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

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

***Abstract*- Cloud computing provides a data storage system which enables the users to be less dependent on the client system and provides an architecture to upload the data in a cloud that can be shared by multiple users. Databases and application are buried in the cloud server, which is outside the ability to control of the data owner. Attacks from opponent are difficult to stop in cloud storage and also increased number of queries will release more information to the cloud server. 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. Secure analysis shows privacy of numerical information is strongly protected against two cloud providers. Extensive security and performance analysis are shows that our proposed algorithm is high efficient.**

***Index Terms*- Cloud Computing; Database; Two-cloud architecture; Privacy Preservation; Query. 28ui**

1. INTRODUCTION

Cloud storage enables users to remotely store their data and provides security through authentication of the user. Cloud computing can allow a user to access applications and data from any computer at any time since they are stored on a remote server. It also decreases the need for companies to purchase top-of-the-line servers and hardware or hire people to run them. By centralizing memory, bandwidth, storage & processing in an off-site environment for a fee, cloud computing can significantly reduce cost. 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.

In our proposed system will develop novel security approach for outsourced database. We utilize two non-colluding clouds in which the application, database and web service is divided into two non-colluding clouds, cloud A and cloud B. In the single cloud storage, there is no solution for cloud failure and there may be chances of data lost due cloud failure. By using two cloud architecture, Data can be divide into different forms and store it onto two different clouds. We propose a novel homomorphic algorithm using which 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. Two

cloud architecture will be used in industrial data storage and also useful in banking storage for storing the data of different branches. In the proposed system, bank application is considered in which cloud A consist encrypted database (HomoEncDB), bank application, XML service, encryption or decryption process and cloud B contain web services of keys, database of keys for encryption or decryption (HomoKeysDB) as shown fig.1. If any cloud provider or unauthorized user wants to gain that private data was unable to get the whole data because different parts of data are stored onto different clouds. Two cloud architecture will prevent data lost and provide high privacy and double security to cloud users with less computation cost and minimum time. Henceforth, perceiving just a single cloud can't help uncover private data.

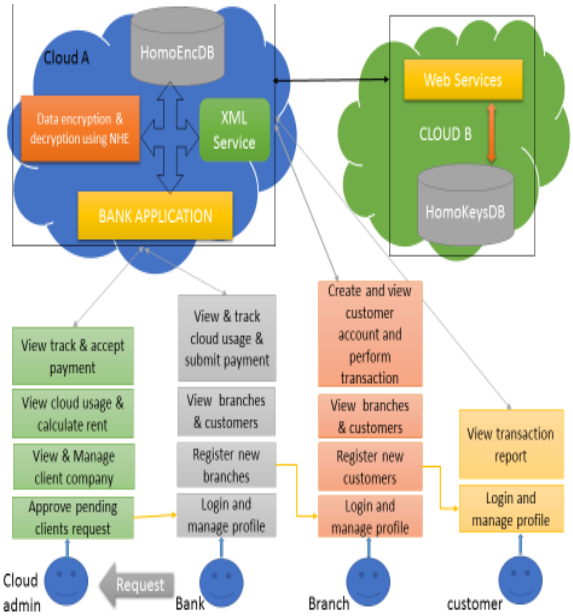

Fig. 1. Proposed system model with bank application

Algorithm:

A. Encryption:

- Take input as number or text
- If input is text then convert it into ascii value n=ascii of text
- Else n=input number
- Reverse the number n
- Generate key k=random (1, (length(number)/2))
- If k is not even, convert it into even number
- Divide n by k
- Final result will be the cipher number
- If input is a text value convert the number into char
- Store the cipher text or number into database

B. Decryption:

- Take cipher text or number as input from database
- If input is text then convert it into ascii value n=ascii of text
- Else n=input number
- Get key k
- N=n*k
- Reverse N such that length(N)=length of original cipher which is stored in database
- N=(Reverse(N))
- N is original number
- If input is text, convert N into char

2. EXPERIMENTAL EVALUATION

On one hand, both of the two clouds will respond with correct information in the interactions of our proposed scheme (honest); on the other hand, the clouds try their best to obtain private information from the data that they process (curious). From the perspective of privacy assurance, here the data not only include permanently stored information (i.e., database), but also each temporary query request (i.e., queries). Additionally and importantly, as the assumption in some existing works, we assume that the two clouds A and B are non-colluding. Cloud A follows the protocol to add required obfuscation to protect privacy against cloud B, so that cloud B cannot obtain additional private information in the interactions with cloud A. No private information is delivered beyond the scopes of protocols. All the data received by cloud A is encrypted and the computation steps are all performed in the ciphertext domain, and because of the semantic security of proposed algorithm, cloud A cannot deduce any private information unless cloud B colludes with it. Cloud B cannot infer any private information from cloud A's input as long as blinding factors are properly generated, and cloud A and B are non-colluding. The application database (HomoEncDB) is encrypted with the proposed homomorphic encryption algorithm. The data of each table is encrypted with the key of that table. The key is

generated according to the row present in that table. The generated key is stored in another database (HomoKeysDB) which is stored in cloud B. If the key is used frequently then it will automatically store in XML file for frequent use. The privacy of database is preserved because one database is fully encrypted format and both the cloud is non-colluding. Both the cloud administrator is unable to retrieved the useful information from their respective database. Cloud B database has the key in encrypted format and it have no mean to have keys because key is changes according to number of row in the table and the tables are stored in application database. If user search the information or fire the query, each time application has to retrieve the key from cloud B and encrypt the query with the key of that table on which it will be fired. So the clients frequent queries will not inevitably and gradually reveal some private information on data statistic properties. The result of the query is in the encrypted format. The result of query is stored in dataset. Then again the application has to retrieved the key from cloud B database or XML file. The result of the query is decrypted and then it will present to the user in the plaintext. Thus, data and queries of the outsourced database should be protected against the cloud service provider. Hence our system is more efficient than the existing system. In the existing system, Paillier's homomorphic encryption makes up a large proportion of the cost. While in proposed algorithm, the encryption and decryption cost is negligible. It increases the security of our system. The values of the plaintext changes and size also change according to our proposed algorithm. For example, we use surya korde with age 36 it will gives the following cipher-text. The length of plaintext and cipher-text is given.

Values Before Encryption

Name : Surya Korde [Length : 11]

Age : 36 [Length : 2]

Mobile : 9824364755 [Length : 10]

Values After Encryption

Name : MTg5fDE4OHwyMDJ8MTk5fDE2M3wxMjB8MTg1fDIwOXwyMDJ8MjA1fDE3MQ== [Length : 60]

Age : 124 [Length : 3]

Mobile : 9824364843 [Length : 10]

The processing of different items is independent and can be implemented in parallel. The efficiency increases linearly to the number of parallel processes. For the storage overhead, in our proposed scheme, cloud B only keeps table key, and Cloud A stores all the encrypted data. The mainly storage overhead of both schemes is on the encrypted data, which is consistent with the actual situation. In our proposed scheme, both stored data and query logic are partitioned into two parts. This improves the privacy preservation of numeric related query, while the complexity increases too. But it is acceptable, as the increase in overhead is small and the security has been greatly improved. In the existing system, Paillier's homomorphic encryption makes up a large proportion of the cost. While in proposed algorithm, the encryption and decryption cost is negligible. In our scheme, to perform subtraction, multiplication and addition of the items in cloud A data, a key is required form cloud B, therefore the delay is linear to the number of the items.

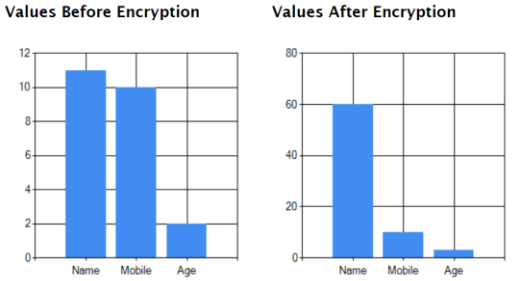

Fig. 2. Proposed system evaluation.

3. COCLUSION

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. Privacy of information is strongly protected against cloud providers in our proposed scheme. Extensive security and performance analysis are shows that our proposed algorithm is high efficient. Our proposed scheme is more efficient due to parallel processing. Our proposed scheme is support all query operations, including "SUM or AVG".

REFERANCES

[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.