iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** IV **Month of publication:** April 2018

DOI: http://doi.org/10.22214/ijraset.2018.4438

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

Implementation of Secure Homomorphic Encryption on Cloud Database using Two-cloud Architecture

Nivedita W. Wasankar[1], Prof. A.V. Deorankar[2]
[1]*M. Tech. Scholar,* [2]*Assistant Professor, Department of Computer Science and Engineering Government College of Engineering, Amravati (MH) India*

Abstract: ***Now a day, cloud computing is a fastest growing technology. It allows users to use or access different applications, store information. File access assure in real technique to the file protection due to untrusted cloud servers. Databases and application are buried in the cloud server, which is outside the ability to control of the data proprietor. Attacks from opponent are difficult to stop in cloud storage. Similarly, increased number of queries will release more information to the cloud server. The various researchers have done numerous work for these issues. In this paper, we propose a two-cloud architecture for secure database that provide privacy preservation to various queries. Privacy of information is strongly protected against cloud providers in our proposed scheme.***
Keywords: Cloud Computing, Database, Two-cloud architecture, Privacy Preservation, Query.

I. INTRODUCTION

Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management and reduce the cost for both the enterprises and individual users. A cloud client, such as an IT enterprise, wants to outsource its database to the cloud, which contains valuable and sensitive information (e.g. transaction records, account information, disease information), and then access to the database (e.g. SELECT, UPDATE, etc.).

Cloud provider may be honest-but-curious. they try to obtain private information for his own benefits. Even, they cloud may forward such sensitive information to the business competitors for profit, which is an unacceptable operating risk. Encryption are needed before outsourcing sensitive data to cloud such as database system. clients' frequent queries will certainly and gradually expose some private information on data statistic properties. Data and queries of the outsourced database should be protected against the cloud service provider.

There is one approach to mitigate the security risk of privacy leakage is to encrypt the private data and hide the query/access patterns. CryptDB is allows query processing over encrypted databases. CryptDB receives queries from the application server, secures them and sends them to the database server. Then, it will receive encrypted data from the database, decrypts it and sends to application server to be sent to the requester. It uses a few instruments to accomplish this security functionality like OPE.

One of the devices being the Order preserving encryption (OPE) is a deterministic encryption scheme whose encryption function preserves numerical ordering of the plaintexts. This means that comparing encrypted data returns the same result than comparing the original data. This permits to order encrypted data without the need of decryption. This cryptosystem is useful in databases, in which record fields are encrypted because it permits to range queries. order preserving encryption(OPE) will expose some private information to the cloud service provider as it is designed to preserve the order on ciphertexts so the order information of the data, the statistical properties and the access pattern will be leaked. Hence, the objective of security protection of the outsourced information to a cloud server is polished by partitioning the sensitive knowledge into two parts and store them in two non-colluding clouds.

Furthermore, a secure database service architecture is permitted by utilizing two non-colluding clouds in which the application, database and web service is divided into two clouds. Henceforth, perceiving just a single cloud can't help uncover private data.

A. Motivation

Privacy is most important issue in cloud storage services. Many researchers unable to protect private data by some technique. Everybody has some private and confidential data that they don't share to any one likewise all IT enterprises and organizations has numerous private information. they don't want to share the information to anybody. If any of the information of the organization's is

leaked, there is trouble for their organization. With the goal that we are turning on protection of the sensitive information and provide protection.

B. Objectives

To develop a security module for cloud database using homomorphic encryption technique, To develop a separate cloud for encryption and decryption process, To implement secure key generation and storage system on 2nd cloud (processing cloud), To implement XML temporary storage for frequently required data which reduces time taken by encryption and decryption process as well as database operations, To implement secure encrypted queries processing system including complex and aggregate queries like SUM, AVG etc., To implement parallel execution which reduces processing time.

II. RELATED WORK

Popa et al. [2] proposed CryptDB, which allows query processing over encrypted databases. CryptDB is a system that provides practical and provable confidentiality for applications backed by SQL databases. It works by executing SQL queries over encrypted data using a collection of efficient SQL-aware encryption schemes. The database managed by the cloud provider, but database items are encrypted with keys that are only known by the data owner. SQL queries run over the encrypted database using a collection of operations such as equality checks and order comparisons. CryptDB uses encryption schemes that allow such comparisons to be made on ciphertexts. CryptDB represents a weak attacker model because it assumes the existence of a trusted cloud-based application server and proxy. Nevertheless, CryptDB represents an interesting position on the trade-off between functionality and confidentiality from cloud providers.

Bohli et al. introduce various multi-cloud distinct architectures and discussed according to their security and privacy capabilities and prospects. As a multi-cloud architecture offer some level of security by providing one or more of the fallowing four strategies, as explained in [3]:

1) *Data replication:* Data is duplicated in multiple clouds and also synchronized whenever a change happens. Replication provide data integrity and availability because of multiple copies of the same data.
2) *Data partitioning:* Partitioning is split the data between multiple clouds such that no cloud provider can get a meaningful insights of the data that it is hosting. It provides data privacy and allow data accessing only for authorized clients.
3) *Application partitioning into tiers:* It provide security by partitioning application system into tiers which allow the separation between application logic and data which gives protection of data leakage due to application logic flow.
4) *Application Fragmentation:* Fragmenting the application which allow the application to be saved on multiple clouds. It provides a distributed fine grained fragments among multiple clouds. Thus, cloud provider cannot gain access to whole application which provide data confidentiality.

In 2004, R. Agrawal et al. [4] propose the encryption of data that belonging to a integers subset [p_{min}, p_{max}], although they suggested that the possibility of treating floating point values as if they were integers, since positives maintain the same order, and for the negatives, that have the order reversed, only need to subtract the resulting integer from the largest negative. In their proposed method, data transforms for follow certain statistical distribution into ciphertexts. It maintains order and follows a different distribution, chosen by the user. To generate the encryption function, they convert all the data to encrypt, the list of distribution samples has to be emulated. The auxiliary information is necessary for the encryption and decryption of data. It will be generated from all these samples. To model the distributions, data need to be partitioned in buckets. Linear interpolation will be used in it. During encrypting, data is first convert in a uniform distribution and then transformed into the target distribution. One of the drawbacks of their propose cryptosystem is key generation. While it is relatively small and its generation is linear to size of the database. After a key generation, if a large amount of data is added to the database, it will be necessary to choose a new key and re-encrypt the database.

Boldyreva et al. [5] The cryptography community provided first formal security guarantee of order-preserving encryption. Boldyreva et al. introduce notion in distinguish ability under ordered chosen plaintext attack. They also prove that no other stateless scheme can achieve this notion. It settles for an encryption scheme with less security of a random order-preserving function. This scheme requires only to store a key on the client. Then they state that a random order-preserving function can achieve the security property of window one-wayness [6]. Furthermore, they provide a scheme that achieves IND-OCPA security, but only requires all plaintexts in advance. If all plaintexts are known in advances, their order can be determined.

FHE allows an unlimited number of operations with unlimited number of times. The first plausible and achievable Fully Homomorphic Encryption (FHE) scheme was introduced by Craig Gentry in 2009 [7], that evaluates an arbitrary number of additions and multiplications and thus calculate any type of function on encrypted data. It is based on ideal-lattices in math and it is

not only a description of the scheme, but also a powerful framework for achieving FHE. However, it is conceptually and practically not a realistic scheme. Different FHE schemes demonstrated that FHE still needs to be improved significantly to be practical on every platform as they are very expensive for real-life applications because of the bootstrapping part, which is the intermediate refreshing procedure of a processed ciphertext.

Ramachandram et al. [8] describe the partially homomorphic encryption which is a form of encryption that allows for some computations to be performed on the ciphertext without decrypting the ciphertext. The result of the operations is returned as an encrypted result, which when decrypted is the same as if some operation was performed on the plaintext. There are many form of partially homomorphic cryptosystems that allow for some specific operations to be performed (namely addition and multiplication). PHE allows only one type of operation with an unlimited number of times (i.e. no bound on the number of usages). In other words, PHE schemes can only be used for particular applications, whose algorithms include only addition or multiplication operation.

III. PROPOSED SYSTEM

A. Existing System

In existing system, two cloud architecture for database security is proposed. The two clouds are non-colluding clouds, one cloud eg cloud A manages application and encrypted database whereas cloud B manages database processing as well as encryption and decryption process. In existing system entire focus of the research is on the security of database outsourced by the tenants. For database security, Paillier Cryptographic Algorithm is proposed. For numeric range queries like between, <, >, OR, AND cloud B fetches data greater than 0 from cloud A database to cloud B and then fire range queries on decrypted data. Existing system is not enable to process aggregate queries like SUM, AVG etc.

B. Proposed System

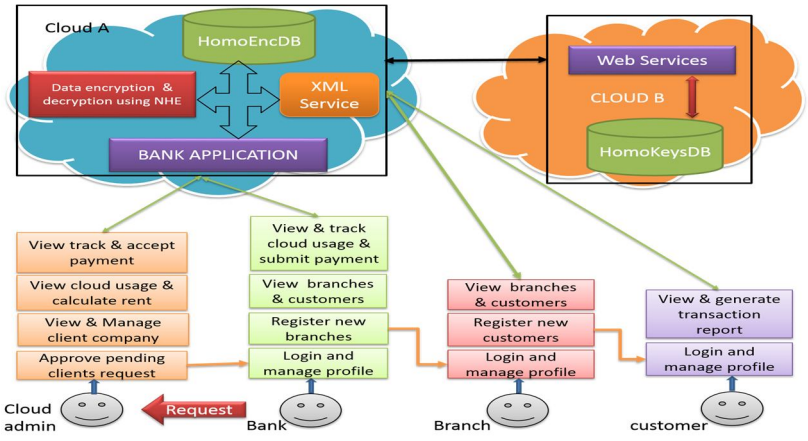
In our proposed system will develop novel security approach for outsourced database. we propose two non-colluding clouds, cloud A and cloud B. Cloud A will manages application and databases, encryption/decryption of data whereas cloud B will manage database queries processing as well as web services for key of databases. We propose a novel homomorphic algorithm using which we can reduce processing time as well as temporal execution memory. For numeric data decryption, existing approach requires too much space and execution time. Therefore, to increase performance of our system we propose a new technique in which we will create run time sql functions in cloud A. The sql server functions will save our time as well as memory. The proposed system will fire range and aggregate queries at a time. In the proposed system bank application is considered in which cloud A consist encrypted database (HomoEncDB), bank application, XML service, encryption/decryption process and cloud B contain web services of encryption/decryption key, database of keys for encryption/ decryption (HomoKeysDB) as shown in fig 1.

IV. ALGORITHMS

This proposed system performs encryption as well as decryption on the documents or files that can be used to secure our data. Therefore, following encryption and decryption algorithm is used for this system to provide the security of sensitive data.

A. Encryption

1) take input as no or text
2) If input is text, then convert it into ascii value n=ascii of text
3) Else n=input no



4) Reverse the no n
5) Generate key k=random (1, (length(no)/2))
6) If k is not even, convert it into even no
7) Divide n by k
8) Final result will be the cipher no
9) If input is a text value convert the no into char
10) Store the cipher text/no into database

B. Decryption

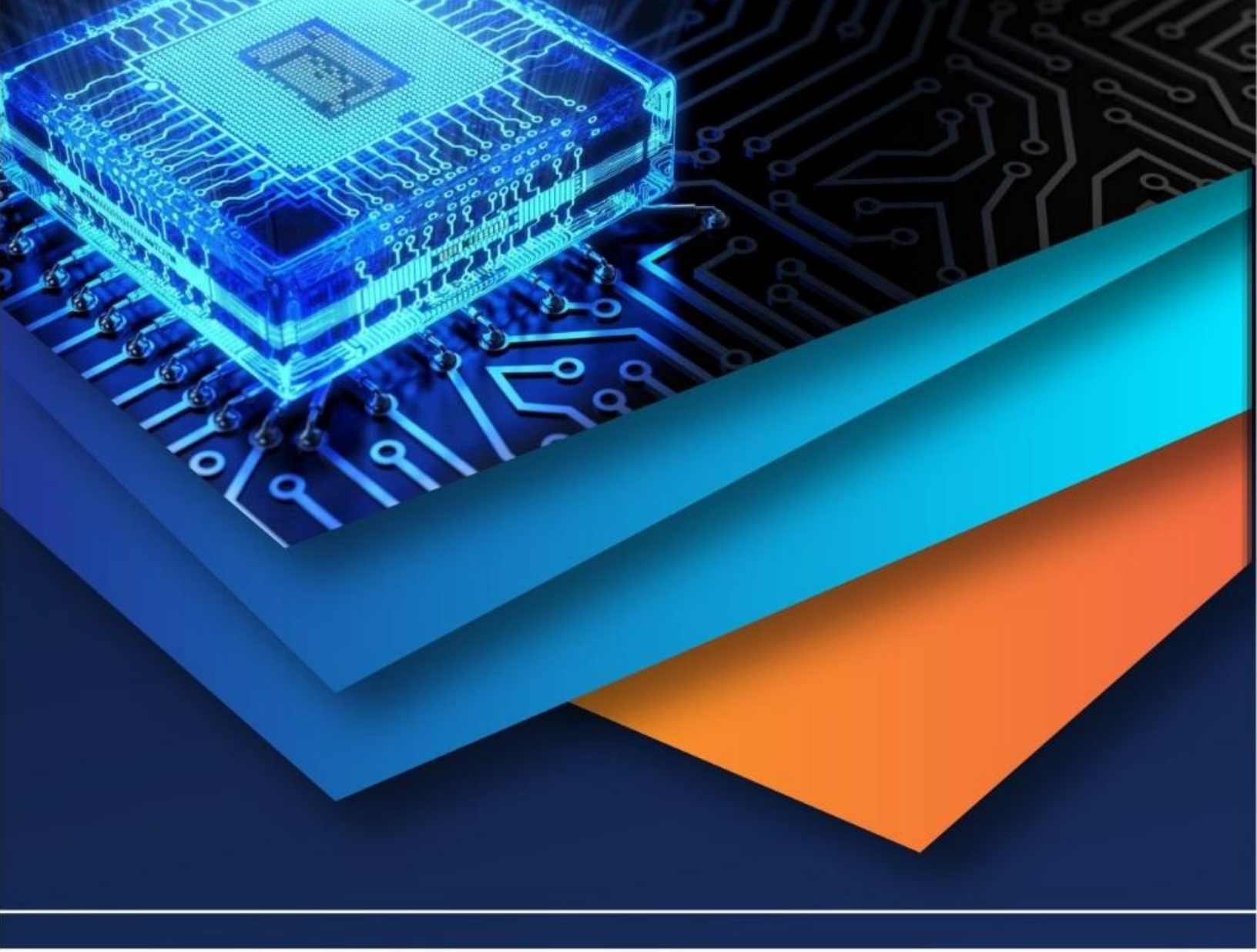
1) take cipher text/no as input from database
2) If input is text, then convert it into ascii value n=ascii of text
3) Else n=input no
4) Get key k
5) N=n*k
6) Reverse N s.t length(N)=length of original cipher which is stored in database
7) N=(Reverse(N))
8) N is original no
9) If input is text, convert N into char

V. CONCLUSIONS

In this paper, we have studied the various techniques and protocols associated with the privacy preservation of the outsourced data to the external cloud server. The CryptDB which fundamentally includes utilizing the range queries productively finished the encrypted information utilizing a novel SQL-aware encryption system. However, some data is still exposed to the cloud server. The order preserving encryption is one of the tools used by the CryptDB which enables comparison operations to be specifically connected on encrypted information, without decrypting the operands. But encryption of non-numeric information isn't possible with this tool. Later the multi-cloud architecture was introduced which introduced the idea of partitioning the sensitive information and query logic into two different non-colluding clouds which don't have the knowledge about each other. In the proposed system, we present a two-cloud architecture with a homomorphic algorithm for outsourced database service, which ensures the privacy preservation of data contents, statistical properties and query assess pattern. Our proposed scheme is more efficient due to parallel processing. Our proposed scheme is support all query operations, including "SUM/AVG".

REFERENCES

[1] Kaiping Xue, Shaohua Li, Jianan Hong, Yingjie Xue, Nenghai Yu, and Peilin Hong,"Two-Cloud Secure Database for Numeric-Related SQL Range Queries with Privacy Preserving", Information Forensics and Security, IEEE Transactions on, pp. 1556–6013, 2016 (DOI 10.1109/TIFS.2017.2675864).
[2] R. A. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: protecting confidentiality with encrypted query processing," in Proceedings of the 23rd ACM Symposium on Operating Systems Principles. ACM, 2011, pp. 85–100.
[3] J.-M. Bohli, N. Gruschka, M. Jensen, L. Iacono, and N. Marnau, "Security and privacy-enhancing multicloud architectures," Dependable and Secure Computing, IEEE Transactions on, vol. 10, no. 4, pp. 212–224, July 2013.
[4] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu, "Order Preserving Encryption for Numeric Data". In ACM SIGMOD international conference on Management of data, pages 563–574, 2004.
[5] Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-Preserving Symmetric Encryption. In 28th Annual International Conference on Advances in Cryptology: The Theory and Applications of Cryptographic Techniques -EUROCRYPT'09, pages 224–241, 2009.
[6] Boldyreva, A., Chenette, N., and O'Neill, "Order-preserving encryption revisited: improved security analysis and alternative solutions". In Proceedings of the 31st International Conference on Advances in Cryptology (2011), CRYPTO.
[7] Gentry, C, "A fully homomorphic encryption scheme", Doctoral Dissertation, Stanford University, 2009.
[8] S. Ramachandram, R.Sridevi, P. Srivani, "A Survey Report On Partially Homomorphic Encryption Techniques In Cloud Computing" International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 12, December – 2013.
[9] Paillier, P.,"Public-key cryptosystems based on composite degree residuosity classes", In Eurocrypt, 1999.
[10] M. A. AlZain, E. Pardede, B. Soh, and J. A. Thom, "Cloud computing security: from single to multi-clouds," in Proceedings of the 45th Hawaii International Conference on System Science (HICSS2012). IEEE, 2012, pp. 5490–5499.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY