Supply Chain and Finance, pp. 273-281
P.M. Pardalos, A. Migdalas and G. Baourakis, Editors
World Scientific Publishing, River Edge, NJ, U.S.A., 2004

CHAPTER 16

MINING ENCRYPTED DATA

B. Boutsinas

Department of Business Administration,
University of Patras Artificial Intelligence Research Center (UPAIRC),
University of Patras, GR-26500 Rio, Patras, Greece,
Tel: +30-610-997845, Fax: +30-610-996327
E-mail: vutsinas@bma.upatras.gr

G. C. Meletiou

T.E.I. of Epirus, P.O. Box 110, GR–47100 Arta and UPAIRC, Greece,
Tel: +30-6810-26825, Fax: +30-6810-75839
E-mail: gmelet@teiep.gr

M. N. Vrahatis

Department of Mathematics, UPAIRC,
University of Patras, GR-26500 Patras, Greece,
Tel: +30-610-997374, Fax: +30-610-992965
E-mail: vrahatis@math.upatras.gr

Business and scientific organizations, nowadays, own databases containing confidential information that needs to be analyzed, through data mining techniques, in order to support their planning activities. The need for privacy is imposed due to, either legal restrictions (for medical and socio-economic databases), or the unwillingness of business organizations to share their data which are considered as a valuable asset. Despite the diffusion of data mining techniques, the key problem of confidentiality has not been considered until very recently. In this chapter we address the issue of mining encrypted data, in order to both protect confidential information and to allow knowledge discovery. More specifically, we consider a scenario where a company having private databases negotiates a deal with a consultant. The company wishes the consultant to analyze its databases through data mining techniques. Yet the

company is not willing to disclose any confidential information.

Keywords: Data mining, cryptography, security, privacy.

1. Introduction

Nowadays business or scientific organizations collect and analyze data, orders of magnitude greater than ever before, in order to support their planning activities. Consequently, considerable attention has been paid in the development of methods that contribute to knowledge discovery in business or scientific databases, using data mining techniques.[11] The new generation of data mining techniques are now applied to a variety of real life applications ranging from recognizing scenes to stock market analysis. Specifically, mining financial data presents special challenges.

Usually, data mining rules can be used either to classify data into predefined classes that are described by a set of concepts-attributes (classification), or to partition a set of patterns into disjoint and homogeneous clusters (clustering), or to represent frequent patterns in data in the form of dependencies among concepts-attributes (associations). Data mining algorithms typically are based on systematic search in large hypotheses spaces.

Business or scientific databases contain confidential information. The need for privacy is either due to legal restrictions (for medical and socioeconomic databases) or due to the unwillingness of business organizations to expose their data, which are considered a valuable asset.

Despite the diffusion of data mining techniques, the key problem of confidentiality has not been addressed until very recently. In Ref. 7 ways through which data mining techniques can be used in a business setting to provide business competitors with an advantage, are presented. In Ref. 8 a technique to prevent the disclosure of confidential information by releasing only samples of the original data, independently of any specific data mining algorithm, is provided. In Refs. 2, 10 the authors propose to prevent the disclosure of confidential information, when association rules are to be extracted, by artificially decreasing the significance of these rules. In Ref. 13, the authors consider the scenario in which two parties owning private databases wish to run a classification data mining algorithm on the union of their databases, without revealing any confidential information. Similarly, in Ref. 9 the author addresses the issue of privacy preserving in distributed data mining, where organizations may be willing to share data mining association rules, but not the source data.

In this chapter we address the issue of mining encrypted data, in order

to both protect confidential information and to allow knowledge discovery. More specifically, we consider a scenario in which a company having private databases negotiates a deal with a consultant. The company wishes the consultant to analyze its databases using data mining techniques, yet it is unwilling to disclose any confidential information. We address the problem by encrypting the private data and allowing the consultant to apply the data mining techniques on the encrypted data. Then, the consultant provides the company with the extracted data mining rules. Finally, the company decrypts those rules before use. Note that the decrypted rules should be the same as the rules that would be extracted from the original data. We investigate the applicability of certain cryptography techniques to the above scenario when either classification, clustering or association rules are to be extracted.

It has to be mentioned that, in our approach, the sender coincides with the receiver. In other words the encoder and the decoder are the same. (The protocol: Alice composes the plaintext; Alice encrypts it; Alice sends the ciphertext for data mining processing; Alice receives the encrypted answer; Alice decrypts and recovers the answer). Thus, we intend to propose "an Alice to Alice" cryptography for privacy preserving data mining.

To this end, we encrypt the data by applying a proper cryptosystem. We focus on the main question, which is the choice of the appropriate cryptosystem by taking under consideration that each attribute value, no matter where it is located in the original table of data (plaintext) has to be encrypted with the same sequence of symbols in the ciphertext.

A direct solution is for the set of all possible attribute values to play the role of the alphabet and the cryptosystem to be "mono-alphabetic". Notice that, in real life applications, the plaintext has no content as well as the cardinality of the "alphabet" is very large which is a serious problem for the cryptoanalyst (enemy). Of course, in the case of a text in the English language, a mono-alphabetic cryptosystem is based just on a permutation of the 26 symbols and thus it cannot resist to a frequency analysis attack. However, the case of encrypting business data is complicated. Attribute values can represent customer characteristics, product codes, etc. The result of a frequency analysis attack is unpredictable. A mono-alphabetic cryptosystem may resist, but this is not guaranteed. Accepting this risk seems to be a bad premise to build on. On the other hand, the idea to develop a cryptosystem which is based on a permutation of k symbols ($1000 \leqslant k \leqslant 20000$) is primitive.

In this chapter, we propose an alternative methodology based on dis-

tributed data mining techniques. In particular, to each attribute from the original table of data (the plaintext) a set of possible encryptions is assigned $E_i = (c_{i1}, c_{i2}, \ldots, c_{ik})$. Of course $i \neq j$ implies $E_i \cap E_j = \emptyset$. Each time a_i changes to one of the $c_{i1}, c_{i2}, \ldots, c_{ik}$.

In the rest of the chapter we first present the proposed methodology and then we briefly discuss some preliminary issues concerning distributed data mining techniques. The chapter closes with some concluding remarks.

2. The Proposed Methodology

The main acting agents of the protocol are, "Alice" that represents a business or scientific organization and "Bob" that represents a data mining consultant who handles the data mining process. Alice owns a database with fields and field values that correspond to attributes and attribute values referred by the data mining rules. Attribute and attribute values may describe, for instance, a profit related behavior of the customers of a company that need to be classified/clustered or products sold together in a transaction that need to be examined for existing dependencies. Attribute values, irrespective of what they represent, have to be encrypted. We consider a great number of such attribute values denoted by $g_1, \cdots, g_M$ and we also set $G = \{g_1, \cdots, g_M\}$. Trivially, each g_i can be represented as an integer $i : 1 \leqslant i \leqslant M$ denoting its index. Alternatively, since each g_i has a label like "good customer" or "driver" or "tomato", this label can be transformed to an integer. For instance, a label can be transformed to a string of bits with the help of ASCII code, in turn, each string corresponds to a number (integer). As a result, in both of the above cases each g_i can be represented as a small integer.

The proposed methodology is as follows:

During the first step, Alice selects and preprocesses the appropriate data and organizes it into relational tables. A relational table is supposed to be two dimensional, however it can be represented as one dimensional considering it in a *row major order* or in a *column major order*.

During the second, third and fourth step, encryption takes place. We propose two different encryption techniques which are described in detail in subsections 2.1 and 2.2. Note that, both encryption techniques are based on symmetric keys $r_i, 1 \leqslant i \leqslant s$. The s different keys r_i are repeated periodically for every record of any Q_j. Thus, the m-th record of any Q_j is encrypted using the key r_i, where $i = m(\mathrm{mod}\, s) + 1$. It is this characteristic

encrypted data mining algorithm {
1. Alice collects data organized into relational tables $Q_1, Q_2, \cdots$
2. Alice obtains a "small" number of symmetric keys (or random numbers) $r_i, 1 \leqslant i \leqslant s$, e.g. $s = 4$
3. Alice obtains either a public key E and a private key D or a secret key X
4. Alice encrypts relational tables $Q_1, Q_2, \cdots$ as follows:
   $\{\ Q_j \rightarrow ENCRYPTION_{I or II} x(r_i(Q_j)) = C_j\ \}$
5. Alice sends $C_1, C_2, \cdots$ to Bob. Data mining performed. Bob returns the obtained rules to Alice
6. Alice decrypts the rules.
}

that, later, supports the decryption of the extracted rules using distributed data mining algorithms.

At the fifth step, Alice sends the encrypted tables to Bob. Bob applies the proper data mining algorithm to the encrypted tables and a number of data mining rules are extracted. Of course, attribute and attribute values appeared in the rules are encrypted. Then Bob returns these rules to Alice.

During the final step, Alice decrypts the rules. Of course, after decryption, there will be rules concerning the same attributes and attribute values which, however, extracted from different subsets of the initial table. Thus, Alice synthesizes the final set of rules combining the corresponding rules by using distributed data mining algorithms, as it will be described in subsection 2.3.

2.1. *Encryption Technique I – The RSA Cryptosystem*

The first encryption technique is based on the RSA cryptosystem.[16] Two large primes p and q are selected and their product $N = p \cdot q$ is computed. Then e and d, the public and private key, respectively, are selected. These keys have to satisfy the following relation:

$$e \cdot d \equiv 1 \bmod [(p-1) \cdot (q-1)].$$

By r_i, $1 \leqslant i \leqslant s$ we denote the s random numbers which are chosen for the encryption.

Assume that k is the least integer such that $g \leqslant 2^k$ for all $g \in G$. Then the r_i's have to satisfy:

(1) $\quad 0 < r_i \leqslant N - 1 - 2^k,$

(2) For all $i_1, i_2 : 1 \leqslant i_1, i_2 \leqslant s$ holds that $r_{i_1} \oplus r_{i_1} > 2^{k+1}$, where $\oplus$ denotes the exclusive or operator.

Encryption:

$$Q_j \longmapsto (r_i \oplus Q_j)^e \bmod N = C_j.$$

Decryption:

$$C_j \longmapsto (C_j^d \bmod N) \oplus r_i.$$

Remark 1: Condition (1) is required for the encryption and description processes to be invertible.

Remark 2: If $g_1, g_2 \in G$, $g_1 \neq g_2$ then the encryptions are different. On the contrary assume that:

$$(g_1 \oplus r_1)^e \equiv (g_2 \oplus r_2)^e \bmod N,$$

then

$$g_1 \oplus r_1 = g_2 \oplus r_2 \;\Rightarrow\; g_1 \oplus g_2 = r_1 \oplus r_2,$$

which is a contradiction since:

$$r_1 \oplus r_2 > 2^{k+1} \geqslant g_1 \oplus g_2.$$

2.2. *Encryption Technique II – Using a Symmetric Cryptosystem*

The second encryption technique is based on the Discrete Logarithm Problem.[14] Let p be a large prime, $g < p$ for $g \in G$. By x we denote Alice's secret key, $0 < x \leqslant p-2$. By $r_i, 1 \leqslant i \leqslant s$ we denote s random symmetric keys, $0 < r_i \leqslant p-2$.

In the case of Q_j being an entry of the table consider:

Encryption:

$$Q_j \longmapsto Q_j^{r_i \cdot x} \bmod p = C_j.$$

Decryption:

$$C_j \longmapsto C_j^{(r_i \cdot x)^{-1}} \bmod p = Q_j.$$

Remark 3: For α a primitive element $\bmod p$ consider the pair $(\alpha^r, Q^{r \cdot x})$. Although it contains some "partial" information related to the random key r, r cannot be recovered from α^r.

Remark 4: Assume that $Q_1 \neq Q_2$. Then

$$(\alpha^{r_1}, Q_1^{r_1 \cdot x}) \neq (\alpha^{r_2}, Q_2^{r_2 \cdot x}).$$

2.3. *Distributed Data Mining*

Currently, data mining systems focus on real life problems that usually involve huge volumes of data. Therefore, one of the main challenges of these systems is to devise means to handle data that are substantially larger than available main memory on a single processor. Several approaches to this problem, are reported in the literature.

An obvious approach is to parallelize the data mining algorithms. This approach requires the transformation of the data mining algorithm to an optimized parallel algorithm suited to a specific architecture (e.g. Refs. 1, 12, 19). Another approach is based on windowing techniques, where data miners are supplied with a small subset of data, having a fixed size, the *window* (e.g. Refs. 5, 15, 18). Iteratively, the window is updated with new data of the remaining set, until a predefined accuracy is met.

An alternative approach is based on partitioning the initial data set into subsets, applying a data mining algorithm in parallel to these subsets and synthesizing the final data mining rules from the partial results.

For instance, in Ref. 6 a classification technique is proposed, called meta-learning, that exploits different classifiers supplied with different subsets of the data, in parallel. Then, partial results are combined, in the sense that the predictions of these classifiers are used to construct the final prediction. In Ref. 4, a classification methodology is proposed, for combining partial classification rules. It is, actually, a two-phase process. First, a number of classifiers are trained, each with a different subset of the data. Then, the trained classifiers are used in the construction of a new training data set, substantially smaller than the initial one. The latter data set is used to train the final classifier through an iterative process, that is guided by thresholds concerning the size of this data set and the achieved accuracy. In Ref. 3 an iterative clustering process is proposed that is based on partitioning a sample of data into subsets. In a first phase, each subset is given as an input to a clustering algorithm. The partial results form a dataset that it is partitioned into clusters, the meta-clusters, during a second phase. Under certain circumstances, meta-clusters are considered as the final clusters. Finally, in Ref. 17 an algorithm for extracting association rules is presented that is based on a logical division of the database into non-overlapping partitions. The partitions are considered one at a time and all associations for that partition are generated. Then, these associations are merged to generate a set of all potential associations. Finally, the actual associations are identified.

The latter approach, based on partitioning the initial data set, can be applied during the last step of the proposed methodology that concerns the decryption of the encrypted data mining rules. As mentioned earlier, the m-th record of any Q_j is encrypted using the key r_i, where $i = m(\mathrm{mod}\, s) + 1$. Thus, every Q_j can be partitioned into s subsets, where in every subset any g_k is encrypted using the same r_i. Notice, also, that the encryption of any g_k of a subset is different its encryptions in different subsets. After decrypting the extracted data mining rules, the obtained rules, partitioned by the subset they originated from, are identical to the partial rules that would be obtained by partitioning the initial data set into s subsets and applying a data mining algorithm in parallel to these subsets. Therefore, the key idea is that the rules obtained from each subset, after decryption, can be combined in order to construct the final set of rules, by using the distributed data mining algorithms mentioned above. Thus, Alice will obtain the final set of rules without revealing any confidential information to Bob.

3. Conclusions and Future Research

We have proposed a novel methodology for mining encrypted data. Such a methodology is very useful when the owner of the data is wishes to prevent the disclosure of confidential information.

Various cryptosystems have been tested. The obvious solutions based on a mono-alphabetic cryptosystem may not resist a frequency analysis attack. We have proposed two alternatives that perfectly fit with the problem requirements. In both cases, the decryption phase is based on distributed data mining algorithms.

Notice that distributed data mining algorithms may not maintain the accuracy that would be achieved by a simple data mining algorithm supplied with all the data. In other words, Alice may obtain rules which are not so accurate as they would be if she was not using distributed data mining algorithms, for instance in the case of mono-alphabetic cryptosystems. However, the loss in accuracy (if any) is usually small enough. Thus, the proposed methodology is considered to be acceptable.

References

1. R. Agrawal and J.C. Shafer. Parallel Mining of Association Rules: Design, implementation and experience. *IEEE Trans. on Knowledge and Data Engineering*, **8**(6):962–969 (1996).
2. M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios. Dis-

closure Limitation of Sensitive Rules. *Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange*, Chicago, 45–52 (1999).
3. B. Boutsinas and T. Gnardellis. On Distributing the clustering process. *Pattern Recognition Letters*, Elsevier Science Publishers B.V., **23**(8), 999-1008 (2002).
4. B. Boutsinas, G. Prassas, and G. Antzoulatos. On scaling up classification algorithms, submitted (2002).
5. P.S. Bradley, U.M. Fayyad , and C. Reina. Scaling Clustering Algorithms to Large Databases. *Proceedings of the 4th Int. Conf. on Knowledge Discovery and Data Mining*, 9–15 (1998).
6. P. Chan and S. Stolfo. Meta-learning for multistrategy and parallel learning. *Proceedings of the 2nd Int. Work. Multistrategy learning*, 150–165 (1993).
7. C. Clifton and D. Marks. Security and Privacy Implication of Data Mining. *Proceedings of the 1996 ACM Workshop on Data Mining and Knowledge Discovery* (1996).
8. C. Clifton. Protecting against Data Mining through Samples. *Proceedings of the 13th IFIP Conference on Database Security*, Seattle, Washington (1999).
9. C. Clifton. *Privacy Preserving Distributed Data Mining.* (2001)
10. E. Dasseni, V. Verykios, A. Elmagarmid, and E. Bertino. Hiding Association Rules by Using Confidence and Support. *LNCS* 2137, 369-383 (2001).
11. U.M. Fayyad, G. Piatetsky-Shapiro and P. Smyth. *Advances in Knowledge Discovery and Data Mining.* AAAI Press/MIT Press (1996).
12. X. Li and Z. Fang. Parallel clustering algorithms. *Parallel Computing*, **11**, 275–290 (1989).
13. Y. Lindell and B. Pinkas. Privacy Preserving Data Mining. *Advances in Cryptology- CRYPTO '00, LNCS* 1880, 36-53 (2000).
14. S.C. Pohlig and M. Hellman. An Improved Algorithm for Computing Logarithms over $GF(p)$ and its Cryptographic Significance. *IEEE Transactions on Information Theory*, **24**, 106–110 (1978).
15. F. Provost and V. Kolluri. Scaling Up Inductive Algorithms: An Overview. *Proceedings of the 3rd Proceedings of the Knowledge Discovery and Data Mining*, 239–242 (1997).
16. R. Rivest, A. Shamir, and L. Adlemann. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM*, **21**, 120–126 (1978).
17. A. Savasere, E. Omiecinski, and S. Navathe. An Efficient Algorithm for Mining Association Rules in Large Databases. *Proceedings of the 21th IEEE International Conference on Very Large Databases* (1995).
18. H. Toivonen. Sampling large databases for finding association rules. *Proceedings of the 22th IEEE International Conference on Very Large Databases*, India, 134–145 (1996).
19. X. Zhang, M. Mckenna, J. Mesirov, and D. Waltz. An efficient implementation of the backpropagation algorithm on the connection machine CM-2. Tech. Rep. RL89-1, Thinking Machines Corp. (1989).