IPHV8I1035X

International Journal Of Advanced Research and Innovation -Vol.8, Issue .I

ISSN Online: 2319 – 9253

Print: 2319 – 9245

IMPROVED DISTRIBUTED INDEPENDENT ACCESS TO ENCRYPTED CLOUD DATABASES

[#1]P.SNEHA, M.Tech Student,
[#2]D.VENKATESHWARLU, Associate Professor,
Dept of CSE,
VIDYA JYOTHI INSTITUTE OF TECHNOLOGY, HYDERABAD, TELANGANA, INDIA.

Abstract: Since data in cloud will be placed anywhere, because of the critical nature of the applications, it is important that clouds be secure. The major security challenge with clouds is that the owner of the data may not have control of where the data is placed. This is because if one wants to exploit the benefits of using cloud computing. As the information of the imperceptive users is outsourced and concerning is reserve hector to the salver suitable to high storage and processing. As the bovine convention is increased in adequate territory head take reference to is a rebellious undertaking to oblige mainstay and auditing to the stored matter in the clod-like server. The wish of the movement is to power a unheard-of fib digress integrates cloud database help with data secretiveness and the alternative of executing concurrent operations on encrypted data. The tiny fabrication has the dormant financial statement of omitting middleman proxies focus square footage the scalability ,availability and elasticity properties that are intrinsic in cloud-based solutions. The adeptness of the tiny fable is evaluated scan digest analyses and enough extremist frugal based on a venerable execution vocation to the TPC-C important case for different numbers of clients and network latencies.

We propose SecureDBaaS as the first solution that allows cloud tenants to take full advantage of DBaaS qualities, such as availability, reliability, and elastic scalability, without exposing unencrypted data to the cloud provider. The architecture design was motivated by goal: to allow multiple, independent, and geographically distributed clients to execute concurrent operations on encrypted data, including SQL statements that modify the database structure.

Keywords- *Cloud, security, confidentiality, SecureDBaaS, database.*

I.INTRODUCTION

Cloud based services are becoming popular as they focus on high availability and scalability at low cost. While providing high availability and scalability, placing critical data to cloud poses many security issues. For avoiding these security issues the data are stored in the cloud database in an encrypted format. The encrypted cloud database allows the execution of

SQL operations by selecting the encryption schemes that support SQL operators. Encrypted cloud database permits different types of accesses such as distributed, concurrent, and independent. One of the architecture that supports these three kinds of access is SecureDBaaS, which was proposed by Luca Ferretti et al [1]. The SecureDBaaS architecture supports multiple and independent clients to execute concurrent SQL operations on encrypted data. Data consistency should be maintained by leveraging concurrency control mechanisms used in DBMS engines.

This survey explains the various concurrency control protocols that can be used in the encrypted cloud database. The applications need 1SR if data is replicated. Hence, to guarantee the merits of cloud, it is essential to provide high scalability, availability, low cost and data with strong consistency, which is able to dynamically adapt to system conditions. Self optimizing one copy serializability (SO-1SR) is the concurrency control protocol that dynamically optimizes all stages of transaction execution on replicated data in the cloud database [2]. Current DBMSs supported by cloud providers allows relaxed consistency guarantees which in turn increase the design complexity of applications [3]. The second concurrency control protocol is the snapshot isolation (SI) which provides increased concurrency in cloud environment when compared to 1SR [4]. Transactions are read from the snapshot, reads are never blocked because of write locks which in turn increases concurrency. SI does not allow many of the inconsistencies, but allows write skew anomalies. SI allows transaction inversions. To avoid transaction inversions strong consistency guarantee is required, i.e. strong SI (SSI).

The third concurrency control protocol is the session consistency (SC) [5]. Session consistency is a different variety of eventual consistency. The system provides read your writes consistency inside each session. Session consistency is at a low cost while considering response time and transaction cost.

The cost based concurrency control in the cloud is the C_3 i.e. cost-based adaptive concurrency control in cloud [6]. C_3 dynamically switch between strong consistency level and weak consistency level of transactions in a cloud database according to the cost at runtime. It is built on the top of 1SR and SSI.

II. SECURE DBAAS

SecureDBaaS (Secure database as a service) architecture proposed by Luca Ferretti et al supports multiple clients and clients which are geographically distributed to execute the independent and concurrent operation on encrypted data in the remote database [1]. SecureDBaaS also guarantees data confidentiality and cloud level consistency. This architecture eliminates the intermediate server between the cloud database and client in order to provide availability and scalability [7].

SecureDBaaS is the architecture that supports the concurrent execution of operations in the encrypted cloud database. The existing proxy based architecture constraints the multiple and distributed clients to access data concurrently from the same database. The data consistency during the concurrent access of data and metadata can be assured by using some isolation mechanisms or the concurrency control protocols in the cloud database. SecureDBaaS allows the execution of concurrent SQL operations (INSERT, DELETE, SELECT, UPDATE) from multiple and distributed clients. In order to provide data confidentiality the tenant data and metadata should be in an encrypted format. For this reason, clients convert plaintext SQL statements into SQL statements that support transactions and isolation mechanisms allowed in cloud databases [8]. The solutions for the consistency issues lies in the five contexts: (1) data manipulation (2) modification of structures (3) altering table (4) modification of secure type (5) unrestricted operations.

2.1. *Architecture design*

The architecture design of SecureDBaaS consists of one or more client machines with SecureDBaaS installed and cloud database. This client is responsible for the connection of a user to the cloud DBaaS to perform SQL operations. The SecureDBaaS manages plaintext data, metadata, encrypted data and encrypted metadata. The plaintext data includes the data user wants to save in cloud DBaaS [9]. In order to avoid

the confidentiality issues, multiple cryptographic approaches are used to convert plaintext data to encrypted form for storing in cloud database. The metadata includes information needed to encrypt or decrypt data. Moreover, metadata is also stored in an encrypted format [10].

Encryption Schemes:

The encryption schemes supported by SecureDBaaS [11] are:

(1) Plain: it supports the storage of unencrypted data in the cloud and allows all types of SQL operations.

(2) OPE: order preserving encryption permits the execution of inequality and range queries on encrypted data.

(3) Det: it permits the execution of equality and aggregation operators on encrypted data.

(4) Random: it assures highest confidentiality level.

But it restricts all SQL operators.

2.2. *Implementation*

SecureDBaaS client consists of five components:

Operation parser software: Is responsible for the conversion of receiving plain text SQL command into intermediate form which is processed later by other modules.

Encryption engine: Is responsible for all kinds of encryption and decryption operations specified in the metadata of SecureDBaaS.

Metadata manager: it manages metadata local copies and assures its consistency.

Query writer: it translates the query in intermediate form from the operation parser into SQL statements that can be executed by the cloud database over encrypted data.

Database connector: it acts as an interface between client and remote DBMS.

III. CLOUD COMPUTING SECURITY THREATS AND SOLUTION

Distributed Data: - This mechanism is used to share the data of the user in networks while their roaming when the user need. Data distributed among different locations, need concurrent access of an encrypted data. To preserve data privacy and stability of the user data; we have to eliminate the intermediary server between the user and the cloud provider. Among different providers may taking advantage of secret sharing. Without intermediate server data distribution can done in secure level [1].

Privacy issues: - A Privacy issue is one of the main issues for the data user who stored their data in the cloud environments

[2]. Every user may want their personal data in private manner. Sometimes cloud provider compromise the data to the malicious attackers, so the problem may occur for the data user. With the use of external provider data may loss, so user must make sure who is accessing the data and who is maintaining the server at every time to protect their data. For this privacy issues user can encrypt the data so no one can access the data.

Encryption is one of the best methods to protect the data. Encryption is based on embedding the text into some format it may be ciphertext, audio embedding process. Control issues: - Controlling the data from the unauthorized is one of the main issues for outsourced data in a cloud. Physical control is one of the best methods for the control mechanism and at the same time every time physical control is not a possible one from the unauthorized one [9, 10]. When compare to physical scheme an automatic control mechanism can provide a secure one in the possible of every time. Visualization is one of the important one to control the users data and maintain control over access to user resources. This control mechanism is ability to control the deployed applications and potentially application of the user. Concurrent and independent access: - Concurrently and independently access in a cloud in important one for a cloud database service, protecting data privacy to the user data by allowing a cloud database to perform concurrent operations over an encrypted data, for eliminating a trusted broker or trusted proxy [11]. For this concurrency and independent model Secure Database as a Service (SDBaaS) integrate cloud database with secure provider manner for data Privacy and security. Concurrency model is used to Read/Write operation with the user database in a secure manner [12].

Identity and Access management: - In cloud computing data is stored in distributed location with a many client and run in extraction process with large amount of data of client information. To accessing the data over network may occur an untrustful problem because of increasing no. of attackers in networks, so who anyone can access our data without our permission which is called hacking process. To control the unauthorized access we provide a mechanism called access control tool, to control the data over distributed networks [13, 14]. Access control works in the bases of authenticate the authorized user with a sigh on mechanisms. It provides a data access matrix to monitor the accessing data limits. Here we provide a mechanism to access the data in limited manner which is controlled by the data user. Identity mechanism is used to the unauthorized one by sign on of instant user when an actual user is signed in. this mechanism is used to manage the multiple user in a network.

IV.IMPLEMENTATION

4.1 Data Management:

Cloud database acts as service provider for tenants. The cloud is created first for the system. All information or data store in the relational database. So for creating tables and column we have to access it with SQL query only.

4.2 Metadata Management:

Metadata generated by SecureDBaaS contain all the information that is necessary to manage SQL statements over the encrypted database in a way transparent to the user. Metadata management strategies represent an original idea because SecureDBaaS is the first architecture storing all metadata in the untrusted cloud database together with the encrypted tenant data. SecureDBaaS uses two types of metadata.

• Database metadata are related to the whole database. There is only one instance of this metadata type for each database.

• Table metadata are associated with one secure table. Each table metadata contains all information that is necessary to encrypt and decrypt data of the associated secure table.

This design choice makes it possible to identify which metadata type is required to execute any SQL statement so that a SecureDBaaS client needs to fetch only the metadata related to the secure table/s that is/are involved in the SQL statement.

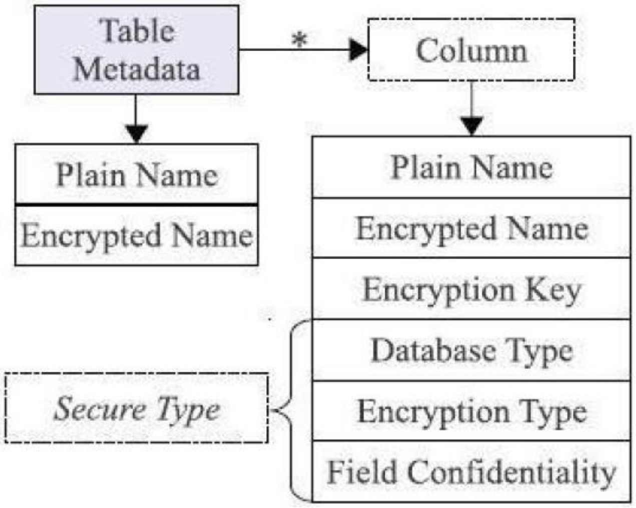


Fig.1. Structure of table metadata.

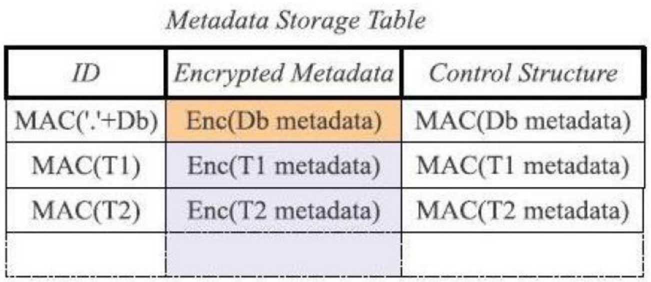
This design choice minimizes the amount of metadata that each SecureDBaaS client has to fetch from the untrusted cloud database, thus reducing bandwidth consumption and processing time. Moreover, it allows multiple clients to access independently metadata related to different secure tables.

Database metadata contain the encryption keys that are used for the secure types. A different encryption key is associated with all the possible combinations of data type and encryption type. Hence, the database metadata represent a key ring and do not contain any information about tenant data. The structure of a table metadata is represented in Fig. 1. Table metadata contain the name of the related secure table and the unencrypted name of the related plaintext table. Moreover, table metadata include column metadata for each column of the related secure table. Each column metadata contain the following information.

- Plain name: the name of the corresponding column of the plaintext table.
- Coded name: the name of the column of the secure table. This is the only information that links a column to the corresponding plaintext column because column names of secure tables are randomly generated.
- Secure type: the secure type of the column. This allows a SecureDBaaS client to be informed about the data type and the encryption policies associated with a column.
- Encryption key: the key used to encrypt and decrypt all the data stored in the column.

SecureDBaaS stores metadata in the metadata storage table that is located in the untrusted cloud as the database. This is an original choice that augmnts flexibility, but opens two novel issues in terms of efficient data retrieval and data confidentiality. To allow SecureDBaaS clients to manipulate metadata through SQL statements, we save database and table metadata in a tabular form. Even metadata confidentiality is guaranteed through encryption. The structure of the metadata storage table is shown in Fig. 2 This table uses one row for the database metadata, and one row for each table metadata.

Database and table metadata are encrypted through the same encryption key before being saved. This encryption key is called a master key. Only trusted clients that already know the master key can decrypt the metadata and acquire information that is necessary to encrypt and decrypt tenant data. Each metadata can be retrieved by clients through an associated ID, which is the primary key of the metadata storage table. This ID is computed by applying a Message Authentication Code (MAC) function to the name of the object (database or table) described by the corresponding row. The use of a deterministic MAC function allows clients to retrieve the metadata of a given table by knowing its plaintext name. This mechanism has the further benefit of allowing clients to access each metadata independently, which is an important feature in concurrent environments. In addition, SecureDBaaS clients can use caching policies to reduce the bandwidth overhead.



Metadata Storage Table

ID	Encrypted Metadata	Control Structure
MAC('.'+Db)	Enc(Db metadata)	MAC(Db metadata)
MAC(T1)	Enc(T1 metadata)	MAC(T1 metadata)
MAC(T2)	Enc(T2 metadata)	MAC(T2 metadata)

Fig.2. Organization of database metadata and table metadata in the metadata storage table.

4.3 Algorithms:

Encryption algorithms are applied to encrypt the database. There are various encryption algorithms symmetric and asymmetric, but we will apply symmetric algorithm which proved key distribution only once to all tenants there will be no different private key related to every user.

V. CONCLUSION

In this paper, we have discussed concurrent and independent access to encrypted cloud databases, proposes an innovative architecture that guarantees confidentiality of data stored in public cloud databases. The proposed system will not require modifications to the cloud database, and it will be immediately applicable to existing cloudDBaaS. Resolve problem of single point failure and a bottleneck limiting availability and scalability of cloud database services.

In this survey we present threats and solution of security and privacy for the cloud user. Cloud computing is one of the important for the cloud user to access the data through network at any where, so they were worried about the security problem of their personal data. Their personal data are maintained by the cloud provider. Providing security to their data is most important one for the user from the provider. Sometime user may afraid from the provider also because they may leak the data by compromise with the untrusted one.

REFERENCES

[1] L. Ferretti, M. Colajanni, and M. Marchetti, “Distributed, Concurrent, and Independent Access to Encrypted Cloud Databases,” IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 437–446, Feb. 2014.

[2] I. Fetai and H. Schuldt, “SO-1SR: towards a selfoptimizing one-copy serializability protocol for data management in the cloud,” in Proceedings of the fifth international workshop on Cloud data management, 2013, pp. 11–18.

[3] C. Curino, E. P. Jones, R. A. Popa, N. Malviya, E. Wu, S. Madden, H. Balakrishnan, and N. Zeldovich, "Relational cloud: A database-as-aservice for the cloud," 2011.

[4] K. Daudjee and K. Salem, "Lazy database replication with snapshot isolation," in Proceedings of the 32nd international conference on Very large data bases, 2006, pp. 715–726.

[5] T. Kraska, M. Hentschel, G. Alonso, and D. Kossmann, "Consistency Rationing in the Cloud: Pay only when it matters," Proc. VLDB Endow., vol. 2, no. 1, pp. 253–264, 2009.

[6] I. Fetai and H. Schuldt, "Cost-based data consistency in a data-as-a-service cloud environment," in Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on, 2012, pp. 526–533.

[7] Y. Lu and G. Tsudik, "Enhancing data privacy in the cloud," in Trust Management V, Springer, 2011, pp. 117–132.

[8] L. Ferretti, M. Colajanni, and M. Marchetti, "Supporting security and consistency for cloud database," in Cyberspace Safety and Security, Springer, 2012, pp. 179–193.

[9] H. Hacigumus, B. Iyer, and S. Mehrotra, "Providing database as a service," in Data Engineering, 2002. Proceedings. 18th International Conference on, 2002, pp. 29–38.

[10]K. P. Puttaswamy, C. Kruegel, and B. Y. Zhao, "Silverline: toward data confidentiality in storage-intensive cloud applications," in Proceedings of the 2nd ACM Symposium on Cloud Computing, 2011, p. 10.

[11]L. Ferretti, F. Pierazzi, M. Colajanni, and M. Marchetti, "Security and confidentiality solutions for public cloud database services," in SECURWARE 2013, The Seventh International Conference on Emerging Security Information,
Systems and Technologies, 2013, pp. 36–42.

[12]L. Ferretti, M. Colajanni, M. Marchetti, and A. E. Scaruffi, "Transparent Access on Encrypted Data Distributed over Multiple Cloud Infrastructures," in CLOUD COMPUTING 2013, The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization, 2013, pp. 201–207.

[13] J. G. U. Berkeley and others, "A Critique of ANSI SQL Isolation Levels," Online Verfügbar Http131107, vol. 65.

[14]A. J. Feldman, W. P. Zeller, M. J. Freedman, and E. W. Felten, "SPORC: Group Collaboration using Untrusted Cloud Resources.," in OSDI, 2010, vol. 10, pp. 337–350.

[15]M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and others, "A view of cloud computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.

[16]W. Jansen, T. Grance, and others, "Guidelines on security and privacy in public cloud computing," NIST Spec. Publ., vol. 800, p. 144, 2011.

[17]C. Almond, "A practical guide to cloud computing security," White Pap. Accent. Microsoft, 2009.

[18]S. Hildenbrand, D. Kossmann, T. Sanamrad, C. Binnig, F. Faerber, J. Woehler, D. Kossmann, and D. Kossmann, Query Processing on Encrypted Data in the Cloud by. ETH, Department of Computer Science, 2011.

[19]Y. Sun, J. Zhang, Y. Xiong, and G. Zhu, "Data Security and Privacy in Cloud Computing," Int. J. Distrib. Sens. Netw., vol. 2014, pp. 1–9, 2014.