Two-Cloud Secure Database with High Level of Privacy Preservation

Nikhil Kumar[1], Tapasya Thakur[2], Priyanka Verma[3]

[1,2,3]*Student, Department of Computer Engineering, D. Y. Patil College of Engineering, Pune, India*

***Abstract*: The cloud foretells a new era of cloud computing where applications services are provided through the Internet. Cloud Computing is a flexible, cost-effective, and proven delivery platform for providing business or consumer IT services over the Internet. Industries and individuals outsource database to realize convenient and low-cost applications and services. 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. Apart from its benefits, the service is also waiving users' physical proprietorship of their outsourced data, which inevitably threatens the security of the data in the cloud. In order to overcome this problem and attain a secure and trustworthy cloud storage service, we proposed a two-cloud architecture for a secure database with various protocols for provision of privacy preservation.**

***Keywords*: database, cloud computing, range query, privacy preserving.**

1. Introduction

With the rapid growth in cloud industry, several trends are opening up the era of cloud computing, which is an internet-based development and use of computer technology. Cloud Computing is defines as the aggregation of computing as a utility and software as a service where applications are delivered as service over the internet. The processors together with the Software as a Service (SaaS) computing architecture is transforming data centres into pools. Moving data into the cloud has reduced the users' burden since they don't have to care about the direct hardware management or the infrastructure maintenance. This gradually reduces the cost for both high-end enterprises and individual users. The concept of offloading data and computation in cloud computing is used to address the inherent problems in mobile computing by using resource providers other than the mobile device itself to host mobile applications. Some of the pioneers of cloud computing vendors are Amazon Simple Storage (S3) and Amazon Elastic Compute Cloud (EC2). It is important to ensure secure and reliable data transmission in the cloud, however, since the cloud service provider is not entirely trustworthy it becomes integral to perform encryption before outsourcing sensitive data from database to the cloud.

2. Proposed system

In the proposed model, the admin works on the client side and the two clouds (Cloud A and Cloud B) works on the server side performing computational and storage services. From the perspective of maintaining data security, the two clouds are assumed to be non-colluding with each other. These two clouds work together to operate on clients' queries/requests and also they follow a series of intersection protocols to maintain data privacy. All this adds to the factor of making the data stored in the cloud hack-proof.

In our model, the knowledge about the queries and the database is partitioned and stored in the clouds A and B respectively. To conduct a secure database, data is encrypted using various encryption techniques and stored in cloud A, whereas, the cloud B contains the private keys required to access the data stored in the database which can only be given to the authorized personnel. Every time the user poses a query, the corresponding knowledge comprises the actual data content and the relative processing logic. We utilize a prototype of knowledge partition, dividing application logic into two parts, which was first proposed by Bohli et al. In [7]. The application logic is divided into two parts and each cloud knows a part of it but no idea which cloud contains what segment of the query logic.

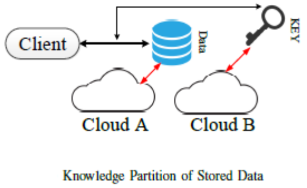

Fig. 1. Knowledge partition of stored data

3. System architecture and security requirements

Representative network architecture for cloud storage service architecture has the following modules:

Data Owner: industries and individuals outsource database to realize convenient and low-cost applications and services, it can be an enterprise wanting to outsource its database to the cloud, which contains valuable and sensitive information. Data owner outsource database in an encrypted format to protect their data contents from intruder.

Data User: Data user will download needed files from cloud which are already uploaded in cloud by data owner. In the cloud the data is in encrypted format, so to view the original data contents i.e. the decrypted format, the user has to get permission (key) from the owner. The user analyzes the query request and figures out how many columns are involved. In our project, user sends the encrypted query request to cloud A and gets the desired file by using search engine with range queries.

Cloud Service Provider (CSP): CSP can view the functionalities of all the modules. They are responsible for keeping the data available and accessible, and the physical environment protected and running. People and organizations buy or lease storage capacity from the providers to store user, organization, or application data. In our project, we have two non-colluding clouds which provide high level security to the data contents.

Administrator: admin is responsible for providing authentication to both data owner and data user, Admin will send secret key to the data user via the registered mail ID for authentication purpose. Users can login using the secret key exclusively and not by any other means. Introducing this aspect provides access to authorized personnel only and can notify if an intruder tries to leak sensitive information.

In cloud data storage, a user stores his data through a CSP into a set of cloud servers, which are running in a parallel, cooperative and partitioned manner. Cloud's backbone layer consists of physical servers and switches. The cloud Service Provider is responsible to run, manage and upgrade cloud hardware resources according to the requirements of users.

Fuzzy query over encrypted data is becoming a popular topic, since in practical scenarios, some query requests usually want to retrieve data with similar, rather than exactly same indexes[11], [12]. Fuzzy searchable encryption has been introduced for cloud computing in many literatures, such as [5]. These schemes deal with the issue that search keywords allows small-scaled distinction in character/numeric level. Specifically for numerical keywords, the query predicate can get numerical records within a range. Some schemes targeted at spatial query, especially knn [2], [3], [5], [8], [9], which focus on the distance between the query vector and the data. They usually inquire about certain spatial objects (or several numerical attributes) related to the others within a certain distance. Range query has been proposed for that purpose. However, such existing range query schemes are not suitable for practical secure database due to high storage overhead to maintain the corresponding ciphertext. Subsequently, order preserving encryption (OPE) has been introduced to provide numeric-related range query in structured database, such as CryptDB. OPE preserves the order of values in encryption field, while hiding the actual values. Until now, OPE has been developed to increase both efficiency and security.

Bohli et al. [1] proposed a multi-cloud architecture, which can protect the private information of many outsourced services, including database. The main contribution is the introduction of four knowledge partition patterns among multiple cloud service providers: (1) Replication of applications, (2) Partition of application system into tiers, (3) Partition of application logic into fragments, and (4) Partition of application data into fragments. The knowledge is partitioned into two fragments, respectively stored in one cloud, who is assumed to be non-colluding to another cloud. Therefore, no cloud can get any private information in such multi-cloud architecture. However, Bohli et al. [6] have not provided a detailed scheme or realization for database.

There are various cryptographic techniques to support numeric-related operations (e.g. addition, multiplication, XOR) upon the encryption field. Paillier cryptosystem [41] is one of the most popular techniques that provides addition homomorphic, which means: if two integers a and b are encrypted with a same key k into two ciphertexts (be denoted as Ek(a) and Ek(b)), there exists an operation (refer to as "⊗"), such that

$$Ek(a) \otimes Ek(b) = Ek(a + b)$$

Paillier cryptographic algorithm consists of:

- Key generation: Two large and independent prime numbers p and q are randomly selected. Then we compute $n = p \bullet q$ and $\mu = \lambda - 1 \bmod n$, where λ is the least common multiple of p and q, and commonly $\lambda = \mathrm{lcm}(p-1, q-1)$. The public key (PK) is n, and the private key (SK) is (λ, μ).
- Encryption: Let m be the integer to be encrypted. Firstly, we select a random number $r \in Z\ast\ n2$, and then the ciphertext of m can be computed.
- Decryption: Let the ciphertext $c = E(m;r)$. The plaintext m can be recovered.

4. Related work

In our plan, two clouds have been assigned distinctive tasks in the database system: Cloud A provides the main storage service and has the encrypted database while Cloud B performs the computational tasks and oversees whether each numerical record satisfies the clients' query request with its own security key. This is done keeping in mind the access of private data to authenticated users only. We have assumed the two clouds to be non-colluding so the application logic is divided into two parts in the proposed plan, where each cloud knows a part of the logic. As we move further, we will analyze that possessing a single part of knowledge is not suffice to reveal the data privacy or the queries. On the basis of this architecture, our model provides a proposition for query of numeric-related data with privacy preserving. The client can easily pose a query for retrieving data from the cloud with predicates like BETWEEN, ">", "<" for one column or other condition combinations for multiple column data retrieval.

1. *Table creation:* Firstly, the client rents the cloud service to outsource his/her database system to the cloud. For the protection of the private data in the cloud, a certain procedure is implemented before uploading the data into the cloud.

The client selects a symmetric key K randomly for every column in the table and then uses the key K to encrypt that particular column name Ti (1 ≤ i ≤ m , where m is the total column number of the table), The encrypted name is denoted as E(Ti), assuming each encrypted name of equal length.

Also, each row value should also be encrypted. Since, in this paper we are taking only numeric-related data into consideration, the client generates a public/private key pair denoted as PK and SK and for every numeric-related value x, the client uses PK to encrypt it:

X=E(x, PK),

and the client should keep a track of the total row values or items of the table N. After completion of encrypting the entire table along with its content, the table is sent to Cloud A with public key PK and the private key SK is securely sent to Cloud B.

2. *Query Request:* For retrieving some data from the outsourced database, the client will have to first generate a SQL query. After the plaintext query request is generated, it will be modified into an encrypted one as follows:

- Encrypt the column name: the cloud computes the column name (Ti) with the symmetric key K.
- Encrypt the range value boundary value: The range boundary value a is encrypted. In Pallier cryptosystem with the help of public key PK by the client.
- Generate the token: The client first analyzes the request and finds out how many columns are involved. The client uses the private key SK to sign in the data. The corresponding token sign(TNO||CN||N||T) is generated related to the columns involved in the query, where TNO is the token serial number, CN is the total columns involved, N is the total items in the table and T is the timestamp.
- Send the query request: finally, the client send the query request to Cloud A with signed token, where the actual data is stored in the database

SELECT * FROM table WHERE E(Ti)> A,

3. *Item send:* Cloud A performs three steps before sending the data to cloud B after it gets the column name E(Tj):

- Number Comparison: For every item Xj = Tij in the column, Cloud A selects a random positive number rj and Ꞓj individually, where 0 ≤ Ꞓj < rj and then calculates:

⟦X^'⟧_(j= (X_j/A)) r_j.E(-Ꞓ_j,PK)

The decryption result of the above equation is (xj – a). Rj-Ꞓj with the help of additive homophobic property of Paillier Cryptosystem.

- Items Shuffling: Cloud A randomly shuffles the column L to column L' so as to secure the data in case of repeated pattern detection. Cloud A should also be able to securely store the mapping of new items between the shuffled and original columns in a new column, say, M.

Finally, Cloud A removes the column name E(Ti) from the column L', and sends it to Cloud B together with the token received from the client.

- Index Send: Cloud B receives the token from Cloud A and first verifies if the received token is legit or not, whether the token has expired or has it been used before in a time period. Cloud B then checks the column with cloud A and finally if the request is authenticated, cloud B decrypts each item as follows:

xj'= D(Xj'', SK),

4. *Query Response:* Cloud A receives the index column L" and for every item j' in it, cloud A looks up in the column M for the mapping information and retrieves the index j in the original column. Then the corresponding rows in the table are sent to the client as response. After receiving the response, client uses his/her private key (SK) to obtain the required items present in the rows received and remove unrequited data.

Privacy Preservation in repeated queries:

Repeated queries from the client can significantly increase the chances of private data leakage. The clouds could figure out the query patterns and might collect more statistical information about the data. However, our model can gratuitously reduce the data leakage in such cases.

1. Cloud A: cloud A can learn more and more about the private data stored in the database through repeated queries but in this strategy it is curbed as follows: Multiple query requests are crossing over multiple columns and simple query requests are passed for database access. In such a scenario, Cloud A only receives the final index result of the column comparisons from Cloud B and doesn't get the original comparison result of each one column. Cloud A cannot make modifications to the tokens received; if it does any, Cloud B will find that unmatched with the token it sent to cloud A. Also, each token is signed by the client through the secret key (SK) it has. Hence, Cloud A can't generate new tokens neither can it modify them.
2. Cloud B: Every involved column name is removed before sending to cloud B and since random numbers are given to each item, the cloud cannot guess whether previous queries belonged to items on the same column or different ones. Item shuffling also aids in not distinguishing the items in repeated queries.

5. Conclusion

In this report, a two-cloud architecture with a series of interaction protocols for outsourced database service has been proposed, which insures the privacy preservation of the data contents, statistical properties and query configuration. Also supports the range queries, it not only protects the confidentiality of static data, but also addresses possible privacy leakage in statistical properties or after a large number of query operations. Security analysis shows that the scheme can meet the privacy-preservation requirements.

Furthermore, performance evaluation result shows that the suggested system is effective. For future work, consider to further enhance the security while ensuring practicality, and it will poke out our proposed scheme to support more operations, such as sum/Avg.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph et al., "A view of cloud computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

[2] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," IEEE Transactions on Services Computing, vol. 5, no. 2, pp. 220–232, 2012.

[3] K. Xue and P. Hong, "A dynamic secure group sharing framework in public cloud computing," IEEE Transactions on Cloud Computing, vol. 2, no. 4, pp. 459–470, 2014.

[4] J.W. Rittinghouse and J. F. Ransome, Cloud computing: implementation, management, and security. CRC press, 2016.

[5] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," Future Generation Computer Systems, vol. 28, no. 3, pp. 583–592, 2012.

[6] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," Wireless Communications and Mobile Computing, vol. 13, no. 18, pp. 1587–1611, 2013.

[7] 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.

[8] C. Curino, E. P. Jones, R. A. Popa, N. Malviya et al., "Relational cloud: A database-as-a-service for the cloud," 2011, http://hdl.handle.net/1721.1/62241.

[9] D. Boneh, D. Gupta, I. Mironov, and A. Sahai, "Hosting services on an untrusted cloud," in Advances in Cryptology-EUROCRYPT 2015. Springer, 2015, pp. 404–436.

[10] X. Chen, J. Li, J. Weng, J. Ma, and W. Lou, "Verifiable computation over large database with incremental updates," IEEE Transactions on Computers, vol. 65, no. 10, pp. 3184–3195, 2016.

[11] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New publicly verifiable databases with efficient updates," IEEE Transactions on Dependable and Secure Computing, vol. 12, no. 5, pp. 546–556, 2015.

[12] S. Benabbas, R. Gennaro, and Y. Vahlis, "Verifiable delegation of computation over large datasets," in Annual Cryptology Conference. Springer, 2011.