Privacy Preserving Tree Augmented Naïve Bayesian Multi–party Implementation on Horizontally Partitioned Databases

Maria Eleni Skarkala, Manolis Maragoudakis, Stefanos Gritzalis, and Lilian Mitrou

Department of Information and Communication Systems Engineering, University of the Aegean, Karlovassi, Samos 83200, Greece
{mes,mmarag,sgritz,l.mitrou}@aegean.gr

Abstract. The evolution of new technologies and the spread of the Internet have led to the exchange and elaboration of massive amounts of data. Simultaneously, intelligent systems that parse and analyze patterns within data are gaining popularity. Many of these data contain sensitive information, a fact that leads to serious concerns on how such data should be managed and used from data mining techniques. Extracting knowledge from statistical databases is an essential step towards deploying intelligent systems that assist in making decisions, but also must preserve the privacy of parties involved. In this paper, we present a novel privacy preserving data mining algorithm from statistical databases that are horizontally partitioned. The novelty lies to the multi-candidate election schema and its capabilities of being a basic foundation for a privacy preserving Tree Augmented Naïve Bayesian (TAN) classifier, in order to obviate disclosure of personal information.

Keywords: Privacy, Distributed data mining, Horizontally partitioned databases, Tree Augmented Naïve Bayes, Homomorphic encryption.

1 Introduction

Technological progress has led to ever-increasing storage, retrieval and processing of data collections stored in large–scale databases. Statistical databases with financial, medical or social data have usually been exploited for analysis and discovery of useful patterns. Database owners wish to share the data contained therein, on the premise there is no leakage of sensitive information. A crucial issue arises when this information can be misused for various reasons in favor of some expectant aggressors [8]. Data is usually distributed across several parties, thus the usage of secure protocols is required for sharing information. In order to efface possible disclosure of sensitive data, resulting in privacy violation while data mining processes are applied, various techniques have been proposed. A privacy preserving data mining algorithm should support the following features; prevent disclosure of sensitive information, resist to potential security holes that many traditional data mining algorithms pose, not degrade access and use of non-sensitive information, be useful for large volume of data and not have exponential computational complexity. An algorithm to be effective

S. Furnell, C. Lambrinoudakis, and G. Pernul (Eds.): TrustBus 2011, LNCS 6863, pp. 62–73, 2011.

must simultaneously manage large number of participants owning large databases, but ensuring at the same time that personal data are not revealed to other parties, or to a trusted third party (Miner). Involved databases can be either horizontally [10, 12, 24, 28] or vertically [20, 23] partitioned, with each party holding its own sensitive data. In the former case each party holds a different set of records but a unified set of attributes, while in the latter case, different sets of attributes for the same recordset are distributed to participants [1]. The parties involved are considered to be mutually mistrustful and in some cases are curious to learn information about other participant's data. If a party does not deviate from a protocol during its execution and sends its data, then it is considered semi-honest, but in case it sends specific inputs in order to discover other participant's data, it is considered malicious. In real world applications, the former case behaviors are more often. The problem of privacy preservation has been addressed in different ways such as randomization, perturbation and k-anonymity. Several techniques that have been proposed using data encryption are based in the idea of Yao [25]. Secure multi party computation [11], an extension of Yao's idea, is also widely used to prevent leakage of any information other than the final results. The contribution of cryptography is essential as the original data are not transformed in any way, like randomization or transformation methods do, a fact that can lead to inaccurate outcomes [28].

In this paper, we present a novel protocol which utilizes a robust Bayesian algorithm that unlike the widely used Naïve Bayesian classifier is not based on the unrealistic assumption about the independence of attribute variables given the class. More specifically, we propose the privacy preserving version of the Tree Augmented Naïve Bayesian classifier used by our protocol which aims to extract global information from statistical databases horizontally partitioned. In comparison with work [28] which uses only binary attributes' type, our approach uses databases containing numerical (included binary) but also nominal data. The protocol presented in this work was developed in a client-server (C2S) environment and the participants can only be connected with the Miner, making communication among them unfeasible. Privacy is preserved using cryptographic techniques exploiting homomorphic primitive first proposed by Yang [24], through which the Miner who collects the data of at least three semi-trusted parties is unable to identify the original records. Because of some "curiosity", the protocol requires the existence of at least three participants in order to maintain privacy, as to the subsequent analysis of the results sensitive information in the model of two parties may be leaking. The contribution of the present work lies within the exploitation of a variation of the multi-candidate selection model, used for mining frequencies of attribute-class vectors in a secure and efficient manner, and inspired from the work of [14]. We stress that the protocol presented is a sketch of a generic privacy preserving data mining scheme in the fully distributed setting, where there are k-out-of-l selections [4].

The rest of the paper is structured as follows. Next section addresses previous work on the topic. Section 3 introduces the security and design requirements and analyzes the proposed protocol, while section 4 presents the results from experiments carried out and the total evaluation of the protocol. This paper closes with some conclusions drawn from this study.

2 Related Work

A categorization of privacy preserving data mining algorithms is presented in [22]. The algorithms were categorized in five segments; apportionment of data, modification of data, data mining algorithm for which the privacy technique has been designed for, type of data that need to be protected from disclosure and technique adopted for the preservation of privacy. The authors in [1] explored various methods such as randomization, k-anonymity and transformations for hiding personal information. Randomization and cryptography as privacy preservation techniques have been studied widely by researchers. The first method was used in association rules [20] and decision trees [3] for vertically and horizontally partitioned data respectively. The technique proposed in [3] was based on the reconstruction of data. The privacy provided is measured by the facility of finding the factual data of a modified attribute. This measure is suffering from inconsistencies with regards to the distribution of the actual and the transformed data. Instead, the work [2] uses the entropy of information as a measure of privacy and thus solves the problem in [3].

The second method was applied in models which most of them are based in the idea of Yao [25], extended by Goldreich [11] who studied the secure multi-party computation problem (SMC). This approach is widely used in distributed environments, where parties wish to estimate through a function, with their data as input, the final mining results, but in a way that privacy is ensured. Cryptographic techniques have been applied for horizontally [10] partitioned databases to build decision trees [13, 17], Naive Bayesian classifiers [12, 24, 26, 21] and Association Discovery Rules [10], or for vertically partitioned data to construct association rules [7, 23] and Naïve Bayesian classifiers [21]. One technique based on cryptography proposed in [10], where the first participant transmits a number of frequencies and a random value to her neighbor, encrypted. Another method proposed by Clifton [6] on distributed environments, is the local execution of data mining methods. Then the results are sent to a trusted third party who combines the results of each participant to obtain the final results. This technique, however, can lead to inaccurate outcomes [12]. Naive Bayes classification was employed in many researches [12, 24, 26] because of its simplicity and straightforward approach. Simplified Bayesian Networks have also been used for data mining processes by either applying the Tree Augmented Naïve Bayes (TAN) [28] or K2 [23] algorithm as structure search methods. The authors in [28] use an algebraic technique to perturb original data. Our approach uses cryptographic techniques to build a simplified Bayesian Network using TAN as search algorithm. Such networks are considered more efficient in relation to Naïve Bayes classifiers as they take into account the dependency among databases' attributes.

A tool used in the literature is the homomorphic primitive first used in the work of Yang et.al. [24]. Our protocol employs the Paillier cryptosystem [16] in which this primitive is applied, which can assure both privacy and accurate results. As a conclusion, while randomization methods are efficient, on the other hand they are not completely secure, and the results can be inaccurate. Unlike, cryptographic methods are secure and the results are more accurate, but are lagging in terms of efficiency.

3 System Description

Privacy preservation emerges nowadays, as collections of data are daily exchanged. Data mining techniques that used to derive statistics from distributed databases must ensure that personal data will not be disclosed. This work aims to develop a mining algorithm which extracts accurate results, while privacy is preserved using efficient encryption that satisfies the essential security requirements. A Miner contributes in the creation of a classification model by collecting from at least three parties the overall frequencies of each value per attribute in relation to each class value. The attributes' type can be either numerical (binary data are included) or categorical. These frequencies are encrypted using asymmetric cryptography [16] which exploits the homomorphic primitive, ensuring that sensitive data remain secret and only aggregated results can be exported. The communication among parties is infeasible as the only data flow is between each party and the Miner, in order to prevent any collusion attacks. As mentioned, much of the work is based on the study of [14] and thus, many of the motivating features used as well as theoretical requirements to be met spring up from theirs quotations.

3.1 Tree – Augmented Naïve Bayesian Classifier

The objective and the novelty of this work is the development of a protocol through which global information is extracted using the Tree Augmented Naive Bayesian algorithm [5]. The traditional Naive Bayes algorithm computes the conditional probability of each attribute A_i given the class C during training. When classifying, the Bayes theorem is applied thereafter, to compute the probability of C given a particular instance vector <$A_{1......}$ A_n>, where n is the total number of attributes. This classifier assumes that all attributes are independent given the value of C, an over restrictive and often unrealistic assumption. In order to improve the performance of such classifiers is necessary to alleviate the issue of the independence assumption. Bayesian Networks exploit possible dependencies among attributes in order to compute more efficiently the Bayesian probabilities. However, unrestricted Bayesian networks are not very successful classifiers [15] since they do not have any prior knowledge on the class variable when they are learned from data, resulting in network structures that do not favor classification. An interesting variation of Bayesian networks is the Tree – Augmented Naïve Bayesian classifier (TAN). TANs usually behave more robust as regards to classification since they combine the initial structure of the Naïve Bayes algorithm. This classifier allows the existence of additional edges between attributes that represent the relations among them. The full TAN structure is depicted in Figure 1.

In a TAN network the class variable has no parents and each attribute has as parents the class variable and at most one other attribute. In an augmented structure, an edge from attribute A_i to A_j implies that the influence of A_i on the assessment of the class variable also depends on the value of A_j. The procedure for learning these edges, which is based on a method proposed by Chow and Liu [5], reduces the problem of constructing a maximum likelihood tree to finding a maximal weighted spanning tree in a graph. The problem of finding such a tree is to select a subset of arcs such that the sum of weights attached to the selected arcs is maximized.

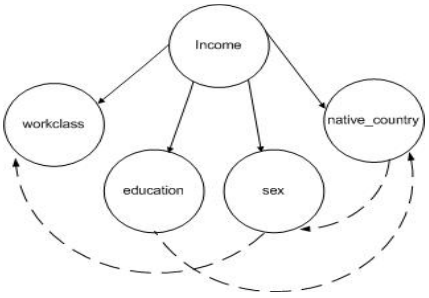

Fig. 1. TAN structure created from simplified dataset “Adult” [19]

The procedure consists of four main steps. At first, for each attribute pair the mutual information is computed using Equation 1, measuring how much information the attribute y provides about x. In the second step an undirected graph is built in which the vertices are the variables in x (the weight of an edge connecting two attributes). In the third step a maximum weighted spanning tree is created while in the final step the undirected tree is transformed to a directed one by choosing a root variable and setting the direction of all edges to be outward from it.

$$I_p (X;Y) = \Sigma_{x,y} P(x, y) \log p(x, y)/p(x)p(y) \tag{1}$$

TAN results are significantly improved over those produced by the classical Naive Bayes classifier and the Bayesian networks as the assumption of independence is removed, but at the same time the robustness and computational complexity are maintained, showing better accuracy [9].

3.2 Paillier Cryptosystem

Paillier algorithm [16] is used for encryption and exploits the additive homomorphic primitive [24] achieving privacy and un-linkability between data and participants’ identities. The Miner and each participant create, at their own side, a key pair of 1024 bits size. The public key of each user is the product N of two random prime numbers (N=p*q), and a random number g which belongs to $Z^*_{n^2}$. The private key is the result of two equations, *lambda* and *mu* (Equation 2 & 3).

$$\text{Lambda} = \text{lcm}(p-1, q-1) = (p-1)*(q-1)/\text{gcd}(p-1, q-1) \tag{2}$$

$$\text{mu} = (L(g^{\text{lambda}} \bmod N^2)^{-1} \bmod N) \text{ where } L(u) = (u-1)/N \tag{3}$$

Generally, if a party j wishes to send frequency i, encrypts the message with the Miner’s public key. Paillier encryption is performed as shown in Equation 4, where M is a random value produced at the Miner’s side and send to each participant encrypted.

$$E[m_{i,j}] = g^{M^{\wedge}i} x^N \pmod{N^2} \tag{4}$$

When at least three participants have sent their data to the Miner, homomorphic primitive is used to calculate the total frequencies by decrypting all the messages received at once. The decrypted message can be written as presented in Equation 5.

$$T = a_0M^0 + a_1M^1 + \ldots\ldots + a_{l-1}M^{l-1}(\mathrm{modN}) \tag{5}$$

3.3 Security Requirements and Possible Threats

In distributed environments every party is considered to be either semi-honest or malicious. Semi-honest adversaries follow the protocol specifications, they do not collude but are curious to learn more information during the execution of the protocol. Malicious adversaries can be either internal or external. Internals deviate from the protocol and send specific inputs in order to infer other parties' private data. An external one tries to impersonate a legal party and then behave as an internal. In order to confront such behaviors in our scheme, external adversaries cannot participate as every party has to send her digital signature which is assigned by a Certification Authority. The Miner and each participant are *mutually authenticated*, thus unauthorized users are excluded and the authorized ones are connected with the literal server indisputably. Internals are restricted to send blank inputs, or missing values, so are not able to gain any further information other than the final results. Three clients must participate in order to prevent any probing attacks and revelation of other participants' data. Semi-honest adversaries cannot learn more information from the final results as they cannot communicate and collude with each other. Data are transmitted only among the Miner and each client. The case in which parties collaborate outside the protocol is not considered in the present work. *Privacy* can be preserved if the requirements of *confidentiality*, *anonymity* and *un-linkability* are fulfilled. Using asymmetric encryption all data exchanged among one party and the Miner are encrypted and only the participant for whom the message is intended for can decrypt it, so eavesdropping attacks or data leaking is infeasible. Anonymity and un-linkability can be achieved as through homomorphic primitive the Miner cannot identify which participant submits specific inputs to the system. The Miner could also be considered as an internal adversary, if he tries to decrypt partial transmitted messages resulting to privacy violation. This is infeasible as the Miner can decrypt only the overall distributions. Paillier cryptosystem at its initial mode is vulnerable to chosen plaintext attacks. The usage of a random variable (in our scheme M value) is important to confront such attacks. *Integrity* mechanisms are implemented in case any active attacker tries to modify the transmitted messages, and cause variations to the final results or even disclosure of sensitive data.

3.4 Protocol Analysis

The proposed protocol combines both privacy preservation, through Paillier cryptosystem that follows the homomorphic model, and data mining capabilities in a fully distributed environment. Our approach is based on the classical homomorphic election model, and particularly on an extension for supporting multi-candidate elections, where each participant has k-out-of-l selections [4]. Both the Miner and each participant possess a key pair for creating digital signatures, in order to be mutually authenticated, which is used only in the first phase of the protocol and is generated by a Certificate Authority. We assume that the Miner is able to obtain all the public keys of each participant, and each participant can retrieve the Miner's public key. All transmitted messages are encrypted with the keys created during the key generation phase of Paillier cryptosystem, and each one includes a SHA-1 digest to confirm that no modification has been accomplished. Figure 2 presents the main procedures that are being carried out by the protocol using the notations given in Table 1. The Miner regroups all data sent by the participants (clients) of the protocol. His purpose is focused on building a TAN classifier in order to extract the final results by finding the correlations among the attributes and the network structure that represents them. These results will be sent later to each one of the three clients who participated in the creation of the mining model.

Initially, the Miner generates the encryption key pair (S_{pu} and S_{pr}) through Paillier key generation phase and an RSA key pair (S_{Dpu} and S_{Dpr}) of 1024 bit and uses MD5 hash function to create the digital signature. We assume that a client is able to obtain the public key S_{Dpu}. Then a random number M is generated which will be sent, in later phase encrypted to every client that is aware of the Miner's password. This variable is used during the encryption of the sensitive data. When a client request connection with the Miner she sends her public key C_{pu} and her digital signature encrypting the C_{pu} key with her private key C_{Dpr}. The Miner decrypts the digital signature with the client's public key C_{Dpu} obtained by the Certificate Authority and creates a digest of the message send (C_{pu}). If the Miner verifies the client's identity, stores the public key C_{pu} of the client. In return the Miner sends his public key S_{pu} and his digital signature encrypted with his private key S_{Dpr}. The client decrypts the Miner's signature with his public key S_{Dpu} and creates a digest of the S_{pu} key, and is now able to verify that she is connected to the legal server. Afterwards she stores the public key S_{pu}. The purpose of this phase is to prevent any unauthorized access to the system. After the completion of this phase, temporary identities are given to each client. In order to proceed to the mining process, the client must be aware of the Miner's password. If she provides the correct one she can possess the random variable M. Once the mutual authentication process is completed the Miner can gather the clients' personal data. A client can participate in the exportation of statistics giving her personal data. However, requires the sensitive data contained in the database can be disclosed, in the notion of verbatim records, neither to the Miner nor to other participants, nor to a malicious one not involved in the protocol. Every client that consents to the creation of the classification model sends each value of the class and each value of every attribute subsequently. These messages are encrypted with the Miner's public key $S_{pu.}$ This procedure is necessary for the Miner to initialize the TAN classifier, after the collection of at least three clients' data. After the completion of the model initialization, the Miner inquires about the frequencies that correspond to the first attribute. Each client sends the count of every value for this specific attribute in relation to every class value. This count is encrypted using the random variable M and the message send contains also the name of the attribute value and the name of the class value that the count is related with.

When the Miner has collected the data from the three clients he proceeds to their decryption, applying homomorphic primitive. Later on, the Miner asks the frequencies for the next attribute and this process is continuing for all attributes. Then the Miner creates the classification model, as described in section 3.1., and the final results are sent to all participating clients.

Table 1. Notations used in protocol

S_{pu}	Server's public key for encryption/decryption
S_{pr}	Server's private key for encryption/decryption
C_{pu}	Client's public key for encryption/decryption
C_{pr}	Client's private key for encryption/decryption
H(m)	SHA-1 hash of message m
Enc{m}k	Encryption of message m with key k
Decr{m}k	Decryption of message m with key k
S_{Dpr}	Server's private key for digital signature
S_{Dpu}	Server's public key for digital signature
C_{Dpu}	Client's public key for digital signature
C_{Dpr}	Client's private key for digital signature

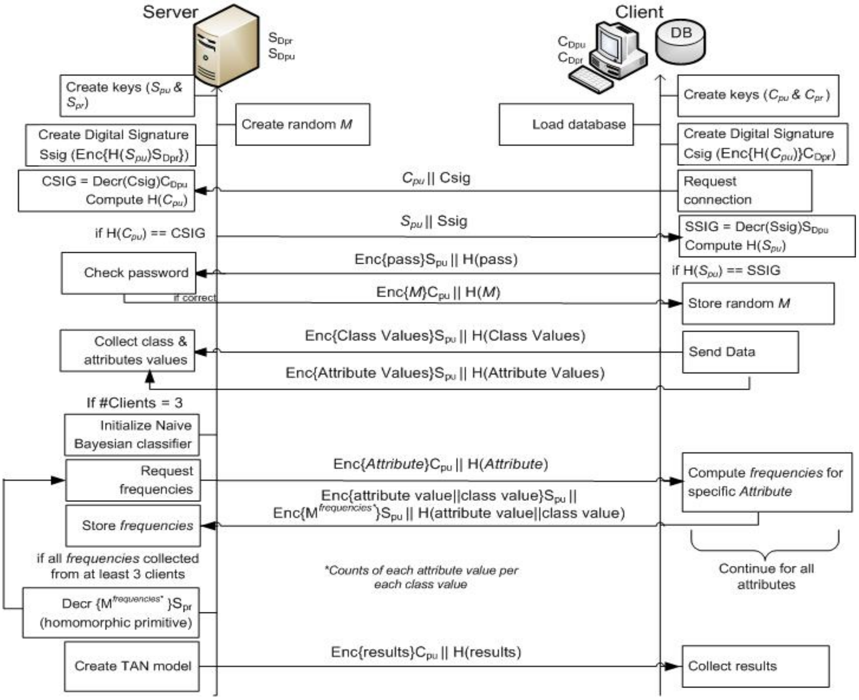
Server
S_{Dpr}
S_{Dpu}
Client
C_{Dpu}
C_{Dpr}
DB
Create keys (S_{pu} & S_{pr})
Create random M
Load database
Create keys (C_{pu} & C_{pr})
Create Digital Signature Ssig (Enc{H(S_{pu})S_{Dpr}})
Create Digital Signature Csig (Enc{H(C_{pu})}C_{Dpr})
CSIG = Decr(Csig)C_{Dpu} Compute H(C_{pu})
C_{pu} || Csig
Request connection
if H(C_{pu}) == CSIG
S_{pu} || Ssig
SSIG = Decr(Ssig)S_{Dpu} Compute H(S_{pu})
Check password
Enc{pass}S_{pu} || H(pass)
if H(S_{pu}) == SSIG
if correct
Enc{M}C_{pu} || H(M)
Store random M
Collect class & attributes values
Enc{Class Values}S_{pu} || H(Class Values)
Send Data
Enc{Attribute Values}S_{pu} || H(Attribute Values)
If #Clients = 3
Initialize Naive Bayesian classifier
Request frequencies
Enc{Attribute}C_{pu} || H(Attribute)
Compute frequencies for specific Attribute
Store frequencies
Enc{attribute value||class value}S_{pu} || Enc{$M^{frequencies^*}$}S_{pu} || H(attribute value||class value)
if all frequencies collected from at least 3 clients
*Counts of each attribute value per each class value
Continue for all attributes
Decr {$M^{frequencies^*}$}S_{pr} (homomorphic primitive)
Create TAN model
Enc{results}C_{pu} || H(results)
Collect results

Fig. 2. Messages exchanged during execution of protocol

4 Evaluation

In this section, the main procedures of our protocol are evaluated as we aim to demonstrate that they portray a low level of computation time but at the same time privacy is preserved. We measure the time needed for the key generation phase, the mean authentication time and login time. In order to measure the performance of our scheme we use three scenarios in which three clients participate and the number of records and attributes varies, and compare the secure modes with the corresponding insecure modes of our proposal. For the secure versions we measure the mean encryption and decryption time of messages exchanged. In section 4.2 we present an evaluation of the TAN classifier using Recall and Precision variables as metrics. The experiments were conducted to a computer system with Intel Core 2 Duo T5750 processor at 2.00 GHz, with 3GB DDR2 RAM. The operating system of the machine is MS Windows 7.

4.1 Experiments

For the measurement of mean authentication time, mean login time and mean key generation time we collected measurements from 50 runs. The mean times are presented in milliseconds (ms). Key generation phase includes the encryption key pair generation and the creation of the RSA digital signature. We assume that each client knows the Miner's S_{Dpu} key and the Miner is aware of all public keys C_{Dpu} of the clients. A client requires 513 ms and 133 ms to create the encryption key pair and the digital signature, respectively. The Miner requires 465 ms for the encryption keypair, 45 ms to create the digital signature, and 68 ms to generate the random variable M. In total, 289 ms are required for the key establishment phase. As authentication time we mean the time that is needed for the Miner and each client to mutually be authenticated. From the measurements we calculate that 29 ms are needed for the mutual authentication and 289 ms for the login phase. In this phase the client sends the Miner's password encrypted with the S_{pu} key and the Miner in return respond with the correctness of the password received by sending the random variable M. The mean login time is significant larger than the mean authentication time as decryption and encryption operations are involved.

In order to evaluate our protocol we examined three scenarios to compute the time that is required for the completion of some main procedures. For each scenario we compare the secure mode and an insecure one. The insecure mode is similar to the secure one described in section 3.4 without the authentication phase and the encryption and decryption procedures. Every message is send in cleartext and the client is authenticated only by providing the correct password to the Miner. We use three different scenarios, each one including different number of attributes and different number of records, and three clients are involved to each setting. The horizontally partitioned database used for these experiments comes from a real dataset [19], which is tailored for each case. For the first scenario each client's database consist of 50 records and 5 attributes, in the second scenario it consists of 100 records and 5 attributes and in the third scenario 100 records and 10 attributes are involved. These sets were selected in order to compare if the performance of the protocol depends on the number of attributes and records. From the outcomes, which are presented in Table 2 in comparison with each insecure scenario, as it was expected the mean times are smaller for the insecure versions as encryption and decryption operations are not involved. The classifier initialization time is low and is increased only if the number of attributes is growing. The higher time is when the Miner collects the frequencies for each attribute from all the clients, and is affected when the number of instances is raising. Regarding to the creation of the TAN model, the time is increased when the number of instances is growing. The time to send the final results to all clients is raising both in cases the number of instances and attributes is increased, but the exact opposite happens to the insecure scenarios. We can conclude that the overall time to complete all the steps is determined by the mining process time which is mostly increased when the number of the attributes is growing.

Because of the different number of characters that have been encrypted and decrypted during the execution of the protocol, we collected from the above scenarios the mean times of all messages being exchanged and evaluate them. The average time to encrypt a message is equal to 60 ms. Similar results obtained for the decryption time, as 70 ms is the average time that resulted. We conclude that the mean times are low, so the Paillier cryptosystem is not only effective but also efficient.

Table 2. Experiments' results

Basic Procedures	**1st scenario**	**1st Insecure**	**2nd scenario**	**2nd Insecure**	**3rd scenario**	**3rd Insecure**
Classifier initialization	13	14	16	16	30	22
Mining Process	31777	563	35502	559	94793	3069
BN creation	39	19	117	29	68	171
Final results	2407	5,4	4258	2,2	4476	1,2

4.2 Classifier Evaluation

The evaluation of the TAN classifier is also an important affair. In order to examine the classifier created by the Miner we calculate two variables, Recall and Precision. Variable Recall is the percentage of records categorized with the correct class in relation to the number of all records with this class. Variable Precision is the percentage of records that have truly a certain class over all the records that were categorized with this class.

Table 3. Classifier evaluation results

	Naïve Bayes Classifier						**TAN classifier**					
Records	**1000**		**2000**		**5000**		**1000**		**2000**		**5000**	
Correct	49		49		50		54		55		56	
Incorrect	51		51		50		46		45		44	
ClassValue	**<=50**	**>50**	**<=50**	**>50**	**<=50**	**>50**	**<=50**	**>50**	**<=50**	**>50**	**<=50**	**>50**
Recall	0,42	0,54	0,48	0,52	0,50	0,8	0,42	0,63	0,52	0,6	0,54	0,6
Precision	0,43	0,53	0,77	0,23	0,47	0,2	0,48	0,57	0,73	0,38	0,73	0,39

Three different sets of data were used as training sets each one holding 1000 records, 2000 records and 5000 records with 14 attributes. As test set, 10% of the training records were kept off the training phase. Our aim is to figure if the created model classifies correctly and more accurate, in relation to the Naïve Bayes classifier, the given records. The results are presented in Table 3. The evaluation has presented fairly good results, from which we can presume that as the training set is getting larger, the Miner classifies more instances correctly, and the TAN model shows better accuracy in relation to the Naïve Bayes model.

5 Conclusion

Classification is considered as a key factor for the detection of hidden information within voluminous data being exchanged. The data stored in databases often contain sensitive information, so possible disclosure during mining processes can compromise fundamental rights of individuals such as privacy or the right to be free from discrimination. This problem can be solved using privacy preserving TAN classifier, which was the purpose of the present work. A protocol developed in a C2S environment where data are horizontally partitioned to participants. Communication among them is infeasible and the only data flow is between a Miner and each participant. Data exchanged during the execution of the protocol, which are the incidences of each attribute value in relation to each class value, are encrypted. The Miner is not in position to know which one of the participants has sent specific frequencies, and so each party remains anonymous. This is achieved by exploiting the homomorphic primitive, and by decrypting the data from at least three parties who consented to participate to the mining process. The data are also been examined for modifications during transmission. From experiments conducted we conclude that the proposed protocol is effective but also efficient. From the security perspective, the cryptographic approach is considered the most appropriate in terms of accuracy of the final results, because the data are not altered in any mode and therefore accurate results are extracted. It was demonstrated that the proposed protocol is safe from possible disclosure of sensitive information. This work focused primarily on ensuring the informational privacy of persons related with.

References

1. Aggarwal, C.C., Yu, P.S.: A General Survey of Privacy-Preserving Data Mining Models and Algorithms. In: Aggarwal, C.C., Yu, P.S. (eds.) Privacy-Preserving Data Mining, pp. 11–52. Springer, US (2008)
2. Agrawal, D., Aggarwal, C.: On the Design and Quantification of Privacy Preserving Data Mining Algorithms. In: 12th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 247–255. ACM, New York (2001)
3. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: 2000 ACM SIGMOD Conference on Management of Data, vol. 29(2), pp. 439–450 (2000)
4. Baudron, O., Fouque, P.-A., Pointcheval, D., Stern, J., Poupard, G.: Practical multi-candidate election system. In: PODC 2001: Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing, pp. 274–283. ACM, New York (2001)
5. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory 14, 462–467 (1968)
6. Clifton, C.: Privacy Preserving Distributed Data Mining. In: 13th European Conference on Machine Learning, pp. 19–23 (2001)
7. Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., Zhu, M.Y.: Tools for Privacy Preserving Distributed Data Mining. ACM SIGKDD Explorations 4(2), 28–34 (2002)

8. Clifton, C., Marks, D.: Security and Privacy Implications of Data Mining. In: Proceedings of the 1996 ACM SIGMOD Workshop on Data Mining and Knowledge Discovery, Montreal, Canada, pp. 15–19 (1996)
9. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning 29(2-3), 131–163 (1997)
10. Kantarcioglu, M., Clifton, C.: Privacy preserving distributed mining of association rules on horizontally partitioned data. IEEE Transactions on Knowledge and Data Engineering 16(9), 1026–1037 (2004)
11. Goldreich, O.: Secure multi-party computation. Working Draft (1998)
12. Kantarcıoglu, M., Vaidya, J.: Privacy Preserving Naive Bayes Classifier for Horizontally Partitioned Data. In: IEEE ICDM Workshop on Privacy Preserving Data Mining, pp. 3–9 (2003)
13. Lindell, Y., Pinkas, B.: Privacy Preserving Data mining. Journal of Cryptology 15(3), 177–206 (2002)
14. Magkos, E., Maragoudakis, M., Chrissikopoulos, V., Gritzalis, S.: Accurate and Large-Scale Privacy-Preserving Data Mining using the Election Paradigm. Data and Knowledge Engineering 68(11), 1224–1236 (2009)
15. Mitchell, T.: Machine Learning. McGrawHill, New York (1997)
16. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
17. Pinkas, B.: Cryptographic techniques for privacy-preserving data mining. ACM SIGKDD Explorations Newsletter 4(2), 12–19 (2002)
18. Sweeney, L.: k-Anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10(5), 557–570 (2002)
19. UC Irvine Machine Learning Repository, `http://archive.ics.uci.edu/ml/index.html`
20. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 639–644 (2002)
21. Vaidya, J., Kantarcioglu, M., Clifton, C.: Privacy-preserving Naive Bayes classification. The VLDB Journal 17(4), 879–898 (2008)
22. Verykios, V., Bertino, E., Fovino, I., Parasiliti Provenza, L., Saygin, Y., Theodoridis, Y.: State-of-the-art in privacy preserving data mining. ACM SIGMOD Record 33(1), 50–57 (2004)
23. Wright, R., Yang, Z.: Privacy-Preserving Bayesian Network Structure Computation on Distributed Heterogeneous Data. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004), Seattle, WA, USA, pp. 713–718 (2004)
24. Yang, Z., Zhong, S., Wright, R.: Privacy-preserving classification of customer data without loss of accuracy. In: SIAM International Conference on Data Mining, SDM 2005 (2005)
25. Yao, A.C.: How to generate and exchange secrets. In: 27th Annual Symposium on Foundations of Computer Science, pp. 162–167 (1986)
26. Yi, X., Zhang, Y.: Privacy-preserving naive Bayes classification on distributed data via semi-trusted mixers. Information Systems 34(3), 371–380 (2009)
27. Zhan, J., Matwin, S., Chang, L.: Privacy-Preserving Naive Bayesian Classification over Horizontally Partitioned Data. Data Mining: Foundation and Practice (118), 529–538 (2008)
28. Zhang, N., Wang, S., Zhao, W.: On a new scheme on privacy-preserving data classification. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 374–383. ACM, NewYork (2005)