A Novel Approach For Privacy Mining Of Generic Basic Association Rules

Waddey Moez, Pascal Poncelet, Sadok Ben Yahia

A Novel Approach For Privacy Mining Of Generic Basic Association Rules

Moez Waddey
Département des Sciences de l'Informatique
Faculté des Sciences de Tunis
Campus Universitaire 1060
Tunis, Tunisia
waddey.moez@gmx.com

Pascal Poncelet
LIRMM
Université Montpellier 2
161 rue Ada
34392 - Cedes 5
Montpellier, France
poncelet@lirmm.fr

Sadok Ben Yahia
Département des Sciences de l'Informatique
Faculté des Sciences de Tunis
Campus Universitaire 1060
Tunis, Tunisia
sadok.benyahia@fst.rnu.tn

ABSTRACT

Data mining can extract important knowledge from large data collections - but sometimes these collections are split among various parties. Privacy concerns may prevent the parties from directly sharing the data. The irony is that data mining *results* rarely violate privacy. The objective of data mining is to generalize across populations rather than reveal information about individuals [10]. Thus, the *true* problem is not data mining, but how data mining is *done*. This paper presents a new scalable algorithm for discovering closed frequent itemsets in distributed environment, using commutative encryption to ensure privacy concerns. We address secure mining of association rules over horizontally partitioned data.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: General; H.2.8 [**Database Management**]: Database applications—*privacy preserving distributed data mining*

General Terms

Algorithms Security

Keywords

Data mining, Association rules mining, Privacy preserving, Commutative encryption

1. INTRODUCTION AND MOTIVATIONS

One of the most studied problems in data mining is the process of discovering frequent itemsets and, consequently, association rules. Discovering hidden patterns from large amounts of data plays an important role in marketing, business, medical analysis, intrusion detection, and other applications where these patterns are of paramount important for strategic decision making [19].

PAVLAD'09, November 6, 2009, Hong Kong, China.

Despite its benefits in various areas, extracted knowledge can also present a threat to privacy and information security if not carried out or used properly. Data mining and data warehousing go hand-in-hand: most tools operate by gathering all data into a central site, then running an algorithm against that data. However, privacy concerns can prevent building a centralized warehouse-data may be distributed among several sites (or parties), none of which are allowed to transfer their data to another site [10]. These recent advances in data mining and machine learning algorithms have introduced a new problem in database security [4, 11]. A Distributed Data Mining (DDM) model assumes that data sources are distributed across multiple sites. The challenge here is: how can we mine the data across the distributed sources securely or without either party disclosing its data to others? In the remainder, we assume homogeneous databases i.e., all sites have the same scheme, but each site stores information on different entities. The goal is to produce association rules with every party's data transactions globally, while preventing the private information to be known by other parties.

The definition of privacy, followed in this line of research, is conceptually simple (as defined in [10] and [9]): *no site should learn anything new from the process of data mining.* Specifically, anything learned during the data mining process must be derivable given one's own data and the final result. In other words, nothing is learned about any other site's data that isn't inherently obvious from the data mining result.

There are several works on privacy preserving association rule extraction, which is interested in securing the mining task. However, the main complaint that can be addressed stands on the fact they generate redundant association rules. Thus, the latter requires much computational effort as well an important communication cost between parties. In this paper, we introduce an approach that advocates to use of a condensed representation for the itemsets during the mining task and a condensed set to represent the association rules that will be generated and develop a communication protocol while fulfilling privacy requirements [9]. The remainder of the paper is organized as follows: In section 2, we first detail the basic notions for frequent closed itemset, as well as the usefulness of the condensed representation adopted in the mining task, and present the condensed representation using benefits of generated association rules. Then, we describe related work on privacy preserving data

mining. Then, describe the work which we have based our communication protocol. Section 3 describes our approach, which can generate the generic base of association rules from the frequent closed set in a distributed environment while preserving the constraints of privacy by using a commutative cryptographic protocol communication that we have designed. Section 4 concludes this paper and points out future perspectives.

2. BACKGROUND AND RELATED WORK

There are several fields where related work is occurring. We first introduce basic definitions for association rule mining. Then, we describe related work in privacy-preserving data mining. After that, we go into specific background work on which this paper relies.

2.1 Association Rule Mining

The problem of association rule mining was initially presented in [2]. The authors in [3] extended and formalized the problem as follows: Let $\mathcal{I} = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called *items*. Let $\mathcal{D}$ be a database of transactions, where each transaction $\mathcal{T}$ is an itemset such that $\mathcal{T} \subseteq \mathcal{I}$. Associated with each transaction is a unique identifier, called its TID. A set of items $X \subset \mathcal{I}$ is called an *itemset*. A transaction $\mathcal{T}$ contains an itemset X, if $X \subseteq \mathcal{T}$. An *association rule* is an implication of the form $X \Rightarrow Y$ where $X \subset \mathcal{I}$, $Y \subset \mathcal{I}$, and $X \cap Y = \varnothing$. The rule $X \Rightarrow Y$ holds in the set of transactions Y with confidence c if $\frac{|X \cup Y|}{|X|} \geq c$ where $|A|$ is the number of occurrences of the set of items A in the set of transactions $\mathcal{D}$. The rule $X \Rightarrow Y$ has support s if $\frac{|X \cup Y|}{\mathcal{N}} \geq s$ where $\mathcal{N}$ is the cardinality of the transaction set $\mathcal{D}$.

Definition 1. (Formal Context) A formal context (or an extraction context) is a triplet $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$, where $\mathcal{O}$ represents a finite set of objects, $\mathcal{I}$ is a finite set of items and $\mathcal{R}$ is a binary (incidence) relation (*i.e.*, $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{I}$). Each couple $(o, i) \in \mathcal{R}$ expresses that the object $o \in \mathcal{O}$ contains the item $i \in \mathcal{I}$.

Definition 2. (Closure Operator) Let $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$ be a data mining context, $\mathcal{O}$ a set of transactions, $\mathcal{I}$ a set of items, and $\mathcal{R}$ a binary relation between transactions and items. For $O \subseteq \mathcal{O}$ and $I \subseteq \mathcal{I}$, we define :

$$f(O) = \{i \in I \mid \forall o \in O, (o, i) \in \mathcal{K}\}$$
$$g(I) = \{o \in O \mid \forall i \in I, (o, i) \in \mathcal{K}\}$$

$f(O)$ associates with O, items common to all transactions $o \in O$, and $g(I)$ associates with I, transactions related to all items $i \in I$. The operators $\gamma = f \circ g$ and $\gamma' = g \circ f$ are the Galois closure operators.

The closure operator γ induces an equivalence relation on the power set of items portioning it into disjoint subsets called *equivalence classes*. The largest element (*w.r.t.* the number of items) in each equivalence class is called a *closed itemset* and is defined as follows:

Definition 3. (Closed Itemset) An itemset $I \subseteq \mathcal{I}$ is said to be closed if and only if $\gamma(I) = I$ [18]. The support of I, denoted by $Supp(I)$, is equal to the number of objects in $\mathcal{K}$ that contain I. I is said to be *frequent* if $Supp(I)$ is greater than or equal to a user-specified minimum support threshold, denoted *Minsup*. The frequency of I in $\mathcal{K}$ is equal to $\frac{Supp(I)}{|\mathcal{O}|}$.

Definition 4. (Minimal Generator) An itemset $g \subseteq \mathcal{I}$ is said to be a minimal generator of a closed itemset f , if and only if $\gamma(g) = f$ and $\nexists\ g_1 \subset g\ s.t. \gamma(g_1) = f$ [6]. Thus, the set MG_f of the minimal generators associated to a closed itemset f is:

$$MG_f = \{g \subseteq \mathcal{I} | \gamma(g) = f \text{ and } \nexists g_1 \subset g\ s.t.\ \gamma(g_1) = f\}$$

The study of the extraction of closed itemsets [6] grasped the interest of the data mining community. Indeed, *frequent closed itemset* (FCI) based algorithms were introduced to mainly tackle two complementary problems. On the one hand, FCI-based algorithms present an effective mining approach for dense extraction contexts. In such contexts, large equivalence classes are obtained. FCIs, standing on the top of the hierarchy induced by each equivalence class, allow to informatively infer the supports of FIs. On the other hand, FCI-based algorithms, which heavily draw on Formal Concept Analysis (FCA) mathematical settings [6][22], present a novel alternative with a clear promise to dramatically reduce, without information loss, the size of the association rule set that can be drawn from both synthetic and real-life datasets. The result of such a reduction is a reasonably-sized subset of association rules that can be seen as an irreducible nucleus of association rules, commonly known as "generic basis" of association rules [17]. A structural survey of FCI-based algorithms is presented in [6]. Therefore, the problem of mining association rules might be reformulated, under the point of view of the FCI-based algorithms, as follows:

1. Discover both distinct "closure systems", *i.e.*, sets of sets which are closed under the intersection operator, namely the FCI set and the FMG set. Also, the upper cover of each FCI should be available.

2. From the information discovered in the first step, *i.e.*, both closure systems and the upper covers, derive generic bases of association rules (from which all remaining rules can be derived).

The CLOSE algorithm [16], is an algorithm of extraction of frequent closed itemset that sweeps the extraction context in a level-wise manner. The set of candidate closed itemsets of an iteration k is the set of closed k-generators of this iteration. The set of 1-generators is initialized with the list of the 1-itemsets in the context during the first iteration 1, as shown algorithm 1.

FF_k	set of k-groups of frequent k-generators. each element of this set has three fields: *generator*, *closure* and *support*.
FFC_k	set of k-groups of candidate k-generators. each element of this set has three fields: *generator*, *closure* and *support*.

Table 1: Notations used in Algorithm 1

Algorithm 1 *CLOSE*: Extraction of Frequent Closed Itemsets

Input: Context $\mathcal{K}$; minimal support: $Minsupp$;
Output: FF_K : units of frequents $K-groups$;
begin
  $FFC_1.generators \leftarrow \{1-itemsets\}$;
  for $(k \leftarrow 1; FFC_K.generators \neq \varnothing; k++)$ **do**
    Gen-Closure(FFC_K);
    foreach *group candidate* $c \in FFC_K$ **do**
      if *(c.support* $\geq Minsupp$*)* **then**
        $FF_k \leftarrow FF_k \cup \{c\}$;
    $FFC_{K+1} \leftarrow$ Gen-Generator(FF_k);
  Result: $\bigcup_K FF_k$;
end

The Gen-Closure procedure receives a *Frequent Closed Candidates* FFC_k unit of candidates k-groups containing the k-generators candidates of the iteration k in argument. It determines the closure of each generator, stored in the *closure* field, and the support stored in the field *support*.

The Gen-Generator procedure receives a FF_k unit of the frequent k-groups in parameter. It turns over FFC_{k+1} of $(k+1)$-candidates groups containing the $(k+1)$-generator which will be used during the iteration $k+1$. The Gen-Generator procedure consists of three phases, all the potentials $(k+1)$-generator are created by using the k-generators in FF_k. The Second and third phases make it possible to remove among these generator those which one knows that the calculation of closure is useless. The second phase removes the potential generators infrequent potential generators and those which are not minimal. The third phase removes among those which the closure is already calculated.

Definition 5. (Generic Basis for Exact Association Rules) Let $\mathcal{FCI}$ be the set of frequent closed itemsets extracted from the context and, for each frequent closed itemset c, let us denote $\mathcal{G}_c$ the set of minimal generators of c. The generic basis for exact association rules, called $\mathcal{GBE}$, is defined as follows:

$$\mathcal{GBE} = \{\mathcal{R} : g \Rightarrow (c-g) \mid c \in \mathcal{FCI} \text{ and } g \in \mathcal{G}_c \text{ and } g \neq c\}$$

2.2 Related Work

Previous work in privacy-preserving data mining has mainly addressed two issues. Within the first issue, the aim is to hide sensitive extracted association rules by distorting or transforming the data. The idea is to limit disclosure of sensitive rules [5]. In particular, attempt to selectively hide some frequent itemsets from databases. The problem here can be stated as follows. Let $\mathcal{D}$ the source database, let $\mathcal{R}$ be a set of significant association rules that are mined from $\mathcal{D}$, and let $\mathcal{R}_\mathcal{H}$ be a set of rules in $\mathcal{R}$ that should be hidden. The issue here is how to transform $\mathcal{D}$ into a database $\mathcal{D}'$ so that all (or maximum number of) rules in $\mathcal{R}$ can still be mined from $\mathcal{D}'$ but for the rules in $\mathcal{R}_\mathcal{H}$. $\mathcal{D}'$ becomes the *released database*. Therefore, this idea is to reduce the support of the rules in $\mathcal{R}_\mathcal{H}$ below the given threshold. There are two kinds of techniques of the hiding algorithm presented in literature, namely *heuristic* and *exact*. Heuristic techniques rely on the optimization of certain sub-goals during the hiding process, while they do not guarantee optimality. The authors in [5] prove that the exact solution is NP-hard.

A number of cryptography-based approaches have been developed in the context of privacy preserving data mining algorithms, to solve problems of the following nature: Two or more parties are interested in conducting a computation based on their private inputs, but neither party is willing to disclose its own output to anybody else. The challenge here is how to carry out such a computation while preserving the privacy of the inputs. This problem is referred to as the *Secure Multiparty Computation* (SMC) problem [14]. Thus, an SMC problem deals with computing a probabilistic function on any input, in a distributed network where each participant holds one of the inputs, ensuring independence of the inputs, correctness of the computation, and that no more information is revealed to a participant in the computation than that's participant's input and output. There have been some cryptography based algorithms as well. Lindell and Pinkas [14] first introduced a SMC technique for classification using the ID3 algorithm, over horizontally partitioned data. Zhan et al. [8] proposed a cryptographic protocol for making the ID3 algorithm privacy preserving over vertically partitioned data. Agrawal et al. [1] presented a technique for computing set intersection, union, and equi-joins for two parties. Lin et al. [13] proposed a secure manner for clustering task using the EM algorithm over horizontally partitioned data. Clifton et al. described protocols for privacy preserving distributed data mining of association rules on horizontally partitioned data [10]. In this paper, we put the focus on the work related to privacy preserving distributed mining of association rules. We will mainly focus on Clifton et al. work [10] since it is highly connected to ours. Unfortunately, this work suffers from a major limitation in terms of performances, since a process of encryption/decryption (based on a commutative encryption protocol) is launched for each rule generated by any site. In the case of real life applications, this constraint is too restrictive and it is of paramount importance to propose a new approach that tackles this issue. In this context that our work has been born, we must use a concise exact representation to circumvent this weakness. Moreover, most work (all work to our knowledge) use Apriori presented by Agrawal et al. [2] as frequent local itemset generation algorithm, or a solution based on Apriori. And that, this algorithm despite its simplicity, several works present in literature have proved most effective solutions both in terms of execution time and in term of spatial complexity.

2.3 Background

In this section, we present basic definitions that will be of use in the remainder.

2.3.1 Secure Multi-party Computation

Substantial work has been done on secure multi-party computation. The key result is that a wide class of computations can be computed securely under reasonable assumptions. We give a brief overview of this work, concentrating on material that is used later in the remainder. The definitions given here are from Clifton et al. [10].

Definition 6. (Semi-Honest Model)A fulfilling the semi-honest model follows the rules of the protocol using its correct input, but is free to later use what it sees during execution of the protocol to compromise security.

This is somewhat realistic in the real world since parties interested in mining data for their mutual benefit will follow the protocol to get correct results. Also, a protocol that is buried in large, complex software can not be easily altered.

A computation is said to be secure wherever the view of each party during the execution of the protocol can be effectively simulated by the input and the output of the party. This is not quite the same as saying that private information is protected. For example, if three parties use a secure protocol to mine frequent (closed) itemsets. A secure protocol still reveals that if a particular item is not supported by an other sites, even if this item appears in the globally supported frequent closed itemsets, then if this item is supported only by a single site. A site can deduce this information by solely looking at its locally supported itemsets and the globally supported itemsets. On the other hand, there is no way to deduce the exact support value of any item of any site, by looking at the globally supported frequent closed itemsets.

With three or more parties, knowing that an item holds globally do not reveals that at least one site supports it, but no site knows which site supportes it (other that, obviously, itself). In summary, a secure multi-party protocol will not reveal more information to a particular party than the information that can be induced by looking at that party's input and the output. There exists a vast body of literature on secure multi-party computation. The respective papers by Yao [24, 25] build foundations for general secure multi-party computation.

2.3.2 Commutative Encryption

Commutative encryption is an important tool that can be used in many privacy-preserving protocols. Commutative encryption is the component of cryptography that deals with secure multi-party computations, which provides effective practice schemes for distributing functions. An encryption algorithm is commutative if the following equations, (1) and (2), hold for any given feasible encryption keys $K_1, \ldots, K_n \in \mathcal{K}$, any m in items domain $\mathcal{M}$, and any permutation of i, j.

$$E_{K_{i_1}}(\ldots E_{K_{i_n}}(M)\ldots) = E_{K_{j_1}}(\ldots E_{K_{j_n}}(M)\ldots) \quad (1)$$

$$\forall M_1, M_2 \in \mathcal{M} \text{ such that } M_1 \neq M_2 \text{ and for given } k, \epsilon < \frac{1}{2^k}$$
$$\Pr[E_{K_{i_1}}(\ldots E_{K_{i_n}}(M_1)\ldots) = E_{K_{i_1}}(\ldots E_{K_{i_n}}(M_2)\ldots)] < \epsilon \quad (2)$$

The work in [12] introduces a secret sharing scheme that is inspired by Shamir's keyless secret communication [15], which relies on the commutative property of modular exponentiation. The idea, however, works with any commutative encryption function fulfilling certain conditions. Shamir in [20] explored the power of commutativity in cryptography. Works, respectively presented in [1] and [7], use commutative cryptography for information sharing across databases, and for privacy in distributed data mining. The topic has received much recent fundamental consideration in [21].

Definition 7. (Commutative Encryption Scheme) Let $\mathcal{M}$ be denoting a message space and $\mathcal{K}$ denoting a key space. A commutative encryption function is a family of bijections $f : \mathcal{M} \times \mathcal{K} \longrightarrow \mathcal{M}$ such that for a given $m \in \mathcal{M}$ we have $f_a \circ f_b(m) = f_b \circ f_a(m)$, for any $a, b \in \mathcal{K}$.

It is easy to show through proposition that modular exponentiation under certain conditions is a commutative encryption function.

Proposition 1. *Choose a prime p. Define $f_a(m) \stackrel{Def}{=} m^a (mod\ p)$. Let $m \in \mathbb{Z}_p$, $a \in \mathbb{Z}_{p-1}$, such that $\gcd(a,\ p-1) = 1$. Then we have:*

- *$f_a(m)$ is a bijection*
- *$f_a(m)$ is commutative*

Proof.
- **$f_a(m)$ is a bijection**. There exists $b \in \mathbb{Z}_{p-1}$ such that $f_b \circ f_a(m) = m$. This is easily seen as follows: given that $a \in \mathbb{Z}_{p-1}$ and $\gcd(a, p-1) = 1$, we can find $b \in \mathbb{Z}_{p-1}$ (using the extended Euclid algorithm) such that $ab = 1 (mod\ p-1)$.

  As a result:

  $$f_b \circ f_a(m) = m^{ab} (mod\ p) = m^{1+k(p-1)} (mod\ p) = m$$

  where we have used the Fermat's theorem, (*i.e.*, $m \in \mathbb{Z}_p$, *then* $m^{(p-1)} = 1\ (mod\ p)$).
- **$f_a(m)$ commutative**. For all positive integers a, b, m, we have $f_a \circ f_b(m) = m^{ab} (mod\ p) = f_b \circ f_a(m)$.

□

The above property is the key step towards the proof of the correctness of a keyless secret sharing scheme that've we in our contribution. The proposed algorithm in Khayat's paper [12] uses the assumption that a large prime p has been made publicly known to all (even adversaries). The prime can be made part of a technical standard for this type of applications and then be published to the world. As a result, the prime p is assumed to be accessible to everyone in an authentificated way.

3. OUR CONTRIBUTION

3.1 Problem Definition

Let $i \geq 3$ be the number of sites. Each site has a private transaction database DB_i. We are given support threshold s. The goal is to discover the generic base of exact association rules, as defined in Section 2.1. No site should be able to learn contents of a transaction at any other site, what items are supported by any other site, or the specific value of support for any items at any other site, unless that information is revealed by knowledge of one's own data and the final result. Furthermore, we are interested in using some cryptographic toolkits to construct a secure multi-party computation protocol to perform this task.

3.2 Global Architecture

Our method follows the general principle presented in algorithms that generate the frequent closed itemset such as CLOSE (presented in section 2.1) in a distributed environment. In our method, a communication protocol will be proposed which ensure the constraints described above in a semi-honest model. We take the commutative encryption protocol as a starting point for our approach. Indeed, the general principle of our approach will be to communicate intermediate results by preserving the security and anonymity with the main site which is given the responsibility to generate the candidates of higher size. At this point, the problem

of mining association rules might be reformulated, under the point of view of the FCI-based algorithms (see section 2.1)

In the following, we describe a new secure architecture associated with our algorithm which takes avantage of commutative encryption as well as work on extraction of generic base from a condensed representation of data, i.e., *closure* in this case. This architecture offers the advantage of being able to carry out the various tasks of the extraction of itemsets while guaranteeing security and anonymity (i.e., no site can have access to private information of the various bases)

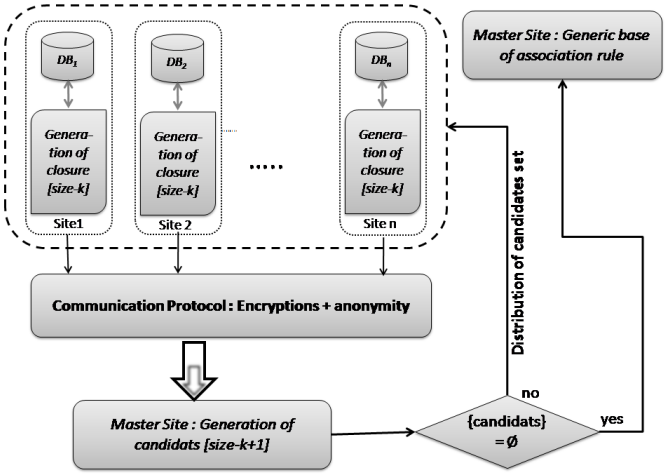

Figure 1: Global Architecture

In the following, we describe the global architecture of our approach as presented by figure 1. Figure 1 represents the global architecture of our solution. Given an algorithm that runs locally which generates *support* and *closure* for a set of candidates, from a context $\mathcal{K}$ and a set of candidate for given size as inputs.

Initially, the initialization process of commutative cryptography protocol will be invoked. Then, the master site,*i.e.*, the site which launches the mining task, distributes the list of 1-itemsets candidates. Therefore, the different sites run, concurrently, a local algorithm described below, which generates their *closure* and *support*. At this step, the commutative encryption protocol is lunched in order to communicate the results to the master site. Now, the master site has at hand the set of the local *closures* as well as the local *supports* of the candidates items. The master site can now generate the global *support* by making the sum of the local *supports*. The global *closure* is calculated by making the intersection of the local *closures*. In this way, the master site can generate the candidates of higher size in the same way as do the $CLOSE$ algorithm briefly presented in section 2.1. Then, at this level, the master site repeats the above steps whenever it can generate candidates of higher size.

The final step of the proposed solution is the generation of the generic base of exact association rules. The master site can performs this task efficiently, since it has already the set of all frequent closed itemset, and their minimal generators. The generation of the generic basis of exact association rules from the frequent closed and minimal generators is treated in previous work [23, 16].

During the first iteration of the algorithm, the set of generators of FFC_1 is initialized with the 1-itemsets of the context $\mathcal{K}$, i.e., all elements of $\mathcal{I}$. Since the context is distributed horizontally, thus all sites share the same attributes.

FF_K	The set of frequent closed itemset of k-size.
FFC_k	The set of frequent closed itemset candidates of k-size.
FFC_K^L	The set of local frequent closed itemset candidates k-size.
FFC_k^G	The set of global frequent closed itemset candidates k-size.

Table 2: Notations used in Algorithm 2

Algorithm 2 Distributed Extraction of Frequent Closed Itemsets with Privacy Preserving

Input: n: number of sites;
$\mathcal{K}$: Extraction context;
$Minsupp$: minimal threshold of support;
$master$: boolean flag : it is set to $true$ if the current site is the master one, otherwise it is set to $false$;

begin
  Initialize(n);
  if $master$ **then**
    $FFC_1.generators \leftarrow \{1 - itemsets\}$;
  for $(k \leftarrow 1; FFC_K.generators \neq \varnothing; k++)$ **do**
    if $master$ **then**
      Distribute(FFC_K, n);
    Receive(FFC_K);
    $FFC_K^L \leftarrow$ Gen-Local(FFC_K);
    CommutativeEncryptionProtocol(FFC_K^L);
    if $master$ **then**
      $FFC_K^G \leftarrow$ Collect(FFC_K^L);
      $FF_K \leftarrow$ Gen-global(FFC_K^G));
      $FFC_{K+1} \leftarrow$ Gen-Generator(FF_k);
  Result: $\bigcup_K FF_k$
end

The Gen-Local procedure receives a *Frequent Closed Candidates* FFC_k unit of candidates k-groups containing the k-generators candidates of the iteration k in argument. It calculates the local *support* and *closure* of each generator. This procedure is running on all parties. The *Commutative Encryption Protocol* procedure receives the set of candidates with their *closures* and *supports*. Therefore, the commutative encryption protocol is executed in order to transfer the results to the master site while ensuring privacy preserving. The *Gen-Global* procedure receives the set of FFC_K^L, obtained by executing the protocol of communication, and generate the global *support* by making a sum of local *supports* as well as, the global *closure* by making the intersection between the local *closures* received previously. Then, the master site can pruned the infrequent itemsets given *minsupp*. At this step, the master site execute the *Gen-Generator* procedure in order to generate the candidates of size $k+1$, it returns the set of this candidates. This

process will be repeated until the *Gen-Generator* procedure generate an empty set.

As a final step, the master site executes a procedure in order to generate the generic base of exact association rules.

3.3 The Communication Protocol

In the following, we present algorithm of commutative encryption protocol whose pseudo-code is presented by algorithm 3. This algorithm is inspired from work in [12, 21].

Algorithm 3 Commutative Encryption Protocol: privately collect messages from parties

Input: n sites; n message m_i owned by sites $P_i, (_{0\leq i\leq n-1})$; P_0 : Master Site

Output: P_0[master site] collects all messages with guaranteeing the security and anonymity

begin

Initializing :

p : a large public prime;

m_i : secret owned by the site $P_i (m \in \mathbb{Z}_p)$; $i \in \{0, 1, 2, \ldots, n-1\}$;

$\forall P_i \in \{P_0, P_1, \ldots, P_{n-1}\}$, P_i has (a_i, b_i) as private key;

$a_i, b_i \in \mathbb{Z}_p$, $\gcd(a_i, p-1) = 1$; $a_i b_i = 1 \ mod(p-1)$;

Locking :

P_i locks the secret by applying $c_i = m_i^{a_i} mod(p)$;

P_i sends c_i to P_{i+1} /* $mod(n)$ */;

[P_{n-1} sends c_{n-1} to P_0];

for $i = 0, 1, 2, \ldots, n-1$ **do**

  P_i locks the secret c_{i-1};

  $c_i = c_{i-1}^{a_i} mod(p)$;

  P_i sends c_i to P_{i+1} /* $mod(n)$ */;

$P_i, i \in \{1, 2, \ldots, n-1\}$; send these messages to P_0 [master sites];

Unlocking :

P_0 sends all messages to P_1 in a random order;

foreach $P_i \in \{P_1, P_2, \ldots, P_{n-1}\}$ **do**

  P_i receives all messages set from site P_{i-1};

  P_i removes his lock from the set of all messages;

  P_i sends the set of messages to P_{i+1};

  /* `should be in an arbitrary sequential order` */ ;

P_0 [master site] receives the set of all messages;

P_0 unlocks the secret *i.e.*, the set of messages

end

Figure 2 shows the trace of execution of the protocol. For the sake of simplicity, we put the focus only on messages from *Site* 3 but in general case, each party participates with a message. Figure 2 explains the various stages of the protocol. Steps ① to ⑥ present the locking phase and steps ⑦ to ⑩ present the unlocking phase. During steps ① to ④, each party locks the message with his own key. Step ⑤ is the last locking phase and then site 3 sends his message to the master site. During step ⑥, the master site collects all the messages from parties (each party participates with one message). After that, in step ⑦ the master site sends the set of messages to site 2 in an arbitrary order. During steps ⑧ to ⑩, each party removes his lock from the set of messages and then sends messages to the next party in an arbitrary order. When the master site receives the set of all messages, these messages are locked only with the master site key. When he removes his own lock and at this step, the master site obtains unlocked messages where he is unable to guess the original sender of a specific message. Noted that, in Figure 2, f_{si} stands for the encryption function (f_{s1}, for example, is the encryption function used by *site* 1).

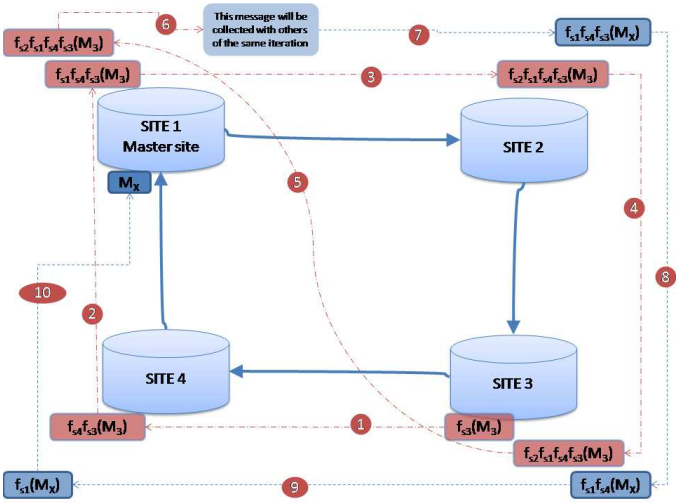

Figure 2: Encryption/Decryption Scheme for message from site 3

Interestingly enough, the initialization phase will be executed only once, even through the protocol will be invoked several times by another system.

At this step, we prove, firstly, that the cryptographic model is safe. Secondly, we show that our protocol preserves the constraints of privacy. Within the first item, the correctness of the scheme can be easily established by using proposition 1 (section 2.3.2), as follows: at the end of a locking phase, the shared secret s' is

$$s' = s^{a_0 a_1 a_2 \ldots a_{n-1}} mod(p)$$

and at the end of the unlocking phase, we have:

$$s'^{(b_0 b_1 \ldots b_{n-1})} mod(p) = s^{(a_0 a_1 a_2 \ldots a_{n-1})(b_0 b_1 b_2 \ldots b_{n-1})} mod(p) = \mathbf{s}$$

Correctness.

As can be seen, the commutativity of the modular exponentiation makes the order of unlocking operations unimportant. The set of secrets s_i at the end is obtained by P_0.

Security.

The security of the scheme heavily draws on the computational difficulty of *discrete logarithm problem* (DLP). An adversary may pursue one of the following goals in breaking the scheme:

- *Total break*: He finds out the secret, thereby by passing the unanimous consent requirement altogether.

- *Partial break*: He finds out the private keys (a_i or equivalently b_i) of one or more of the parties. This enables him to play the part of some parties, stealing their right to consent.

The following arguments establish the security of the scheme:

1. In the first locking loop, from P_i to P_{i+1}, the secret owner has locked the secret s_i ($c_i = s_i^{a_0} mod(p)$) before sending it to P_{i+1}. Note that the adversary has two unknowns to search for: a_i and s_i. The private keys are chosen from a huge space ($a_i, b_i \in \mathbb{Z}_p$). Therefore, an exhaustive search is infeasible, at least in a reasonable time. The adversary has no way of guessing s_i without knowing a_i.

2. In the subsequent loops, the adversary can eavesdrop and reads c_{i-1} in transit from P_{i-1} to P_i, and he can read c_i in transit from P_i to P_{i+1}. To find out a_i, the adversary has to solve a DLP ($c_i = c_{i-1}^{a_i} mod(p)$), which is infeasible, at least in reasonably time. If he succeeds, he could then compute b_i, which is one of the keys necessary for unlocking the shared secret s'. He has to solve a DLP for all loops in order to obtain all b_i's. On the other hand, if the adversary determines any of the b_i's, he can take the role of the corresponding parties and conspire for unauthorized unlocking of the secret. Since solving DLP is computationally hard, the above possibilities are vanishingly improbable.

3. The same type of arguments are valid in the unlocking phase: The shared secret is always locked by at least one party when it transits.

4. In the last unlocking loop, the set of secret is sent to P_0, and it is only locked by a_0. But again, the adversary has to solve a DLP ($s' = c_{n-1}^{b_0} mod(p)$) in order to find out b_0.

5. There is a nonzero possibility that the private keys are by chance selected such that

$$\begin{aligned} a_0 a_1 &= 1 \ mod(p-1) \text{ or} \\ a_0 a_1 a_2 &= 1 \ mod(p-1) \text{ or} \\ &\dots = \dots \\ a_0 a_1 a_2 \dots a_n &= 1 \ mod(p-1) \end{aligned}$$

   This will result in having $c_i = s$ for some i. In that case, the secret s is exposed in transit from P_i to P_{i+1}. However, given the fact that the keys must be chosen from a huge space, the probability of any of the above events is vanishingly small.

Therefore, the security of the scheme was shown to rely on DLP, as claimed in Khayat's paper [12].

At this point, we proved the correctness of our protocol. Now, we show that the protocol preserves privacy. The following arguments can ensure that the protocol preserves the privacy of the scheme:

1. The protocol is safe: The shared secret is always locked by at least one party when it transits. Consequently, the secret is shared only with the parties involved in th mining tasks.

   Now we are interested in parties involved in the task of data mining:

2. During the locking phase, the messages passing between the parties are encrypted at least by the owner of the message. Therefore, others parties can not read these messages. This phase did not violate the constraints of privacy.

3. During the unlocking phase, keeping in mind that the master site did not remove his lock, the set of the messages is at least locked with the master site key. The adversaries have no way of guessing the set of messages.

4. At the end of encryption phase, all parties have already send messages (encrypted by all partie's keys) to the master site. This step is not disturbing to the constraint of privacy.

5. On the basis on the assumption of semi-honest model, all parties following the instructions of the protocol. The protocol requires that a party P_i sends messages to P_{i+1} in a random order. For example, if P_i has received the messages $(M1, M3, M2)$ from P_{i-1}, then P_i can send messages in the following sequence $(M3, M2, M1)$ by example. In this way, at the end of the unlocking step, the order of messages will be necessarily altered while no party, even the master site, is able to guess the original order of messages. This property of the algorithm preserve the anonymity of messages between parties involved in the data mining task. In this way, the protocol ensures anonymity of the messages in the last unlocking stage.

6. During the master site when decrypts messages, in the final stage of unlocking phase, receives messages encrypted only with its key. After decryption, the master site can not guess the original sender of a message. It receives a set of messages, while it knows that the messages are necessarily from the parties involved in this task, but it can not know which message was from party P_1 (for example).

4. CONCLUSION AND FUTURE WORKS

Cryptographic tools can enable data mining that would otherwise be prevented due to security concerns. We have given procedures to mine distributed association rules on horizontally partitioned data. We have shown that distributed association rule mining can be done efficiently under reasonable security assumptions.

The main contributions of this paper is to propose a framework of commutative encryption for privacy preserving association rules mining. For that the randomization methodologies are not good enough to attain the high accuracy and protect clients' information from privacy breach and the malicious attack, we show that how association rule mining can be done in this framework and prove that is secure enough to keep the clients' privacy. Our algorithm is currently under implementation and we plan to carry out thorough experiments to assess to efficiency of the introduced approach vs those of the literature.

5. REFERENCES

[1] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 86–97. ACM Press, 2003.

[2] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data, SIGMOD'93*, pages 207–216, New York, NY, USA, 1993. ACM Press.

[3] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI/MIT Press, 1996.

[4] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM Press, May 2000.

[5] M. Atallah, A. Elmagarmid, M. Ibrahim, E. Bertino, and V. Verykios. Disclosure limitation of sensitive rules. In *Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange (KDEX'99)*, page 45, Washington, DC, USA, 1999. IEEE Computer Society.

[6] S. Ben Yahia, T. Hamrouni, and E. Mephu Nguifo. Frequent closed itemset based algorithms: A thorough structural and analytical survey. *ACM SIGKDD Explorations*, 8(1):93–104, june 2006.

[7] C. Clifton, M. Kantarcioglu, and J. Vaidya. Tools for privacy preserving distributed data mining. *ACM SIGKDD Explorations*, 4:28–34, 2003.

[8] W. Du and Z. Zhan. Building decision tree classifier on private data. In *Proceedings of the IEEE International conference on privacy, security and data mining*, pages 1–8, Darlinghurst, Australia, 2002. Australian Computer Society, Inc.

[9] M. Kantarcioglu. A survey of privacy-preserving methods across horizontally partitioned data. In *Privacy-Preserving Data Mining: Models and Algorithms*, pages 313–335. Springer US, 2008.

[10] 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):1026–1037, september 2004.

[11] M. Kantarcioglu, J. Jin, and C. Clifton. When do data mining results violate privacy? In *Proceedings of the tenth ACM SIGKDD International conference on Knowledge discovery and data mining*, pages 599–604, New York, NY, USA, 2004. ACM.

[12] S. Khayat. Using commutative encryption to share a secret. Cryptology ePrint Archive, Report 2008/356, 2008.

[13] X. Lin, C. Clifton, and M. Zhu. Privacy-preserving clustering with distributed EM mixture modeling. *Knowl. Inf. Syst.*, 8(1):68–81, 2005.

[14] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Journal of Cryptology*, pages 36–54. Springer-Verlag, 2000.

[15] A. Menezes, S.Vanstone, and V. Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., FL, USA, 1996.

[16] N. Pasquier. *Datamining: Algorithmes d'extraction et de réduction des règles d'association dans les bases de données*. PhD thesis, Université Clermont Ferrand II, France, 2000.

[17] N. Pasquier. Mining association rules using formal concept analysis. In *Stumme, G. (Ed.): Working with Conceptual Structures. Contributions to ICCS 2000. Verlag Shaker Aachen*, pages 259–264, 2000.

[18] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24:25–46, 1999.

[19] B. Pinkas. Cryptographic techniques for privacy-preserving data mining. *SIGKDD Explor. Newsl.*, 4(2):12–19, 2002.

[20] A. Shamir. On the power of commutativity in cryptography. In *Proceedings of the 7th Colloquium on Automata, Languages and Programming*, pages 582–595, London, UK, 1980. Springer-Verlag.

[21] S. Weis. *New foundations for efficient authentification, commutative cryptography, and private disjointness testing*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2006.

[22] R. Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In I. Rival, editor, *Ordered Sets*, pages 445–470. Reidel, Dordrecht-Boston, 1982.

[23] S. Ben Yahia, G. GASMI, and E. Mephu Nguifo. A new generic basis of factual and implicative association rules. *Intelligent Data Analysis - An International Journal (IDA)*, 13(4), 2009. to appear.

[24] A. Yao. Protocols for secure computations. In *SFCS '82: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 160–164, Los Alamitos, CA, USA, 1982. IEEE Computer Society.

[25] A. Yao. How to generate and exchange secrets. In *SFCS '86: Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pages 162–167, Los Alamitos, CA, USA, 1986. IEEE Computer Society.