ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

Data Mining Using Secure Homomorphic Encryption

Raunak Joshi[1], Bharat Gutal[2], Rajkumar Ghode[3], Manoj Suryawanshi[4], Prof U.H. Wanaskar[5]

Student, Computer Engineering Department, PVPIT, Pune, India [1,2,3,4]

Assistant Professor, Computer Engineering Department, PVPIT, Pune, India[5]

Abstract: Data Privacy is one of the major issues while storing the Data in a database environment. Data Mining based attacks, a major threat to the data, allows an adversary or an unauthorized user to infer valuable and sensitive information by analyzing the results generated from computation performed on the raw data. This paper presents an approach to mine the data securely using k-means algorithm even in the presence of adversaries. This approach assumes that the data is not stored in a centralized location but is distributed to various hosts. This proposed approach prevents any intermediate data leakage in the process of computation while maintaining the correctness and validity of the data mining process and the end results.

Keywords: Data Mining, Security, K-means, Encryption.

I. INTRODUCTION

With the advancement in technology, industry, and research a large amount of data is being generated which is increasing at an exponential rate. Traditional Data Storage systems are not able to handle Data and also analyzing the Data becomes a challenge and thus it cannot be handled by traditional analytic tools.

Data Mining based attacks, a major threat to the data, allows an adversary or an unauthorized user to infer valuable and sensitive information by analyzing the results generated from computation performed on the raw data.

Data Mining based attacks, a major threat to the data, allows an adversary or an unauthorized user to infer valuable and sensitive information by analyzing the results generated from computation performed on the raw data.

The homomorphic public key encryption is a cryptographic system that allows the performance of a set of operations on the data when they are encoded, resulting in its data appearing in plain text. The approach is able to maintain the correctness and validity of the existing k-means to generate the final results even in the distributed environment

II. PROBLEM DEFINITION AND SCOPE

This Problem statement proposes a secure k-means data mining approach assuming the data to be distributed among different hosts preserving the privacy of the data. The approach is able to maintain the correctness and validity of the existing k-means generate the final results even in the distributed environment. A new approach of modern cryptography, defined as the Homomorphic Encryption allows for the encrypted data to be arbitrarily computed which is a solution that aims to preserve the security, confidentiality and data privacy. This system proposes methods that ensure the confidentiality and privacy in the mining of databases based on fully homomorphic encryption.

III.SYSTEM ARCHITECTURE

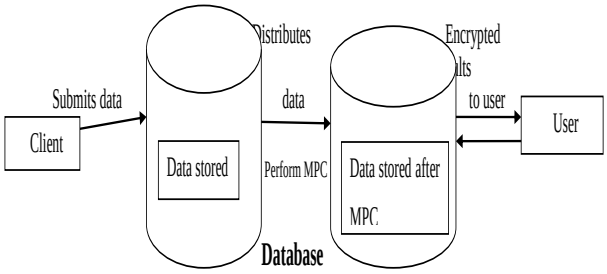

Fig.1. System Architecture

The system architecture follows a certain flow. This flow can be client uploads the data or submits the data in the it then gets stored in the database. Then after performing data mining by applying k-means algorithm it gets distributed among different hosts and after applying MPC that is Multi Party Computation it gets encrypted and it gets stored again in database. If the user is the authorized user then he will be able to decrypt the data using the privacy key. Secure multi-party computation (also known as secure computation or multi-party computation/MPC) is a subfield of cryptography with the goal to create methods for parties to jointly compute a function over their inputs, and keeping these inputs private.

IV.RELEVANT MATHEMATICS ASSOCIATED AND ALGORITHMS

Let us consider a set S

where, S= {U, R, SER, D, N, C, K-means ()}

Here, S: System which includes: U: Set of Users Where U={U1, U2, U3 …, Un}

SER: Server.

R: Set of Request.

Where R= {R1, R2, R3……., Rn}.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

D: Database with horizontal partitions(p1,p2).

N: Number of Cluster. (i.e. 2)

C: Set of Centroid.

Where C={C1, C2,..Cn}.

K-means (N): It is the algorithmic part of the system.

Where N is number of cluster i.e. 2.

K-means Algorithm for data mining

AES Algorithm for homomorphic encryption

K means Algorithm:
K-Means clustering intends to partition n objects into k clusters in which each object belongs to the cluster with the nearest mean. This method produces exactly k different clusters of greatest possible distinction.

Clusters the data into k groups where k is predefined.
Select k points at random as cluster centers.
Assign objects to their closest cluster center according to the Euclidean distance function.
Calculate the centroid or mean of all objects in each cluster. Repeat steps until the same points are assigned to each cluster in consecutive rounds.

AES Algorithm:
KeyExpansions—round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block.

InitialRound
AddRoundKey—each byte of the state is combined with a block of the round key using bitwise xor.

Rounds
SubBytes—it's substitution step where each byte is replaced with a subByte using s-box. S box is matrix given by rijndael.
ShiftRows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.
MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.

AddRoundKey
Final Round (no MixColumns)
SubBytes
ShiftRows
AddRoundKey.

In the SubBytes step: each byte in the state matrix is replaced with a SubByte using an 8-bit substitution box, the Rijndael S-box.

Shift-row step :
The first row is left unchanged. it cyclically shifts the bytes in each row.

MixColumn step:
The four bytes of each column of the state are combined using an invertible linear transformation.

AddRoundKey step:
In the this step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule;

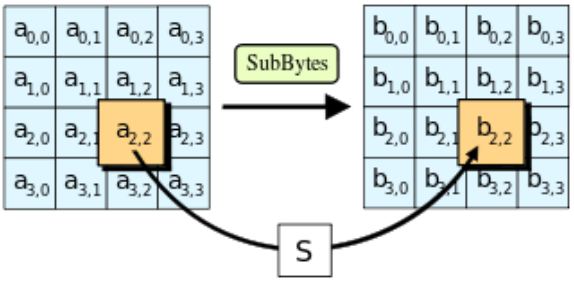

Fig.2. Sub Bytes

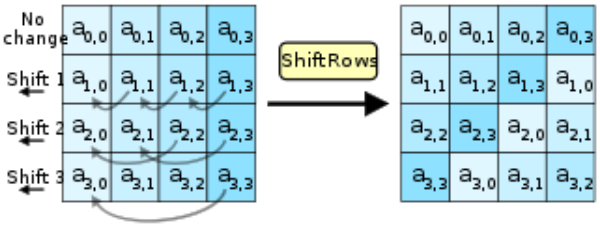

Fig.3. Shift Rows

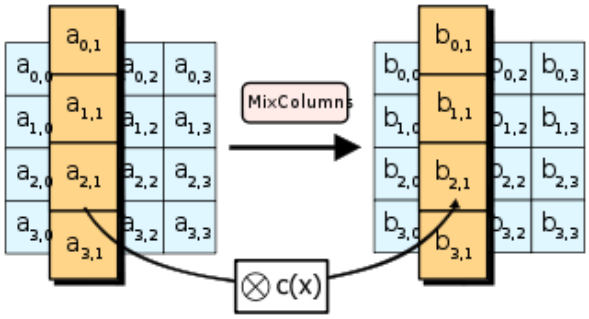

Fig.4. Mix Columns

V. LITERATURE SURVEY

Paper name: Performance of Ring Based Fully Homomorphic Encryption for securing data .
Published Year: 2014
In this paper we have studied homomorphic encryption in database.

Paper name: Homomorphic Encryption-based Secure SIFT for Privacy-Preserving Feature Extraction
Published Year: 2014
In this paper we have studied Privacy Preservancy in Data Mining.

Paper name: Survey on Recent Algorithms for Privacy Preserving Data mining
Published year: 2014
In this paper we have studied we have studied privacy preserving recent algorithms in data mining.

Paper name: Data Storage Security Using Partially Homomorphic Encryption.
Published year: 2013
In this paper we have studied data security using homomorphic encryption

Paper name: The Use of Fully Homomorphic Encryption in Data Mining with Privacy Preserving
Published year: 2015
In this Paper we have studied use of homomorphic encryption in data mining.

DOI 10.17148/IJARCCE.2015.41064

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

VI. PROJECT PROBLEM ASSESSMENT USING NP-HARD

NP-hard (non-deterministic polynomial-time hard), in computational complexity theory, is a class of problems that are, informally, "at least as hard as the hardest problems in NP". More precisely, a problem H is NP-hard when every problem L in NP can be reduced in polynomial time to H.As a consequence, finding a polynomial algorithm to solve any NP-hard problem would give polynomial algorithms for all the problems in NP, which is unlikely as many of them are considered hard.

A common mistake is thinking that the NP in "NP-hard" stands for "non-polynomial". Although it is widely suspected that there are no polynomial-time algorithms for NP-hard problems, this has never been proven. Moreover, the class NP also contains all problems which can be solved in polynomial time.

A decision problem H is NP-hard when for any problem L in NP, there is a polynomial-time reduction from L to H. An equivalent definition is to require that any problem L in NP can be solved in polynomial time by an oracle machine with an oracle for H. Informally, we can think of an algorithm that can call such an oracle machine as a subroutine for solving H, and solves L in polynomial time, if the subroutine call takes only one step to compute.

Another definition is to require that there is a polynomial-time reduction from an NP-complete problem G to H. As any problem L in NP reduces in polynomial time to G, L reduces in turn to H in polynomial time so this new definition implies the previous one. It does not restrict the class NP-hard to decision problems, for instance it also includes search problems, or optimization problems.

If there is a fast solution to the search version of a problem then the problem is said to be Polynomial-time, or P for short. If there is a fast solution to the verification version of a problem then the problem is said to be Non deterministic Polynomial time or NP for short. Clustering algorithms are generally heuristic in nature and are often polynomial in time.

In the k-means problem, we are given a finite set S of points, and integer $k \geqslant 1$, and we want to find k points (centre's) so as to minimize the sum of the square of the Euclidean distance of each point in S to its nearest center. This is done in polynomial time.

There are some complexity-theoretic reasons to believe that cryptography can't be based on NP-completeness. Basically it all boils down to the mismatch between average-case hardness required for cryptography, and worst-case hardness required for NP-hardness. Moreover, many hard algebraic problems which serve as the basis for cryptography are in NP ∩ co-NP. Such problems cannot be NP-complete unless NP = co-NP. Finally, AES is a finite-domain function. It is invertible distinguishable from a random permutation in (large but) constant time. The definitions of P, NP, etc., refer to asymptotic behaviour that is, as the input size grows to infinity. Because of the algebraic structure of AES, it is probably possible to define a "generalized AES" for infinitely many key lengths.

Clustering algorithms are generally heuristic in nature and are often polynomial in time. We are using k means clustering algorithm for data mining and AES algorithm for homomorphic encryption so our project comes under NP hard Problem.

VII. CONCLUSION

Security and privacy is the major issue concerning the clients as well as of services as a lot of confidential and sensitive data is stored which can provide valuable information to an attacker. This proposes a method to solve the privacy issues of the database. It assumes that the user data is distributed on two hosts and performs a combined k-means clustering using the Homomorphic encryption system for security purpose so as to prevent any interpretation of intermediate results by an attacker. The proposed approach can further be extended by adding a digital signature or hashing technique to authenticate the third party so as to prevent an adversary from posing as the third party to host's. Also it can be generalized or extended to more number of hosts if required.

REFERENCES

[1] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A Berkeley view of cloud computing." Dept. Electrical Eng. and Comput. Sciences,
[2] University of California, Berkeley, Rep. UCB/EECS 28 (2009): 13.
[3] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, "Controlling data in the cloud: outsourcing computation without outsourcing control." In Proceedings of the 2009 ACM workshop on Cloud computing security, pp. 85-90. ACM, 2009.
[4] D. J. Solove, "I've got nothing to hide and other misunderstandings of privacy," San Diego L. Rev. 44 (2007): 745.
[5] P. K. Rexer, "Data miner survey highlights the views of 735 dataminers" 2010.
[6] C. Su, F. Bao, J. Zhou, T. Takagi, and K. Sakurai, "Privacy-preserving two-party k-means clustering via secure approximation." InAdvanced Information Networking and Applications Workshops, 2007,
[7] AINAW'07. 21st International Conference on, vol. 1, pp. 385-391. IEEE, 2007.
[8] Md. Riyazuddin , Dr.V.V.S.S.S.Balaram , Md.Afroze , Md.JaffarSadiq , M.D.Zuber. "An Empirical Study on Privacy Preserving Data Mining". International Journal of Engineering Trends and Technology (IJETT). V3(6):687-693 Nov-Dec 2012. ISSN:2231-5381.
[9] The Use of Fully Homomorphic Encryption in Data Mining with Privacy Preserving July 2014A. Laécio A. Costa1, B. Ruy J. G. B. de Queiroz
[10] Q. Lu, Y. Xiong, X. Gong, and W. Huang. "Secure collaborative outsourced data mining with multi-owner in cloud computing." 2012
[11] IEEE 11th International Conference on Trust, Security and Privacy in
[12] Computing and Communications (TrustCom) , IEEE, pp. 100-108, 2012.
[13] S. Owen, A. Robin, T. Dunning, and E. Friedman. Mahout in Action.
[14] Manning Publications, 2012.
[15] R. Bhadauria, R. Borgohain, A. Biswas and S. Sanyal. "Secure Authentication of Cloud Data Mining API " arXiv preprintarXiv:1204.0764, 2012.