Analysis of Privacy Preserving Distributed Data Mining Protocols

By

ZHUOJIA XU

A thesis submitted in fulfilment of the requirements for the degree of

MASTER BY RESEARCH

School of Engineering and Science,

Faculty of Health, Engineering and Science,

VICTORIA UNIVERSITY

2011

Abstract

This thesis studies the features and performance of privacy-preserving distributed data mining protocols published as journal articles and conference proceedings from 1999 to 2009. It examines the topics and settings of various privacy-preserving distributed data mining protocols as well as the performance metrics for evaluation of these protocols.

The framework for analysis of thesis draws on systematic data collection, document encoding, content analysis, protocol classification, criteria identification and performance comparison of privacy-preserving protocols for distributed data mining applications.

We studied and revealed an elaborate taxonomy for classifying privacy-preserving distributed data mining algorithms. Such a classification scheme is built on several dimensions, including secure communication model, data distribution model, data mining algorithms and privacy-preservation techniques. Besides, we have classified these privacy-preserving distributed data mining protocols into one of the mutually exclusive categories and recorded the frequency of protocols in each category as well. Based on this classification scheme, we have characterized each privacy-preserving distributed data mining algorithm according to its feature of dimensions. Therefore, we can compare the performance of protocols in similar or same categories in terms of an array of metrics, namely communication cost, computation cost, communication rounds and scalability. Relative performance of different protocols is also presented.

This thesis, thus, aims to provide a framework for classifying privacy-preserving distributed data mining protocols and compare the performance of different protocols based on the outcome of the classification scheme.

Declaration

"I, Zhuojia Xu, declare that the Master by Research thesis entitled Analysis of Privacy-preserving Distributed Data Mining Protocols is no more than 60,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work".

Signature Date

Acknowledgement

I would like to thank my principal supervisor, Associate Professor Xun Yi and my co-supervisor, Professor Yanchun Zhang, for their inspiration and support throughout the entire project period. They have been of great help and given me many constructive comments and ideas.

I would also like to thank my parents for giving their encouragement and support to me during this master study.

Finally, I want to express my gratitude to my workmates in the Applied Informatics Centre for the discussion and contribution of ideas during my research study. Thank you, MD Kaosar, Xuebing Yang, Mike Ma, Guandong Xu and Yanan Hao for the support and cooperation.

Basic definitions

Algorithm – a finite sequence of instructions, an explicit, step-by-step procedure for solving a problem.

A priori – a classic algorithm for learning association rules on transaction databases.

Boolean vector – a vector with its only possible values being 0 and 1.

Broadcast – refers to transmitting a packet that will be received by every device on the network.

Class label – in classification, the input fields to be classified. It's also referred to as *target field*.

Cryptosystem – a computer system involving cryptography.

Data holder – a user participating in the computation and hold data sets as input.

Decision tree – a predictive model mapping from observations about an item to its conclusion about the target value.

Discrete logarithm – group-theoretic analogous to ordinary logarithm.

Distributed computing – loosely or tightly coupled programs or concurrent process running on multiple processing elements or storage elements.

ElGamal Cryptosystem - is an asymmetric key encryption algorithm for public-key cryptography.

Factorization – decomposition of an object into a product of other objects, called factors.

Frequent itemset – is an itemset whose support is greater than some user-specified minimum support.

Naïve Bayes classifier – a term dealing with a simple probability classifier based on applying Bayes' theorem with strong (naïve) independence assumption.

Polynomial – a finite length of expressions constructed from variables and constants using algebraic operations (addition, subtraction and multiplication)

Protocol – a set of rules for computers to communicate with each other across a network. It is a convention or standard that controls connection, communication and data transfer between computing endpoints.

Random number – a number or sequence exhibiting statistical randomness.

Scalar product – also referred to as dot product. Is an operation of two vectors on real numbers and returns a real-valued scalar quantity.

Scalability – a desirable property of a system, a network or a process, which indicates the ability to either handle growing amounts of work in good manner or to be readily enlarged.

Trusted Third Party (TTP) - is an entity which facilitates interactions between two parties who both trust the third party; they use this trust to secure their own interactions.

List of Figures

List of Tables

List of Algorithms

Contents

Chapter 1 Introduction

1.1 Overview

Nowadays, data management applications have evolved from pure storage and retrieval of information to finding interesting patterns and associations from large amounts of data. With the advancement of Internet and networking technologies, more and more computing applications, including data mining programs, are required to be conducted among multiple data sources that scattered around different spots, and to jointly conduct the computation to reach a common result. However, due to legal constraints and competition edges, privacy issues arise in the area of distributed data mining, thus leading to the interests from research community of both data mining and information security.

This kind of requirements give birth to the field of Privacy-preserving Distributed Data Mining (PPDDM), which aims at solving the following problem: a number of participants want to jointly conduct a data mining task based on the private data sets held by each of the participants. After the task, each participant knows nothing more than their local inputs and the global result of the data mining algorithm.

Three major phases [32] - *Conceptive Stage*, *Deployment Stage* and *Prospective Stage* have been land-marked regarding the progress of this new research area -. During each of these phases, researchers of PPDDM concentrated their efforts on different aspects of this area.

Conceptive Stage starts off with the identification of conflicts between knowledge discovery and privacy concerns as is proposed by O'Leary [65] [66], Fayyad, Piatetsky-Shapiro and Smith [64] [67]. From then on, a debate over how to balance between privacy concerns and accurate results of data mining applications has arisen.

Deployment Stage characterizes the period when a large number of PPDDM algorithms have been developed in various refereed sources, such as [22][40][43]. These algorithms address different privacy related issues in distributed data mining context.

Prospective Stage is a forthcoming period, which is expected to be a popular research area. Researchers in [3][32] have focused more effort on the privacy principles, policies, requirements and standards in order to establish a common framework to evaluate and standardize PPDDM.

One of the major challenges of PPDDM area is to devise a framework of synthesizing and evaluating various protocols and algorithm developed so far, because researchers, developers and practitioners interested in this topic have been confused, to some extent, by the excessive number of techniques developed so far.

In this thesis, our main objective is to put forward a framework to synthesize and characterize currently existing Privacy-preserving Distributed Data Mining (PPDDM) protocols and to provide a standard and systematic approach of understanding PPDDM-related problems, analysing PPDDM requirements, designing effective and efficient PPDDM protocols and undertaking studies on continuous performance improvement. The PPDDM protocols we present and analyze here in this thesis are all assumed to preserve privacy absolutely under certain privacy definition and produce completely accurate data mining outcomes. Our contributions in this thesis can be summarized as follows: 1) we devise a framework to analyse and synthesize currently existing literatures and classify them into logical categories; 2) we illustrate the nature of the problem by characterizing scenarios in PPDDM; 3) we analyse and compare different PPDDM protocols that address the same problem setting to evaluate their

relative performance; 4) we initialize a new potential research area within PPDDM that can contribute to standardization of PPDDM.

The efforts made in this thesis are by no means exhaustive and comprehensive. However, we primarily aim at providing directions to narrow the gap between technical solutions and business requirements of PPDDM algorithms. PPDDM protocols can be characterized as a variety of scenarios or problems, which have distinct requirements and constraints for performance. The performance of PPDDM protocols are generally determined by three parameters: effectiveness, efficiency and privacy degree. Currently, various PPDDM protocols devised with different techniques present different performance indicators on these parameters. Some algorithms are more efficient and secure, but the joint data mining results are greatly affected. Others produce accurate outcomes and absolute security, but the computational or communicational complexity is too high to be accepted. Accordingly, a decision-making strategy is needed to analyse and determine how to work out PPDDM solutions fit for the proposed business problems. Research communities have already set about standardizing such a strategy. This thesis focuses on bringing about a classification framework and evaluation methodology to assist informed and intelligent selection of PPDDM algorithms. Thus, we argue that this piece of work will shed some light on standardized strategy for design and develop PPDDM protocols.

1.2 Keywords

Classification, Evaluation, Privacy, Security, Privacy-preserving, Distributed, Data Mining.

1.3 Topic of the thesis

This thesis introduces the current status of research on privacy-preserving distributed data mining protocols, highlighting the gap between the development of algorithms and the standardization of the algorithms.

Our study, therefore, focuses on the topics of what is the classification scheme of privacy-preserving distributed data mining (PPDDM) protocols, how PPDDM protocols can be classified, what are the evaluation metrics to analyse various PPDDM protocols and how to compare their performance based on a set of common scales.

1.4 Problem description

Designing and developing efficient, effective and secure privacy-preserving distributed data mining protocols is considered one of the most challenging tasks when it comes to the trade-off between privacy, efficiency and complexity. There are no fixed design approaches that can be generalized to address different problem settings. Different distributed data mining applications are formulated into different problem definitions, thus different methods and techniques used for generating solutions. Under such circumstances, the invention of standard to guide effective and efficient development and testing of privacy-preserving data mining protocols has become imperative. The research community of Privacy-preserving data mining have also been aware of this situation and have actually made great progresses and achievements in this field. However, the real problem is that the contributions made so far have aimed at dealing with either centralized scenarios or general scenarios without focus on distributed ones. No solutions have been developed to address standardization issues in the distributed scenario for privacy-preserving data mining

applications. This thesis aims at proposing a framework to satisfy the needs for standardization towards privacy-preserving data mining protocols in distributed scenarios - PPDDM protocols.

Three research questions will be explored in this thesis:

1. What are the design tools of privacy-preserving distributed data mining protocols?
2. How to characterize and classify PPDDM protocols?
3. How to evaluate and compare PPDDM protocols in terms of predefined metrics?

In accordance with the research questions, the work of this thesis can be divided into three parts:

- Provide a summary of protocols and algorithms that serve as design tools of PPDDM protocols;
- Classify current PPDDM protocols in terms of a set of dimensions, including data partitioning model, data mining algorithms, secure communication model, privacy-preserving techniques;
- Evaluate the complexity of PPDDM protocols and compare them based on unified criteria, including communication cost, communication round, computation cost. Communication cost is the total number of bytes of information exchanged among all the sites involved in the distributed data mining tasks; communication round refers to the total number of transfers of information required over the separate databases; computation cost is the total times of CPU operations needed to execute the PPDDM algorithms.

1.5 Justification and motivation

This is an emerging and promising area which has captured interests of both information security and data mining research communities. Various protocols

addressing different problem settings and data mining applications have been developed during the last decade.

However, the lack of a common framework to classify and evaluate privacy-preserving distributed data mining protocols can cause a serial of troubles. Firstly, proposed protocols may be of little value to practical situations. Secondly, new researchers and practitioners can be confused by the variety of techniques and algorithms without learning of their features. Finally, such lack of classification and evaluation work may prevent this area from future development towards standardization.

Thus, after completing this thesis, one will have a clearer understanding of PPDDM area with respect to the design methods, the classification criteria and features as well as the relative performance of different PPDDM protocols.

1.6 Selection of methods

The methods that are used to answer the research questions in this thesis are a combination of literature search and more analytical approaches to classify and evaluate various existing algorithms.

The answer to the first research question will be found in the literature review and background information. We are able to find the relevant information, summarize it and present it in a structured manner.

The second question is answered by our own approach, which is like common process of problem-identifying and problem-solving. We survey the relevant context of PPDM classification area and find a gap to be filled in PPDDM classification work. We propose our scheme to address this problem.

The answer to the third question is presented through analysis of grouped PPDDM protocols and comparison of their performance in terms of certain metrics.

Thus, the methods used in this thesis to address the research questions can be summarized as literature survey, content analysis and performance analysis.

1.7 Organization of the thesis

The rest of the thesis is organized as follows. Chapter 2 provides background materials and related work in privacy-preserving distributed data mining area. In Chapter 3, we discuss some cryptographic primitives and building blocks for PPDDM algorithms. Then we introduce classic data mining algorithms, including classification, association rule mining and clustering, in which privacy concerns are always considered with respect to scientific, business and government applications.

In Chapter 4 and Chapter 5, we go into the solutions of distributed naïve bayes classification and distributed association rules mining, analyse the complexity of different solutions and undertake comparison between them in terms of communication cost, computation cost and communication rounds.

The findings of Chapter 3 - 5 are summarized in Chapter 6, which presents the conclusions we have drawn in this thesis. Chapter 7 addresses future work to be conducted.

Chapter 2 Preliminaries

There are some cryptographic properties or features that are used to create privacy-preserving protocols. These properties include additive homomorphic encryption property and commutative encryption property. Form these rudimental properties we can get some core components that most privacy-preserving distributed data mining protocols are based on. Considering developing privacy-preserving distributed data mining algorithms, the most common challenge is how to obtain the trade-off among accurate data mining results, privacy-preservation of individual records and efficient and scalable algorithms. In order to determine a PPDDM algorithm meeting these design goals, analysis work has to be done in a systematic and comprehensive way.

Several approaches have been proposed to address the privacy protection issues in data mining applications. One method is a reconstruction-based approach which reconstructs the distribution probability of the original dataset and creates a new distribution curve. Another method is a heuristics-based approach that protects individual information by using data perturbation methods, such as blocking, generalizing, aggregating and swapping, etc. These two approaches have a major drawback when dealing with privacy-preserving data mining problems. They trade off between the privacy of the individual information and the correctness of the data mining results. That is, privacy is achieved at the cost of accurate outcome. Besides, such kind of solutions can only tackle centralized data mining applications.

Cryptographic-based approach is an effective way to resolve this accuracy-privacy trade-off. The data mining outcome is absolutely accurate and the privacy of personal information is leaked by no means under predefined security constraints. Therefore, we assume that privacy is ensured and accuracy is maintained when it comes to cryptographic solutions for distributed data mining applications.

This chapter gives a brief review of related work. It provides an overview of encryption properties that privacy-preserving key components make use of, and presents some sub-protocols that most privacy-preserving distributed data mining protocols are based on. We also describe some most common privacy-preserving techniques used to prevent disclosure of private information (e.g. encryption, secret sharing). The last section provides the evaluation criteria we will employ to measure and compare the performance of our selected protocols.

2.1 Data mining algorithms

2.1.1. Classification

Data classification deals with the process of finding the common properties among a set of objects in a database and divides them into different categories. The basic idea of classification techniques is to use some limited set of records, named a training set. In this training set, every object has the same number of items of the real database, and every object has already associated a label identifying its classification. We take Naïve Bayes algorithm as an example.

The Naïve Bayes algorithm gives us a way of combining the prior probability and conditional probabilities in a single formula, which we can use to calculate the probability of each of the possible classifications in turn. Having done this we choose the classification with the largest value.

Naïve Bayes Classification

Given a set of k mutually exclusive and exhaustive classifications c_1, c_2, …, c_k

which have prior probabilities $P(c_1)$, $P(c_2)$, …, $P(c_k)$, respectively, and n Attributes a_1, a_2, …, a_n which for a given instance have values v_1, v_2, …, v_n respectively, the posterior probability of class c_i occurring for the specified instance can be shown to be proportional to

$$P(c_i) \times P(a_1 = v_1 \text{ and } a_2 = v_2 \ldots \text{ and } a_n = v_n | \ c_i)$$

with the assumption that the attributes are independent. Then the value of this expression can be calculated using the product

$$P(c_i) \times P(a_1 = v_1 \mid c_i) \times P(a_2 = v_2 \mid c_i) \times \ldots \times P(a_n = v_n \mid c_i)$$

We calculate this product for each value of i from 1 to k and choose the classification that has the largest value.

Figure 1: The Naïve Bayes Classification algorithm

2.1.2. Association rules

Association rule mining is one of the most important tasks of data mining to find patterns in data. Association rules can be briefly expressed in the form of $X \Rightarrow Y$, where X and Y are sets of items. Association rule mining stems from the analysis of market-basket datasets.

The association rule mining problem can be formally described as follows: let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. A unique identifier, called *TID* is linked to each transaction. A transaction T is said to contain X, a set of some items in I, if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ holds in the transaction set D with confidence c if c% of

transactions in D that contain X also contain Y. The rule $X \Rightarrow Y$ has support s in D if s% of the transactions in D contain $X \cup Y$.

Apriori algorithm is used for generating all the supported itemsets of cardinality at least 2.

1: Create L_1 = set of supported itemsets of cardinality one
2: Set k to 2
3: While ($L_{k-1} \neq \emptyset$) {
4: Create C_k from L_{k-1} (see Figure 3, Generate C_k from L_{k-1})
5: Prune all the itemsets in C_k that are not
6: Supported, to create L_k
7: Increase k by 1
8: }
9: The set of all supported itemsets is $L_1 \cup L_2 \cup \ldots \cup L_k$

Figure 2: The A priori Algorithm

Generates C_k from L_{k-1}

Join Step:

Compare each member of L_{k-1}, say A, with every other member, say B, in turn.

If the first k-2 items in A and B (i.e. all but the last two elements in the two itemsets) are identical, place set $A \cup B$ into C_k.

Prune Step:

For each member c of C_k in turn {

Examine all subsets of c with k-1 elements

Delete c from C_k if any of the subsets is not a member of L_{k-1}

}

Figure 3: The A priori-gen Algorithm

2.1.3. Clustering

Clustering is an effective method to discover data distribution and patterns in underlying datasets. The primary goal of clustering is to learn where the data is dense or sparse in a dataset. Clustering is also considered the most important unsupervised learning problem, as it concerns with finding a structure in a collection of unlabeled data. The general definition of clustering can be stated as:

The process of organizing objects into groups whose members are similar in some way. Although classification is a convenient means for distinguishing groups or classes of objects, it requires the costly collection and labeling of a large set of training records or patterns, which the classifier uses to model each group.

K-means clustering is an exclusive clustering algorithm. Each object is assigned to precisely one of a set of clusters. This method of clustering is started by deciding how many clusters need to be formed from the raw data. This value is called *k*. Generally, the value of k is a small integer, such as 2, 3, 4 or 5.

We next select *k* points. They are treated as the centroids (initial central points) of *k* potential clusters. We can select these points as we wish, but the method may work better if the *k* initial points picked are fairly far apart. Then each point is assigned one by one to the cluster with the nearest centroid. The entire algorithm is summarized in Figure 4.

1. Choose a value of *k*
2. Select *k* objects as initial set of *k* centroids in an arbitrary fashion.
3. Assign each of the objects to the cluster for which it is nearest to the centroid.
4. Recalculate the centroids of the *k* clusters.
5. Repeat step 3 and 4 until the centroids no longer move.

Figure 4: The k-means clustering algorithm

2.2. Privacy-preserving techniques

2.2.1. Public-key encryption scheme

The idea of public-key cryptography [68] was first put forward in 1976. In 1977, Ronald Rivest, Adi Shamir and Leonard Adleman invented the famous RSA Cryptosystem. Several other public-key systems, such as *Elliptic Curve Cryptosystem* and *ElGamal Cryptosystem*, were proposed later on. The security of these public-key cryptosystems is based on different computation problems, such as *Discrete logarithm problem*, *Elliptic curve discrete logarithm problem*, *Factorization problem* and etc.

The idea behind a public-key cryptosystem is to find a cryptosystem where it is computationally infeasible to determine D_k given E_k. The advantage of such a system is to relieve the cost of communication of secret keys as is in a symmetric-key cryptosystem. We take *RSA* as an example to describe a public-key cryptosystem here:

RSA algorithm consists of three steps: key generation, encryption and decryption.

Key generation

1. Assume $n = p\ q$, where p and q are two distinct large primes.
2. Compute $j\ (n) = (p\text{-}1)\ (q\text{-}1)$
3. Choose an integer a, where $1<a< j\ (n)$ and a and $j\ (n)$ are co-prime (share no common divisors other than 1).
4. Compute b, such that $ab\equiv 1\ (\text{mod}\ j\ (\text{n}))$. The public key comprises p, q and the public exponent b. The private key comprises the modulus n and the private exponent a, which is secretly kept.

Encryption

Bob first sent the public key (n, b) to the Alice, who wishes to send message m to Bob.

Alice computes the ciphertext $c \equiv m^b \pmod{n}$, and then transmits c to Bob.

Decryption

Bob receives the c and recovers *m* by making the following computation:

$m \equiv c^a \equiv (m^b)^a \equiv m^{ab} \equiv m^{1+kj\,(n)} \equiv m(m^k)^{j\,(n)} \equiv m \pmod{n}$, since $ab = 1 + kj\ (n)$.

For small *n* of the *RSA Cryptosystem*, it is not secure in practice actually.

2.2.2. Oblivious transfer protocol

Oblivious transfer protocol (often abbreviated as OT) refers to a protocol that a sender sends some information to the receiver, but remains oblivious as to what is received.

The first form of oblivious transfer protocol [69] was presented in 1981. In this form, the sender gives out a message to the receiver with probability ½, while the sender remains oblivious as to whether the receiver gets the message or not. Rabin's oblivious transfer scheme is based on the RSA cryptosystem. A more useful form of oblivious transfer, named 1-out-2 oblivious transfer was invented and used to build protocols for secure multi-party computation. It is generalized to "1-out-of-*n* oblivious transfer" where the user gets exactly one database element without the server getting to know which element was queried. The latter notion of oblivious transfer is a strengthening of private information retrieval where one does not care about database's privacy.

In a 1-out-2 oblivious transfer protocol, the sender has two messages m_0 and m_1, and the receiver has a bit *b*, and the receiver wishes to receive m_b, without the sender learning *b*, while the sender wants to ensure that the receiver receives only one of the two messages. The protocol of Even, Goldreich, and Lempel is general, but can be instantiated using RSA encryption as follows.

1. The sender generates RSA keys, including the modulus N, the public exponent e, and the private exponent d, and picks two random messages x_0 and x_1, and sends N, e, x_0, and x_1 to the receiver.
2. The receiver picks a random message k, encrypts k, and adds x_b to the encryption of k, modulo N, and sends the result q to the sender.
3. The sender computes k_0 to be the decryption of q-x_0 and similarly k_1 to be the decryption of q-x_1, and sends $m_0 + k_0$ and $m_1 + k_1$ to the receiver.

The receiver knows k_b and subtracts this from the corresponding part of the sender's message to obtain m_b.

2.2.3. Secret sharing scheme

Here we present a special type of secret sharing scheme called threshold scheme. We formalize the definition as follows:

Let t, w be positive integers, $t \leq w$. A (t, w)-threshold scheme is a method of sharing a key K among a set of w participants denoted as P, so that t participants can compute the value of K, but no group of (t-1) participants can do so.

Shamir Threshold Scheme [72], invented by Shamir in 1979, is one of the methods to construct such a (t-w)-threshold scheme and describes as follows:

Initialization Phase

1. D chooses w distinct integer denoted as x_i, $1 \leq i \leq w$, $1 \leq x_i \leq n$, where $n \geq w+1$. For $1 \leq i \leq w$, D gives the value x_i to P_i. The values are public.

Share Distribution Phase

2. Suppose D wants to share a key $K \in [1,n]$, D secretly chooses t-1 values at random from $[1,n]$, denoted as $a_1, \ldots, a_{t-1}$.
3. For $1 \leq i \leq w$, D computes $y_i = a(x_i)$, where $a(x) = K + \sum_{j=1}^{t-1} a_j x^j \bmod n$.

4. For $1 \le i \le w$, D gives the share y_i to P_i.

In this scheme, the dealer construct a random polynomial function $a(x)$ of degree at most t-1. The constant value of the function is the key K. Each participant gets a share x_i from the dealer D. They calculate $y_i = a(x_i)$ correspondingly, and obtains the point (x_i, y_i) of the polynomial. In that case, they obtain a set of t functions, from which a group of t participants can jointly determine the polynomial by sharing (x_i, y_i) (i = 1, 2, …, t) and then K is obtained while t-1 participant cannot succeed.

2.2.4. Randomization techniques

Randomization is the process of perturbing the input data to distributed data mining algorithms so that the data values of individual entities are protected from revealing. Several randomization techniques can be identified in privacy preserving data mining algorithms, including adding random numbers, generating random vectors and random permutation of a sequence.

The typical example of randomization approach is the one found in Agrawal-Skrikant algorithm [1]. Data is perturbed in two manners: the value class membership and value distortion. The value class membership is a method that values of an attribute are divided into intervals and the interval in which a value lies is returned instead of the original value. The value distortion method works by adding a random value y_i to each value x_i of an attribute. Then, the original data distribution is reconstructed by the Bayesian approach, i.e., iterating $f_X^{(j)}(a) = \frac{1}{n}\sum_{i=1}^{n} \frac{f_y(w_i - a) f_X^{(j-1)}(a)}{\int_{-\infty}^{\infty} f_y(w_i - z) f_X^{(j-1)}(z) dz}$ until $f_X^{(j)}$ is statistically the same as the original distribution of X (using the c^2 goodness-of-fit test), where $X(=x_1, x_2, \ldots, x_n)$ is the original variable, $Y(=y_1, y_2, \ldots, y_n)$ is an random variable obeying a uniform

distribution between [-u, u], $f_Y(a)$ stands for the density function of Y, $w_i = x_i + y_i$ for i =1, 2, …, n, and for $f_X^{(0)}$ is a uniform distribution. Given a sufficiently large number of samples, $f_X^{(j)}$ can be expected to be very close to the real density function f_X of X after sufficient iterations. Based on the reconstructed distribution, decision trees can be induced [1].

2.3. Design tools

Privacy-preserving distributed data mining problems are normally addressed by means of cryptographic-related techniques, which provide various encryption tools to help protect individual and private information from being revealed when transferred online or communicated among different data sources. Here, we introduce some basic but common techniques in cryptography that can serve as building blocks for more advanced privacy-preserving protocols to tackle distributed data mining applications.

2.3.1. Homomorphic encryption scheme

Homomorphic encryption is a form of encryption where one can perform a specific algebraic operation on the plaintext by performing a different algebraic operation on the ciphertext. In secure computation protocols, we use homomorphic encryption keys to encrypt individual parties' private data so that their joint computation result can be obtained without decrypting the private input. In general, a homomorphic encryption scheme satisfies the following condition: $E(x_1) \cdot \mathrm{E}(x_2) = E(x_1 + x_2)$, where E is an encryption functions; x_1 and x_2 are plaintexts to be encrypted. According to associative property, $E(x_1 + x_2 + \ldots + x_n)$ can be computed as $E(x_1) \cdot E(x_2) \cdot \ldots \cdot E(x_n)$. That is,

$$E(x_1 + x_2 + \ldots + x_n) = E(x_1) \cdot E(x_2) \cdot \ldots \cdot E(x_n)$$

2.3.2 Secure sum

In distributed data mining algorithms, calculating the sum of values from individual sites is a very frequent task. Secure sum [5] assumes three or more parties with no collusion among them. It is also a special case of secure multi-party computations.

The value $n = \sum_{l=1}^{s} n_l$ is assumed to lie between [0…m]. One site, numbered as 1, is designated as the *master* site. The remaining sites are numbered 2…s. Site 1 generates a random number r, uniformly chosen from [0…m]. Site 1 adds r to its local value n_1 and passes (r+n_1) mod m to site 2. Since the value r is uniformly chosen from [1…m], (r+n_1) mod m is also distributed uniformly across this region, so site 2 learns nothing about the actual value of n_1.

Throughout the process from site l = 2…s-1, the algorithm is as follows. Site l receives

$$N = r + \sum_{j=1}^{l-1} n_j \mod m.$$

Since this value is uniformly distributed across [1…m], I learns nothing. Site i computes

$$r + \sum_{j=1}^{l} n_j \mod m = (n_j + N) \mod m$$

and then passes it to site l+1. This process continues until it is passed back to site 1.

2.3.3 Secure Scalar Product

The scalar product, or inner product, of two binary vectors is a commonly used tool in privacy-preserving data mining applications [7].

Notations:

- R_a: first binary vector with n elements (a_1, a_2, …, a_n)
- R_b: second binary vector with n elements (b_1, b_2, …, b_n)
- $R_a \cdot R_b$: product of first and second vector $\sum_{i=1}^{n} a_i b_i$
- (PK, SK): random generated public-private key pair
- r: a random number
- $e(x)$: encryption of x using PK
- $d(x)$: decryption of x using SK

The procedure of this protocol is summarized in Algorithm 1.

Setting: Alice has R_a and Bob has R_b
Goal: Bob learns $R_a \cdot R_b + r$ and Alice learns r
1. Bob generates (PK, SK) of a semantically secure Homomorphic encryption scheme and sends PK to Alice. 2. Bob encrypts his elements using PK and sends the vector ($e(b_1)$, …, $e(b_n)$) To Alice. 3. Alice generates r and encrypts it using PK. 4. Alice computes $Z = e(r) \cdot \prod_{i=1}^{n} y_i$, where $y_i = e(b_i)$ if $a_i = 1$ and $y_i = 1$ if $a_i = 0$. Alice sends Z to Bob. 5. Bob decrypts Z to get $d(Z) = r + \sum_{i=1}^{n} a_i \bullet b_i$ and sends to Alice. 6. Alice gets $\sum_{i=1}^{n} a_i \bullet b_i$ and sends to Bob.

Algorithm 1: Secure scalar product protocol

2.3.4 Secure Frequency Mining Protocol

Here, we present a primitive, which is the most popular in data mining applications. It is named secure frequency mining [54]. This protocol is implemented by additive homomorphic encryption scheme on a variant of Elgamal encryption. We describe the protocol as follows:

Notations:

- G: a group in which discrete logarithm is hard
- g: a generator in G
- U_i: the i^{th} user participating in the computation
- x_i: the first private key generated by the i^{th} party
- y_i: the second private key generated by the i^{th} party
- X_i: g^{xi}, the fist public key for the i^{th} party
- Y_i: g^{yi}, the second public key for the i^{th} party
- X : multiplication of all X_i; $\prod_{i=1}^{n} X_i$
- Y : multiplication of all Y_i; $\prod_{i=1}^{n} Y_i$

Suppose that each user holds a Boolean value d_i, and the miner's goal is to learn $d = \sum_{i=1}^{n} d_i$. The privacy-preserving protocol for the miner to learn d is detailed in Figure 5.

$U_i \rightarrow$ miner: $m_i = g^{di} \cdot X^{yi}$;
$h_i = Y^{xi}$.

Miner: $r = \prod_{i=1}^{n} \frac{m_i}{h_i}$;

for d = 1 to n
If $g^d = r$ **then**
Output d.

Figure 5: Secure frequency mining protocol

Now we prove that when the miner finds $g^d = r$, the value d is the desired sum. Suppose $g^d = r$, then

$$g^d = r = \prod_{i=1}^{n} \frac{m_i}{h_i} = \prod_{i=1}^{n} \frac{g^{d_i} \cdot X^{y_i}}{Y^{x_i}} = \prod_{i=1}^{n} g^{d_i} \cdot \prod_{i=1}^{n} \frac{X^{y_i}}{Y^{x_i}}$$

$$= g^{\sum_{i=1}^{n} d_i} \cdot \prod_{i=1}^{n} \frac{(\prod_{j=1}^{n} X_j)^{y_i}}{(\prod_{j=1}^{n} Y_j)^{x_i}} = g^{\sum_{i=1}^{n} d_i} \cdot \prod_{i=1}^{n} \frac{(g^{\sum_{j=1}^{n} x_j})^{y_i}}{(g^{\sum_{j=1}^{n} y_i})^{x_i}}$$

$$= g^{\sum_{i=1}^{n} d_i} \cdot \frac{g^{\sum_{i=1}^{n}\sum_{j=1}^{n} x_j y_i}}{g^{\sum_{i=1}^{n}\sum_{j=1}^{n} y_j x_i}} = g^{\sum_{i=1}^{n} d_i}.$$

Thus, $g^d = g^{\sum_{i=1}^{n} d_i}$ as desired. For $d = 1$ to n, it is easy to find the value of d.

2.4. Evaluation criteria of PPDDM protocol

In this section of the thesis, we are going to present some metrics that can be used to measure and evaluate privacy-preserving distributed data mining protocols. Researchers have designed and developed some set of measuring metrics regarding privacy-preserving data mining. In [3], Elisa Bertino and Igor Nai Fovino proposed a framework for evaluating PPDM. They devised some general criteria to evaluate the effectiveness and correctness of PPDM algorithms, including efficiency, scalability, data quality, hiding failure and privacy level. In general, privacy-preserving data mining algorithms can be evaluated and analyzed in regard to the work of privacy, complexity and accuracy. These three areas are the major design and testing goals of PPDM algorithms.

In the case of PPDDM, that is, privacy-preserving distributed data mining, cryptographic techniques are commonly employed to protect the privacy of each data holder while still ensuring the result is accurate compared with non-privacy preserving techniques exerted on data mining algorithms. This strategy is quite different from that of reconstruction-based techniques [3] used in centralized data

mining tasks, where a trade-off between the privacy of datasets and accuracy of mining results is unavoidable.

In this thesis, we are not going to quantitatively analyze the privacy and accuracy of these protocols, because their outcomes are designed and proved to be secure and correct under the agreed assumption and privacy definition. Rather, we focus on how well these algorithms perform to achieve the security goals – the efficiency parameter.

Now let's get down to the efficiency parameter in more detail. Efficiency is a metric that is used to assess the resources consumed by a privacy-preserving data mining algorithm. It's also known as the complexity analysis of an algorithm, which represents the ability of the algorithm to execute with good performance that is assessed in terms of time and space, and in case of distributed data mining algorithms, in terms of communication cost and computation cost.

- Time requirements are usually measured in terms of CPU time, or computation cost or even the average number of operations required by the PPDDM techniques. Normally, it is desirable for an algorithm to have a polynomial complexity than to have an exponential one. As is in the case of privacy preserving distributed data mining, it is advisable and practical to confine the execution times of the algorithms to being proportional to that of the non-privacy preserving data mining algorithms. Space requirements are assessed by means of the amount of memory allocated to implement the given algorithm, or the number of items and values assessed in case of privacy preserving data mining.
- Communication requirements are evaluated in terms of the amount of information exchanged among all the sites involved in the distributed data mining tasks. The communication overhead can further be measured by means of communication

rounds, which indicates the synchronizing capability of the distributed system. The unit of measure of communication overhead in thesis is byte.

- Scalability is another important aspect to assess the performance of PPDDM algorithms: it represents the efficiency trends of an algorithm towards the increase in the size of datasets. Thus, this parameter is used to measure both the performance and storage requirements together with the costs of the communications required by a distributed data mining technique when data sizes increase. A PPDDM algorithm must be designed and implemented to be scalable with larger datasets, due to the rapid development of the hardware and storage technology, which enables it possible to store and manage increasingly huge amounts of data.

Therefore, in this thesis, the evaluation metrics of privacy-preserving distributed data mining algorithms are summarized as communication cost, communication rounds, computation cost and scalability.

Chapter 3 Classification of PPDDM Protocols

3.1. Introduction

3.1.1. Overview

Privacy issues arise when distributed data computing applications become popular in private and public sectors. Different data holders across scattered spots want to undertake a joint data mining task to obtain certain global patterns that will benefit them all whilst they each are reluctantly to disclose their private data sets to one another during the execution of the computing. This trick problem is commonly referred to as *privacy-preserving distributed data mining*.

Let us first take a look at two real-world examples of distributed data mining with different privacy constraints:

- **Scenario 1**: Multiple competing supermarkets, each having an extra large set of data records of its customer's buying behaviors, want to conduct data mining on their joint data set for mutual benefit. Since these companies are competitors in the market, they do not want to disclose too much about their customer's information to each other, but they know the results obtained from this collaboration could bring them an advantage over other competitors.
- **Scenario 2**: Success of homeland security aiming to counter terrorism depends on combination of strength across different mission areas, effective international collaboration and information sharing to support coalition in which different organizations and nations must share some, but not all, information. Information privacy thus becomes extremely important; all the parties of the collaboration promise to provide their private data to the collaboration, but neither of them want each other or any other party to learn much about their private data.

The above scenarios describe different PPDDM problems. Each scenario poses a set of challenges. For instance, scenario 1 is a typical example of heterogeneous collaboration, while scenario 2 refers to a task in a homogeneous cooperation setting.

Technology alone cannot address all of the PPDDM scenarios [32]. The above questions can be to some extent addressed if we provide some key requirements to guide development of technical solutions. One alternative is to describe them in terms of general parameters. In [32], some parameters are suggested:

- **Outcome**: Refers to the desired data mining results. For instance, some may look for association rules identifying relationships among attributes, or relationships among customers' buying behaviors in scenario 1, or may even want to classify data as is in scenario 2.
- **Data Distribution**: How are the data available for mining? Are they horizontally distributed or vertically distributed across multiple sites? In the case of horizontally partitioned situation, each data owner holds the same schema of entities in their database, and in vertically partitioned scenario, different sites contain different attributes for every entity.
- **Privacy Preservation**: What specific concerns are required to tackle privacy issues? If the privacy is maintained for every local data holder, the individual privacy or personal identifiable information is ensured, otherwise collective privacy is. Even for personal privacy, privacy level can vary regarding data privacy or data anonymity.

3.1.2. Research questions

Several research questions have been asked about this field: 1) what kinds of options exist for privacy preserving purposes in distributed data mining? 2) Which

method or technique is more popular or prevailing? 3) How to measure the performance of privacy-preserving distributed data mining protocols? We reviewed 60 recent published journal and conference papers from 2000 to 2008 to analyze and demonstrate these problems.

3.2. Related work

There are some works by other researchers in regard to synthesizing and classifying existing privacy-preserving data mining literatures. Vassilios S. Verykios, Elisa Bertino and Igor Nai Fovino [3] propose five dimensions to classify and analyze privacy-preserving data mining algorithms with aims of state-of-the-art. Their classification dimensions are data distribution, data modification, data mining algorithm, data or rule hiding and privacy preservation. Based on their classification dimension, in [3], they proposed a classification taxonomy of existing PPDM algorithms. According to the features of privacy preservation solutions, these algorithms are primarily divided into three categories: heuristic-based, reconstruction-based and cryptography-based. The former two categories deal with centralized database and the last one tackles with distributed database. In [46], Xiaodan Wu et al. presented a simplified taxonomy to consolidate the previous one. They analyzed and summarized existing references, thus putting the taxonomy into practical usage.

Although the scheme and taxonomy by Bertino, Nai and Parasility in [3] provided a comprehensive coverage for privacy-preserving data mining algorithms, it still has two major drawbacks. Firstly, they did not provide us with specific cryptographic techniques used in the cryptographic-based solutions for distributed-DB case. Rather, they merely mentioned encryption techniques. Secondly, in distributed database scenarios, we usually do not pay too much attention to whether raw data or aggregated

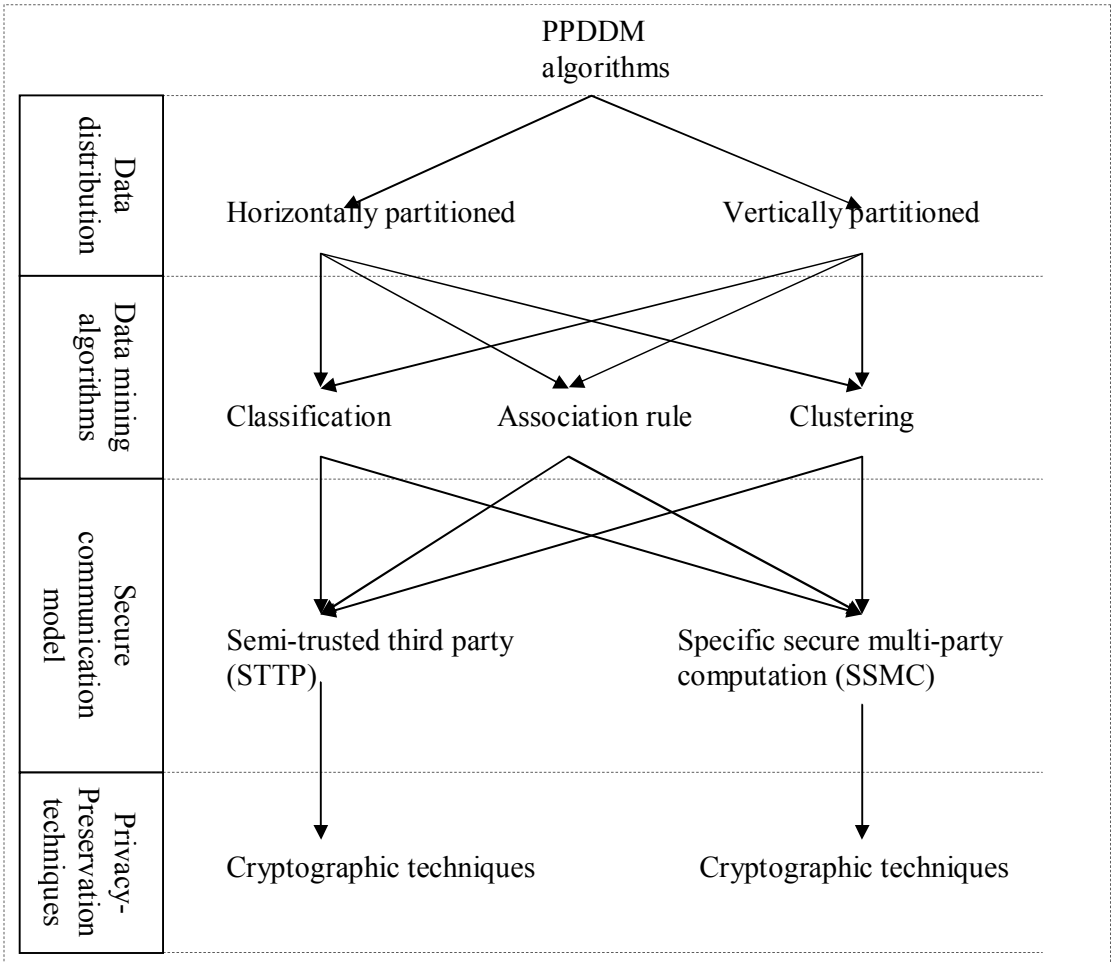
data is hidden, because normally we aim at hiding raw data, which requires a more contingent privacy level. Instead, how data is distributed, namely, horizontally partitioned or vertically partitioned across data sites is the factor that counts and interests us.

3.3 Classification dimensions of PPDDM protocols

In this section, we present a concise classification scheme for PPDDM protocols. In this scheme, four dimensions are identified according to which any privacy preserving distributed data mining problems can be categorized and classified. They are:

- Data partitioning model
- Data mining algorithms
- Secure communication model
- Privacy preserving techniques

We propose a taxonomy regarding PPDDM protocols contained in four levels (see Figure 6). This scheme is different and innovative from current relevant schemes in two ways: 1) it specifically deals with the distributed privacy-preserving data mining protocols while some other schemes do not deal with this area in depth; 2) It includes data distribution models of distributed data mining, including horizontally partitioned and vertically partitioned, which other schemes have not specified clearly; 3) This scheme expands cryptographic techniques used in distributed data mining for privacy protecting purpose, such as encryption, secret sharing, oblivious transfer and etc. Figure 6 depicts a general architecture of how these dimensions interrelated to one another.



Cryptographic techniques = {public-key encryption, oblivious transfer, secret sharing, randomization}

Figure 6: Classification of PPDDM protocols

3.3.1 Data partitioning model

The first dimension is data partitioning model. In distributed data mining scenarios, datasets can be distributed and scattered in different locations in different models. Two basic ways of distribution of datasets are: homogeneous distribution (horizontal partitioning) and heterogeneous distribution (vertical partitioning). These two models are formally defined as follows [44]: assume dataset be D in terms of the entities for which the data is collected and the information that is collected for each entity. Thus, we denote $D \equiv (R, A)$, where R is the data records that are collected for each individual or entity and A is the attribute set that is collected about each record. We assume that there are k different sites, $P_1, \ldots, P_k$ collecting datasets $D_1 \equiv (R_1, A_1), \ldots, D_k \equiv (R_k, A_k)$ respectively.

It assumes that in horizontal partitioning data model, different sites collect the same sort information about different entities. Therefore, in horizontal partitioning, $R_G = \cup_i R_i = R_1 \cup \ldots \cup R_k$, and $A_G = A_1 \ \ldots = A_k$. Such situations exist in real life. For example, two organizations collect very similar information. However, the customer base for each organization tends to be quite different. Figure 7 demonstrates horizontal partitioning of data. The figure shows two medical institutions, New York Hospital and Chicago Medical Centre, each of which collects personal information for their patients. Attributes such as PatientID, Gender, Age, Occupation and Disease are stored in both databases. Merging the two databases together should lead to more accurate predictive models used for medical research activities.

Patient ID	Gender	Age	Occupation	Disease

New York Hospital				
Patient ID	**Gender**	**Age**	**Occupation**	**Disease**
18	Female	35	Manager	Arthritis
73	Male	27	Engineer	Diabetes
…	…	…	…	…
249	Female	56	Teacher	Tuberculosis

Chicago Medical Centre				
Patient ID	**Gender**	**Age**	**Occupation**	**Disease**
291	Female	43	Nurse	Hepatitis
426	Male	19	Student	Chlamydia
…	…	…	…	…
610	Male	62	Retired	Obesity

Figure 7: Horizontal partitioning / Homogeneous distribution of data

On the other hand, vertical partitioning of data assumes that different sites collect different feature sets for the same set of entities. Thus, in vertical partitioning, $R_G = R_1 \ldots = R_k$, and $A_G = \cup_i = A_1 \cup \ldots \cup A_k$. For example, in some city, Walmart collects information about potatoes and tomatoes consumed by a group of customers. Carrefour, it competitor, collects information about chicken, beef and pork bought by the same group of customers. Groceries can be linked to butchers. This linking

information can be used to join the databases. The joint database could then be mined to reveal more useful information about the buying behaviour of that group of customers. Figure 8 demonstrates vertical partitioning of data.

TID	Potato	Tomato	Chicken	Beef	Pork

Walmart		
TID	**Potato**	**Tomato**
ABC	Yes	Yes
MFN	No	No
…	…	…
QYR	No	Yes

Carrefour			
TID	**Chicken**	**Beef**	**Pork**
ABC	No	No	Yes
MFN	No	Yes	No
…	…	…	…
QRY	Yes	Yes	Yes

Figure 8: Vertical partitioning / Heterogeneous distribution of data

A fully distributed setting is a special case of horizontal partitioning when the number of entities, denoted as N, is the same as the number of parties involved in the distributed computing, that is $N = k$. In the case of distributed data mining, if every individual user or participant holds exactly one record of the dataset, which is provided to the joint computation, this scenario is called a fully distributed setting. There has been some more complex and hybrid partitioning models of data, like the partitioning of each entity or each feature is different.

The following table summarizes the relevant references specifying privacy-preserving data mining problems in horizontally partitioned or vertically partitioned environment, respectively.

Data distribution	References
Horizontally partitioned	[2][7][10][14][16][21][22][26][27][39][54][55][57]
Vertically partitioned	[4][6][9][17][18][19][20][23][24][33][34][40][41][42][43][48][51][52][61][63]

Table 1: Summary of data distribution references

3.3.2 Data mining tasks / algorithms

The second dimension is data mining algorithms on which privacy preserving techniques are imposed. Generally, the data mining algorithms under research include classification, association rule mining and clustering. Classification concerns the problem of finding a set of models (or functions) that describe and distinguish the data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. As to classification, there are several different ways to classify a new instance. They are naïve Bayes classifier, decision tree classifier and *k*-nearest neighbor classifier. Association rule mining is the process of discovering associated rules and showing attribute value and conditions that occur frequently in a given set of data. Clustering analysis involves the process of decomposing or partitioning a data set (usually multivariate) into groups so that the points in one group are similar to each other and are as different as possible from the points in other groups. For clustering, the most commonly used algorithms are *k*-means clustering and EM clustering.

The following table summarizes the references regarding the data distribution models that various data mining algorithms are divided into.

Data mining Algorithms	References	
	Horizontally partitioned	Vertically partitioned
Classification	[9][20][21][51][54][57]	[2][7][24][42][43][48][51][52][61]
Association Rules	[10][14][19][22][39][56]	[4][14][23][33][34][40][63]
Clustering	[16][17][18][27]	[6][18][41]

Table 2: Summary of data mining techniques references

3.3.3 Secure communication model

Here, we present our third classification dimension, secure communication model, which generally refers to the interactive relation of the participants joining in the cooperative computation and the roles they play in the whole process of privacy-preserving distributed data mining tasks. Another similar term is "coopetative" model [36]. This term stems from the word "coopetation", which was originally employed in social-economics to describe the situation that competing entities producing the same line of products and services have to cooperate with each other to improve the overall value of their market by means of making decisions based on the joint analysis of their private data. Similarly, distributed data mining tasks commonly feature a scenario where all the data holders participating in the joint computation on their individually private data sets naturally have the desire and interest to obtain the final result of the application. As the proprietary owner of their individual data set, it is understandable that each data holder is reluctant to share private information with other data holders. However, in order to reach the final result of the distributed data

mining, they are ready and motivated to provide inputs to the computation, as long as the privacy requirements are met.

Generally, most practical approaches to solve this scenario is to conduct the secure computation at one or more of the participants or at one or more third parties with the assumption that all of participants are semi-honest [39] and the third parties are semi-trusted participants [39]. Herein, let us give the informal definition of both semi-honest and semi-trusted. In [44], a semi-honest party (i.e. honest but curious) follows the rules of the protocol using its correct input, but is free to learn from what it sees during the execution of the protocol to compromise security. In [56], a third party is semi-trusted if it fulfills the following condition: the third party is trusted to provide some commodities or compute intermediate outcome of the computation based on encrypted input it receives; it follows the execution of the protocol correctly, just like all the other users as well, although it tries to learn and deduce some information from its own input and output.

Under such scenario and assumption, privacy-preserving distributed data mining problems can be solved mainly based on two types of secure computation model: One is based on Semi-trusted Third Party (STTP) model. Theoretically, the general secure multi-party computation protocols can be used to deal with any collaborative data mining problems, yet this kind of solutions are too inefficient when the database is huge in amount and the number of participants is large, due to its intricate and complicated design. On the other hand of the spectrum, the trusted third party (TTP) model is too navie and straightforward, so that the privacy is compromised to a larger extent at the point of the TTP. Therefore, more practical solutions have been put forward in the past few years with respect to how to solve the privacy issues of distributed data mining more efficiently and accurately. Among them, 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). In real world, it is much more feasible to find such a semi-trusted third party than to find a trusted third party. This semi-trusted third party can be implemented by means a miner, a mixer, or a commodity server, that all act in a semi-trusted manner.

The other stream is based on Specific Secure Multi-party Computation under Semi-honest assumption (SSMC). It aims at accomplishing efficient and accurate solution for the PPDDM problems. Under the semi-honest assumption, specific secure multi-party computation protocols are employed to deal with functions commonly used in data mining applications rather than the general secure multi-party computation protocol. These techniques include secure sum, secure set union, secure intersection, secure scalar product, etc. The advantage of such kind of protocols and tools lies in that they are designed to specially fit in with the data mining tasks, instead of any general functions. As the function for secure computation can be identified, the computing complexity is reduced greatly and a linear proportional cost can be obtained.

Clifton and Vaida [5] propose a toolkit of techniques that can be used to address various kinds of PPDDM issues in a practical and performance-friendly manner. The assumption of their ideas and techniques are that all participants of the joint computation conduct the algorithm strictly within themselves, rather than inviting any other external parties of any trust level to assist the computation. They argue that most distributed data mining tasks can be reduced to computing certain simple functions such as sum, average, frequency, sets union, etc with inputs of individual parties, which are the private information that needs protecting. Furthermore, some techniques can be generalized to tackle the computation of these simple functions and the

preservation of private data, thus solving many privacy-preserving distributed data mining problems. These techniques are all built on SSMC model; however the computational complexity is greatly reduced compared with Yao's secure evaluation function [55].

Alex Gurevich and Ehud Gudes [14] propose a new architecture of conducting privacy-preserving distributed association rule mining applications and further extend to general distributed data mining tasks. This architecture is composed of *n* participating databases, a miner which manages the computation and decides which computation to be done, and the calculator which compute without knowing what itemset it computes. The key point of the architecture is that it invites the miner and calculator as third parties to participate in the joint computation while during the process, only the miner and all participants get to know the result of the mining and the calculator only performs auxiliary computations without knowing any information provided and published.

The miner and calculator in the above architecture and model depict a general picture of what we present in this thesis as Semi-trusted third party (STTP). These third parties are semi-honest themselves like those semi-honest participants, but they do not hold any part of the data or provide any private inputs. Their roles are just to manage the data mining process or assist in the computation. They do not collude or collaborate between them or any participants and they not trusted by the participants. Therefore, to our assumption and definition, external parties that possess such properties other than any data holders participating in the distributed mining process are called Semi-trusted third parties.

The following table summarizes the categories of secure computation models (STTP or SSMC) on which papers published from 2000 to 2009 are based:

Secure communication model	References	
	Horizontally partitioned	Vertically partitioned
Semi-trusted third party (STTP)	[2][10][14][16][21][26][27] [39][54][56][57]	[7][14][34][39]
Specific secure multi-party computation (SSMC)	[9][17][18][19][20][22][51]	[4][6][18][23][24][33][40][41] [42][43][48][51][52][61][63]

Table 3: Summary of secure communication model references

3.3.4 Privacy preservation techniques

The fourth dimension to classify PPDDM algorithms are the privacy preserving techniques used to protect the private information communicated among sites and central miner or mixer. These techniques include homomorphic encryption scheme, public key cryptosystem, etc. All of them serve as the building blocks for other more advanced and high-level protocols, such as privacy-preserving frequency mining, privacy-preserving summation, etc.

We present some efficient methods to conduct secure computation in distributed data mining settings. This is by no means an exhaustive list of efficient methods and techniques to achieve privacy preserving data mining protocols. They are, however, sufficient to allow us to present several privacy-preserving solutions for distributed data mining problems.

- Oblivious transfer: refers to a protocol by which a sender sends some information to the receiver, but remains oblivious as to what is received. See Section 2 for more details.

- Public-key encryption: an encryption holds the property of additively homomorphic if the functionality of the encrypted values can be obtained by means of the encryption of the addition of the values, i.e. $E(a) * E(b) = E(a + b)$. See Section 2 for more details.
- Secret sharing: refers to any the method for distributing a secret amongst a group of participants, each of which is allocated a share of the secret. The secret can only be reconstructed when the shares are combined together; individual shares are of no use on their own, i.e. Shamir secret sharing scheme. See Section 2 for more details.
- Randomization: refers to adding a noise to the original data to hide its real value, thus protecting privacy of the data sets.

The following table summarizes the references in relation to the above-mentioned privacy-preserving techniques employed in different data partitioning models of PPDDM.

	PP Techniques	References
H-Partition	Public-key encryption	[10][17][22][23][24][34][39][41][54][56][57][61][63]
	Oblivious transfer	[2][16][18][20][21][43]
	Secret sharing	[9]
	Randomization	[14]
V-Partition	Public-key encryption	[4][8][10][19][23][24][33][40][42][43][48]
	Secret sharing	[6][27]
	Oblivious transfer	N/A
	Randomization	[14][51][52]

Table 4: Summary of privacy preservation techniques references

Chapter 4 Privacy-preserving Distributed Naïve Bayes Classifier

The study on secure distributed classification solutions is an important task. The goal is to have a simple, efficient and privacy-preserving classifier. The ideal would be for all parties to jointly decide on a model and then use the model locally. In this section, we discuss the context of Naïve Bayes classifier. We assume that the data address the problem of building naïve classifier in a horizontally partitioned setting. This means that many parties collect the same set of information about different entities. They do not want to reveal their own instances or the instance to be classified.

We formally define the problem as follows: assume there are m attributes, (A_1, A_2, …, A_m), and one class attribute V, which has a domain of $\{v_1, v_2, \ldots v_p\}$. We also assume that there are n data holders (U_1, …, U_n), where each data holder U_j , $j \in [1, n]$ has a vector denoted (a_{j1}, …, a_{jm}, v_j) which is an instance of the attributes vector (A_1, A_2, …, A_m) and v_j is U_j's class label. In this problem setting, a classifier to classify a new instance is required for selecting the most likely class label v:

$$V = \arg\max \Pr(v^{(l)}) \prod_{i=1}^{m} \Pr(a_i | v^{(l)}),$$

where (a_1, …, a_m) is the attributes vector of the new instance.

4.1. STTP-based public-key encryption solution – *Privacy-preserving Classification of Customer Data without Loss of Accuracy* [54]

In this protocol, a semi-trusted miner is introduced to compute the intermediate value transferred by each participant and ElGamal public-key encryption scheme with additive homomorphic property is employed to achieve privacy-preserving goal. For this protocol, we consider the following scenario: n parties, m attributes and d class values.

4.1.1. Notations

- m: total number of non-class attributes in the dataset
- A: attribute set of dataset ($A_1, A_2, \ldots, A_m$)
- A_i: the i^{th} attribute of A ($1 \leq i \leq m$)
- $\{a_i^{(1)}, \ldots, a_i^{(d)}\}$: value domain of the i^{th} attribute, $1 \leq k \leq d$
- V: the class attribute in the dataset
- $\{v^{(1)}, \ldots, v^{(p)}\}$: value domain of class variable, $1 \leq l \leq p$
- U_j: the j^{th} user participating in the computation, $1 \leq j \leq n$
- n: total number of users participating in the computation
- $(a_{j1}, \ldots, a_{jm}, v_j)$: the j^{th} user's vector of the dataset
- S: the set of privacy-sensitive attributes; $S \subseteq A$

4.1.2. Protocol

A detailed specification of the protocol is given in Algorithm 2.

Input: n parties, m attributes, d class values

Output: naïve bayes classifier

1: Let $a_{ji}^{(k,l)} = 1$ if $(a_{ji}, v_j) = (a_i^{(k)}, v^{(l)})$; 0 otherwise.

2: U_j's private keys: $(x_{ji}^{(k,l)})_{i \in S}$, $(y_{ji}^{(kl)})_{i \in S_{(k,l)}}$

3: Public keys: $X_{ji}^{(k,l)} = g^{x_{ji}^{(k,l)}}$; $Y_{ji}^{(k,l)} = g^{y_{ji}^{(k,l)}}$.

4: $X_i^{(k,l)} = \prod_{1 \le j \le m} X_{ji}^{(k,l)}$; $Y_i^{(k,l)} = \prod_{1 \le j \le m} Y_{ji}^{(k,l)}$.

5: U_j: **for** $i \in S$ **{**

6: $\bar{a}_{ji}^{(k,l)} = g^{a_{ji}^{(k,l)}} \cdot (X_i^{(k,l)})^{y_{ji}^{(k,l)}}$;

7: $h_{ji}^{(k,l)} = (Y_i^{(k,l)})^{x_{ji}^{(k,l)}}$; **}**

8: $Uj \rightarrow$ miner: $(\bar{a}_{ji}^{(k,l)}, h_{ji}^{(k,l)})_i (a_{ji})_{i \in S}, v_j$.

9: Miner: **for** $i \in S$ **{**

10: $r_i^{(k,l)} = \prod_{j=1}^{n} \frac{\bar{a}_{ji}^{(k,l)}}{h_{ji}^{(k,l)}}$;

11: **}**

12: **for** $\#(a_i^{(k)}, v^{(l)}) = 1 \ldots n$ **{**

13: if $g^{\#(a_i^{(k)}, v^{(l)})} = r_i$ break;

14: **}**

15: **for** $i \in S$ **{**

16: count $\#(a_i^{(k)}, v^{(l)})$.

17: **}**

18: **for** $l = 1 \ldots p$ **{**

19: count $\#(v^{(l)})$;

20: **}**

21: Compute the posterior probability based on the frequency counts obtained. Output naïve bayes classifier. (see Section 2.1 for details)

Algorithm 2: STTP-based Privacy-preserving Naïve Bayes classifier

4.1.3. Protocol Analysis: we implicitly assume that the output classifier is encoded in such a way that it contains the frequencies $\#(v^{(l)})$ and $\#(a_i^{(k)}, v^{(l)})$ for all (i, k, l).

Complexity Analysis: n customers and m attributes are involved in this protocol. 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. Assume there are s sensitive attributes, where $s = |S|$. It can be deduced that the computational overhead of each customer, as opposed to a non-private solution, is dps encryptions. In data mining applications, we usually have $n>>dps$; thus,

Communication overhead:

We calculate the communication cost of the protocol by means of calculating the bits of total information exchanged among all sites during the execution of the algorithm. In this case, we note the figure as $O(2.dps.n)N$; the exchange round is constant number 2.

Computation overhead:

The computation cost is measured by means of counting the total number of additional encryption and decryption operations executed in the algorithm. The time cost is $O(2.dps.n)$; the space cost is $3nN$

4.2. SSMC-based public-key encryption solution – *Privacy Preserving Naïve Bayes Classifier for Horizontally Partitioned Data* [20]

In this protocol, we consider the following scenario: there are n parties participating in the computation, m attributes in the dataset, the class variable V has d values. Algorithm 3 illustrates the protocol of generating the output of classifier on horizontally partitioned dataset in privacy preserving manner. When it comes to the case of numeric attributes, we deal with it by first converting the numeric attributes to nominal attributes and then running the protocol. Thus, in this section we only discuss the case of nominal attribute.

4.2.1. Notations

- n: number of parties participating the computation
- d: number of class variable values in the dataset
- m: number of attributes in the dataset
- v_j: the ith value of class variable, $1 \leq j \leq d$
- c_{yz}^{x} : the number of instances with party Px having class y and attribute value z
- a_{y}^{x} : the number of instances with party Px having class y
- p_{yz} : the probability of an instance having class y and attribute value z

4.2.2. Protocol

Input: n parties, m attributes, d class values
Output: Naïve bayes classifier
1: **for** (class values $y = v_1 \ldots v_d$) **{**
2: **for** ($i = 1 \ldots k$) **{**
3: $\forall z$, Pi locally computes c_{yz}^{i}
4: Pi locally computes a_{y}^{i}
5: **}**
6: **}**
7: $\forall (y, z)$, All parties calculate $c_{yz} = \sum_{i=1}^{k} c_{yz}^{i}$ using secure sum protocol (see Section 2.4)
8: $\forall y$, All parties calculate $a_y = \sum_{i=1}^{k} a_{y}^{i}$ using secure sum protocol
9: All parties calculate $p_{yz} = c_{yz}/a_y$
10: Output naïve bayes classifier (see Section 2.1 for details)

Algorithm 3: SSMC-based Privacy-preserving Naïve Bayes classifier

4.2.3. Protocol Analysis: since all the model parameters are completely present with all the parties, evaluation is no problem at all. The party which wants to evaluate an instance simply uses the Naïve Bayes evaluation procedure locally to classify the

instance. The other parties have no interaction in the process. Thus, there is no question of privacy being compromised.

Performance analysis: 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. In this case, the communication overhead is noted as O(*dps.n*)*N*; the communication round is $n + 1$, a number proportional to the number of sites.

The computation cost is measured in terms of the total number of additional encryption and decryption operations executed in the algorithm. In this protocol, the computation overhead is O(*dps.n*); space complexity is *nN,* that is the amount of storage consumed by parameters of the algorithm.

4.3. Performance comparison

Here, we provide a graphic illustration of both algorithms in terms of their communication cost, computation cost and communication rounds. Our experiment is done on the environment of a PC with a 1GH*z* processor and 512MB memory under NetBSD. The simulations of the protocols are implemented in the C#.NET programming language. The length of cryptographic key is 512 bits. The dataset we used is a synthetic control chart time series data taken from the UCT Machine Learning repository. The dataset consists of 60 rows and 6 columns. It is attached in the Appendix of this thesis. 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 comparison and 3, 5, 7, 9 and 11 participating sites respectively for the comparison of computation cost and communication rounds. The first test is to see how much communication overhead

STTP-based protocol brings and how the communication overhead SSMC-based protocol compares with the former one. The total amounts of transmissions caused by the protocols with respect to the number of parties are depicted in Figure 9. The communication rounds of each protocol are displayed in Figure 10. As expected from the formula in Section 4.1.3 and Section 4.2.3, STTP-based protocol incurs a constant communication round, which is 2, while SSMC-based protocol incurs *n* rounds. The second test that we have performed is to analyze and compare the computational overheads brought by STTP-based protocol and SSMC-based protocol. Execution times of the protocols with respect to the number of parties are shown in Figure 11. The communication cost, communication rounds and computation cost for both protocols are recorded, compared and presented below.

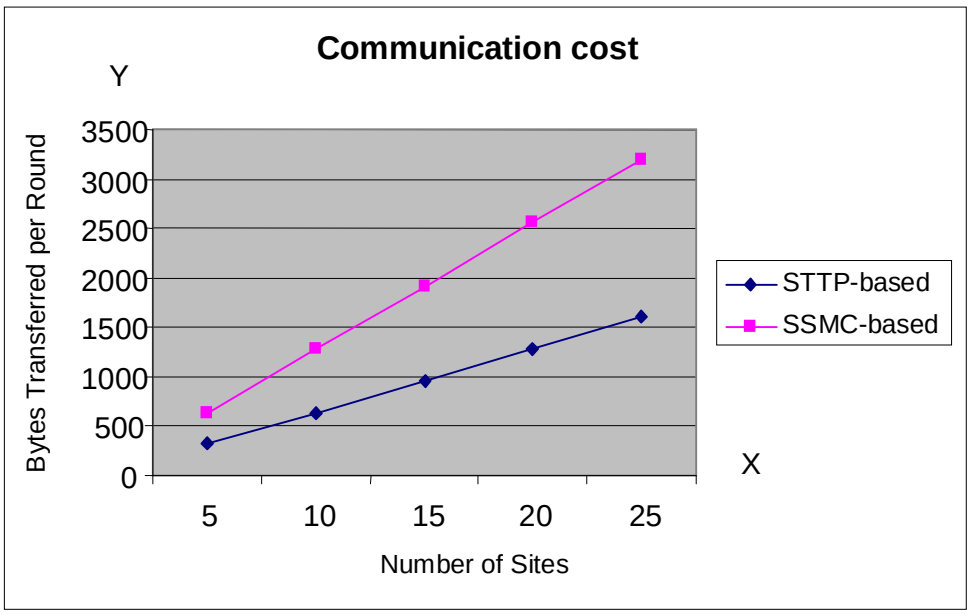


Figure 9: Communication cost comparison for classification

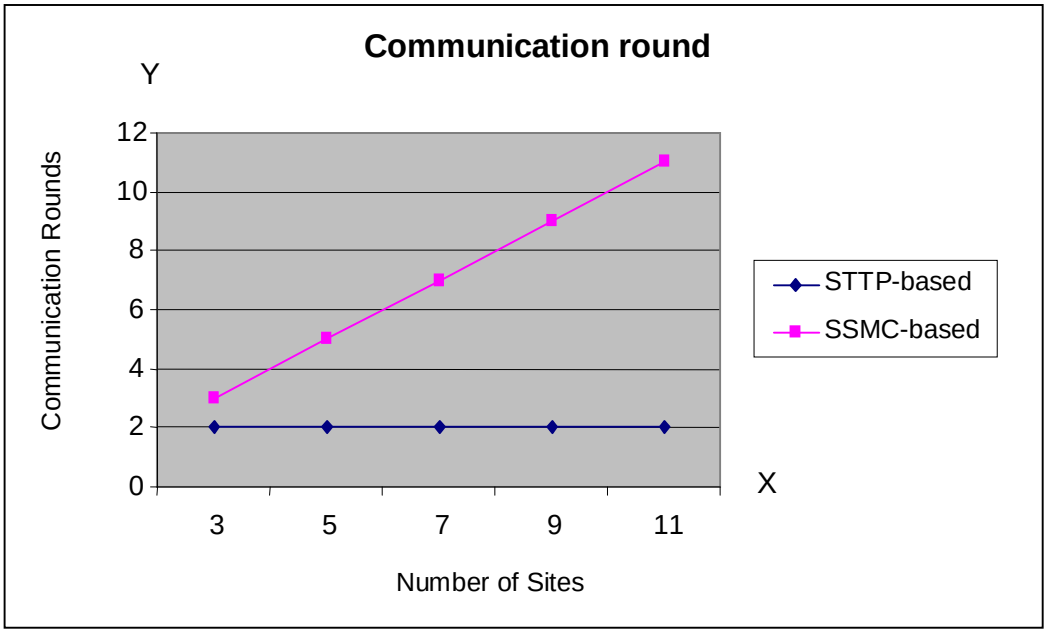


Figure 10: Communication round comparison for classification

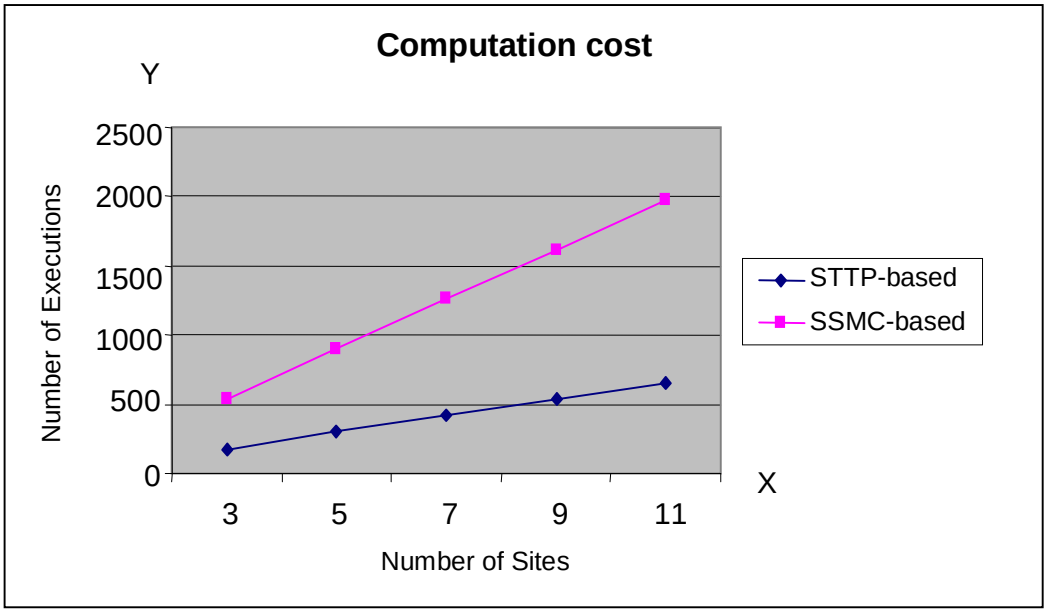


Figure 11: Computation cost comparison for classification

From the information presented in the above graphs, we can clearly identify that the communication, computation cost and the communication rounds of Wright's STTP-

based protocol are all lower than those of Vaidya's SSMC-based protocol. Therefore, through the analysis and comparison based on our evaluation framework, we conclude that Wright's protocol dominates Vaidya's protocol in the overall performance.

Chapter 5 Privacy-preserving Distributed Association Rule Mining

This section addresses the problem of computing association rules within such a distributed scenario, in which one of the distributed custodians of data are allowed to transfer their data to another site. Here, we assume homogeneous / horizontal (see Section 3.1 for more details) databases: all sites have the same schema, but each site has information on different entities.

This problem of distributed association rule mining can be formulated as follows: assume that a transaction database DB is horizontally partitioned among n sites (namely S_1, S_2, …, S_n) where $DB = DB_1 \cup DB_2 \cup \ldots \cup DB_n$ and DB_i resides at side S_i ($1 \leq i \leq n$). The itemset X has local support count of $X.sup_i$ at site S_i if $X.sup_i$ of the transactions contains X. The global support count of X is given as $X.sup = \sum_{i=1}^{n} X.sup_i$. An itemset X is globally supported if $X.sup \geq s \times (\sum_{i=1}^{n} |DBi|)$. Global confidence of a rule $X \Rightarrow Y$ can be derived by $\{X \cup Y\}.sup/X.sup$. Thus, association rules can be constructed based on these rules with their global confidence greater than the minimum confidence level.

5.1. STTP-based public-key encryption solution – *Privacy-preserving distributed association rule mining via semi-trusted mixer* [56]

This protocol is performed in four steps: the first step is setup phase. During this phase, all users exchanged a secret key among themselves based on group key agreement protocol [71]. Details of key agreement protocols will not be demonstrated in this thesis. The second step is to find all global frequent sets of items on the basis of local frequent sets of items. In this phase, a priori algorithm [70] is utilized for sorting out all local frequent sets of item. During the third step, global support counts of all

frequent item sets are discovered. In the fourth step, rules are formulated out of global frequent sets of items above the minimum confidence threshold. These steps are operated one by one in a sequential order.

5.1.1. Notations

- n: # parties attending the joint computation
- U_i: the i^{th} user attending the joint computation
- DB_i: local dataset held by U_i
- P_i: the set of locally frequent items in DB_i
- S_{min}: global minimum support of candidate itemsets
- (E_k, D_k): secret key encryption (DES or AES encryption)
- K: secret encryption key
- (N, g): public key of Paillier public-key cryptosysem
- (p, q): private key of Paillier public-key cryptosystem; $N = p\ q$
- $l = lcm(p\text{-}1, q\text{-}1)$

5.1.2. Protocol

Step 1: Finding candidate items

Input: $P_1, P_2, \ldots, P_n$ $(n \geq 3)$, the minimum support is s_{min}, the encryption key is K

Output: $C_1 = \mathbf{U}_{i=1}^{n} P_i$

for U_i, $i = 1 \ldots n$ {
  $P_i = \emptyset$, $E_k(P_i) = \emptyset$
  for $j = 1 \ldots |I|$ {
    if $F_i(j) \geq s_{min} |DB_i|$ {
      $P_i = P_i \cup \{ij\}$, $E_k(p_i) = E_k(p_i) \cup \{E_k(ij)\}$
    }
      M $\leftarrow E_k(P_i)$
  }
}
$M1 = \emptyset$
for $i = 1 \ldots n$ {
  $M_1 = M_1 \cup E_k(P_i)$
}
for U_i, $i = 1 \ldots n$ {
  $C_1 = \emptyset$
  for each $X \in M_1$ {
    $C_1 = C_1 \cup \{D_k(X)\}$
  }
  Return $C_1 = \cup_{i=1}^{n} P_i$
}

Algorithm 4: Finding candidate items

Step 2: Finding the global support count of an itemset A

Inputs: $p_1, p_2, \ldots, p_n$ ($n \geq 3$), p_i is the local support count of the the itemset in DB_i, public key (N,g), $N = pq$, private key (p,q), or $l = lcm(p\text{-}1, q\text{-}1)$

Output: $F(A) = \sum_{i=1}^{n} p_i$

for U_i, $i = 1 \ldots n$ {

Randomly choose $r_i \in Z_N^*$,

$E_g(p_i) = g^{p_i} r_i^N \pmod{N^2}$

M $\leftarrow E_g(p_i)$

}

$M_2 = 1$

for $i = 1 \ldots n$ {

$M_2 = M_2 * E_g(p_i) \pmod{N^2}$

}

for U_i, $i \in (1,n)$ {

$$F(A) = \frac{(M_2^l \pmod{N^2}) - 1)/N}{(g^l \pmod{N^2}) - 1/N} \pmod{N}$$

Return $F(A) = \sum_{i=1}^{n} p_i$

}

Algorithm 5: Finding global support count of itemsets

5.1.3. Protocol Analysis: In protocol 1, each user has two communications with the mixer: (1) Each user U_i sends the encrypted candidate items (which are frequent in DB_i) to the mixer; (2) The mixer broadcasts the mixed encrypted candidate items, i.e. the union of encrypted candidate items from all users. In protocol 2, each user also has two communications with the mixer: (1) Each user U_i sends the encrypted local support count of an itemset in DB_i to the mixer; (2) The mixer broadcasts the mixed encrypted global support count of the itemset, i.e., the product of encrypted local support counts from all users.

Complexity analysis: In protocol 1, assume that the size of a ciphertext $E_k(a_{ij})$ (of a standard secret key cryptosystem) is l bits, then the communication cost of each user

U_i is $(|P_i| + |C_1|)l$ bits, and the total communication cost in protocol 1 is $(\sum_{i=1}^{n}|P_i| + |C_1|)l$ bits. The computation cost for each user U_i is $2|P_i|$ (secret key) encryptions plus $|C_1|$ (secret key) decryptions, and the computation cost for the mixer is $\sum_{i=1}^{n}|P_i|$ (secret key) decryptions plus set union. In protocol 2, assume that the size of N is L bits, i.e., $L = \log_2 N$, then the size of a cipher text in Paillier cryptosystem is $2L$ bits. In this case, the communication cost for each user U_i is $4L$ bits and the total communication cost of the mixer is $(2n+2)L$ bits. The computation cost for each user is one Paillier encryption, on Paillier decryption and $2L/l$ (secret key) encryptions, while the computation cost for the mixer is $2nL/l$ (secret key) decryptions and n-1 modular multiplications.

5.2. SSMC-based public-encryption solution – *Privacy-preserving Distributed Mining of Association Rules on Horizontally Partitioned Data* [21]

This protocol comprises two algorithms that run sequentially to form the whole distributed association rule mining protocols. The first sub-protocol, like the counterpart in the previous section, intends to find the global frequent itemsets. The second sub-protocol aims at obtaining the global support count of all frequent itemsets. The following sections will illustrate and analyze the protocol in more details.

5.2.1. Notations

- N: number of sites participating in the computation
- $LL_i(k)$: locally large itemset of the i^{th} site
- $LLe_i(k)$: encryption of locally large itemset of the i^{th} site
- RS: *RuleSet*, set of items and rules merged

- x_r: random integer chosen from a uniform distribution over 0…m-1
- m: $m \geq 2 * |DB|$
- f: randomly selected itemset from F
- $CG(k)$: the union of k locally large itemsets
- F: random itemsets

5.2.2. Protocol

Step 1: Finding secure union of large itemsets of size k

Input: N sites numbered 1, 2, …, N, $N \geq 3$, F is set of non-itemset
Output: globally large k itemsets $RS_{(k)}$
for site $I = 1 \ldots n$ **{**
  Generate $LL_i(k)$ as in steps 1 and 2 of FDM algorithm
  $LLe_i(k) = \emptyset$
  For each $X \in LL_i(k)$ **{**
    $LLe_i(k) = LLe_i(k) \cup \{E_i(X)\}$
  }
  For $j = |LLe_i(k)| + 1$ to $|CG(k)|$ **{**
    $LLe_i(k) = LLe_i(k) \cup \{E_i(f)\}$
  }
}
for $j = 0 \ldots N\text{-}1$ **{**
  if $j = 0$ **{**
    site $i \rightarrow LLe_i(k)$ to site $(i+1) \bmod N$
  }
}
Each site $I \rightarrow LLe_{i+1 \bmod N}$ to site i mod 2
site 0: $RS_1 \leftarrow \cup_{j=1}^{\lceil (N-1)/2 \rceil} LLe_{(2j-1)}(k)$
site 1: $RS_2 \leftarrow \cup_{j=0}^{\lceil (N-1)/2 \rceil} LLe_{2j}(k)$
site 1 $\rightarrow RS_1$ to site 0
site 0: $RS \leftarrow RS_0 \cup RS_1$
for $i = 0 \ldots N\text{-}1$ **{**
  Site i decrypts items in RS using D_i
  Site i sends permuted RS to site $i+1 \bmod N$
}
site N-1 decrypts items in RS using $D_{N\text{-}1}$
$RS_{(k)} = RS - F$
site N-1 $\rightarrow RS(k)$ to sites 0…N-2

Algorithm 6: Finding secure union of large itemsets

Step 2: Finding global support counts

Input: N sites numbered 1, 2, …, N, $m \geq 2*|DB|$
Output: all globally large itemsets
$rs = \emptyset$
At site 0:
for each $r \in$ *candidate_set* **{**
$t = r.sup_i - s * |DB_i| + x_r \pmod m$;
$rs = rs \cup \{(r,t)\}$;
}
Send rs to site 1;
for i = 1 to N-2 **{**
for each $(r,t) \in rs$ do
$\bar{t} = r.sup_i - s * |DB_i| + t \pmod m$;
$rs = rs - \{(r,t)\} \cup \{(r,\bar{t})\}$;
}
Send rs to site i+1;
}
At site N-1:
for each $(r,t) \in$ *rule_set* **{**
$\bar{t} = r.sup_i - s * |DB_i| + t \pmod m$;
if $(\bar{t} - x_r) \pmod m > 0$ **{**
Multi-cast r as a globally large itemset.
}
}

Algorithm 7: Securely finding global support counts

5.2.3. Protocol Analysis: In this protocol, the number of sites is N. Let the total number of locally large candidate itemsets be $|CG_{i(k)}|$, and the number of candidates that can be directly generated by the globally large (k-1) itemsets be $|CG_{(k)}|$. The excess support of an itemset X can be represented in $m = [log_2 (2*|DB|)]$ bits. Let t be the number of bits in the output of the encryption of an itemset. A lower bound on t is $\log_2 (|CG_{(k)}|)$; based on current encryption standards t = 512 is a more appropriate value.

Performance Analysis: The total communication cost for protocol 1 is $O(t^* |CG_{(k)}|*N^2)$ bits, and that of Protocol 2 is $O(m*|\cup_i LL_{i(k)}|*(N+t))$ bits. The computation cost in protocol 1 is $O(t^3*|CG_{(k)}|*N^2)$, where t is the number of bits in the encryption key. The

computation cost in protocol 2 is $O(t^3*|CG_{(k)}|*m)$ for the secure comparison at the end of the protocol.

5.3. Performance comparison

The following graphs are illustrations of the comparisons of the performance of these two protocols in terms of their communication cost and computation cost. Our experiment is done on the environment of a PC with a 1GHz processor and 512MB memory under NetBSD. The simulations of the protocols are implemented in the C#.NET programming language. The length of cryptographic key is 512 bits. The dataset we used for test is the Heart Disease Multivariate dataset consisting of 76 attributes and 293 instances. Due to the large amount of data, the full data set is not included in this thesis and can be referred to through the URL in [73]. We have performed two tests with the datasets. The performance is measured in the case of 3, 5, 7, 9 and 11 sites participating in the joint computation. The first test is to see how much communication overhead STTP-based protocol brings and how the communication overhead SSMC-based protocol compares with the former one. The total amounts of transmissions caused by the protocols with respect to the number of parties are depicted in Figure 12. The communication rounds of each protocol are displayed in Figure 13. As expected from the formula in Section 5.1.3 and Section 5.2.3, STTP-based protocol incurs a constant communication round, which is 2, while SSMC-based protocol incurs n rounds. The second test that we have performed is to analyze and compare the computational overheads brought by STTP-based protocol and SSMC-based protocol. Execution times of the protocols with respect to the number of parties are shown in Figure 14. The communication cost, communication

rounds and computation cost for both protocols are recorded, compared and presented below.

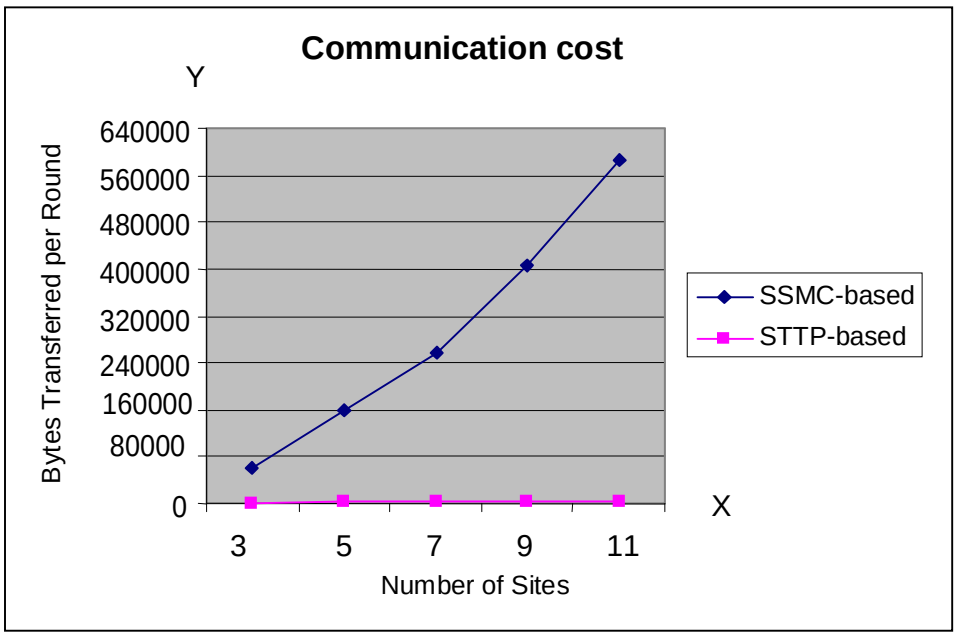


Figure 12: Communication cost comparison for association rules

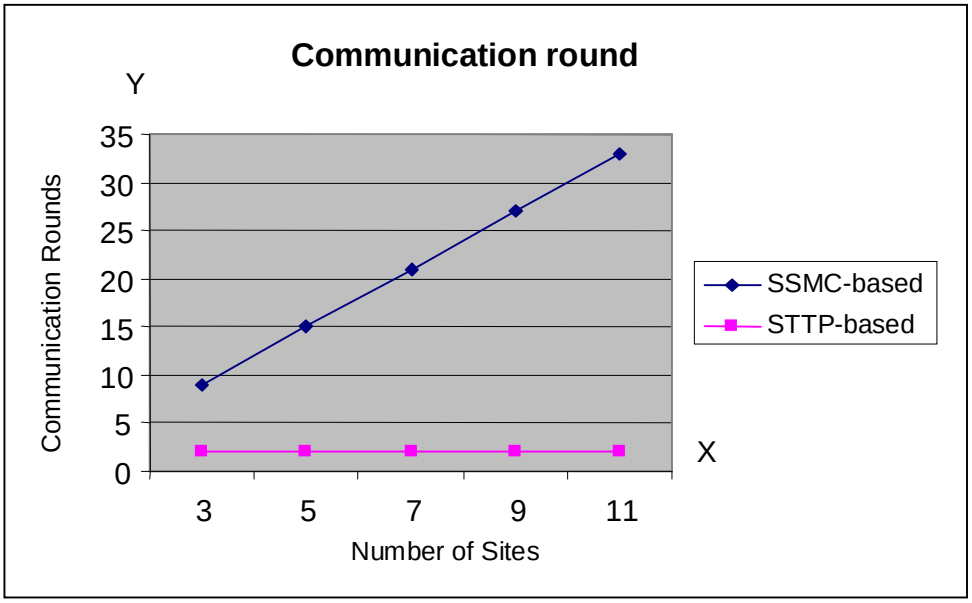


Figure 13: Communication round comparison for association rules

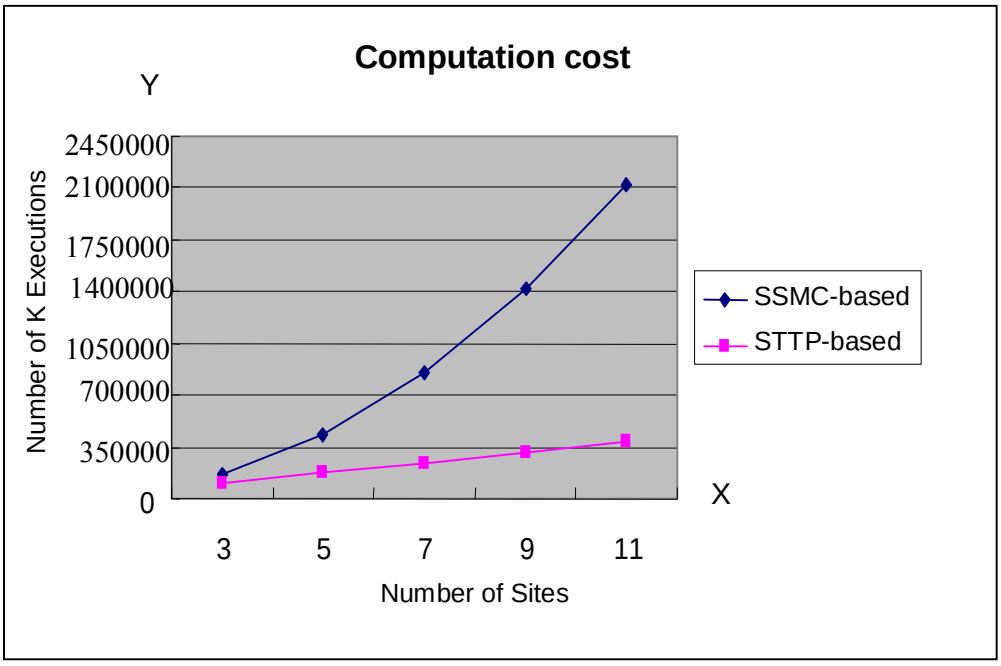


Figure 14: Computation cost comparison for association rules

From the information presented in the above graphs, we can clearly identify that the communication, computation cost and the communication rounds of the Xun Yi's STTP-based protocol are all lower than those of Clifton's SSMC-based protocol. Therefore, through the analysis and comparison based on our evaluation framework, we can conclude that Xun Yi's protocol dominates Clifton's protocol in the overall performance.

Chapter 6 Conclusion

The purpose of this master thesis is to organize some designing methods of PPDDM protocols, to classify privacy-preserving distributed data mining protocols by means of certain dimensions and to compare the performance of PPDDM protocols with a set of evaluation metrics. After identifying the classification scheme and relative performance of various PPDDM protocols, we are able to design more quality protocols that meet specific business needs with respect to the its privacy, accuracy and efficiency.

We will make some conclusions regarding the research questions in Section 1.6.

6.1 Design methods of PPDDM protocols

Most PPDDM protocols can be reduced to some sub-protocols. As we have seen in Section 2.4, there are three sub-protocols that serve as the key component in the design of PPDDM protocols. They are:

- secure sum protocol
- secure frequency mining protocol
- secure scalar product protocol

Common privacy-preserving distributed data mining problems can be solved by means computing the sum of the frequencies of data values of each attributes in each dataset or the scalar product of Boolean vectors that represent database transactions. In solving this kind of small components, we are able to design and develop secure, effective and efficient privacy-preserving data mining protocols.

6.2 Classification scheme of PPDDM protocols

As we have seen in Chapter 3, PPDDM protocols can be classified into mutually exclusive categories in terms of a set of classifications such as:

- Secure communication model
- Data partitioning model
- Data mining algorithms
- Privacy preservation techniques

Such a classification scheme can effectively cover all current PPDDM protocols and put each protocol into one of the categories. Each category represents a combination of different values in each dimension. As is in our case, 2*2*3*4 = 48 categories can be identified altogether. With any combination of two dimensions, we have surveyed the presence of protocols and drew a reference table describing them.

6.3 Evaluation of PPDDM protocols

We have conducted an evaluation work on various PPDDM protocols in terms of its performance complexity. A set of evaluation metrics - communication cost, computation cost, communication round and scalability have been set up. Our evaluation strategy is to calculate the overall overhead of the protocol measured in terms of number of bits exchanged. Each protocol in the predefined category has been represented in the form of a numerical figure with its performance. Based on that, comparison of the performance of protocols that fall into the same category is carried out by means of demonstrating in line chart.

Assessing the relative performance of PPDDM algorithms is a very difficult task, as it is often the case that no single algorithm outperforms others on all criteria. Also, for maximum flexibility, we rate that relative merit of individual module that

comprised by the PPDM algorithm. The rating is given in three different levels – high, medium and low. In Table 5, we are able to summarize the results and the general principles.

Elements	Complexity	Bytes exchanged	Communication round	Scalability
Secure Communication Model:				
STTP	Low	High	Low	High
SSMC	High	Low	High	Low
Data Mining Tasks:				
Classification	Low	N/A	N/A	High
Association Rule	Low	N/A	N/A	High
Clustering	High	N/A	N/A	Low
Privacy Preserving Technique:				
Homomorphic encryption	Low	Medium	N/A	High
Oblivious transfer	High	High	N/A	Low
Secret sharing	Medium	Low	N/A	High
Randomization	Low	Low	N/A	High

Table 5: Relative performance of PPDDM protocols

Chapter 7 Future work

Researches on privacy-preserving distributed data mining have gone through several stages and will continue to progress in the next few years. Issues such as, standardization of PPDDM protocols, secure multi-party computation approaches under malicious model and game-theoretical framework of PPDDM will be the hot spots in this research area.

Standardization issues in privacy-preserving distributed data mining cover a wide range of topics, including a common framework of PPDDM with respect to privacy definitions, principles, policies and requirements as well as more effective and precise evaluation metrics regarding efficiency, privacy and complexity of PPDDM algorithms.

Currently, most cryptographic solutions to PPDDM problems are constructed and analysed with the assumption of semi-honest model. However, in real world applications, the case of pure semi-honest scenario is rare. Most parties should be regarded as malicious users, that is, they can deliberately provide false information or corrupt the execution of the algorithm. Research work into this area has gained great momentum and requires further efforts to clarify.

Game theoretical approach is another emerging field that aims to tackle privacy-preserving distributed data mining problems. This kind of solution characterizes PPDDM problems by means of 'coopetative' models in social economic field. It defines the behaviour of the parties based on the assumption of rational selection, which maximize one's own utility rather than simply honest or malicious. This area is a very proposing one, the framework of which has been proposed, yet the solution and evaluation work is still open for further investigation.

Appendix

Synthetic Control Chart

Data Set Characteristics: Time-Series
Attribute Characteristics: Real
Number of Instances: 60

28.7812 34.4632 31.3381 31.2834 28.9207 33.7596
24.8923 25.7415 27.5532 32.8217 27.8789 31.5926
31.3987 30.6316 26.3983 24.2905 27.8613 28.5491
25.7743 30.5262 35.4209 25.6033 27.9734 25.2702
27.1798 29.2498 33.6928 25.6264 24.6555 28.9446
25.5067 29.7929 28.0765 34.4812 33.8345 27.6671
28.6989 29.2101 30.9291 34.6229 31.4138 28.4636
30.9493 34.3179 35.5674 34.8829 30.6691 35.2667
35.2538 34.6402 35.7584 28.5518 25.6518 29.6442
29.1734 31.5089 33.1944 35.6177 31.5892 35.1223
35.2623 35.6805 31.0851 30.2589 24.1366 27.0766
26.7115 24.0969 30.0213 29.9423 31.5461 33.7673
34.2296 32.8783 24.2457 26.9036 25.8677 34.0545
30.1451 28.2025 26.5217 26.3509 35.4694 33.4757
24.0543 33.0039 35.4925 28.5634 28.7661 32.5147
28.1944 30.7994 33.9014 26.7178 27.5378 28.0809
26.0377 28.3229 27.8169 26.0755 35.8312 26.7637
35.4815 26.8952 34.7511 36.0264 26.2202 30.3771
27.0997 31.5199 30.8038 28.5932 26.6983 32.5261
29.4965 28.1316 29.4459 31.4876 32.5261 34.8466
25.7946 33.9116 25.3323 24.4356 26.9818 25.3804
33.4655 30.7767 35.9042 35.4448 27.8881 24.0044
31.7006 31.3466 34.3405 33.6582 34.7008 29.0794
32.2613 27.3329 30.6942 27.3329 33.7185 30.3973
30.2137 33.5576 34.5969 35.8645 32.4596 30.8967
34.2127 25.6697 24.0059 31.6246 28.0684 25.6268
33.0777 32.9058 26.4093 25.7531 25.1916 24.8237
26.1229 27.5393 27.8286 32.7989 32.2584 33.4875
34.1522 28.7753 32.6374 28.1436 31.2739 26.1861
26.0528 26.6546 25.6327 30.1965 30.5483 34.0332
26.8518 35.4224 27.0112 24.2167 27.0174 33.6189
29.3166 26.3116 27.7893 28.1943 35.7062 31.5904
35.1227 25.1472 32.4178 29.6882 25.0212 34.6611
28.1731 28.4389 32.2253 33.9017 35.7554 34.0295
25.5897 35.4172 28.1573 24.0632 33.7019 30.8997
33.0274 30.1664 34.7195 26.2819 26.4512 27.5253
25.7733 30.8156 27.1798 31.8126 30.3624 34.5414

34.9666 27.9138 28.6666 29.0522 33.2348 31.0326
27.2146 25.2048 34.4779 30.5769 32.0237 24.368
29.9681 29.0023 30.4124 34.6798 26.3895 25.9224
34.4988 30.9151 31.0631 35.3471 35.4036 29.5457
32.5771 34.6846 33.8951 32.1611 30.2299 25.5001
31.5544 27.4875 31.6382 33.7431 30.5938 28.4246
30.9239 30.6985 26.9554 30.2883 29.2382 31.7165
33.6876 34.2003 35.4782 25.0712 28.2744 35.3555
27.4772 33.4699 33.2165 28.2722 32.1262 35.4932
32.8206 33.6873 26.9483 30.8083 28.2708 26.5647
33.1673 35.5424 35.1414 29.9838 29.1135 24.5773
28.4681 35.4415 25.6554 28.3725 25.7072 29.978
26.9161 30.7253 30.9511 29.6203 34.9325 35.0063
24.1182 26.5063 29.8101 24.7781 30.7051 30.0393
24.9684 34.7265 32.8871 28.9516 25.0455 29.6056
25.1186 25.9547 33.2881 29.5138 29.4147 24.4418
31.3965 29.0775 30.8773 25.9966 29.7929 24.6621
26.0285 26.5547 32.1637 34.9435 30.7851 28.2194
26.2081 32.9091 29.9049 35.2788 25.4861 31.5948
35.5825 28.1943 32.7483 28.2917 30.6059 25.1913
27.1566 25.8174 28.7136 31.3756 31.5648 30.2067
30.9518 30.6834 25.0323 35.2149 25.1843 35.2237
32.7251 34.6791 24.0879 32.3513 28.7592 33.6939

Bibliography

1. R. Agrawal, R. Srikant, Privacy-preserving data mining, in Proceedings of *ACM SIGMOD International Conference on Management of Data*, pp. 439-450, 2000.

2. A. Amirbekyan and V. Estivill-Castro. Privacy-Preserving k-NN for Small and Large Data Sets. In: *Proceedings of The Seventh IEEE International Conference*, pp. 699-704, Omaha NE. October 28-31, 2007.

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

4. C. Clifton. Privacy Preserving Distributed Data Mining. *13th European Conference on Machine Learning*, pp. 19-23, 2001.

5. C. Clifton, M.Kantarcioglu, X. Lin, J. Vaidya and M. Zhu. Tools for privacy preserving distributed data mining. SIGKDD Explorations, 4(2): pp. 28-34, Jan 2003.

6. M.C. Doganay, T.B. Pedersen, Y. Saygin, E. Savas and A. Levi. Distributed privacy preserving k-means clustering with additive secret sharing. In: Proceedings of the 2008 international workshop on Privacy and anonymity in information society, pp. 3-11. Nantes, France. 2008.

7. W. Du and Z. Zhan. Building decision tree classifier on private data. *Proceedings of the IEEE international conference on Privacy, security and data mining – Volume 14*, pp. 1-8, 2002.

8. C. Dwork and K. Nissim. Privacy-preserving data mining on vertically partitioned databases. In *Advances in Cryptology – Proceedings of CRYPTO 2004*, volume 3152 of Lecture Notes in Computer Science, Santa Barbara, California, August 2004.

9．F. Emekci, O. D. Sahin, D. Agrawal and A.E. Abbadi. Privacy preserving decision tree learning over multiple parties. *Data & Knowledge Engineering*, 63: pp. 348-361, 2007.

10．J. Frieser and L. Popelinsky. DIODA: Secure mining in horizontally partitioned data. In: *Proceedings of P&S Issues in DM ws. At ECML/PKDD*, pp. 12, 2004.

11．T. Fukazawa, J. Wang, T. Takata and M. Miyazaki. An Effective Distributed Privacy-Preserving Data Mining Algorithm. In: *Proceedings of the Fifth International Conference on Intelligent Data Engineering and Automated Learning*, pp. 320-325, 2004.

12．B. Gilburd, A. Schuster and R. Wolff. K-TTP: A New Privacy Model for Large-Scale Distributed Environments. *International Conference on Knowledge Discovery and Data Mining*, pp. 563-568, Seattle, Washington, USA, August 22-25, 2004.

13．B. Goethals, S. Laur, H. Lipmaa and T. Mielikainen. On Private Scalar Product Computation for Privacy-Preserving Data Mining. *Information Security and Cryptology—ICISC 2004: 7th International Conference*, pp. 104-120, Seoul, Korea, December 2-3, 2004: Revised Selected Papers, 2005.

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

15．A. Iliev and S. Smith. More efficient secure function evaluation using tiny trusted third parties. Technical report, Technical Report TR2005-551, Dartmouth College, Computer Science, Hanover, NH, USA, July 2005. http://www.cs.dartmouth.edu/reports/abstract/TR2005-551.

16. A.Inan, Y. Saygin, E. Savas, A.A. Hintoglu, A. Levi. Privacy preserving clustering on horizontally partitioned data. In: *Privacy Preserving Clustering on Horizontally Partitioned Data*, *IEEE Computer Society*, pp. 95, Los Alamitos. 2006.

17. G. Jagannathan, K. Pillaipakkamnatt and R. Wright. A New Privacy-Preserving Distributed k-Clustering Algorithm. *Proceedings of the Sixth SIAM International Conference on Data Mining*, pp. 492-496, 2006.

18. G. Jagannathan and R. Wright. Privacy-Preserving Distributed k-Means Clustering over Arbitrarily Partitioned Data. *Conference on Knowledge Discovery in Data*, pp. 593-599, Chicago, Illinois, USA, August 21-24, 2005.

19. M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. In: *The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'02)*, pp. 24-31, Madison, Wisconsin, June 2, 2002.

20. M. Kantarcioglu and J.Vaidya. Privacy preserving naïve Bayes classifier for horizontally partitioned data. In *IEEE ICDM Workshop on Privacy Preserving Data Mining*, Melbourne, FL, pp. 3-9, November 2003.

21. M. Kantarcioglu and C. Clifton. Privately computing a distributed k-nn classifier. *PKDD*, v. 3202, LNCS, pp. 279-290., 2004.

21. M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge and Data Engineering*, 16(9): pp. 1026-1037, 2004.

23．V. Kapoor, P. Poncelet, F. Trousset and M. Teisseire. Privacy preserving sequential pattern mining in distributed databases. *Proceedings of the 15th ACM international conference on Information and knowledge management*, pp. 758-767, 2006.

24．O. Kardes, R. Ryger, R. Wright and J. Feigenbaum. Implementing Privacy-Preserving Bayesian-Net Discovery for Vertically Partitioned Data. *Proceedings of the ICDM Workshop on Privacy and Security Aspects of Data Mining, Houston, TX*, 2005.

25．H. Kargupta, K. Liu, S. Datta, J. Ryan and K. Sivakumar. Homeland security and privacy sensitive data mining from multi-party distributed resources. *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on*, 2, 2003.

26．H. Kargupta, K. Liu and J. Ryan. Privacy Sensitive Distributed Data Mining from Multi-Party Data. *Proc. 1st NSF/NIJ Symp. Intelligence and Security Informatics*, pp. 336-342, Springer, 2003.

27．S.V. Kaya, T.B. Pedersen, E.Savas and Y. Saygin. Efficient privacy preserving distributed clustering based on secret sharing. In *LNAI 4819 PAKDD 2007*, pp. 280-291, Springer, 2007.

28．X. Lin, C. Clifton and M. Zhu. Privacy-preserving clustering with distributed EM mixture modeling. Knowledge and Information System, 8 (1): pp. 68-81, 2005.

29．S. Merugu and J. Ghosh. Privacy-preserving distributed clustering using generative models. *Data Mining*, *2003*. *ICDM 2003. Third IEEE International Conference on*, pp. 211-218, 2003.

30. S. Merugu and J. Ghosh. A privacy-sensitive approach to distributed clustering. *Pattern Recognition Letters*, 26 (4): pp. 399-410, 2005.

31. M. Naor and B. Pinkas. Computationally Secure Oblivious Transfer. *Journal of Cryptology* (2005) 18: pp. 1-35, 2005.

32. S. Oliveira and O. Zaiane. Toward Standardization in Privacy-Preserving Data Mining. *ACM SIGKDD 3rd Workshop on Data Mining Standards*, pp. 7-17, 2004.

33. W. Ouyang and Q. Huang. Privacy Preserving Association Rules Mining Based on Secure Two-Party Computation. *LECTURE NOTES IN CONTROL AND INFORMATION SCIENCES*, pp. 344-969, 2006.

34. W. Ouyang and Q. Huang. Privacy Preserving Sequential Pattern Mining Based on Secure Multi-party Computation. *Information Acquisition, 2006 IEEE International Conference on*, pp. 149-154, 2006.

35. C. Park and S. Chee. On Private Scalar Product Computation for Privacy-Preserving Data Mining. *The 7th Annual International Conference in Information Security and Cryptology (ICISC 2004)*, 3506: 104-120, 2004.

36. T. B. Pedersen, Y. Saygm and E. Savas. Secret sharing vs. Encryption-based Techniques For Privacy Preserving Data Mining. *Joint UNECE/Eurostat work session on statistical data confidentiality*, Manchester, United Kingdom, December 17-19, 2007.

37. A. Schuster, R. Wolff and B. Gilburd. Privacy-preserving association rule mining in large-scale distributed systems. *Cluster Computing and Grid, 2004. CCGrid 2004. IEEE International Symposium on*, pp. 411-418, 2004.

38. C. Su, F. Bao, J. Zhou, T. Takagi and K. Sakurai. Privacy-Preserving Two-Party K-Means Clustering via Secure Approximation. *Advanced Information Networking and*

Applications Workshops, 2007, AINAW'07. 21st International Conference on Volume 1, pp. 385-391, May 21-23, 2007.

39. C. Su and K. Sakurai. Secure Computation Over Distributed Databases. *IPSJ Journal*, Vol. 0, No. 0, 2005.

40. J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In: *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 639-644, Edmonton, Alberta, Canada, July 23-26, 2002. [Online]. Available: http://doi.acm.org/10.1145/775047.775142

41. J. Vaidya and C.Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In KDD 2003. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 206-215. ACM Press, New York, 2003.

42. J. Vaidya and C. Clifton. Privacy preserving naïve Bayes classifier on vertically partitioned data. In: *2004 SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, April, pp. 22-24, 2004.

43. J. Vaidya and C. Clifton. Privacy-Preserving Decision Trees over Vertically Partitioned Data. *Data and Applications Security XIX: 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, pp. 139-152, 2008.

44. J. Vaidya, Y. Michael Zhu and C.W. Clifton. Privacy Preserving Data Mining. Boston, MA: Springer Science + Business Media, Inc., 2006.

45. A. Veloso, W. Meira Jr, S. Parthasarathy and M. de Carvalho. Efficient, Accurate and Privacy-Preserving Data Mining for Frequent Itemsets in Distributed Databases. *Proceedings of the Brazilian Symposium on Databases*, pp. 281-292, 2003.

46. V.S. Verykios, E. Bertino, I.N. Fovino, L.P. Provenza, Y. Saygin and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *ACM SIGMOD Record*, 33 (1), 2004.

47. W. Wang, B. Deng and Z. Li. Application of Oblivious Transfer Protocol in Distributed Data Mining with Privacy-preserving. *Data, Privacy, and E-Commerce, 2007. ISDPE 2007. The First International Symposium on*, pp. 283-285, 2007.

48. R. Wright and Z. Yang. Privacy-preserving Bayesian network structure computation on distributed heterogeneous data. *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 713-718, 2004.

49. R. Wright, Z. Yang and S. Zhong. Distributed Data Mining Protocols for Privacy: A Review of some Recent Results. *MADNES* 2005, *LECTURE NOTES IN COMPUTER SCIENCE*, 4074, pp. 67-69, 2006.

50. X. Wu, C. Chu, Y. Wang, F. Liu and D. Yue. Privacy Preserving Data Mining Research: Current Status and Key Issues. *LECTURE NOTES IN COMPUTER SCIENCE*, 4489: pp. 762, 2007.

51. L. Xiong, S. Chitti and L. Liu. k Nearest Neighbor Classification across Multiple Private Databases. *CIKM'06*, pp. 840-841, Arlington, Virginia, USA, November 5-11, 2006

52. L. Xiong, S. Chitti and L. Liu. Mining Multiple Private Databases Using a k*NN* classifier. *SAC'07*, pp. 435-440, Seoul, Korea, March 11-15, 2007.

53．Z. Yang and R. Wright. Privacy-Preserving Computation of Bayesian Networks on Vertically Partitioned Data. *IEEE Transactions on Data Knowledge Engineering*, 18(9): pp. 1253-1264, April 2006.

54．Z. Yang, S. Zhong, 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.

55. A.C. Yao. Protocols for Secure Computations. In Proceedings: *23rd IEEE Symposium, on Foundations of Computer Science*, pp. 160-164, Chicago, 1982.

56．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.

57．X. Yi and Y. Zhang. Privacy-preserving naïve Bayes classification on distributed data via semi-trusted mixers. *Information Systems*, Volume 34, Issue 3, pp. 371-380, May 2009.

58．H. Yu, X. Jiang and J. Vaidya. Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data. *Proceedings of the 2006 ACM symposium on Applied computing*, pp. 603-610, 2006.

59．H. Yu, J. Vaidya and X. Jiang. Privacy preserving svm classification on vertically partitioned data. *Advances in Knowledge Discovery and Data Mining*, Volume 3918, pp. 647-656, 2006.

60．J. Zhan and L. Chang. Privacy-preserving collaborative data mining. *Foundations and Novel Approaches in Data Mining*, Volume 9, pp. 213-227, Springer, 2006.

61．J. Zhan, L. Chang and S. Matwin. Privacy Preserving K-nearest Neighbor Classification. *International Journal of Network Security*, 1 (1): pp. 46-51, 2005.

62．J. Zhan, L. Chang, S. Matwin, et al. Privacy-Preserving Collaborative Sequential Pattern Mining. *Workshop on Link Analysis, Counter-terrorism, and Privacy in conjunction with SIAM Int. Conf. on Data Mining*, pp. 61-72, 2004.

63．J. Zhan, S. Matwin and L. Chang. Privacy-Preserving Collaborative Association Rule Mining. *Journal of Network and Computer Application*, Volume 30, Issue 3, pp. 1216-1227, August 2007.

64. D.E. O'Leary. Knowledge Discovery as a Threat to Database Security. In G. Piatetsky-Shapiro and W.J. Frawley (editors): *Knowledge Discovery in Databases*. AAAI/MIT Press, pp. 507-516. Menlo Park, CA, 1991.

65. D.E. O'Leary. Some Privacy Issues in Knowledge Discovery: The OECD Personal Privacy Guidelines. *IEEE EXPERT*, 10(2): pp. 48-52, April 1995.

66. G. Piatetsky-Shapiro. Knowledge Discovery in Personal Data vs. Privacy: A Mini-Symposium. *IEEE EXPERT*, 10(2): pp. 46-47, 1995.

67. U.M. Fayyad, G. Piatetsky-Shapiro and P. Smyth. From Data Mining to Knowledge Discovery: An Overview. In *Advances in Knowledge Discovery and Data Mining*. MIT Press, pp. 1-34. Cambridge, MA, 1996.

68. Wikipedia – Public-key Cryptosystem
http://en.wikipedia.org/wiki/Public-key_cryptography

69. Wikipedia – Oblivious transfer
http://en.wikipedia.org/wiki/Oblivious_transfer

70. Wikipedia – A priori algorithm
http://en.wikipedia.org/wiki/Apriori_algorithm

71. Wikipedia – Key agreement protocol
http://en.wikipedia.org/wiki/Key-agreement_protocol

72. Wikipedia – Shamir secret sharing
http://en.wikipedia.org/wiki/Shamir's_Secret_Sharing

73. http://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/cleveland.data