Multi-User Encrypted SQL Operation on Cloud Database Services in Real Distributed Environment

R. S. Khatawkar[1], N. M. Sawant[2], V. V. Pottiger[3]

[1]*ME- CSE, SKNSCOE, Korti, Pandharpur, Solapur University, Solapur, India*

[2]Assistant Professor, *SKNSCOE, Korti Pandharpur, Solapur University, Solapur, India*

[3]Assistant Professor, *NBNSCOE, Solapur, Solapur University, Solapur, MS, India*

[1]reshmakhatawkar32@gmail.com, [2]namdev.sawant@sknscoe.ac.in, [3]vinayak.pottigar@gmail.com

Abstract* - In a cloud context, the critical information is placed in infrastructure of untrusted third parties, so ensuring data confidentiality stored in cloud is an important factor. The cloud database is strictly related to parameters such as service availability, scalability and security and of data confidentiality. The parameters like security and availability of its platform, are ensure by any cloud provider. But to guarantee confidentiality of the information stored in cloud databases is an open research problem. The researcher suggest some preliminary issues on encrypted data through SQL operations. The proposed scheme for data confidentiality contains authentication, authorization, data encryption, key management policies. But these policies addresses the issues related to typical threat scenarios for cloud database services. So to avoid the threat issues through authentication and authorization data encryption, key management some mechanism is needed. Access control mechanism is used for guaranteeing confidentiality of data and metadata.

Keywords -Access control, Encryption, Database, etc

I. INTRODUCTION

Cloud computing is an emerging paradigm for large scale infrastructures. Now a days there are so many cloud services are available. The cloud database uses platform service of cloud computing. Cloud database services are differ by the perception of confidentiality risks when information is stored in cloud infrastructures. To guarantee data confidentiality and data isolation for cloud databases the cryptography is used. But the cryptography addresses some issues when there is need to perform computations over encrypted data. Cryptography gives partial solutions to data confidentiality. So to give complete and separate solution for data confidentiality and data isolation 'Multi-User relational Encrypted Database (MuteDB)', is used that guarantees data confidentiality by executing SQL operations on encrypted data and by enforcing access control policies. Guaranteeing confidentiality and isolation of data stored in cloud database infrastructures that are subject to two types of threats:1. Guaranteeing data confidentiality in the cloud against external attackers, cloud insiders,2. Tenant(group of user with specific privileges) insider can disclose its credentials to cloud insider. MuteDB is used to protect data against external attackers, cloud insiders and tenant in-siders, and against collusion between these roles. The DBA is in charge of translating the access control policies into an access control matrix used by MuteDB.

The DBA client takes as its input the original plaintext database, and produces the encrypted tenant data. The DBA distributes unique secret keys to the users at the creation of their accounts according to the access control matrix. These keys enable the users to access (decrypt) all and only the subsets of encrypted data corresponding to the structures on which the users have legitimate access. The MuteDB allow to store metadata in the cloud database together with encrypted data. This approach allows each client to access metadata directly and concurrently through standard SQL operations, The MuteDB stores all metadata in three tables. 1.The database tokens table- stores all information related to the encryption enforcement scheme. 2.The database encryption table –store all information related to the algorithms and keys used to encrypt resources. 3.The users tokens table -stores all information related to the users credentials. The authenticated user can access the encrypted data through encrypted SQL operations. The metadata is responsible for converting plaintext SQL operation into encrypted SQL operations. So, MuteDB guaranteeing efficient retrieval of database metadata that are stored in an encrypted form in the cloud database.

II. LITERATURE SURVEY

Many confidentiality solutions exist for cloud storage services but they do not support the execution of SQL operations on encrypted data. Other techniques guaranteeing data confidentiality through encryption managed by the cloud provider, standard database methods and policy enforcement strategies are not acceptable because modern threat models assume that a cloud provider employee could access tenant data. [1] MuteDB is more related to proposals performing operations on encrypted databases, and enforcing access control at the encryption level. [2] Some interesting solutions for enforcing access control policies on outsourced information are proposed. The Secure DBaaS (secure database as service) is used that supports the execution of concurrent and independent operations to the remote encrypted database from many geographically distributed clients as in any unencrypted DBaaS setup. To achieve these goals, Secure DBaaS integrates existing cryptographic schemes, isolation mechanisms, and novel strategies for management of encrypted metadata on the untrusted cloud database. [3] In this paper, the clients issue SQL queries through one trusted proxy managing all encryption and decryption operations, and forwarding them to the encrypted cloud database. Proposed system avoid a similar approach because any architecture relying on one intermediate server limits the availability and elasticity of a cloud database service. Moreover, from the access control perspective, the proposed solutions are similar to that of an internally managed infrastructure where a trusted proxy stores all encryption and decryption keys, and clients access the encrypted database transparently. [4] The privacy, security and trust issues associated with cloud computing in more detail, together with related legal concerns.

There is necessarily some overlap and interdependency between such issues, but nevertheless we believe it is still helpful to categorise these issues in this way. In doing this, we take into account the alternative delivery and deployment models for cloud computing, as these influences the risks involved. [5] MONOMI is a system for securely executing analytical workloads over sensitive data on an untrusted database server. MONOMI works by encrypting the entire database and running queries over the encrypted data. MONOMI introduces split client/server query execution, which can execute arbitrarily complex queries over encrypted data, as well as several techniques that improve performance for such workloads, including per-row precomputation, space-efficient encryption, grouped homomorphic addition.

III. PROPOSED SYSTEM

A. Plaintext database model:

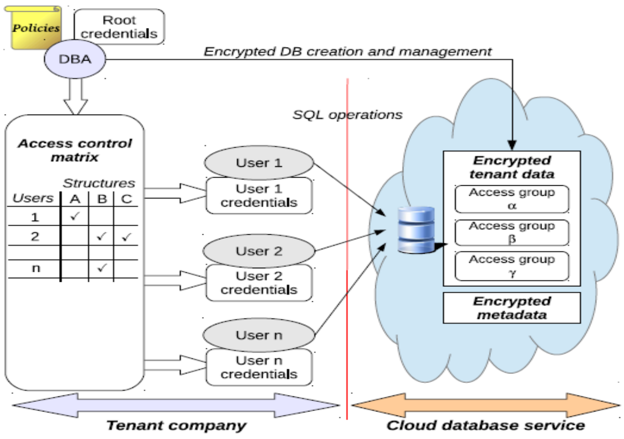
Plaintext most commonly meant message text in the language of the communicating parties. The original definition implied that the message could be read by a human being, the modern definition emphasizes that a person using a computer could easily interpret the data. Thus, in a significant sense, plaintext is the 'normal' representation of data before any action has been taken to conceal, compress, or 'digest' it. The proposed plaintext database model is a poset that extends the structure poset S, with the resources R, a structure s<S associated with a resource r< R is a parent of the resource r (s > r).

B. Access control:

Access control a is way of limiting access to a system or to physical or virtual resources. In computing, access control is a process by which users are granted access and certain privileges to systems, resources or information.. The access control mechanism for muteDB defined as- Let R be the set of resources that represent plain text tenant data, S the set of plaintext database structures, E the set of encrypted tenant data, U the set of users, and K the set of encryption keys. A is access control matrix where, for each user u P U and for each structure s P S, there exists a binary authorization rule a that defines whether an access to s by u is denied or allowed.

C. Encrypted database model:

Database encryption is the process of converting data, within a database. In plaintext format into meaningless cipher text by the means of a suitable algorithm. For each plaintext table, the MuteDB DBA client generates the corresponding encrypted table and a unique encryption key. The name of the encrypted table is computed by encrypting the name of the plaintext table through that key. The encryption algorithm used for encrypting the table names is a standard AES algorithm in a deterministic mode (e.g., CBC with constant initialization vector). In such a way, only the users that know the plaintext table name and the corresponding encryption key are able to compute the name of the encrypted table. The deterministic scheme is preferred because it allows a correspondence between plaintext and encrypted tables and improves the efficiency of the query translation process.



Users	A	B	C
1	✓		
2		✓	✓
n		✓	

Fig 1: MuteDB Architecture

D. Metadata management:

Database metadata include all information allowing a MuteDB client to translate plaintext SQL operations into operations working on the encrypted database. The Mute DB alternative is to store metadata in the cloud database together with encrypted tenant data. This approach allows each client to access metadata directly and concurrently through standard SQL operations, thus avoiding system bottlenecks and single point of failures at the tenant side. MuteDB proposes a new metadata management strategy that enforces access control policies at the encryption level, by generating a different encryption key for each user and by ensuring that each user is able to decrypt all and only encrypted data on which he/she has legitimate access.

E. MuteDB:

The MuteDB DBA client, that is the application for the creation and management of the encrypted database. All database users can issue SQL operations directly to the cloud database from distributed locations by executing a Mute DB client on their machines. The entire set of data are stored in an encrypted form in the cloud database. The cloud database engine can execute queries on encrypted data without accessing any decryption keys. Even metadata that are necessary to manage encryption strategies are considered critical information, hence Mute

DB stores them encrypted in the cloud database: the DBA and the tenant users can efficiently retrieve metadata through standard SQL queries. We refer to the encrypted forms of tenant data and metadata as encrypted tenant data and encrypted metadata.

IV. LIMITATION

The proposed architecture is specifically designed for cloud database scenarios. MuteDB does not rely on any intermediate trusted server that could become a system bottleneck and a single point of failure. In a cloud database scenario, the malicious operations of a tenant insider are limited by access control policies, but these policies cannot prevent the possibility that a tenant insider discloses its credentials including its decryption key(s) to a cloud insider.

The current implementation of the MuteDB prototype includes all the encryption algorithms that are necessary to support each SQL operation on the encrypted database columns.

V. COMPARISION

[1] The proposed system Guarantees data confidentiality in the cloud against external attackers, cloud insiders, under the assumption that they do not collude, and it can be achieved through MuteDB.

[2] The secureDBaas is used to support access to multiple, independent, and geographically distributed clients and to execute concurrent operations on encrypted data.

[3] CryptDB is a system that provides practical and provable confidentiality against external attackers. CryptDB can also chain encryption keys to user passwords, so that a data item can be decrypted only by using the password of one of the users with access to that data.

[4] The description about how security, trust and privacy issues occur in the context of cloud computing and discuss ways in which they may be addressed.

[5] The proposed system consist of securely executing analytical workloads over sensitive data on an untrusted database server that named as MONAMI .

VI. SCOPE

The proposed system, support the execution of SQL operations on encrypted data. Other techniques guaranteeing data confidentiality through encryption managed by MuteDB is as follows.

CryptDB: protecting confidentiality with encrypted query processing. In this mechanism clients issue SQL queries through one trusted proxy managing all encryption and decryption operations, and forwarding them to the encrypted cloud database.

Adjustable query-based encryption:

This technique is used when some encryption schemes leak more information than others about the data to the DBMS server.To implement these adjustments efficiently, CryptDB uses onions of encryption.Chain encryption keys to user passwords In this technique each data item in the database can be decrypted only through a chain of keys rooted in the password of one of the users with, access to that data.

VII. CONCLUSION

The MuteDB architecture for cloud database services that guarantees the data confidentiality through SQL-aware encryption algorithms and data isolation through access control enforcement based on encryption and key derivation techniques. These solutions allow MuteDB to address threat issues that are relevant for cloud services including risks of information leakage due to collusions between cloud provider employees and users.

REFERENCES

[1] Luca Ferretti, Fabio Pierazzi, Michele Colajanni, and Mirco Marchetti” Scalable architecture for multi-user encrypted SQL operations on cloud database services”IEEE TRANSACTION ON CLOUD COMPUTING VOL:PP NO:99 YEAR 2014.

[2] 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, pp. 437–446,2014.

[3] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan,“CryptDB: protecting confidentiality with encrypted query processing,” in Proc. 23rd ACM Symp. Operating Systems Principles,Oct. 2011, pp. 85–100.

[4] S. Pearson and A. Benameur, “Privacy, security and trust issues arising from cloud computing,” in Proc. 2010 IEEE Int’l Conf. Cloud Computing Technology and Science, Nov.-Dec. 2010, pp. 693 – 702.

[5] S. Tu, M. Kaashoek, S. Madden, and N. Zeldovich, “Processing analytical queries over encrypted data,” in Proc. 39th Int’l Conf. Very Large Data Bases, Aug. 2013, pp. 289–300.

[6] E. Damiani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia,S. Paraboschi, and P.Samarati, “Key management for multi-user encrypted databases,” in Proc. ACM Workshop Storage Security and Survivability, Nov. 2005, pp. 74 – 83.

[7] L. M. Vaquero, L. Rodero-Merino, and R. Buyya, “Dynamically scaling applications in the cloud,” ACM SIGCOMM Computer Communication Review, vol. 41, no. 1, pp. 45–52, 2011.

[8] M. R. Asghar, G. Russello, B. Crispo, and M. Ion, “Supporting complex queries and access policies for multi-user encrypted databases,” in Proc. 2013 ACM Workshop on Cloud computing security, Nov. 2013, pp. 77–88.

[9] O. Goldreich, Foundations of Cryptography: Volume 2, Basic Appli- cations Cambridge university press, 2004.

[10] H. Hacig¨um¨ us¸, B. Iyer, C. Li, and S. Mehrotra, “Executing sql over encrypted data in the database-service- provider model,” in Proc.2002 ACM SIGMOD Int’l Conf. Management of data, Jun. 2002, pp.216–227.