International Journal On Engineering Technology and Sciences – IJETS™
ISSN(P): 2349-3968, ISSN (O): 2349-3976
Volume V, Issue X, October - 2018

Multi Cloud Secure Database for SQL Queries with Privacy Preserving

Shimina P
Computer Science
MIT Anjarakkandi
Shimina.p@gmail.com

Rashmila G V
Computer Science
MIT Anjarakkandi
rashmilagv@gmail.com

Abstract

The database hosted and processed in cloud server is beyond the control of data owners. The current schemes available for database are vulnerable to privacy leakage to cloud server. Most of them are based on single cloud architecture and traditional encryption schemes. Increased numbers of queries, statistical properties, access pattern, etc. are some of the practical challenges. Here propose multi cloud architecture for secure database, with a series of intersection protocols that provide privacy preservation to various numeric related queries. Here provide a new client Remote Data Possession Checking Protocol in Cloud Storage protocol based on homomorphic hash function. The new scheme is provably secure against forgery attack, replace attack and replay attack based on a typical security model.

1. Introduction

Cloud computing has been considered as a new model of enterprise IT infrastructure, which can organize huge resource of computing, storage and applications, and enable users to enjoy ubiquitous, convenient and on-demand network access to a shared pool of configurable computing resources with great efficiency and minimal economic overhead. Attracted by these appealing features, both individuals and enterprises are motivated to outsource their data to the cloud, instead of purchasing software and hardware to manage the data

Despite of the various advantages of cloud services, outsourcing sensitive information (such as e-mails, personal health records, company finance data, government documents, etc.) to remote servers brings privacy concerns. The cloud service providers (CSPs) that keep the data for users may access users sensitive information without authorization. A general approach to protect the data confidentiality is to encrypt the data before outsourcing.

The idea suggested by J. M. Bohli et al[1]. on reducing the risk for data and applications in a public cloud is the simultaneous usage of multiple clouds. Order-preserving encryption scheme for numeric data that allows any comparison operation to be directly applied on encrypted data. This scheme handles updates gracefully and new values can be added without requiring changes in the encryption of other values. It allows standard database indexes to be built over encrypted tables and can easily be integrated with existing database systems.

The main contribution of this paper is a multi non-colluding cloud architecture for secure database service where data is stored in one cloud and the knowledge of query pattern is partitioned into two parts, and knowing only one cannot reveal any private information and a series of intersection protocols such protocols will not expose order-related information to any of the two non-colluding clouds. In addition to securing the data contents, this scheme also well preserves the privacy of logical relationship among data contents, such as data order, the privacy of the statistical properties and query pattern

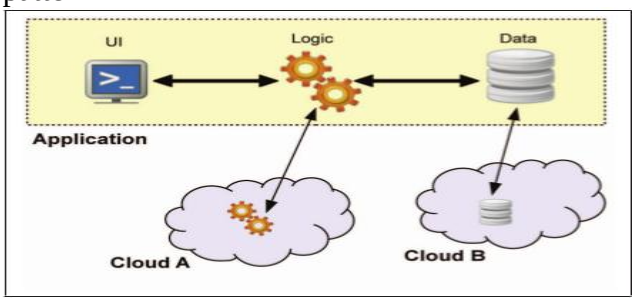

Fig 1. Partition of application system into tiers

2. Related work

The idea suggested by J. M. Bohli et al. on reducing the risk for data and applications in a public cloud is the simultaneous usage of multiple clouds. An attacker that has access to the cloud storage component is able to take snapshots or alter data in the storage. This might be done once, multiple times, or continuously. An attacker that also has access to the processing logic of the cloud can also modify the functions and their input and output data. The basic underlying idea is to use multiple distinct clouds at the same time to mitigate the risks of malicious data manipulation, disclosure, and process tampering. CryptDB[2] provide a secure remote database application, CryptDB addresses two threats: (i) A curious database administrator (DBA) who tries to learn private data by snooping on the DBMS server. (ii) An adversary that gains complete control of application and DBMS servers. Crypt DB ensure the confidentiality of logged out users data. Crypt DBs approach is to execute queries over encrypted data, and the key insight that makes it practical is that SQL uses a well-defined set of operators, each of which can be supported efficiently over encrypted data. The disadvantages of this scheme are of encryption key is twice as large as the number of unique values in the database. Also the updates are problematic.

CypherDB: which is based on re-designing the processor architecture to support arbitrary computation on encrypted data. CypherDB cloud model, involves two parties: a cloud service provider (CSP) and a database owner. The database owner exports an encrypted database to the CSP for future querying. The CSP hosts the database server and provides storage and database administration service to the database owner.

OPE(Order-preserving Encryption)[3] can be used to provide drop-in security in application like databases. Here an order-preserving encryption scheme for numeric data is presented that allows any comparison operation to be directly applied on encrypted data. This scheme handles updates gracefully and new values can be added without requiring changes in the encryption of other values. It allows standard database indexes to be built over encrypted tables and can easily be integrated with existing database systems.

Currently trending is a movement towards ”multi-clouds” from the long followed single cloud approach[5]. It is said that dealing with ”single cloud” providers is predicted to become less popular with customers due to risks of service availability failure and the possibility of malicious insiders in the single cloud. A cloud provider offers many services that can benefit its customers, such as fast access to their data from any location, scalability, pay-for-use, data storage, data recovery, protection against hackers, on-demand security controls, and use of the network and infrastructure facilities. In different cloud service models, the security responsibility between users and providers is different. According to Amazon, their EC2 addresses security control in relation to physical, environmental, and virtualization security, whereas, the users remain responsible for addressing security control of the IT system including the operating systems, applications and data.

Resources in the cloud are accessed through the internet; therefore even if the cloud provider focuses on security in the cloud infrastructure, the data is still transmitted to the users through networks which may be insecure. As a result, internet security problems will affect the cloud, with greater risks due to valuable resources stored within the cloud and cloud vulnerability. Encryption techniques and secure protocols are not sufficient to protect data transmission in the cloud. It is being argued that information privacy is not guaranteed in AmazonS3. Data authentication which assures that the returned data is the same as the stored data is extremely important. So claims exist as though instead of

following Amazons advice that organizations encrypt data before storing them in Amazon S3, organizations should use HMAC technology or a digital signature to ensure data is not modified by Amazon S3.

3. Proposed system

The proposed secure database system includes a database administrator, and two non-colluding clouds and also a Integrity checker system for checking the availability and integrity of the storage-outsourced data,the database administrator can be implemented on a clients side . The two clouds (refer to Cloud A and Cloud B), as the servers side, provide the storage and the computation service. Fig. 2 briefly depicts the architecture of the outsourced secure database system in this scheme.

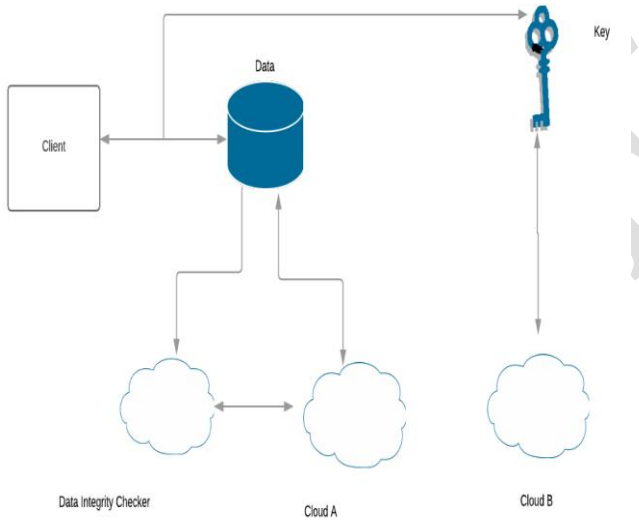

Fig. 2. Multi-Cloud Database Architecture

The knowledge of stored database and queries is partitioned into two parts, respectively stored in one cloud ie cloud A and Cloud B respectively. The mechanism guarantees that knowing either of these two parts cannot obtain any useful privacy information. As shown in Fig 2. to conduct a secure database, data are encrypted and outsourced to be stored in one cloud (Cloud A), and the private keys are stored in the other one (Cloud B).

Based on the two-cloud architecture, this scheme provides an approach to query numeric-related data with privacy preservation. The client can retrieve the desired data from the cloud, when the query predicates contain operators like >,< and BETWEEN for one column, or even diverse condition combinations over one or more columns. For example, the client wants to retrieve items from the table, whose column Ti should be greater than a constant a (i.e., SELECT * FROM table WHERE (Ti > a). In this scheme, it is resolved by figuring out the sign of each value of (Ti(j) > (a), in which j traverses all rows of the whole table. If the result is greater than 0, the relevant item satisfies the query predicate. These procedures are executed in the encryption field, so that the privacy is strongly preserved. Meanwhile, each column name Ti must be encrypted.

Accordingly, if the operator is reversed, i.e., the predicate becomes Ti < a, the corresponding operation is (a - Ti(j)).

The remaining phases are similar as the above mentioned case. Meanwhile, if the predicate is BETWEEN a and b(SELECT * FROM table WHERE Ti BETWEEN a AND b), the result is the intersection of Ti > a and Ti < b. For the predicate =a, it is treated as a special case of the operator BETWEEN, where the retrieved items are intersection set Ti > a - 1and Ti < a + 1. Additionally, the operator of COMBINATION is another one that combines predicates with Boolean expression with ∨ and ∧ :

3.1. The Basic Scheme for Operator ”<”

Cloud A permanently stores the clients encrypted database, and it also keeps the public keys related to the encrypted items in the database. Cloud B keeps the relevant private keys and undertakes the main task of computation. The proposed scheme is composed of Table Creation and Query Protocol. The intersection procedure of Query Protocol

consists of four parts: Query Request, Item Send, Index Send, and Query Response, along with necessary computation operations as depicted in figure

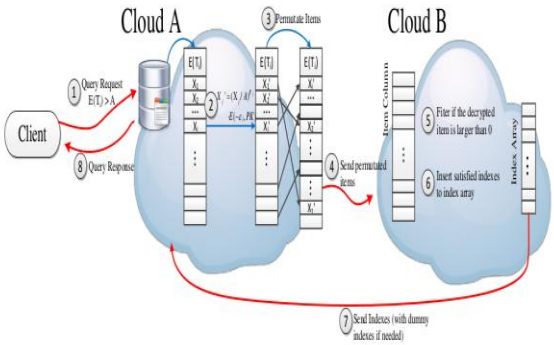

Fig 3.The Query Protocol.

Table Creation: Symmetric key, K is used to encrypt column name, E(Ti). For each item, it's values in multiple columns are encrypted. I.e. X=E(x,PK) The encrypted table along with public key PK is sent to Cloud A. The private key, SK is sent to Cloud B. For multiple tables in a database, table names can be encrypted in the same way that column names are encrypted.

Query Request: When the client wants to retrieve some data from the outsourced database, he/she firstly generates a SQL query (e.g. SELECT * FROM table WHERE Ti > a). After the plaintext query request is generated, it will be modified to an encrypted query following these steps:

Encrypt the column name : Using Symmetric Key K client Encrypt the column name, E(Ti)

Encrypt the range boundary value: The client encrypts the range boundary value a with the public key PK in Pallier cryptosystem. The encrypted boundary value is denoted as A

Generate the token: The client analyzes the query request and figures out how many columns are involved. Then, the client generates the corresponding token Sign(TNO||CN||N ||T), where TNO is the token serial number, and CN is the number of involved columns, N is the total item number in the table, and T is the current timestamp. All these data are signed by the clients private key SK.

Send the query request: This consists of the encrypted query request along with the token.

Item Send: Cloud A finds the column named E(Ti). Before sending the items to Cloud B, it implements the following three phases: **Number Comparison:** For each Xj , A selects a random positive integer, and computes

$$X'_j = \left(\frac{X_j}{A}\right)^{r_j}.E(-\varepsilon_j, PK)$$

All are stored in another temporary column (named L).

Items Shuffling: Cloud A makes a random item shuffling to generate L'. Cloud A securely stores the mapping of items between L' and L in column M. Cloud A removes column name E(Ti) from L' and sends it to cloud B along with the Token.

Index send: Cloud B first verifies the token. Cloud B checks the column from A to make sure that the column number and the item number are consistent with these corresponding values in the token. If the request is authorized, then Cloud B decrypts each item. For each decrypted item if it is greater than zero the index j' is inserted into a new index array L" Additionally, from the aspect of privacy preservation, then Cloud B appends a certain number of dummy indexes and inserts them to the random positions of the new index array L" . Finally, Cloud B returns the final index array L" to Cloud A.

3.2. Integrity of Data in the Cloud

It is a collection of four polynomial-time algorithms (Key-Gen, Sig Gen, Gen Proof, Check Proof), the details are shown as follows:

KeyGen: This is a probabilistic key generation algorithm that is run by the client. It takes a security parameter as input, and returns public key pk and secret key sk.

SigGen: This algorithm is also run by the client to generate the verification meta data . It takes as input private key sk, file F which is an

ordered collection of blocks, and outputs the signature set

GenProof: The cloud server runs this algorithm. It takes as input a file F, its signatures, and a challenge chal. According to the specified block index in chal, it outputs a data integrity proof P.

CheckProof: The client or the Integrity checker runs this algorithm in order to verify a proof of data storage correctness. It takes as input the public key pk, the challenge chal and the proof P. It outputs 1, if the integrity of the data file is intact, or 0 otherwise. Running a public checking protocol consists of two phases,

Setup and Check: The client first invokes KeyGen to initialize his secret key and public key. Then, he generates the verification metadata by executing SigGen. Client stores his data file F and the verification metadata at the cloud server, and removes them from his local memory.

Check: The TPA sends a challenge message chal to the cloud server, and verifies the integrity of the data file F at time of this check. According to the challenge chal, the cloud server can compute a integrity proof from a function of the stored data file F and its verification metadata by invoking GenProof. Upon receiving the proof, the TPA can verify it by invoking CheckProof.

4. Conclusion

Presented a two-cloud architecture with a series of interaction protocols for outsourced database service, which ensures the privacy preservation of data contents, statistical properties and query pattern. At the same time, with the support of range queries, it not only protects the confidentiality of static data, but also addresses potential privacy leakage in statistical properties or after large number of query processes. This scheme employs a homomorphic hash function to verify the integrity for the files stored on remote server, and reduces the storage costs and computation costs of the data owner

References

[1] J.-M. Bohli, N. Gruschka, M. Jensen, L. L. Iacono, and N. Marnau,"Security and privacy-enhancing multicloud architectures," IEEE Transactions on Dependable and Secure Computing, vol. 10, no. 4, pp. 212224,

[2] R. A. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB:protecting confidentiality with encrypted query processing," in Proceedings of the 23rd encryption for numeric data,"

[3] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data",in proceedings of the 2004 ACM SIGMOD international conference on management data

[4] Bony H. K. Chen, Paul Y. S. Cheung, Peter Y. K. Cheung, and Yu-Kwong Kwok, "CypherDB: A Novel Architecture for Outsourcing Secure Database Processing" in 2015 IEEE Transactions on CloudComputing.

[5] M. A. AlZain, E. Pardede, B. Soh, and J. A. Thom, "Cloud computing security: from single to multi-clouds," in Proceedings of the 45th Hawai International Conference on System Science (HICSS2012). IEEE, 2012,