Privacy Preserving Sequential Pattern Mining in Distributed Databases

V. Kapoor*
Netaji Subhas Institute of Technology
Azad Hind Fauj Marg
Dwarka, New Delhi-110045, India

P. Poncelet †
F. Trousset
EMA-LGI2P/Site EERIE
Parc Scientifique Georges Besse
30035 Nîmes Cedex, France

M. Teisseire‡
LIRMM UMR CNRS 5506
161 Rue Ada, 34392
Montpellier Cedex 5, France

ABSTRACT

Research in the areas of privacy preserving techniques in databases and subsequently in privacy enhancement technologies have witnessed an explosive growth-spurt in recent years. This escalation has been fueled by the growing mistrust of individuals towards organizations collecting and disbursing their Personally Identifiable Information (PII). Digital repositories have become increasingly susceptible to intentional or unintentional abuse, resulting in organizations to be liable under the privacy legislations that are being adopted by governments the world over. These privacy concerns have necessitated new advancements in the field of distributed data mining wherein, collaborating parties may be legally bound not to reveal the private information of their customers. In this paper, we present a new algorithm PRIPSEP (PRI*vacy* P*reserving* SE*quential* P*atterns*) for the mining of sequential patterns from distributed databases while preserving privacy. A salient feature of PRIPSEP is that due to its flexibility it is more pertinent to mining operations for real world applications in terms of efficiency and functionality. Under some reasonable assumptions, we prove that our architecture and protocol employed by our algorithm for multi-party computation is secure.

Categories and Subject Descriptors: H.2.8 [Database Applications]: Data Mining

General Terms: Algorithms.

Keywords: Privacy Mining.

1. INTRODUCTION

The increasing use of multi-database technology, such as computer communication networks and distributed, federated and homogeneous multi-database systems, has led to the development of many large distributed transactional databases. For decision-making, large organizations might need to mine these multiple databases located at disparate branches and locations. Particularly, as the Web is rapidly becoming an information flood, individuals and organizations can take into account low-cost information and knowledge on the Internet while making decisions. Although this large data enables in the improvement of the quality of decisions, it also generates a significant challenge in the form of efficiently identifying quality knowledge from multi-databases [20, 25].

Therefore, large corporations may have to confront the multiple data-source problem. For example, a retail-chain with numerous franchisees might wish to collaboratively mine the union of all the transactional data. Each of the smaller transactional databases could contain information regarding the purchasing history of the same set of common customers transacting through online portals or real stores. However, the greater challenge of these computations can be the additional constraint of adhering to stringent privacy requirements laid down by the formulation of new laws such as HIPAA [15]. These regulatory policies have been the driving force behind the increased consciousness in organizations towards the protection of privacy. Consequently, there has been a paradigm shift towards the creation of privacy-aware infrastructures, which entail all aspects, ranging from data-collection to analysis [3].

Conventionally, data mining has operated on a data-warehousing model of gathering all data into a central site, then running an algorithm against that data. Privacy considerations may prevent this generic approach. Hence, privacy preserving data mining has gained recognition among academia and organizations as an important and unalienable area, especially for highly sensitive data such as health-records. If data mining is to be performed on these sensitive datasets, due attention must be given to the privacy requirements. However, conventional sequential pattern mining methods based on support do not preserve privacy and are ineffective for global pattern mining from multiple data sources.

Traditionally, Secure Multi-Party Protocols (SMC) have been employed for the secure computation for any generic functions. However, the complexity and overhead of such secure protocols would be prohibitive for complex data mining tasks such as the discovery of sequential patterns. Hence, to alleviate the communication and bandwidth overhead of the Oblivious Transfer (i.e. the protocol by which sender sends some information to the receiver, but remains oblivious as

*email: vkapoor@cse.iitd.ernet.in. This work was performed as part of an internship at the LGI2P Research Center.

†emails: {Pascal.Poncelet,trousset}@ema.fr

‡email: teisseire@lirmm.fr

CIKM'06, November 5–11, 2006, Arlington, Virginia, USA.

to what is sent) required between parties in an SMC, we employ an alternative architecture consisting of semi-honest and non-colluding sites [12]. This tradeoff between security and efficiency is reasonable as none of the participating sites learn the intermediate or the final results of the calculus. Furthermore, due to the uniform random noise in the datasets, the private information of any individual is also guarded from any possible leak.

In this paper, we present an alternative privacy preserving data mining approach - PRIPSEP, for finding sequential patterns in the distributed databases of a large integrated organization. Our novel algorithm, PRIPSEP is useful for mining sequential patterns via collaboration between disparate parties, employing the secure architecture, performing the secure operations via the underlying protocols.

Organization: The remainder of this paper is organized as follows. Section 2 goes deeper into presenting the problem statement and provides an extensive description of the problem at hand. In Section 3, we present an overview of the related work and give our motivation for a new approach. Section 4 describes our proposed solution with the description of the architecture and the algorithms for secure multiparty protocols. Finally, Section 5 concludes the paper with a roadmap for future work.

2. PROBLEM STATEMENT

In this section, we give the formal definition of the problem of privacy preserving collaborative sequential pattern mining. First, we provide a brief overview of the traditional frequent pattern mining problem by summarizing the formal description introduced in [1] and extended in [18]. Subsequently, we extend the problem by considering distributed databases. Finally, we formally define the problem of privacy preserving sequential pattern mining.

2.1 Mining of Sequential Patterns

Let DB be a database containing a set of customer transactions where each transaction T consists of a customer-id(CID), a transaction time(TID) and a set of items involved in the transaction.

Let $I = \{i_1, i_2 ... i_m\}$ be a set of literals called items. An itemset is a non-empty set of items. A sequence S is a set of itemsets ordered according to their timestamp. It is denoted by $< s_1\ s_2\ ...s_n >$, where $s_j, j \in 1...n$, is an itemset. In the rest of the paper we will consider that itemsets are merely reduced to items. Nevertheless all the proposal could be easily extended to deal with itemsets. A k-sequence is a sequence of k items (or of length k). A sequence $S' =< s'_1\ s'_2 ...\ s'_n >$ is a subsequence of another sequence $S =< s_1\ s_2\ ...\ s_m >$, denoted $S' \prec S$, if there exist integers $i_1 < i_2 < ... i_j ... < i_n$ such that $s'_1 \subseteq s_{i_1}$, $s'_2 \subseteq si_2$, ... $s'_n \subseteq si_n$.

All transactions from the same customer are grouped together and sorted in increasing order and are called a data sequence. A support value (denoted $supp(S)$) for a sequence gives its number of distinct occurrences in DB. Nevertheless, a sequence in a data sequence is taken into account only once to compute the support even if several occurrences are discovered. In other words, the support of a sequence is defined as the fraction of total distinct data sequences that contain S. A data sequence contains a sequence S if S is a subsequence of the data sequence. In order to decide whether a sequence is frequent or not, a minimum support value (denoted *minsupp*) is specified by the user, and the sequence is said to be *frequent* if the condition $supp(S) \geq minsupp$ holds. Given a database of customer transactions, the problem of sequential pattern mining is to find all the sequences whose support is greater than a specified threshold (minimum support). Each of these represents a sequential pattern, also called a frequent sequence.

2.2 From Collaborative to Privacy Preserving Sequential Pattern Mining

Let DB be a database such as $DB = DB_1 \bigcup DB_2 ... \bigcup DB_D$. For simplicity, we consider that all databases DB_1, DB_2 ... DB_D share the same number of customers (CIDs), which is N. We also consider that for each customer in the databases, the number of transaction times (TIDs), K, is the same[1]. As we extend the data representation scheme from the SPAM approach [2], we consider that all transactions are depicted in the form of vertical bitmaps, which we denote as vectors for clarity in mathematical formulae.

DEFINITION 1. *Let V_i^j be a vector where j and i correspond respectively to the i^{th} item and the j^{th} database. V_i^j is defined as follows: $V_i^j = [C_1^{i,j} ... C_N^{i,j}]$ where for $u \in \{1..N\}$, $C_u^{i,j} = [T_1^{i,j,u}, ..., T_K^{i,j,u}]$. $T_{v=\{1..K\}}^{i,j,u}$ corresponds to the transaction list of the customer u, from the database DB_j and the item i. It is a K length bit string that has the v^{th} bit as one if the customer u has bought the item i from the database DB_j.*

Given a set of databases $DB_1, DB_2 ... DB_D$ containing customer transactions, the problem of collaborative sequential pattern mining is to find all the sequences whose support is greater than a specified threshold (minimum support). Furthermore, the problem of privacy-preserving collaborative sequential pattern mining is to discover sequential patterns embedded in the union of databases by considering that the parties do not want to share their private datasets with each other.

In order to illustrate this further, let us consider the following example.

EXAMPLE 1. *Let us assume that three retail franchisees Alice, Bob and Carol wish to securely extract the sequential patterns in the union of their databases without disclosing the identities of any individual customers. Each item is provided with its timestamp (C.f. table 1).*

CID	Alice	Bob	Carol
1	$(1)_1$ $(3)_5$	$(2)_2$	$(7)_4$
2	$(2)_4$	$(1)_3$	$(3)_6$
3	$(2)_6$ $(3)_7$		$(1)_2$ $(7)_3$

Table 1: An example of distributed databases sorted by CID

Let us assume that the minimal support value is set to 50%. From the three distributed databases, we can infer that item (1) is not frequent in any one of the individual databases. However, by considering the union of all databases (C.f. table 2 where the superscript depicts the original database,

[1]This constraint has been considered purely for readability reasons. All the described algorithms could be easily extended to incorporate customer sequences that do not have the same number of TIDs.

CID	Sequences
1	$(1)^A_1\ (2)^B_2\ (7)^C_4\ (3)^A_5$
2	$(1)^B_3\ (2)^A_4\ (3)^C_6$
3	$(1)^C_2\ (7)^C_3\ (2)^A_6\ (3)^A_7$

Table 2: Sequences for each customer in the union of all databases

where the item is derived from), we obtain that the sequence $< (1)(2)(3) >$ *is frequent. By considering the constraints for privacy, this sequence has to be obtained by considering Alice, Bob and Carol are not at liberty to disclose the private transactional history of any of the customers.*

3. RELATED WORK

In this section we focus on the various research work closely related to the domain of privacy preserving data mining and sequential patterns.

Sequential Patterns: Since its introduction, more than a decade ago, the sequential pattern mining problem has received a great deal of attention and numerous algorithms have been defined to efficiently find such patterns (e.g. GSP [18], PSP [14], PrefixSpan [16], SPADE [23], FreeSpan[10], SPAM [2]). Our data representation scheme has been extended from the SPAM algorithm [2], wherein for efficient counting, each customer's transactions are represented by a vertical bitmap.

Privacy Preserving Data Mining: Recently, there has been a spate of work addressing privacy preserving data mining [17, 5]. This wide area of research includes classification techniques [7], association rule mining [8], and clustering [11] with privacy constraints. In early work on privacy-preserving data mining, Lindell and Pinkas [13] propose a solution to privacy-preserving classification problem using oblivious transfer protocol, a powerful tool developed by SMC research. The techniques based on SMC for efficiently dealing with large data sets have been addressed in [19], where a solution to the association rule mining problem for the case of two parties was proposed. Recently, a novel secure architecture has been proposed in [12], where the security and accuracy of the data mining results are guaranteed with improved efficiency.

Secure Multi-Party Computation: A Secure Multi-party Computation (SMC) problem deals with computing any function on any input, in a distributed network where each participant holds one of the inputs, while ensuring that no more information is revealed to a participant in the computation than can be inferred from that participants input and output. Secure two party computation was first investigated by Yao [21, 22] and was later generalized to multi-party computation (e.g. [6, 9, 4]). It has been proved that for any polynomial function, there is a secure multiparty computation solution [9, 4]. The approach used is as follows: the function f to be computed is firstly represented as a combinatorial circuit, and then the parties run a short protocol for every gate in the circuit. Every participant gets corresponding shares of the input wires and the output wires for every gate. While this approach is appealing in its generality and simplicity, the protocols it generates depend on the size of the circuit. This size depends on the size of the input (which might be huge as in a data mining application), and on the complexity of expressing f as a circuit (for example, a naive multiplication circuit is quadratic in the size of its inputs). Hence this approach, is highly impractical for large datasets and complicated computations necessary in complex data mining tasks. Our shift away from a traditional SMC approach has been motivated by [12], describing the limitations of highly secure, yet practically unviable protocols.

Previous Work: The research area of privacy preserving sequential pattern mining lies largely unexplored with only one seminal paper [24]. Zhan et al. have proposed an approach, which entails the transformation of the databases of each collaborating party, followed by the execution of a secure protocol, which results in the preservation of privacy, as well as the correct results. Theoretically, the approach is robust and secure, however, it has serious limitations relating to the initial constraints assumed while developing the approach. It has been proposed that each of the collaborating parties carries a unique inventory. For instance, considering our previous example and not taking into account the possibility of items being shared among the distributed parties, we do not arrive at the complete results. An item such as (1), which is not supported by enough customers in one individual database will not appear in the final results. This assumption causes serious limitation for real applications where item sharing between different databases is imperative as well as a fundamental requirement as shown earlier. Moreover, employing their new data representation scheme for sequential data, the same customer buying the same item more than once from the same database but with a different TID is not permissible. One other drawback of mapping each item to a unique code is the additional overhead incurred while sorting the databases, which might be significant for large databases.

4. THE PRIPSEP APPROACH

In this section, we propose our novel approach for privacy preserving sequential pattern mining in distributed and collaborative databases. Firstly we focus only on collaborative sequential pattern mining in order to clearly explain our methodology. This approach is extended in the next section in order to consider privacy requirements and finally we propose a new algorithm and underlying protocols within the secure architecture.

4.1 Collaborative sequential pattern mining

4.1.1 An overview

As previously seen in Section 2, the challenge with collaborative mining lies in the fact that we have to deal with different databases where the order of items is not known beforehand (e.g. item (7) of the CID 1 in Carol's database occurs before item (3) in Alice's database).

For brevity, we consider the Data Miner performing the generating and verifying phases of candidate sequences similar to the Apriori-based algorithms. We assume that the candidate generation is performed conventionally by combining the k-1 frequent sequences in order to generate k-candidate sequences (e.g. C.f. GSP [18] generation phase). We extend the verification phase as follows. As we have to deal with disparate distributed databases, we assume that the Miner could request information from the D original databases in order to obtain a vector corresponding to the specific item

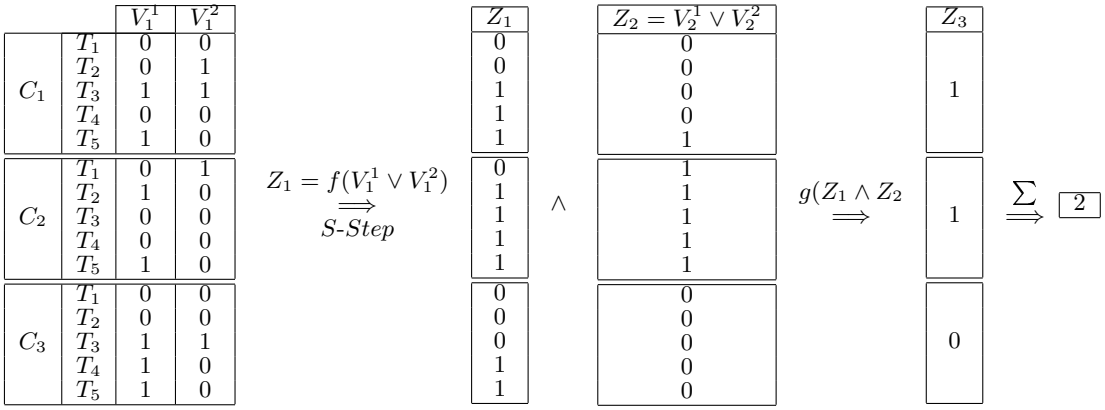

Figure 1: Processing of vectors for collaborative mining

i, i.e. $V_i^{[1..D]}$ for any candidate sequence.
Let us consider that we are provided with two databases, namely DB_1 and DB_2. These databases contain transactions for three customers and each customer has five transaction times or TIDs. The process aims at finding the support value for the sequence $< (1)(2) >$ in the set of all customers of the two databases. First, we extract from DB_1, the vector corresponding to the item (1), i.e. V_1^1, and from DB_2 the vector V_1^2 (left part of figure 1). From the given vectors, two key operations have to be performed: (i) bitwise OR of the two vectors, and (ii) transforming the result in order to check if it could be followed by (2). These two vectors are merged together by applying a bitwise operator ($\vee$): $V_1^1 \vee V_1^2$. For the second operation, similar to the S-step process of the SPAM algorithm, we consider a function that transforms the vector(bitmap). For each customer, following the occurrence of the first bit with value one, every subsequent bit in the vector is flagged as one. However, since we have to deal with different databases as well as efficiency issues, we consider that these two operations are performed through the f function defined below to obtain a new vector $Z_1 = f(V_1^1 \vee V_1^2)$.

Definition 2. *Let us consider a vector V_i^j for a database j and an item i. V_i^j is defined as follows: $V_i^j = (C_1^{i,j}...C_N^{i,j})$ where for $u \in \{1..N\}$, $C_u^{i,j} = (T_1^{i,j,u}, ..., T_K^{i,j,u})$. K stands for the number of TIDs and N corresponds to the number of CIDs. For brevity, we denote this vector as V. Let $f : [0,1]^{N \times K} \rightarrow [0,1]^{N \times K}$ be a function such that: $f(V) = f(C_1...C_N) = [f_c(C_1)f_c(C_2)...f_c(C_N)]$. For each $u \in \{1..N\}$, we have: $f_c(C_u) = \begin{vmatrix} 0 \\ T_1^u \\ T_1^u \vee T_2^u \\ T_1^u \vee T_2^u \vee T_3^u \\ ... \\ T_1^u \vee ... \vee T_{k-1}^u \end{vmatrix}$ where $\vee$ is a bitwise operator. We can notice that $Card(V) = N \times K$, $Card(C_u) = K$, $Card(f(V)) = N \times K$.*

Let $g : [0,1]^{N \times K} \rightarrow [0,1]^N$ be a function such that: $g(V) = g(C_1...C_N) = [g_c(C_1)g_c(C_2)...g_c(C_N)]$. For each $u \in \{1..N\}$, we have: $g_c(C_u) = 1$ if there exists at least one bit with value 1 in the customer transactions. It can be noted that $Card(g(V)) = N$.

In conjunction with the computation of the function f, the vectors corresponding to the item (2) are extracted from DB_1 and DB_2 (V_2^1 and V_2^2 respectively). Similar to the previous step the vector ($Z_2 = V_2^1 \vee V_2^2$) is computed. Following that, the bitwise operator $\wedge$ is used to calculate $Z_1 \wedge Z_2$ and the g function is used to calculate the count for each customer, for the sequence $< (1)(2) >$, i.e. $Z_3 = g(f(V_1^1 \vee V_1^2) \wedge (V_2^1 \vee V_2^2))$. As the resulting vector Z_3 has a cardinality corresponding to the number of customers, i.e. N, the last operation to be performed is a summation of the number of bits with the value 1 in the vector Z_3. This is performed by the $\sum$ operation.

4.1.2 The collaborative support counting algorithm

The Collaborative Frequency algorithm (see Algorithm 1) has been developed as follows. For each item i of the candidate sequence to be tested, a new vector X_i is generated by applying the $\vee$ bitwise operator on all the corresponding vectors from the original databases. Hence, by considering the result of the previous operation, the f function is applied, followed by the bitwise operator $\wedge$ for each item. At the end of this iteration, a new vector Z of cardinality $N \times K$ is produced. Consequently, the g function is applied to the intermediate result for generating a vector of cardinality N, i.e. Y. Finally, the number of bits which are 1 in Y are summated to compute the final value of support.
Complexity: Let $V_s = N \times K$ be the size of the vectors which are sent and S be the candidate sequence to be verified. The transfers that are performed by the algorithm are: $(V_s \times D \times |S|)$ for $\bigvee$ and $(V_s \times |S|)$ for both the f function and $\bigwedge$ operation. There are $(N(K-2))$ $\bigvee$ computations performed by f. If f is already available, i.e. precomputed and stored, we have (N) $\bigvee$ operations, otherwise $(N(K-1))$ $\bigvee$ operations are performed by g.

4.2 From collaborative to privacy-preserving sequential pattern mining

4.2.1 A brief overview of the architecture

In this section we describe an architecture where secure multi-party techniques developed in the cryptographic domain can be easily extended for data mining purposes[12]. Previous work [9] has described that Secure Multi-party protocols can be used directly to solve with total security, any