of cipher text from plain text is described below.

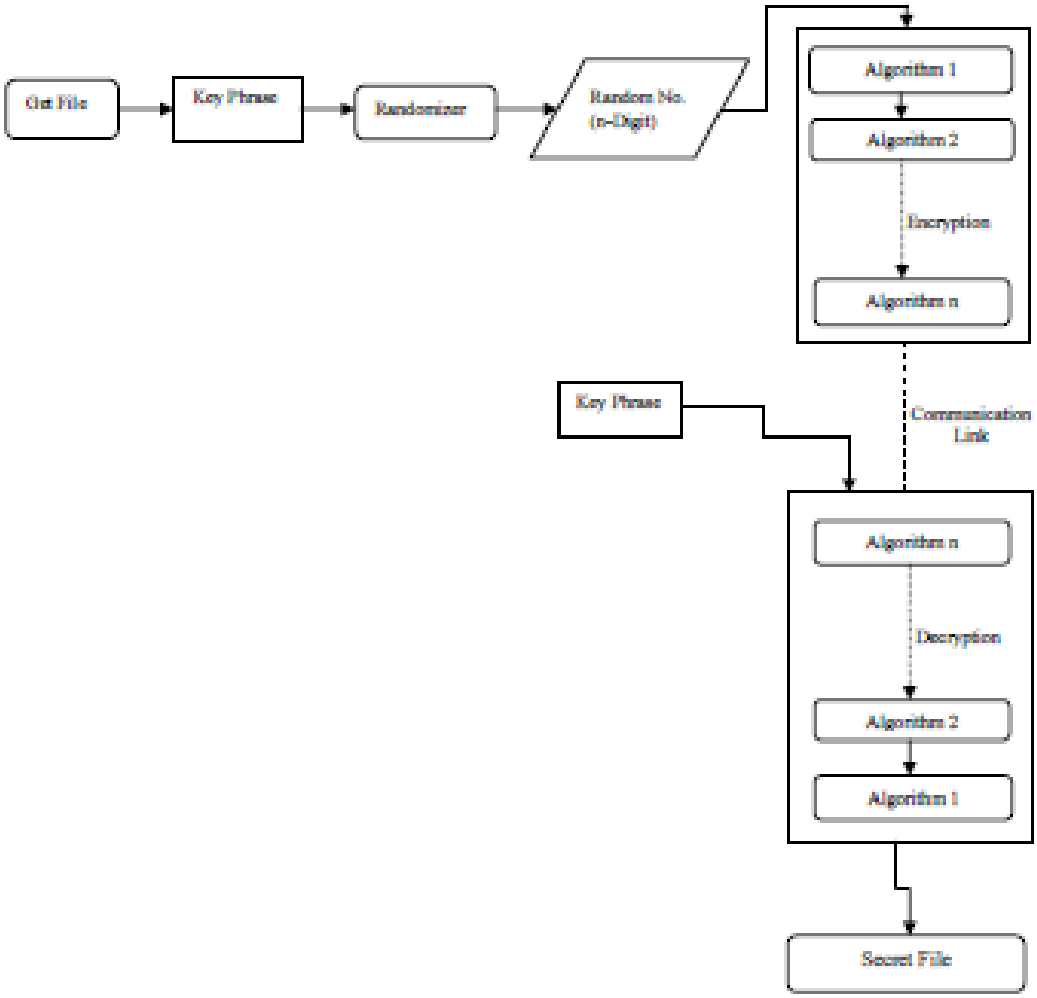

Figure 10: Flow design of Multilevel Encryption and Decryption system

Table-4 : Hospital Patient Database

Name	DOB	SSN	Sex	Zipcode	Disease
John	21/01/1976	657853925	Male	53715	Heart Disease
Robina	13/04/1986	142736281	Female	53715	Hepatitis
Sumit	28/02/1976	543926154	Male	53703	Bronchitis
Smith	21/01/1976	452915372	Male	53703	Broken Arm
Jenifer	13/04/1986	453824193	Female	53706	Flu
Kenighani	28/02/1976	748935291	Female	53706	Hang Nail

Moreover, the dataset depicted in table 4 can also be viewed as an information system. Let I, U, A, and V be an information system with U representing the universe of samples, A the set of attributes and V the values of all the attributes. Let A1, A2, and A3 be subsets of the set of attributes A, of the type identifiers, quasi-identifier, and private respectively. i.e.

A= A1 U A2 U A3,

such that A1 ∩ A2 = A2 ∩ A3 = A1 ∩ A3 = A1 ∩ A2 ∩ A3 = ϕ

Let f be a function mapping and ordered pair (x, a) ∈ UxA to an element in V.

i.e. f : UxA → V such that f (x, a) ∈ V, ∀x ∈ U and ∀a ∈ A

Let V_a, called the domain of a ∈A be the set of all values of the attribute a.

Then $V=U_{\forall a \epsilon A} V_a$.

Therefore, corresponding a partition A = {A1, A2, A3}, a partition of the domains of all the attributes may be obtained

V= {V_{A1}, V_{A2}, V_{A3}} i.e. ∀a ϵA1, f(x,a) ϵ V_{A1} or V_{A1} = $U_{\forall a \epsilon A} V_a$

Similarly, let the subsets V_{A2} and V_{A3} be the blocks of the partition of V corresponding the blocks A2 and A3.

Let |A| = k, Then the multidimensional database U is primarily an array of k-dimensional vectors corresponding to a set of attributes. A partition of I is considered to be the set of subsystems corresponding to each block of the partition of

A = {A1, A2, A3}, say I_{A1}, I_{A2}, and I_{A3}, i.e. I = < $I_{A1}|I_{A2}|I_{A3}$ >

where each I_{Ai} = (U, Ai, V_{Ai}) for iϵ{1,2,3}

8.1. Description of cryptographic algorithms

Encryption is a well known technology for protecting sensitive data. Use of the combination of Public and Private Key encryption to hide the sensitive data of users, and cipher text retrieval algorithms is given in Appendix A. These algorithms (DES, AES, Blowfish and RSA) are standard algorithms being used by different researchers.

IX. EXPERIMENTAL RESULTS AND ANALYSIS OF THE CRYPTOSYSTEMS USED IN THE WORK

We run our experiments to compare the performance of various cryptographic algorithms used in our work. Here we consider the inputs as text files with sizes range from 50 KB to 22300 KB. The encryption/decryption algorithms have been used to carry out different experiments and evaluating their performance in a fully controlled manner.

Main purpose here is to calculate the encryption and decryption speed of each algorithm for different input sizes. Their implementation is tried to optimize the maximum performance for the algorithms. The throughput for encryption as well as decryption is calculated one by one. Encryption time is used to calculate the throughput of an encryption scheme. The throughput of the encryption scheme is calculated by dividing the total plaintext in KB by total encryption time in Second for each algorithm shown in Table 5. If the throughput value is increased, the power consumption of this encryption technique is decreased. Similar procedure has been followed to calculate the throughput of decryption scheme.

$$\text{Encryption throughput (KB/Sec.)} = \frac{\sum \text{input files size}}{\sum \text{Encryption Computation Time}}$$

$$\text{Decryption throughput (KB/Sec.)} = \frac{\sum \text{input files size}}{\sum \text{Decryption Computation Time}}$$

The performance metrics are analyzed by the following

(a) Encryption/decryption time.

(b) CPU processing time – in the form of throughput.

Table 5: Execution time of encryption/decryption process for all algorithms

Input size(KB)	DES		AES		Blowfish		RSA		MLED	
	ENC	DEC	ENC	DEC	ENC	DEC	ENC	DEC	ENC	DEC
50	31	51	56	64	38	38	55	55	125	153
108	35	40	40	57	45	29	46	48	120	133
246	46	50	110	75	43	64	89	73	199	210
320	80	73	162	147	44	90	119	105	286	310
695	86	88	165	144	47	91	157	157	298	356
781	145	131	212	152	66	96	179	169	423	369
900	241	250	260	172	66	103	269	173	567	427
5500	248	240	258	170	118	100	541	382	624	436
7311	1692	1690	1365	880	105	139	961	880	3162	2960
22300	1716	1716	1366	883	152	137	1441	961	3234	3100
Average Time(Sec)	432	432.9	399.4	274	72	88.7	385.7	300	323.4	845.4
Throughput(KB/Sec)	8.64	12.6	9.35	13.6	51	42.11	9.9	12.72	4.22	4.51

The performance of the multilevel encryption and decryption system (MLED) is analyzed against DES, AES, Blowfish and RSA cryptosystem to provide the security. Moreover, CPU processing time i.e. the throughput of our proposed work is also compared with existing methodologies shown in the Table 5. From the above table it is seen that the CPU processing time i.e. the throughput is comparatively less in case of MLED in both encryption and decryption.

The encryption computation time (ECT) is the time which taken by the algorithms to produce the cipher text from the plain text. The encryption time can be used to calculate the Encryption Throughput of the algorithms.

The decryption computation time (DCT) is the time taken by the algorithms to produce the plain text from the cipher text. The decryption time can be used to calculate the Decryption Throughput of the algorithms. The encryption computation time and decryption computation comparison is shown in the Figure 15 and Figure 16.

The encryption and decryption throughput of MLED is compared with the encryption and decryption throughput of DES, AES, Blowfish and RSA. This is shown in Figure 17 and Table 6.

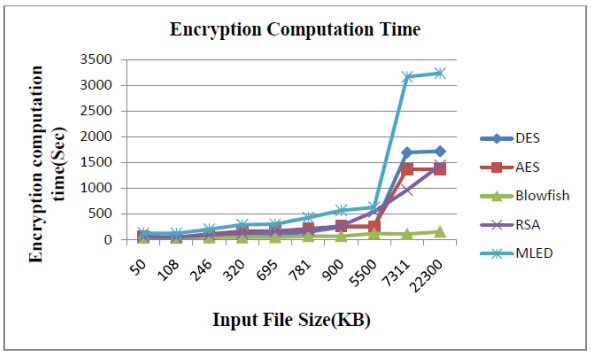

Figure 15: Encryption Computation Time

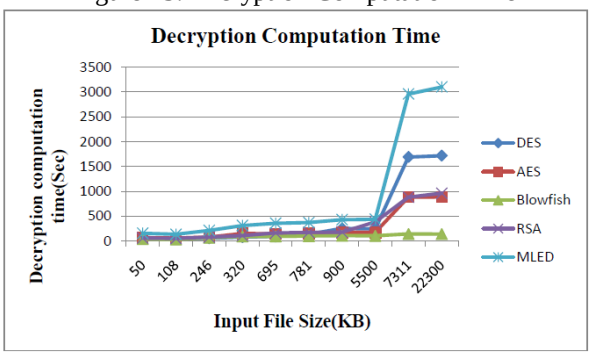

Figure 16: Decryption Computation Time

From both figures 15 and 16 it is seen that ECT and DCT both increase slowly with the increase of input file size (up to 5500 KB). When input file size gets increased from 5500KB to more ECT and DCT suddenly rise more comparing to all other algorithms which satisfy the condition that MLED algorithm uses all above algorithms together. In other words MLED takes more time to compute ECT and DCT. However from table 6 and figure 17 it is observed that encryption & decryption throughput is less with respect to all other encryption algorithms.

Table 6: Encryption & Decryption Throughput

Algorithm	Encryption Throughput (KB/sec)	Decryption Throughput(KB/ sec)
DES	8.64	12.6
AES	9.35	13.6
Blowfish	51.0	42.11
RSA	9.9	12.72
MLED	4.22	6.93

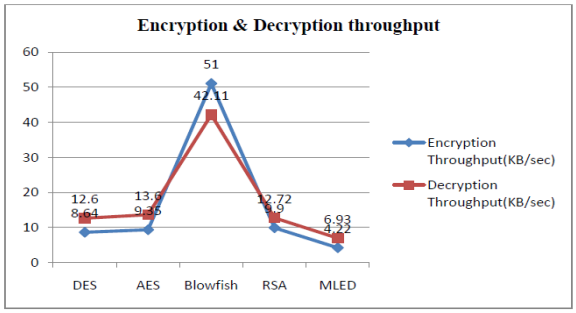

Figure 17: Comparison of Encryption & Decryption throughput

9.1. Privacy preservation Analysis

When most of the operations are jointly performed there is a need of more secured protocols which can maintain privacy and assure correctness. In this chapter we have defined a secured protocol framework for computation in section-7 and proposed a multilevel encryption to be performed before sending inputs for computation. The schematic diagram for trusted third party computation is shown in Figure 18 for better understanding of the concept.

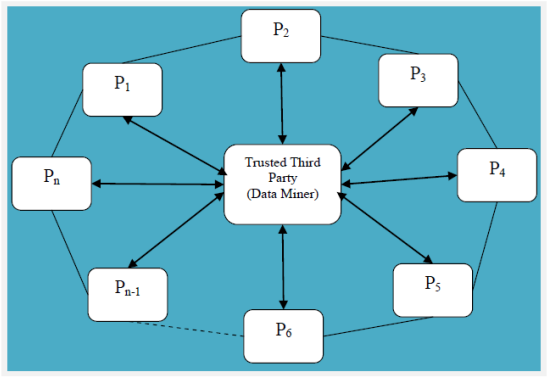

Figure 18: Schematic diagram of TTP in distributed environment

Every party sends their information through multi level encryption and decryption technique to TTP for computation as per the direction. Even though all parties are linked each other through www, no party can able to know about the information of other parties during communication to TTP. For secured data communication protocol uses RSA algorithm for sharing keys between parties and TTP. This ensures that parties send their data in encrypted form to TTP in order to maintain privacy and security of inputs. Due to sensitivity of data, no party will share it's information with other parties involved in computation. Parties provide their inputs to TTP for computation. The parties learn nothing but the results. The miner (TTP) learns global data mining results but nothing more than local databases in secure computation.

In practice, participants in a privacy-preserving protocol might behave maliciously in order to gain maximum benefits from others. But under no circumstances the system discloses the individual information. Hence privacy is maintained during data communication to TTP along with security. So we have focused on the design of an efficient mining protocol by using MLED that remains secure and private even if some of the parties behave maliciously.

9.2. Cost analysis

The performance is analysed and measured in terms of the communication cost and computation cost.

Consider the total number of parties participating in the computation is n and the total number of non-class

attributes in the dataset is *m*. The size of each non-class attribute is *d* and the domain size of class label is *p*. Also, the set of privacy-sensitive attributes is $S \subseteq A$, where *A* is attribute set of dataset $(A_1,A_2,...,A_m)$; A_i is the i^{th} attribute of *A* for $1 \leq i \leq m$, and $\{a_i^{(1)}, \ldots, a_i^{(d)}\}$ is the value domain of the i^{th} attribute. Assume there are *s* sensitive attributes, where $s = |S|$. It can be deduced that the computational overhead of each party is *dps* encryptions. In data mining applications, we usually have $n > dps$.

The communication cost of the protocol can be measured by means of calculating the total bits of information exchanged among all sites during the execution of the algorithm. If *k* is the number of rounds, in TTP the communication overhead is noted as $O(k.dps.n)N$, where N denotes the number of bits transferred.

The computation cost is measured by means of counting the total number of encryption and decryption operations executed in the algorithm. In this case, the computation overhead is $O(k.dps.n)$ and the space cost is $3nN$, that is the amount of storage consumed by parameters of the algorithm.

The dataset we used is a real dataset "Hospital Patient Database". The dataset consists of 80 rows and 6 columns (sample shown in table 4 and details given in Appendix II). We have performed two tests with the datasets. The performance is recorded and measured in the case of 5, 10, 15, 20 and 25 participating sites respectively for communication cost and 3, 5, 7, 9 and 11 participating sites respectively for the computation cost. The first test is conducted to see how much communication overhead of TTP-based protocol and other protocols being used for communication. The total amounts of transmissions caused by the protocols with respect to the number of parties are depicted in Table 7 and Figure 19. The second test that we have performed is to analyze and compare the computational overheads brought by TTP-based protocol and other protocols. Execution times of the protocols with respect to the number of parties are shown in Table 8 and Figure 20.

Table-7: Communication overhead

No. of participating Sites	Bytes transferred per round		
	TTP	STTP	SSMC
5	250	300	600
10	500	650	1250
15	750	1000	1900
20	1000	1300	2600
25	1200	1600	3200

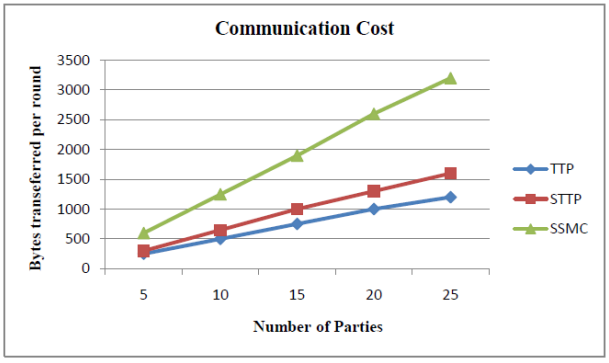

Figure 19: Communication overhead analysis

Table-8: Computation overhead

No. of Sites	Execution Time in ms		
	TTP	STTP	SSMC
3	200	300	600
5	300	500	900
7	400	800	1200
9	500	1100	1550
11	700	1400	2000

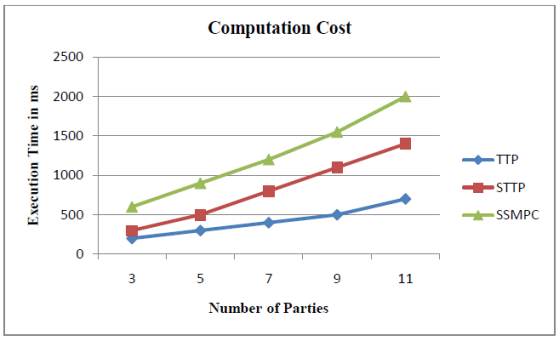

Figure 20: Computation overhead analysis

Figure 19, 20 and table 7, 8 show the communication and computation cost of TTP along with other protocols. It is observed from both figures that communication and computation cost is less with comparison to semi trusted third party and specific secure multiparty computation protocols. In both cases when number of parties is increased, communication cost and computation cost get increased slowly for all protocols, but TTP shows less cost. Hence TTP shows better performance in increase of number of participating parties in data mining activities.

As MLED involves with multiple encryption and decryption algorithms, complexity would be more than single algorithm. However, complexity will vary based on number of keys and length of the key. Hence security and privacy is more in our proposed algorithm.

X. CONCLUSION AND FUTURE WORK

A detailed discussion of strategies has been presented for attainable solutions. PPDDM as a new field of study is combining knowledge and approaches for data mining

and cryptography work. PPDDM offers a general application view but at the same time it also brings many problems to be answered. This is very helpful because of real life issues that are dealing with privacy with respect to information more often than before. In this chapter a study of the broad areas of Secure Communication Pattern (SCP) and the underlying scopes in distributed data mining process with respect to privacy preserving is presented. The fundamental techniques for the SCP have been discussed. These approaches are proposed on Specific Secure Multi-party Computation model and Semi-Trusted Third Party model. Therefore, providing the possibility of reasonable application from SCP methodologies in PPDDM based on needs is of influential findings of this chapter. In addition, the Trusted Third Party (TTP) computing model has also been discussed for secure communication pattern using multi level cryptography technique. The use of MLED and RSA algorithms pre-processes the data required for privacy preservation has been proposed and discussed in the work. The data are encrypted before communicating to the trusted third party who applies data mining functions for further processing. The privacy analysis, computation & communication cost analysis in the trusted third party model have been discussed and compared with other models. During data communication to trusted third party (Data miner) each party's data are encrypted multiply which maintains the privacy and security more with comparison to STTP and SSMPC. From observations it is seen that data handling through TTP performs well. Finally it can be said that this work proposes a novel approach to secure sensitive and private information with multiple levels of privacy for data mining in a distributed environment.

We have considered only balanced data set i.e., same data size. We have not verified using imbalanced data set. Hence imbalanced data set can be taken into consideration further. Moreover, this approach can also be extended for various real time data sets for generalizing the concept.

REFERENCES

[1] V. Torra, "Privacy in Data Mining", Data Mining and Knowledge Discovery Handbook, 687-716,2010

[2] M. Byrd and C. Franke , "The State of Distributed Data Mining", ECS265 Project Report, UC Davis,Davis CA., USA, 2007

[3] J. Lee Brickell, "Privacy-Preserving Computation for Data Mining", Ph.D. Thesis, The University of Texas at Austin, may 2009

[4] J. Domingo-Ferrer and V. Torra, "Privacy in Data Mining", Data Mining and Knowledge Discovery,11(2): 117-119, 2005

[5] O. V´yborn´y, "Time, Data Mining and Security", Ph.D. Thesis Proposal, Faculty of Informatics,Masaryk University, Sep. 10, 2006

[6] X. Wu, Ch. H. Chu, Y. Wang, F. Liu and D. Yue, "Privacy Preserving Data Mining Research: Current Status and Key Issues", Springer-Verlag Berlin Heidelberg, ICCS 2007, Part III, LNCS 4489, pp. 762–772, 2007

[7] J. Wang ,Y. Luo ,Y. Zhao and J. Le, "A Survey on Privacy Preserving Data Mining", First International Workshop on Database Technology and Applications, 2009

[8] Y. Lindell and B. Pinkas, "Secure Multiparty Computation for Privacy-Preserving Data Mining",Journal of Privacy and Confidentiality: Vol. 1: Iss. 1, Article 5, 2009

[9] H. Kargupta, P. Chan, "Advances in Distributed and Parallel Knowledge Discovery", MIT, AAAI Press, Cambridge, New York, 2000

[10] N. Zhang, S. Wang, W. Zhao, "A new scheme on privacy-preserving data classification", in:Proceedings of KDD'05, pp. 374–383, 2005

[11] X. Yi and Y. Zhang, "Privacy-preserving naïve Bayes classification on distributed data via semitrusted mixers", Information Systems, Volume 34, Issue 3, pp. 371-380, May 2009

[12] Zh. XU, "Analysis of Privacy Preserving Distributed Data Mining Protocols", Ph.D. Thesis, Victoria University, 2011

[13] D. K. Mishra, P. Trivedi and S. Shukla, "A Glance at Secure Multiparty Computation for Privacy Preserving Data Mining", International Journal on Computer Science and Engineering Vol.1(3), 171-175, 2009

[14] W. Du and M. J. Atallah, "Secure Multi-party Computations and Privacy Preservation: Results and Open Problems", In Proceedings of New Security Paradigms Workshop, pages 11-20, Cloudcroft, New Mexico, USA, September 11-13, 2001

[15] X. Ge and J. Zhu, "Privacy Preserving Data Mining", New Fundamental Technologies in Data Mining, ISBN 978-953-307-547-1, Published: January 21, under CC BY-NC-SA 3.0 license, Chapter 29,DOI: 10.5772/13364, 2011

[16] P. Wang, "Survey on Privacy Preserving Data Mining", International Journal of Digital Content Technology and its Applications, Volume 4, Number 9, December 2010

[17] P. Kamakshi and A. Vinaya Babu, "Preserving Privacy and Sharing the Data in Distributed Environment using Cryptographic Technique on Perturbed data", Jornal of Computing, Volume 2, Issue 4, ISSN 2151-9617, April 2010

[18] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin and M. Y. Zhu, "Tools For Privacy Preserving

Distributed Data Mining", SIGKDD Explorations, 4(2):28-34, 2002

[19] S. C. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance", IEEE Transactions on Information Theory, IT-24:106–110, 1978

[20] Ch. C. Aggarwal and Ph. S. Yu , "Privacy-Preserving Data Mining: Models and Algorithms", Advances in Database Systems, Vol. 34, ISBN 978-0-387-70992-5, 2008

[21] S. J. Rizvi and J. R. Haritsa, "Maintaining Data Privacy in Association Rule Mining", In Proc. of the 28th International Conference on Very Large Data Bases (VLDB'02), Hong Kong, China, 2002.

[22] Z. Yang, S. Zhong and R. Wright, "Privacy-preserving Classification of Customer Data without Loss of Accuracy", In: Proceedings of the Fifth SIAM International Conference on Data Mining, pp.92-102, Newport Beach, CA, April 21-23, 2005

[23] A. Gurevich and E. Gudes, "Privacy preserving Data Mining Algorithms without the use of Secure Computation or Perturbation", In: 10 th International Database Engineering and Applications Symposium (IDEAS'06), pp. 121-128, 2006

[24] X. Yi and Y. Zhang, "Privacy-Preserving Distributed Association Rule Mining via Semi-Trusted Mixer", In: Data and Knowledge Engineering, vol. 63, no. 2, pp. 550-567, 2007

[25] Md. G. Kaosar and X. Yi, "Semi-Trusted Mixer Based Privacy Preserving Distributed Data Mining for Resource Constrained Devices", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, No. 1, April 2010

[26] W. Du and Z. Zhan, "Building Decision Tree Classifier on Private Data", In: Proc. Of the IEEE ICDM Workshop on Privacy, Security and Data Mining (PSDM'02), Maebashi City, Japan, 1-8, Dec. 2002

[27] D. Wenliang and Z. Zhijun, "A Practical Approach to Solve Secure Multiparty Computation Problems", In: Pro. of the Int. Conf. on Computer Networks and Mobile Computing (ICCNMC'03),2003

[28] M. Hussein, A. El-Sisi and N. Ismail, "Fast Cryptographic Privacy Preserving Association Rules Mining on Distributed Homogenous Data Base", KES 2008, Part II, LNCS / LNAI 5178, pp. 607-616,(DOI: 10.1007/978-3-540-85565-1-75), 2008

[29] M. Kantarcıoglu and C. Clifton, "Privately Computing a Distributed k-nn Classifier", In Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, LNAI 3202, pp. 279–290, 2004

[30] D. Beaver, "Commodity-based cryptography", In Proceedings of the 29th Annual ACM Symposium on Theory of Computing, El Paso, TX USA, May 4-6 1997

[31] D. Beaver, "Server-assisted cryptography", In Proceedings of the 1998 New Security Paradigms Workshop, Charlottesville, VA USA, September 22-26 1998

[32] E. Bertino, I.N. Fovino and L.P. Provenza, "A Framework for Evaluating Privacy Preserving Data Mining Algorithms", Data Mining and Knowledge Discovery, 11 (2): pp. 121-154, 2005

[33] P.Kamakshi , A.Vinaya Babu; "Preserving Privacy and Sharing the Data in Distributed Environment using Cryptographic Technique on Perturbed data"; Journal of Computing, Volume 2, Issue 4, April 2010, Issn 2151-9617

[34] N V Muthu Lakshmi, K Sandhya Rani; "Privacy Preserving Association Rule Mining Without Trusted Party For Horizontally Partitioned Databases"; International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.2, No.2, March 2012

[35] Lindell Y., Pinkas B."Privacy preserving Data Mining" CRYPTO 2000.

[36] Yu.H.,Vaidya J.,Jiang X."Privacy preserving SVM Classification on vertically partitioned data" PAKDD conference, 2006.

[37] Ming-Syan Chen, Jiawei Han,Yu, P.S. (1996), Data mining: an overview from a database perspective, IEEE Transactions on Knowledge and Data Engineering, Vol. 8 No. 6, pp 866 – 883.

[38] A.C Yao(1986), How to generate and exchange secrets, In proceedings of the 27th IEEE Symposium on Foundations of Computer Science, pp 162-167.

[39] Y Lindell and B pinkas (2000), Privacy preserving data mining, In Proc. O CRYPTO'00, pp3654. Springer-Verlag 2000.

[40] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, and Michael Y. Zhu(2003), Tools for privacy preserving distributed data mining, SIGKDD Explorations, Vol. 4, No. 2 pp 1-7.

[41] M. Kantarcioglu and C. Clifto (2004). Privacy-preserving distributed mining of association rules on horizontally partitioned data. In IEEE Transactions on Knowledge and Data

Engineering Journal, volume 16(9), pp. 1026-1037.

[42] Verykios, V.S., Bertino, E., Nai Fovino, I., Parasiliti, L., Saygin, Y., and Theodoridis, Y. (2004), State-of- the-art in privacy preserving data mining, SIGMOD Record, 33(1):50–57.

[43] Elisa Bertino , Igor Nai Fovino Loredana Parasiliti Provenza (2005), A Framework for Evaluating Privacy Preserving Data Mining Algorithms, Data Mining and Knowledge Discovery, Vol. 11, 121–154.

[44] Chin-Chen Chang, Jieh-Shan Yeh, and Yu-Chiang Li (2006), Privacy-Preserving Mining of Association Rules on DistributedDatabases, IJCSNS International Journal of Computer Science and Network Security, Vol.6 No.11.

[45] Alex Gurevich, Ehud Gudes (2006), Privacy preserving data mining algorithms without the use of secure computation or perturbation, 10th international database Engineering and Applications Symposium IDEAS06 IEEE.

[46] Mahmoud Hussein, Ashraf El-Sisi,and Nabil Ismail (2008), Fast Cryptographic Privacy Preserving Association Rules Mining on Distributed Homogenous Data Base, I. Lovrek, R.J. Howlett, and L.C. Jain (Eds.): KES 2008,Part II,LNAI 5178, pp. 607–616, 2008.© SpringerVerlag Berlin Heidelberg.

[47] Jian Wang, Yongcheng Luo, Yan Zhao, Jiajin Le(2009), A Survey on Privacy Preserving Data Mining, First International Workshop on Database Technology and Applications, pp. 111-114.

[48] Fuad Al-Yarimi, Sonajharia Minz; "Multilevel Privacy Preserving in Distributed Environment using Cryptographic Technique" ; Proceedings of the World Congress on Engineering 2012 Vol I, WCE 2012, July 4 - 6, 2012, London, U.K. ISBN: 978-988-19251-3-8; ISSN: 2078-0958

Apendix A

8.1.1. Data Encryption Standard (DES)

DES (Data Encryption Standard) algorithm purpose is to provide a standard method for protecting sensitive commercial and unclassified data. In this same key used for encryption and decryption process .

DES algorithm consists of the following steps

1. DES accepts an input of 64-bit long plaintext and 56-bitkey (8 bits of parity) and produce output of 64 bit block.
2. The plaintext block has to shift the bits around.
3. The 8 parity bits are removed from the key by subjecting the key to its Key Permutation.
4. The plaintext and key will processed by following

(i) The key is split into two 28 halves

(ii) Each half of the key is shifted (rotated) by one or two bits, depending on the round.

(iii) The halves are recombined and subject to a compression permutation to reduce the key from 56 bits to 48 bits. This compressed keys used to encrypt this round's plaintext block.

(iv) The rotated key halves from step 2 are used in next round.

(v) The data block is split into two 32-bit halves.

(vi) One half is subject to an expansion permutation to increase its size to 48 bits.

(vii) Output of step 6 is exclusive-OR'ed with the 48-itcompressed key from step 3.

(viii) Output of step 7 is fed into an S-box, which substitutes key bits and reduces the 48- bit block back down to 32-bits.

(ix) Output of step 8 is subject to a P-box to permute the bits.

(x) The output from the P-box is exclusive-OR'ed with other half of the data block. The two data halves are swapped and become the next round's input.

8.1.2. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) algorithm not only for security but also for great speed. Both hardware and software implementation are faster still. New encryption standard recommended by NIST to replace DES. Encrypts data blocks of 128 bits in 10, 12 and 14 round depending on key size. It can be implemented on various platforms specially in small devices. It is carefully tested for many security applications.

Algorithm Steps

These steps used to encrypt 128-bit block

1. The set of round keys from the cipher key.
2. Initialize state array and add the initial round key to the starting state array.

3. Perform round = 1 to 9 : Execute Usual Round.
4. Execute Final Round.
5. Corresponding cipher text chunk output of Final Round Step

Usual Round : Execute the following operations which are described above.

1. Sub Bytes
2. Shift Rows
3. Mix Columns
4. Add Round Key , using K(round)

Final Round: Execute the following operations which are described above.

1. Sub Bytes
2. Shift Rows
3. Add Round Key, using K(10)

Encryption : Each round consists of the following four steps:

a) Sub Bytes : The first transformation, Sub Bytes, is used at the encryption site. To substitute a byte, we interpret the byte as two hexadecimal digits.

b) Shift Rows : In the encryption, the transformation is called Shift Rows.

c) MixColumns : The Mix Columns transformation operates at the column level; it transforms each column of the state to a new column.

d) Add Round Key : Add Round Key proceeds one column at a time. Add Round Key adds a round key word with each state column matrix; the operation in Add Round Key is matrix addition.

The last step consists of XORing the output of the previous three steps with four words from the key schedule. And the last round for encryption does not involve the "Mix columns" step.

Decryption: Decryption involves reversing all the steps taken in encryption using inverse functions like a) Inverse shift rows, b) Inverse substitute bytes, c) Add round key, and d) Inverse mix columns.

The third step consists of XORing the output of the previous two steps with four words from the key schedule. And the last round for decryption does not involve the "Inverse mix columns" step.

8.1.3. Rivest-Shamir-Adleman (RSA)

RSA is widely used Public-Key algorithm. RSA firstly described in 1977. In our proposed work, we are using RSA algorithm to encrypt the data to provide security so that only the concerned user can access it.

RSA algorithm involves these steps:

1. Key Generation
2. Encryption
3. Decryption

Key Generation

Before the data is encrypted, Key generation should be done.

Steps:

1. Generate a public/private key pair :
2. Generate two large distinct primes p and q
3. Compute $n = pq$ and $\varphi = (p-1)(q-1)$
4. Select an e , $1 < e < \varphi$, relatively prime to φ.
5. Compute the unique integer d , $1 < d < \varphi$ where $ed \equiv_{\varphi} 1$.
6. Return public key (n, e) and private key d

Encryption

Encryption is the process of converting original plain text (data) into cipher text (data).

Encryption with key (n , e)

1. Represent the message as an integer $m \in \{0, \ldots, n-1\}$
2. Compute $c = m^e \bmod n$

Decryption

Decryption is the process of converting the cipher text (data) to the original plain text(data).

Decryption with key d : compute $m = c^d \bmod n$.

8.1.4. Blowfish Encryption Algorithm:

Blowfish was designed in 1993 by Bruce Scheier as a fast, alternative to existing encryption algorithms such AES, DES and 3 DES etc.

Blowfish is a symmetric block encryption algorithm designed in consideration with,

- Fast **:** It encrypts data on large 32-bit microprocessors at a rate of 26 clock cycles per byte.
- Compact**:** It can run in less than 5K of memory.
- Simple: It uses addition, XOR, lookup table with 32-bit operands.
- Secure: The key length is variable ,it can be in the range of 32- 448 bits: default 128 bits key length.
- It is suitable for applications where the key does not change often, like communication link or an automatic file encryptor.
- Unpatented and royality-free.

Description of Algorithm:

Blowfish symmetric block cipher algorithm encrypts block data of 64-bits at a time.it will follows the feistel network and this algorithm is divided into two parts.

1. Key-expansion
2. Data Encryption

Key-expansion:

It will converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes. Blowfish uses large number of subkeys. These keys are generate earlier to any data encryption or decryption.

The p-array consists of 18, 32-bit subkeys:

P1,P2,……..,P18

Four 32-bit S-Boxes consists of 256 entries each:

S1,0, S1,1,……. S1,255

S2,0, S2,1,…….. S2,255

S3,0, S3,1,…….. S3,255

S4,0, S4,1,.....S4,255

Generating the Subkeys:

The subkeys are calculated using the Blowfish algorithm

1. Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the hexadecimal digits of pi (less the initial 3): P1 = 0x243f6a88, P2 = 0x85a308d3, P3 = 0x13198a2e, P4 = 0x03707344, etc.
2. XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P14). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits. (For every short key, there is at least one equivalent longer key; for example, if A is a 64-bit key, then AA, AAA, etc., are equivalent keys.)
3. Encrypt the all-zero string with the Blowfish algorithm, using the subkeys described in steps (1) and (2).
4. Replace P1 and P2 with the output of step (3).
5. Encrypt the output of step (3) using the Blowfish algorithm with the modified subkeys.
6. Replace P3 and P4 with the output of step (5).
7. Continue the process, replacing all entries of the P array, and then all four S-boxes in order, with the output of the continuously changing Blowfish algorithm.

In total, 521 iterations are required to generate all required subkeys. Applications can store the subkeys rather than execute this derivation process multiple times.

Data Encryption:

It is having a function to iterate 16 times of network. Each round consists of key-dependent permutation and a key and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookup tables for each round.

Blowfish Encryption Algorithm
Divide x into two 32-bit halves: xL, xR
For i = 1to 16
xL = XL XOR Pi
xR = F(XL) XOR xR
Swap XL and xR
Swap XL and xR (Undo the last swap.)
xR = xR XOR P17
xL = xL XOR P18
Recombine xL and xR