Privacy-Preserving Mining of Association Rules from Outsourced Transaction Databases

PONNA SANTHOSHI¹, M. ESWARA RAO²

¹PG Scholar, Sri KS Raju Institute of Technology and Sciences, Hyderabad, TS, India.
²Associate Professor, Sri KS Raju Institute of Technology and Sciences, Hyderabad, TS, India.

Abstract: Inspired by developments such as cloud computing, there has been considerable recent interest in the paradigm of data mining-as-service. A company (data owner) lacking in expertise or computational resources can outsource its mining needs to a third party service provider (server). However, both the items and the association rules of the outsourced database are considered private property of the corporation (data owner). To protect corporate privacy, the data owner transforms its data and ships it to the server, sends mining queries to the server, and recovers the true patterns from the extracted patterns received from the server. In this paper, we study the problem of outsourcing the association rule mining task within a corporate privacy-preserving framework. We propose a scheme for privacy preserving outsourced mining and show that the owner can recover the true patterns as well as their support by maintaining a compact synopsis.

Keywords: Encrypt/Decrypt (ED), Privacy-Preserving Framework.

I. INTRODUCTION

In recent years, there has been considerable interest in the data mining-as-service paradigm for enabling organizations with limited computational resources and/or data mining expertise to outsource their data mining needs to a third party service provider [2,9,6,5,13]. As an example, the operational transactional data from various outlets of Safeway, a grocery chain operating in the US and Canada, can be shipped to a third party which provides mining service for Safeway. The Safeway management need not employ an in-house team of data mining experts. Besides, they can cut down their local data management requirements because periodically data is shipped to the service provider who is in charge of maintaining it and conducting mining on it in response to requests from Safeway’s business analysts. In this example, Safeway, the client, is a Data owner and the service provider is referred to as the server. One of the main issues with this paradigm is that the server has access to valuable data of the owner and may learn sensitive information from it. E.g., by looking at the transactions, the server (or an intruder who gains access to the server) can learn which items are co-purchased, and in turn, the mined patterns. However, both the transactions and the mined patterns are the property of Safeway and should remain safe from the server. This problem of protecting important private information of organizations/companies is referred to as “corporate privacy”.

Unlike personal privacy, which only considers the protection of the personal information recorded about individuals, corporate privacy requires that both the individual items and the patterns of the collection of data items are regarded as corporate assets and thus must be protected. In this paper, we study the problem of outsourcing the association rule mining task within a corporate privacy-preserving framework. We propose an encryption scheme which enables privacy guarantees, and show some preliminary results obtained applying this model over large-scale, real-life transaction databases. The architecture behind our model is illustrated in Fig 1. The client/owner encrypts its data using an encrypt/decrypt (ED) module, which can be treated as a “black box” from its perspective. This module is responsible for transforming the input data into an encrypted database. The server conducts data mining and sends the (encrypted) patterns to the owner. Our encryption scheme has the property that the returned supports are not true supports. The ED module recovers the true identity of the returned patterns as well their true supports.

Fig. 1. Architecture of Mining-as-Service Paradigm.

II. RELATED WORK

In this section we outline the work on privacy-preserving data publishing and mining. Privacy-preserving data publishing (PPDP): The idea is that data is published with appropriate
PONNA SANTHOSHI, M. ESWARA RAO

suppression, generalization, distortion, and/or decomposition such that individual privacy is not compromised and yet the published data is useful for mining. Privacy-preserving data mining (PPDM): The main model here is that private data is collected from a number of sources by a collector for the purpose of consolidating the data and conducting mining. The collector is not trusted, so data is subjected to a random perturbation as it is collected. This body of work was pioneered by and has been followed up by several papers since. Privacy-preserving pattern publishing (PPP): The central question is how to publish results of mining such as frequent patterns without revealing any sensitive information about the underlying data, but the resulting patterns are disclosed. A key distinction between our problem and the above mentioned PPDM problems is that, in our setting, not only the underlying data but also the mined results are not intended for sharing and must remain private. Similar to our work, first, they utilize a one-to-n item mapping together with non-deterministic addition of cipher items to protect the identification of individual items. Second, they assume that the adversary may possess some prior knowledge of frequency of the item sets, which can be used to decipher the encrypted items. While our attack model focuses on single items with the assumption that the attacker knows the exact frequency of every single item. The major issue left open is a formal protection result: their security analysis is entirely conducted empirically on various synthetic datasets.

III. PRIVACY MODEL

We let D denote the original TDB that the owner has. To protect the identification of individual items, the owner applies an encryption function to D and transforms it to D*, the encrypted database. We refer to items in D as plain items and items in D* as cipher items. The term item shall mean plain item by default. The notions of plain item sets, plain transactions, plain patterns, and their cipher counterparts are defined in the obvious way. We use I to denote the set of plain items and E to refer to the set of cipher items.

A. Adversary Knowledge

The server or an intruder (attacker) who gains access to the database may possess some background knowledge using which they can conduct attacks on the encrypted database D* in order to make inferences. We adopt a conservative model and assume that the attacker knows exactly the set of (plain) items I in the original transaction database D and their true supports in D, i.e., suppD(i), i ∈ I. The attacker may have access to similar data from a competing company, may read published reports, etc. Moreover we assume the attacker has access to the encrypted database D*. Thus, he also knows the set of cipher items and their support in D*, i.e., supp D*(e), e ∈ E. In this paper we propose an encryption scheme based on: (i) replacing each plain item in D by a 1-1 substitution cipher (ii) adding fake transactions to the database. In particular, no new items are added. We assume the attacker knows this and thus he knows that |E | = |I|. We also assume the attacker knows the details of our encryption algorithm.

B. Encryption

In this section, we introduce the encryption scheme, which transforms a TDB D into its encrypted version D*. Our scheme is parametric w.r.t. k > 0 and consists of three main steps: (1) using 1-1 substitution ciphers for each plain item; (2) using a specific ite k-grouping method; (3) using a method for adding new fake transactions for achieving k-privacy. The encryption scheme is a countermeasure to the item-based and set-based attacks discussed in Sec. 4: since the attacker knows the exact support of each item, we create a k-private D*, such that the cipher items cannot be broken based on their support.

k-Grouping Method: Given the items support table, several strategies can be adopted to cluster the items into groups of k. We assume the item support table is sorted in descending order of support and refer to cipher items in this order as e1, e2, etc. To obtain the formal protection that itemsets (or transactions) cannot be cracked with a probability higher than 1/k, we need to use only grouping methods that yield groups of items that are unsupported in D.

C. Decryption

When the client requests the execution of a pattern mining query to the server, specifying a minimum support threshold í, the server returns the computed frequent patterns from D*. Clearly, for every itemset S and its corresponding cipher itemset E, we have suppD*(S) ≤ suppD*(E). Therefore, our encryption scheme guarantees that all itemsets frequent in D will be returned, in cipher version, by the server. But additional patterns frequent in D*, but not in D, are returned as well. For each cipher pattern E returned by the server together with suppD*(E), the ED module trivially recovers the corresponding plain pattern S as follows: suppD*(S) = suppD*(E)− suppD\(\Delta\)D(E).

IV. CONCLUSION AND FUTURE WORK

We studied the problem of (corporate) privacy-preserving outsourcing of association rule mining. Our encryption scheme is based on 1-1 substitutions and fake transactions such that the transformed database satisfies k-anonymity w.r.t. items and item sets. Moreover we showed some preliminary empirical results that are encouraging; naturally, there are many interesting open issues to be investigated. The next steps include: (1) The study of a formal analysis based on our attack model and the proof that the probability that an item set can be broken by the server can always be controlled to be below a threshold chosen by the owner, by setting the anonymity threshold k; (2) The complexity analysis of our encryption/decryption scheme; (3) The definition of an strategy for incrementally maintaining the synopsis at the client side against updates in the form of appends. (4) scheme using large real data set with different sparsity/density properties to understand how it works in different settings; (5) The analysis of the scalability of the proposed approach and comparison of the execution time of the mining step with that
Privacy-Preserving Mining of Association Rules from Outsourced Transaction Databases

of the decryption step by using a mining algorithm like FP-growth, that could be more efficient than an apriori-based algorithm.

V. REFERENCES