International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

A Secure and Efficient Way of Accessing Encrypted Cloud Databases Using Adaptive Encryption Scheme

Seethal K S[1], Siddana Gowda[2]

[1]Department of Computer Science and Engineering Mangalore Institute of Technology and Engineering (MITE) Moodbidri, Mangalore, Karnataka Email: seethalks555@gmail.com

[2]G R Assistant Professor Department of Computer Science and Engineering Mangalore Institute of Technology and Engineering (MITE) Moodbidri, Mangalore, Karnataka Email: siddanagowda@mite.ac.in

Abstract: ***Now-a-days cloud computing is showing consistent growth in the field of computing. Users can utilize these services on pay-per-use basis. Database-as-a-service is a one type cloud computing service. DBAAS service provides users with database access without need for setting up physical configuration or installing software. But main drawback of DBAAS is, when data is exchanged in cloud, there exists the problem of disclosure of privacy. The idea is to build privacy preserving storage model where data sharing services can update and control the access and limit the usage of their shared data. Preserving privacy is an important issue for cloud computing and it needs to be considered at every phase of design. This paper proposes a metadata based data segregation and storage methodology along with an encryption technique to provide additional security. Here additional security is provided by SQL aware encryption scheme or adaptive encryption scheme. This would serve as a helping note in the progress of strengthening the privacy preserving approaches in cloud computing. This paper also compares both encryption schemes.***

Keywords: Cloud computing, database-as-a-service, secure database-as-a-service, metadata, SQL aware encryption.

1. Introduction

CLOUD computing technology is a service-based, Internet-centric, safe, convenient data storage and network computing service. It is an internet-based model for enabling a convenient and on-demand network access to a shared pool of configurable computing resources. One of the important service of cloud computing is database as a service. Database-as-a-Service (DBaaS) is a service that is managed by a cloud operator that supports applications, without the application team assuming responsibility for traditional database administration functions. With a DBaaS, the application developers should not need to be database experts, nor should they have to hire a database administrator (DBA) to maintain the database.

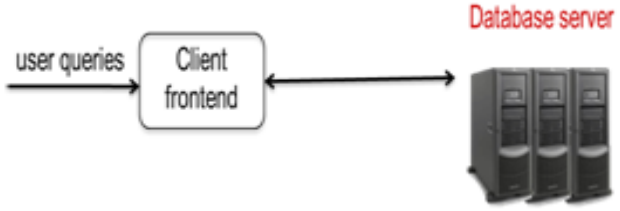

Figure 1.1: Database-as-a-service

True DBaaS will be achieved when application developers can simply call a database service and it works without even having to consider the database. The ultimate goal of a DBaaS is that the customer doesn't have to think about the database. Today, cloud users don't have to think about server instances, storage and networking, they just work. Virtualization enables clouds to provide these services to customers while automating much of the traditional pain of buying, installing, configuring and managing these capabilities. Now database virtualization is doing the same thing for the cloud database and it is being provided as Database as a Service (DBaaS).

This project proposes secureDBaaS. Here all databases are encrypted and stored in the cloud. It allows multiple and distributed users can access their own databases concurrently and independently. Each user uses adaptive encryption scheme for encrypting databases. SecureDBaaS guarantees confidentiality of information at rest, in motion and in use when data are managed through cloud database services. SecureDBaaS eliminate any intermediate proxy server, so a user can achieve availability, scalability and elasticity of DBaaS. Same as confidentiality secureDBaaS maintain the concurrency. The clients access the encrypted database through sql queries and decrypt the database through corresponding algorithms. The rest of the paper is organized as follows: Section II represents literature survey, section III represents problem definition, section IV provides system architecture, section V represents the methodology, section VI, VII and VIII represents results and discussion, conclusion and future scope of the paper.

2. Literature Survey

In large data centre, cloud computing moves the application software and databases, where the management of data and services are not reliable. This unique attribute poses many security challenges. To realize the tremendous potential, business must address the privacy questions raised by the new computing model. The metadata based storage model is based on the information which is valuable only as long as the fragments of the information are related to each other. In [1] proposed a secureDBaaS. Here all client data encrypted with metadata and stored encrypted client data and

encrypted metadata in untrusted cloud. Here confidentiality provided by sql aware encryption algorithms, that allow execution of sql over encrypted data. But it does not support all sql operators. In [2] proposes database as a service. Database as a Service (DBaaS) is a cloud-based approach to the storage and management of structured data. Database service provider provides seamless mechanisms for organizations to create, store, and access their databases. Moreover, the entire responsibility of database management, i.e., database backup, administration, restoration, and database reorganization to reclaim space or to restore preferable arrangement of data, migration from one database version to the next without impacting availability will befall such an organization. Users wishing to access data will now access it using the hardware and software at the service provider instead of their own organization's computing infrastructure. The cloud storage is untrusted, so confidentiality is the main challenge in database as a service. Different approaches are used to provide confidentiality like distributing data among different providers and sharing secret keys. But here the cloud providers can reconstruct the data. In [3] propose data shares are generated uniformly across a domain to prevent information leakage about the outsourced data. Here used the idea of secret sharing to split a database relation into several parts, in the granularity of attribute values, and outsource each part to an honest but curious server. They proposed a searchable secret sharing scheme in which the ordering relation between values is preserved in their corresponding shares, while the distribution of shares is different from the original data distribution. But here does not require the use of multiple cloud providers, and makes use of SQL-aware encryption algorithms to support the execution of most common SQL operations on encrypted data. In [4] proposes transparent cryptographic file system. It is a cryptographic distributed file system. It lets users' access sensitive file stored on remote server in a secure way. It combines ever dropping and data tempering both at the server and over the network using encryption and message digest. This approach is not applicable in secureDBaaS because here assuming that cloud provider is untrusted. In [6] propose CryptDB, a system that provides a practical and strong level of confidentiality. CryptDB meets its goals using three ideas: running queries efficiently over encrypted data using a novel SQL-aware encryption strategy, dynamically adjusting the encryption level using onions of encryption to minimize the information revealed to the untrusted DBMS server, and chaining encryption keys to user passwords in a way that allows only authorized users to gain access to encrypted data. In [7] propose solution to store data at the service provider after encrypting it which can be only decrypt by the owner. But these two approaches applicable to multi-tier web application. It makes several drawbacks. In [8] proposed an alternative technique to homomorphic encryption functions to support aggregation queries over encrypted tuples in the Database-as-a-Server Model. But this uses proxy based architecture. A proxy-based architecture requiring that any client operation should pass through one intermediate server, in which multiple clients, typically distributed among different locations, need concurrent access to data stored in the same DBMS.

3. Problem Definition

Database-as-a-service is a secondary cloud computing service model. Users can access the database from cloud without need to installation of software and configuring hardware. The DBAAS reduces the maintenance cost in organization. The main drawback is lack of control over the database. Cloud providers are untrusted, whatever data storing in database, there is no security for data, and cloud providers can hack the data. This paper considers database-as-a-service and providing security for stored databases. Paper proposes secure database-as-a-service (SDBAAS).

4. System Architecture

SecureDBaaS allow multiple and independent clients to access encrypted database from untrusted cloud server with confidentiality. Figure (3.1) shows the architecture of secureDBaas.

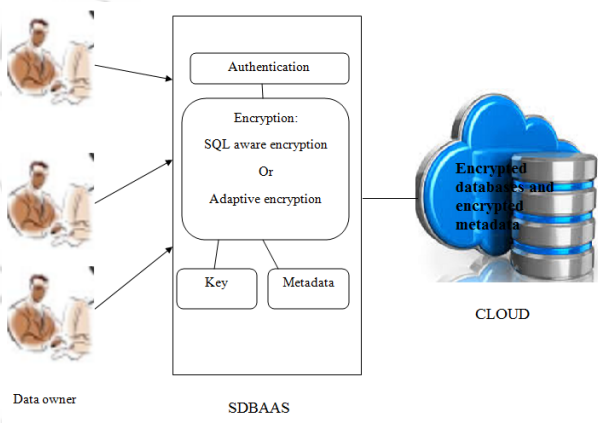

Figure 3.1: Architecture of SDBAAS

Multiple numbers of users can store their databases, access their databases using SDBaaS. Consider an organization, there will be multiple users using client machine. Each client machine installs SDBAAS, so each user can create their databases; provide security by encrypting databases using SQL aware encryption scheme or adaptive encryption scheme. Metadata contains information related to encryption and decryption. Each users have their own metadata. Also each users encrypt the database using information from metadata and encrypt metadata using their private or master key. Then they can store their encrypted metadata and encrypted databases to cloud. Whenever user wants their database, provide sql queries to cloud. SecureDBaaS analyzes the original operation to identify which tables are involved and to retrieve their metadata from the cloud database. The metadata are decrypted through the master key and their information is used to translate the original plain SQL into a query that operates on the encrypted database.

Translated operations contain neither plaintext database (table and column names) nor plaintext tenant data. Nevertheless, they are valid SQL operations that the SecureDBaaS client can issue to the cloud database. Translated operations are then executed by the cloud

database over the encrypted tenant data. This paper mainly compares the SQL aware encryption scheme and adaptive encryption scheme.

a) SQL aware encryption scheme.

In SDBAAS all table are encrypted in different manner. That is for each column we assign a secure type (data type, encryption algorithm, field confidentiality). Data type means type of column example int, char etc. Encryption algorithm means with encryption algorithm to be used to encrypt the corresponding column. Field confidentiality is three types:

- COL: Assigned to column which are encrypted using one key.
- MCOL: Assigned to two or more columns which are encrypted using same key
- DBC: Entire table or database is encrypted using same key.

Based on each secure type, encrypt column using different SQL aware encryption methods:

- Random (RND): It is the most secure encryption because it does not reveal any information about the original plain value. It does not support any SQL operator, and it is used only for data retrieval. An efficient construction of RND is to use a block cipher like AES or Blowfish in CBC mode together with a random initialization vector (IV). Mostly use AES, except for integer value, because the 128-bit block size of AES would cause the cipher text to be significantly longer.
- Plain : it does not encrypt data; it is useful to support all SQL operators on non confidential data.
- Deterministic (DET): It deterministically encrypts data, so that equality of plaintext data is preserved. It supports the equality operator. DET has a slightly weaker guarantee, yet it still provides strong security: it leaks only which encrypted values correspond to the same data value, by deterministically generating the same cipher text for the same plaintext. This encryption layer allows the server to perform equality checks, which means it can perform selects with equality predicates, equality joins, GROUP BY, COUNT, DISTINCT, etc. DET implements using AES in CBC mode.
- OPE(order preserving encryption): it preserves in the encrypted values the numerical order of the original unencrypted data. It supports the comparison SQL operators (i.e., =,<, ≤,>,≥).
- Homomorphic encryption (HOM): It support sum operators.
- Word search (SEARCH): SEARCH is used to perform searches on encrypted text to support operations such as SQL's LIKE operator.

The main drawback of SQL aware encryption scheme is, we need to decide at the design time which sql operations support each column. If we introduce a new SQL operation, it will prevent the execution.

b) Adaptive encryption scheme.

Adaptive encryption scheme avoids the drawback of SQL aware encryption scheme. It supports run time at any SQL operation. Here each encryption method organized as layered manner called onions. Each column encrypted using two or more onions that is using two or more encryption algorithm.

Adaptive encryption scheme consist of:

- Onion-Eq: it supports the equality operator, and integrates Plain, Det and Rand layers.
- Onion-Ord: it supports the comparison operators (i.e., =,<,≤,>,≥), and integrates Plain, Ope and Rand layers.
- Onion-Sum: it supports the sum operator, and integrates Plain, Sum and Rand layers.
- Onion-Search: it support the string equality operator (LIKE), and integrates the Plain, Search and Rand layers.
- Onion-Single-Layer: this is a special type of onion that supports only one encryption layer.

Each plaintext column is converted into one or more encrypted columns, each one corresponding to an onion. Each plaintext value is encrypted through all the layers of its onions.

5. Methodology

We can consider an example related with a customer database in a bank consisting of customer's information along with his credit card information. Every bank will have much type of confidential data, for maintaining these data, must have database and should expert with DBMS. So SDBAAS will help the manager to store every database into cloud after providing security. The schema for storing such information will be in the form of tables. Some tables containing personal information of the user and some tables containing information regarding to credit cards and will be mapped using their ids. This particular information can be stored in a bankDB database as follows: Customertable(CustomerId, CustomerName, CustomerAddress, CustomerPhone, CustomerDOB).Membershiptable(CustomerId,Password, PasswordQuestion, PasswordAnswer). Creditcardtable (CardId, CreditcardNo, CardExpiryDate, CVVNo) Customer_Creditcardtable containing (CustomerId, CardId). Any user can register into this web based bank application. User will register with his personal information. Then system will provide account number and customer ID for each customer. This will stored in Customertable. Then he can login to the application with username and password, perform transactions like transfer money, pay bills ect. The manager will save every user's transactions in the table and upload into cloud after encryption with SQL aware encryption scheme or adaptive encryption scheme. Whenever user or manager needs information, provide SQL commands. The SDBAAS provide corresponding result after downloading and decrypting the databases.

6. Result and Discussion

The SDBAAS concept implemented in bank application. The bank manager or admin in the application will encrypt every days transactions stored in tables using sql aware encryption and adaptive encryption scheme. SDBAAS application allow any user to register and login to online

bank application, and the user can deposit amount, withdraw the amount, transfer amount from one account to another account, pay bills etc. Every transaction will automatically stored in database. Admin of the application will encrypt the stored details and upload in to cloud. When admin use sql aware encryption for encrypting the tables, it support only some sql operators like insert, select, delete, update. Admin will provide sql queries for accessing tables, but it support only these sql operators like insert, select, delete, and update. When the admin use adaptive encryption scheme for encrypting details it support insert, select, delete, update, GROUP BY, SUM and COUNT.

7. Conclusion

DBaaS is one of the main service of cloud computing. This project proposed secure DBaaS, which allow multiple cloud users can access database concurrently and independently over the encrypted data. It also maintains the confidentiality of data because it does not rely on proxy server. It provides same availability, scalability and elasticity of traditional DBaaS. It provides confidentiality to the cloud database using adaptive encryption architecture. This project also compares the performance of sql aware encryption scheme and adaptive encryption scheme.

The paper presented a privacy preserving data storage to cloud. Metadata based model will take some quantifiable effort to be implemented in real time, it provides necessary solution for an environment like cloud computing. This paper is extended the outcome of Metadata based model. In this model, by applying an encryption technique, the privacy of the data can be preserved more efficiently.

8. Future Scope

SDBAAS provide security for stored databases in cloud. Security is provided by two encryption scheme like sql aware encryption and adaptive encryption scheme. Sql aware encryption scheme support only insert, select, delete and update sql operators. But adaptive encryption scheme support insert, select, delete, update, GROUP BY, SUM, COUNT. But both encryption schemes take some time to complete the process. Also both will not support every sql operators like JOIN, logical operators like BETWEEN, ANY, EXIST etc. The future work is concentrated to support every sql operators including logical operators.

Reference

[1] L. Ferretti, M. Colajanni, and M. Marchetti, "Distributed, concurrent, and independent access to encrypted cloud databases," IEEE Trans. Parallel and Distributed Systems, vol. 25, no. 2, Feb. 2014
[2] H. Hacigu¨mu¨ s¸, B. Iyer, and S. Mehrotra, "Providing Database as a Service," Proc. 18th IEEE Int'l Conf. Data Eng., Feb. 2002.
[3] M. Hadavi, E. Damiani, R. Jalili, S. Cimato, and Z. Ganjei, "AS5: A Secure Searchable Secret Sharing Scheme for Privacy Preserving Database Outsourcing, Proc. Fifth Int'l Workshop Autonomous and Spontaneous Security, Sept. 2013.
[4] G. Cattaneo, L. Catuogno, A.D. Sorbo, and P. Persiano, "The Design and Implementation of a Transparent Cryptographic File System For Unix," Pro FREENIX Track: 2001 USENIX Ann. Technical Conf., Apr. 2001.
[5] E. Damiani, S.D.C. Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati, "Balancing Confidentiality and Efficiency in Untrusted Relational Dbmss," Proc. Tenth ACM Conf. Computer and Comm. Security, Oct. 2003.
[6] R.A. Popa, C.M.S. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Protecting Confidentiality with Encrypted Query Processing," Proc. 23rd ACM Symp. Operating Systems Principles, Oct. 2011.
[7] H. Hacigu¨mu¨ s¸, B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over Encrypted Data in the Database-Service-Provider Model," Proc. ACM SIGMOD Int'l Conf. Management Data, June 2002.
[8] E. Mykletun and G. Tsudik, "Aggregation Queries in the Database-as-a-Service Model," Proc. 20th Ann. IFIP WG 11.3 Working Conf. Data and Applications Security, July/Aug. 2006.
[9] L. Ferretti, M. Colajanni, and M. Marchetti, "Supporting Security and Consistency for Cloud Database," Proc. Fourth Int'l Symp. Cyberspace Safety and Security.
[10] http//: www.cloudsolutions.com.
[11] Rackspace, "Rackspace Cloud Database," "http//:wwww.rackspace.com/cloud/database Mar 2014.
[12] Amazon, "Amazon Web Services Blog," http://aws.typepad.com/aws/price reduction/, Mar. 2014.
[13] Amazon RDS Pricing, "Amazon Relational DatabasePricing,"http://aws.amazon.com/rds/pricing, Mar. 2014.
[14] EnterpriseDB, "Postgres Plus Cloud DatabasePricing,http://www.enterprisedb.com/cloud-database/pricing-amazon, Mar.2014.
[15] L. Ferretti, M. Colajanni, and M. Marchetti, "Access control enforcement of query-aware encrypted cloud databases," in Proc. Fifth IEEE Int'l Conf. Cloud Computing Technology and Science, Dec. 2013.

Author Profile

Seethal .K S completed the bachelor's degree in Information Technology from University of Calicut and presently pursuing Masters in Engineering in Computer Network Engineering at Mangalore Institute of Technology, Mangalore

Mr.Siddana Gowda G.R Designation: Asst Professor Qualification : B.E, M.Tech (CS&E). Published paper named "Image retrieval using semantics of query image" in two journals namely IJDRET and JCER .