IJCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Volume-4, Special Issue-3, May 2016 E-ISSN: 2347-2693

A Unified Framework for Cloud Computing using AES and k-NN Classifier

VARUN K H[1] and GIRISHA G S[2]

[1] *Dept. of ISE, BNMIT, Bengaluru*
[2] *Dept. of ISE, BNMIT, Bengaluru*

Available online at: www.ijcseonline.org

Abstract— Data Mining is a way to distillate knowledge from large data sets. Classification consists of predicting a certain outcome based on the given input. Cloud provides the customers to store large amount of data. When classification is done on such large data sets we will know the true potential. But the problem with cloud is that the data is outsourced and anybody can access the data. This has made majority of companies not use the services of cloud. These companies need to give security to customer's data. One of the ways to provide security to data is by using encryption. But classification cannot be done on encrypted data. This paper addresses the Data Mining over Encrypted Data (DMED) problem. We use the AES and the k-NN classifier to propose a unified framework to provide confidentiality of data.

Keywords— AES, k-NN classifier, Data Mining over Encrupted Data

I. INTRODUCTION

The Cloud has become very advantageous in the recent years. The organization does not have to worry about the maintenance of data as it is taken care by the cloud. Many types of organizations like medicine, banking, and insurance, scientific research etc. can make use of the cloud paradigm. But these organizations contain sensitive information of the customers and cannot outsource the data i.e. to protect confidentiality. One way of protecting the confidentiality is to encrypt the data.

Example: Suppose an insurance company wants to use the cloud services to store the customer data. The owner can outsource the data on the cloud. But since the cloud is considered to be semi-honest. The owner will encrypt the data and the associated data mining algorithm. When the agent wants to classify the customers based on the risk. This will be done based on the classification algorithm. To classify the customer records the agent will have to generate a class label A with all the details of the customer. But this A contains confidential information which should be encrypted before sending it to the cloud.

The above example gives a general idea of the situation of the companies which use cloud for data mining. So the general solution is to decrypt the data and to perform data mining, but by doing so the confidentiality of the data will be lost. So we will have to perform data mining on encrypted data. The best solution we propose in this paper is to use homomorphic encryption. Assuming the customer data is outsourced.

A. *Problem Statement*

Say Tarun (owner) has a relational table X of n records $a_1, a_2, \ldots\ldots., a_n$ and m+1 attributes. Let $a_{p,q}$ be the element of pth row and qth column. To protect the confidentiality Tarun encrypts the table attribute wise, he computes $E_{pk}(a_{p,q})$ for $1 \le p \le n$ and $1 \le q \le m+1$, where column (m+1) contains class labels. We consider the used encryption scheme to be semantically secure. X' consists of the encrypted relational data.

Say Danush (agent) be an authenticated person who wants to perform the classification on the records $r = < r_1, r_2, \ldots\ldots\ldots, r_n >$ by applying k-NN classification method on X'. We refer to the above method as a unified framework for Cloud Computing. We perform the classification using k-NN represented as

$$k - NN(X', r) \rightarrow cq$$

Where cq refers to the class labels, X' refers to the encrypted database and r refers to the query.

B. *Our Contributions*

In this paper we propose a unified framework of both the AES and K-NN classifier to perform cloud computing on encrypted relational data. Here we need to make sure that Tarun only can encrypt and outsource the customer's relational data. He should not be participating in any of the further computations regarding cloud computing. Danush should only be able to give a query r with some attributes and he should not be able to view or gain access to any of the customers data. Tarun should also not know the query of Danush to provide confidentiality. So here we are providing confidentiality to both the owner and the agent data. The cloud should also not be able to access the customer data or the query sent by either of the people. This is done because we consider the cloud to be semi-honest (honest but curious). The cloud by keeping track of the owner's participation in the cloud can derive some data access patters and may be able to guess the customers data. For example if a person is communicating about some cardiology related data then it can be derived that the person is a heart patient.

The rest of the paper is organized as follows, the section 2 contains the literature survey of the paper. This section is followed by section 3 which has the k-NN algorithm used to realize the above framework. The section 4 contains a detailed explanation of the conflicts occurring in the paper. The system architecture of the framework is explained in section 5. The performance aspects of the paper is explained in section 6. Finally, section 7 contains the conclusion and the future work related to the paper.

II. LITERATURE SURVEY

As we have described earlier it may seem right to use homomorphic encryption on relational data [1]. As it makes it easier to perform Data mining over encrypted data. But that is not the case, because implementation of such algorithms are very costly. The research of using such algorithms over optimal cost is still in progress. Even if we use such algorithm in today's high end machines it will take at least 30 seconds to execute a single query [2].

Instead of that here we are using the combination of both AES and K-NN algorithms to perform data mining over encrypted data. Here we need two parties to perform data mining as described in Shamir's scheme [3]. The only difference is that, there they need three parties in contrast to our work.

A. Classification of Confidential Data

The notion of classification of confidential data was first suggested by Agrawal and Srikant [4], Lindell and Pinkas [5]. Classification of confidential data could be divided into two Data perturbation and Data distribution. The first method of data perturbation was proposed first but it would be useless on confidential data (e.g., [6]). The second method was to use a simple classifier to perform classification. But here the assumption was that the data is distributed among multiple parties as explained in Shamir's scheme (e.g., [7]). So we determined that even this was not working over semantically secure data.

B. Processing Encrypted Data

Processing of Encrypted data can be very difficult when it comes to data mining e.g., [8]. Here we need to make sure that the data that has been hosted by the owner or in this case say Tarun should be made secure. In no case should the data be revealed to the cloud. This is some confidential data of the customers. So in the previous papers it was proposed to minimize the participation of the owner. This was done by making the customer only to host the data of the customers and was not given permission to edit it. This can be one solution, but it only results in redundant tuples in the relational table. Also we know that fully homomorphic algorithms cannot be a practical solution [9]. So we use the AES algorithm which also has the properties of a homomorphic algorithm to do encryption. We do the data mining operations on this data thus removing the overhead caused by fully homomorphic algorithms.

The query of Danush should also be protected as told earlier so that the owner will not learn his query and try to manipulate data while hosting. So we make sure that the owner will only participate in hosting the data and will not take part in any further activities. The query is protected from the cloud by not performing the data mining directly on the data in the cloud. Instead we derive some data from the cloud without either the owner or the cloud knowing it and perform it on that data. So if any person looks at the data it will look useless, as it will not contain any confidential data of the customer.

C. AES Cryptosystem

AES is a symmetric cryptosystem. It is an iterative system. It consists of four steps substitute bytes, shift rows, mix columns and add round key. The input data is into blocks of 128 bits and the encryption is performed on it. The first step is to substitute bytes from a 16X16 S-Box. The first 4 bits represent the row and the second 4 bits the column of the S-Box. The second step is to shift rows. The first row is not shifted. The second row is shifted 1 byte to its left. The third row is shifted 2 bytes to its left. The fourth row is shifted 3 bytes to its left. This will be the output for the next step. The third step is to mix columns, here the 4X4 matrix is multiplied with another constant matrix. In the last step the matrix is X-ORed with the secret key for that step. The details explained are demonstrated clearly in Fig 1.

The AES cryptosystem has different key length for different number of iterations. But AES can be made homomorphic by using the same key for all encryptions. As we use the same key and information to encryption, it will always give us the same cipher text. But the advantage of AES is that it is semantically secure [10], [11], i.e. even if a person knows the cipher text he will not be able to derive the plain text. Decryption of data in AES is the same as encryption, except to follow the steps in the reverse order.

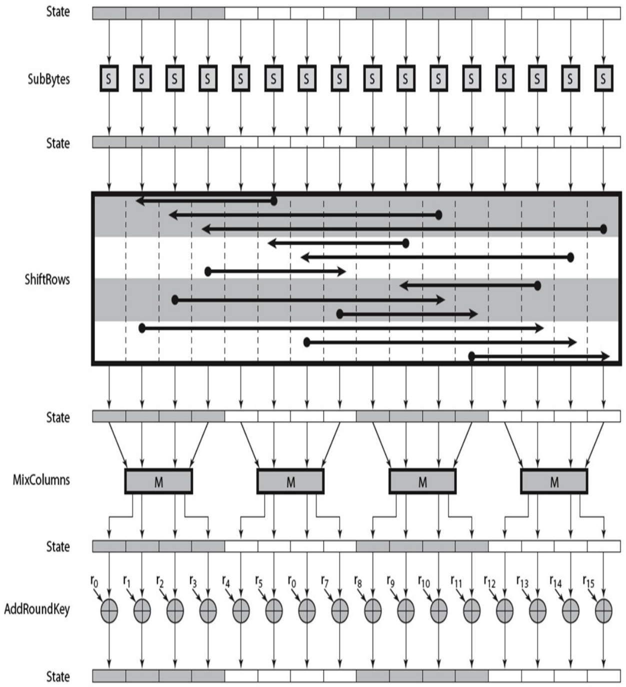
State
SubBytes
S
State
ShiftRows
State
MixColumns
M
State
r_0 r_1 r_2 r_3 r_4 r_5 r_0 r_7 r_8 r_9 r_{10} r_{11} r_{12} r_{13} r_{14} r_{15}
AddRoundKey
State

Fig 1. AES algorithm steps to encrypt data

III. ESSENTIALS FOR SEMANTIC SECURITY

The other algorithm which is used in conjunction with the AES is the K-NN algorithm. The K-NN algorithm takes the encrypted data and the query as input to perform the classification. Classification is discussed extensively in this paper as it is one of the most basic operations of data mining.

Let us consider that Danush has a query q which he has to process on the encrypted database X'. The attributes of Danush's query can be represented as r. The K-NN algorithm takes the encrypted database and the query as inputs to perform the classification. Here we consider two parties as explained earlier C1 and C2. C1 ha the encrypted database X' and p, while C2 has the secret key of the AES algorithm. Now when Danush enters his query to provide confidentiality to his query, we encrypt it using the AES algorithm. The encrypted query is now forwarded to C1 for further processing.

Further C1 receives the query from Danush and will compute the Euclidean distance between the classes where $(x, y) = \sqrt{(x1 - y1)^2 + (x2 - y2)^2}$. Then the decision tree is constructed by using the distances between the classes as the primary parameter.

Algorithm 1 . $k - NN(X', r) \rightarrow cq$

```
Require : $C1$ has and $p$; $C2$ has $sk$; Danush has $q$
1 : Danush :
(a). Compute $E_{pk}(q_j)$ , for $1 \leq j \leq m$
(b). Send $E_{pk}(q) = < E_{pk}(q_1), ........, E_{pk}(q_m) >$ to $C1$
2 : $C1$ and $C2$ :
(a). $C1$ receives $E_{pk}(q)$ from Danush
(b). for $i = 1$ to $n$ do :
    • $Epk(x_i) \leftarrow SSED(Epk(q); Epk(t_i))$
    • $[x_i] \leftarrow SBD(Epk(x_i))$
3 : for $s = 1$ to $k$ do :
$C1$ and $C2$ :
    • $([x_{\min}]; E_{pk}(I); E_{pk}(c')) \leftarrow (\theta 1; \ldots; \theta n)$,
      where $\theta i = ([di]; Epk(I_{ti}); Epk(t_{i,m+1}))$
    • $E_{pk}(c_s') \leftarrow E_{pk}(c')$
4 : $SCMC_k(E_{pk}(c_1'), ........, E_{pk}(c_k'))$
```

Then the data is sorted by calculating the minimum of n numbers. The sorted data is stored in the encrypted format. The secure computation of majority classes is used to calculate the majority of the class labels near to a particular node.

The secure computation of majority classes is used to compute the most nearest nodes to the selected node. It is also used to securely relay the results to the end user i.e. in this case Danush.

The data is received from the previous algorithm, whereas we already known that C1 now is aware of the class labels from the previous algorithm and the secret key is known by C2. Here we have another feature where we can provide complete security to some nodes. For example, we can make the class labels with zero risk unavailable for classification. The final output after secure frequency is a decision tree with all the class labels in a sorted format.

The second feature of the SCMC is to securely traverse the data. Here to transfer the decision tree from C1 to Danush, we multiply every class label with a real number. We send the product γ_q to C2 and real number to Danush. C2 will decrypt the received data and send it to Danush. Finally Danush will receive γ_q' from C1 and rq from C2. He will take the subtraction of both the numbers and take a mod n. The resulting output is the result to the query.

Algorithm 2. $SCMC_k(E_{pk}(c_1'), ..., E_{pk}(c_k')) \rightarrow cq$

```
Require: $< E_{pk}(c_1); \ldots; E_{pk}(c_w) >, < E_{pk}(c_1'); \ldots; E_{pk}(c_k') >$ are known only to C1; sk is known only to C2
1: C1 and C2:
    (a). $< E_{pk}(f(c_1)); \ldots; E_{pk}(f(c_w)) > \leftarrow SF(L; L')$, where $L =< E_{pk}(c_1); \ldots; E_{pk}(c_w) >, L' =< E_{pk}(c_1'); \ldots; E_{pk}(c_k') >$
    (b). for $i = 1$ to $w$ do:
        • $[f(c_i)] \leftarrow SBD(Epk(f(c_i)))$
2: C1:
    (a). $\gamma_q \leftarrow Epk(cq) * Epk(rq)$, where $rq \in_R Z_N$
    (b). Send $\gamma_q$ to C2 and rq to Danush
3: C2:
    (a). Receive $\gamma q$ from C1
    (b). $\gamma_q' \leftarrow Dsk(\gamma_q)$; send $\gamma_q'$ to Danush
4: Danush:
    (a). Receive rq from C1 and $\gamma_q'$ from C2
    (b). $cq \leftarrow \gamma_q' - rq \bmod N$
```

The second feature of the SCMC is to securely traverse the data. Here to transfer the decision tree from C1 to Danush, we multiply every class label with a real number. We send the product γ_q to C2 and real number to Danush. C2 will decrypt the received data and send it to Danush. Finally Danush will receive γ_q' from C1 and rq from C2. He will take the subtraction of both the numbers and take a mod n. The resulting output is the result to the query.

IV. MUTUAL PROTECTION OF DATA

In this protocol we consider that we have two parties' C1 and C2. The only problem is that at any point of time any single party may divert from the protocol and try to manipulate data. The encryption of data may provide the required security. But what if they try to manipulate the data during the computation. For example for C1 to initiate the k-NN with modified inputs and to exit midway from the protocol after achieving the required information. However that is not possible in this case as neither parties have enough information to single handedly carry out the whole operation. Here we have distributed the data between parties to avoid such malicious manipulations.

But if both the parties are trying to manipulate the data then it is meaningless to secure it. So the above holds true only if a single party is malicious. The best way to avoid such behavior is to let the honest party to completely compute the data with zero knowledge of the information. Also it is impossible to try to authenticate the parties or data at each step as it will significantly increase the cost of the project.

Another alternative which can be implemented in the future is to not just distribute the data but also at random intervals change the roles. For example if one party C1 is taking care of processing the data and the other C2 authenticating the data. After some time make C1 authenticate the data and C2 to perform the computation.

V. SYSTEM ARCHITECTURE

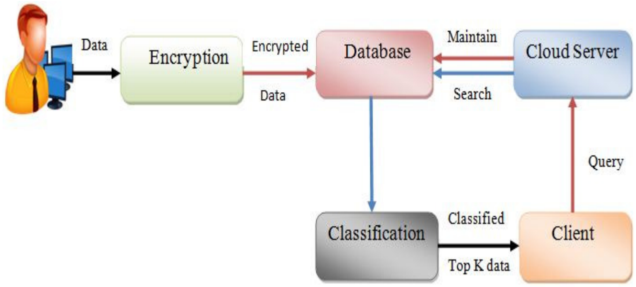

Fig 2. System architecture

As shown in the figure 2, the owner will host the data on the cloud in an encrypted format. We are using the AES cryptosystem to encrypt the data. The role of the owner will end there. The agent can query the cloud for top k classified data. The cloud is using the k-NN classifier to classify the data present on the cloud. The cloud will classify the data and reply back to the agent with specific results.

VI. PERFORMANCE

The computational cost of K-NN increases linearly with the value of k. As we increase the value of k from 3 to 30 we could observe the increase in time. The total complexity of the algorithm is $O(n*(m+k*\log_2 n))$. It can be determined that most of the computational cost is taken up by the k-NN algorithm. Only a little amount is taken by the SCMC algorithm. We have taken a real data set which consists of 524 records and 4 attributes. We encrypt this dataset attribute wise using the AES algorithm. It can also be noted that the AES algorithm also takes up a very large chunk of time as it is performing the encryption attribute wise.

We have to mention that this algorithm is not practically efficient. It takes a lot of time for every transaction even on a very high end machine. But there is a way to increase the efficiency even in the slightest. We will have to use parallelization. But all the above computed results is without using parallelization.

VII. CONCLUSION AND FUTURE WORK

There are a lot of classification algorithms which have been proposed in the past decade. But for a scenario where classification has to be done on an outsourced data this unified k Nearest Neighbors classifier is the best. This will perform its classification on the encrypted data by providing the required semantic security to the data. The efficiency of the algorithm can be improved by providing the minimum amount of security. This will help the organizations in outsourcing the data without the problem of cloud security and will help in utilizing all the advantages of the cloud.

As we have told before there might be a problem of malicious parties. So there is a need to make the parties do their mutual work at specific time intervals. The problem which persists is that of efficiency, as the proposed work will add on to the present efficiency and will make the work useless. So there is a need to research to deploy the above work without disturbing the efficiency.

REFERENCES

[1] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proc. 41st Annu. ACM Sympos. Theory Comput., 2009, pp. 169–178.

[2] C. Gentry and S. Halevi, "Implementing gentry's fully-homomorphic encryption scheme," in Proc. 30th Annu. Int. Conf. Theory Appl. Cryptographic Techn.: Adv. Cryptol., 2011, pp. 129–148.

[3] A. Shamir, "How to share a secret," Commun. ACM, vol. 22, pp. 612–613, 1979.

[4] R. Agrawal and R. Srikant, "Privacy-preserving data mining," ACM Sigmod Rec., vol. 29, pp. 439–450, 2000.

[5] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in Proc. 20th Annu. Int. Cryptol. Conf. Adv. Cryptol., 2000, pp. 36–54.

[6] P. Zhang, Y. Tong, S. Tang, and D. Yang, "Privacy preserving Naive Bayes classification," in Proc. 1st Int. Conf. Adv. Data Mining Appl., 2005, pp. 744–752.

[7] L. Xiong, S. Chitti, and L. Liu, "K nearest neighbor classification across multiple private databases," in Proc. 15th ACM Int. Conf. Inform. Knowl. Manage., 2006, pp. 840–841.

[8] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2004, pp. 563–574.

[9] X. Xiao, F. Li, and B. Yao, "Secure nearest neighbor revisited," in Proc. IEEE Int. Conf. Data Eng., 2013, pp. 733–744.

[10] Shivlal Mewada, Sharma Pradeep, Gautam S.S., "Classification of Efficient Symmetric Key Cryptography Algorithms", International Journal of Computer Science and Information Security (IJCSIS) USA, Vol. 14, No. 2, pp (105-110), Feb 2016 .ISSN: 1947-5500

[11] Shivlal Mewada, Pradeep Sharma, S.S Gautam, "Exploration of Efficient Symmetric AES Algorithm", Ist IEEE Symposium on Colossal Data Analysis and Networking (CDAN-2016)", Mar 18th -19th, 2016. ISBN: 978-1-5090-0669-4

[12] B. K. Samanthula, Y. Elmehdwi, and W. Jiang, "k-nearest neighbor classification over semantically secure encrypted relational data," eprint arXiv:1403.5001, 2014.

Author Profile

Varun K H received the bachelor's degree in Computer Science and Engineering from the Visvesvaraya Technological University, Belagavi, India, in 2013. He is a currently persuing his MTech from Visvesvaraya Technological University Belagavi, India. His interests include applied cryptography, personal privacy and data security in the fields of social networks, cloud computing.

Girisha G S received the bachelor's Degree in Electronics & Communication Engineering from Kuvempu University, 1996, the MTech in Computer Science from Visvesvaraya Technological University, in 2001. He is currently a research student at BNMIT. He has presented 2 international , 7 national conference papers. He is an Associate professor in Information Science engineering department.