INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY
VOLUME 4 ISSUE 2 – APRIL 2015 - ISSN: 2349 - 9303

An efficient concurrent access on cloud database using secureDBAAS

G. Elakkia.,(M.E)
Dept of Computer Science and Engineering
RVS Faculty of Engineering, Coimbatore.
R.Elakkia10@gmail.com

Prof. S. K. Mouleeswaran.,(Ph.d)
Dept of Computer Science and Engineering
RVS Faculty of Engineering, Coimbatore.
Meetmoulee@gmail.com

Abstract—Cloud services provide high availability and scalability, but they raise many concerns about data confidentiality. SecureDBaas guarantees data Confidentiality by allowing a database server for execute SQL operation over encrypts data and the possibility of executing concurrent operation on encrypts data. It's supporting geographically distributed clients to connect with an encrypt database, and for execute an independent operation including those modifying the database structure. The proposed architecture has the advantage of eliminating proxies that limit the several properties that are intrinsic in cloud-based solutions. SecureDBaas that support the execution of concurrent and independent operation for the remote database from many geographically distributed clients. It is compatible for the most popular relational database server, and it is applicable for different DBMS implementation. It provides guarantees for data confidentiality by allowing a cloud database server for execute SQL operation over encrypts data.

Index Terms — Database, SecureDBaaS, Cloud, Security

1 INTRODUCTION

1.1 CLOUD DATABASE

A database is accessible to clients from the cloud and delivered for users via the Internet from a cloud database provider's server. It also known as Database-as-a-Service (DBaaS) [8], cloud databases can use cloud computing in order for optimized scaling, high availability, and multi-tenancy and resource allocation process. A cloud database can be a traditional database such as a MYSQL or SQL Server database that has been used for cloud use. Cloud databases can offer several benefit over their traditional system, including accessibility, automatic failover and fast recovery from system failures, automated on-the-go scaling, minimal investment and maintenance.

Cloud DBMS (CDBMS) as a distributed database that delivers a query service across multiple distributed database nodes located in multiple geographically-distributed data centers [6]. A query can originate from anywhere; from a PC within the corporation, which is connected by a fast line for the local data center, from a PC in the home via a VPN line, from a laptop via a Wi-Fi connection, or from a smart phone via a 3G or 4G connection. For that reason we represent a query here as coming "through the Internet," implying that the response will possibly travel through the Internet too.

2 EXISTING SYSTEM

Existing System contains A The proxy based cloud database, that requires any client operation should pass through one intermediate server, but it is not suitable for cloud-based system, in which multiple clients typically distributed from different locations, and need concurrent access in order for store data in the same DBMS. If a proxy fails, then we cannot perform transaction between client and server and Policy inconsistencies during policy updates due for the consistency model.

Encrypting blocks of data instead of each data item [8]. Whenever a data item that belongs for a block is needed, then the trusted proxy needs for collect the whole block, for decrypt it, and filter out unnecessary data that belong for the same block. Then it requires heavy modification of the original SQL operation produced by each client. Thereby it causes significant overheads on both the DBMS server and the trusted proxy. Prevent the cloud providers for read a portion of data, but information can be rearrange by the cloud providers and allow execution of operation over encrypts data. These functions' preserve data confidentiality, where DBMS is not trusted [3]. It requires modified DBMS software used by the cloud providers.

Some DBMS engines encrypt data at the file system level. It means transparent data Encryption feature. This feature makes it possible for build a trusted DBMS over untrusted storage.

Focused on different usage contexts, including data manipulation, modification for the database structure and does not provide data confidentiality for database as a service. A proxy that characterizes and facilitates the implementation of a SecureDBaaS, and it is applicable only multi tier web application. It causes several drawbacks. Since the proxy is trusted, and It function's cannot be outsourced for an untrusted the cloud providers. Hence, a proxy is implemented and managed by the cloud tenant. Properties of SecureDBaaS service that are availability, scalability, and elasticity are then bounded by trusted proxy of availability, scalability, and

elasticity; it becomes a single point of failure and a system bottleneck. A proxy is meant for implemented and managed by cloud tenant. There are several solutions ensuring confidentiality for the storage as a service by using SQL aware encryption [13]

3 PROPOSED SYSTEM

3.1 SecureDBaas

SecureDBaaS allow multiple and independent clients for connect directly for the untrusted cloud DBaaS without any intermediate server a cloud database service from an untrusted DBaaS provider [8].

SecureDBaaS architecture is created for cloud platforms and does not involve any intermediary a proxy or broker server between the client and the cloud provider.

Tenant then deploys one or more machines (Client 1 through N) and installs a SecureDBaaS client. So that client allows a cloud user for connect with the cloud DBaaS, in order for read and write data, and modify the database tables after creation. SecureDBaaS includes plain text data, modified data, metadata, and modified metadata. Plain text data contain information and tenant wants to store and processes it remotely in the cloud DBaaS. SecureDBaaS contains several cryptographic techniques for transform plain data into modified tenant data and modified tenant data structures because the names of the tables and their columns must be modified in for other form. SecureDBaaS clients contain sets of metadata in order for encrypt and decrypt data. Even metadata is encrypts and stored in the cloud DBaaS.

SecureDBaaS that supports the execution of concurrent and independent operation for the remote encrypts database from many geographically distributed clients as in any encrypts DBaaS. It is compatible with the most popular relational database server and also different DBMS implementations.

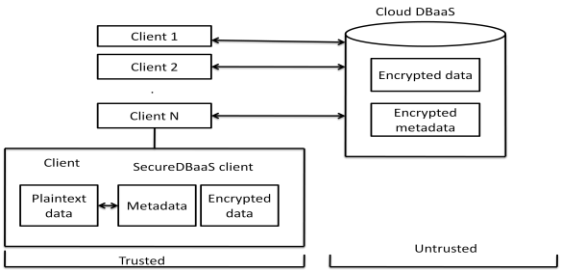

Figure 3.1 SecureDBaas Architecture

3.2 Data and Metadata Management

The tenant's data are saved in a relational database in order for preserve the confidentiality of the stored data and database structure because table and column names may yield information about saved data. Encrypts tenant data are stored through secure tables into the cloud database. For allow execution of SQL statements, each plain text data table transforms into a secure table because the cloud database is not trusted. Secure table is generated by encrypting the name of the corresponding plaintext data table. Table names are encrypts by means of the same encryption algorithm and an encryption key that is known for all the Secure DBaaS clients. Hence, the encrypted names can be computed from the plain data name. On the other hand, column names of secure tables are generated by SecureDBaaS; if different plain data tables have columns with the same name, then the names of the columns of corresponding secure tables is different. This design choice improves confidentiality by preventing an adversarial cloud database from guessing relations among different secure tables through the identification of columns having the same encrypts name.

- Column (COL) is the default confidentiality level, it is used when SQL statements operate on one column; these columns is encrypts through encryption key that is not used by any other column.
- Multicolumn (MCOL) operation is referred by join operator's, foreign keys, and other operation involving two columns; and these columns is encrypts through the same key.

Metadata generated by SecureDBaaS contains all the information that is necessary for manage SQL statements over the encrypted database. Metadata management represents an original idea because SecureDBaaS architecture storing all metadata in the untrusted cloud database.

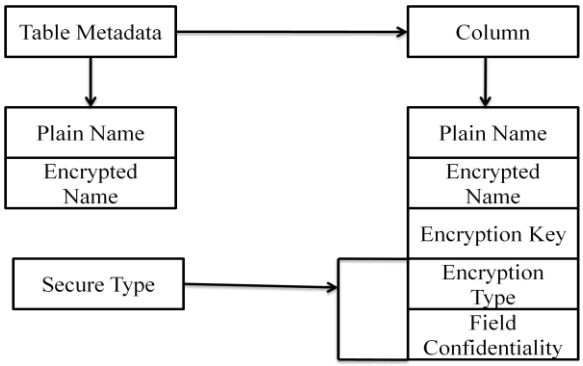

Figure 4.2 Structure of Table Metadata

3.3 Sequential and Concurrent SQL Operations

The SQL operation in SecureDBaaS is considering an initial process in which the cloud database is accessed by one client. Client initially connected for the cloud DBaaS for the authentication process. SecureDBaaS relies on authentication and authorization mechanisms provided by the original DBMS server. After the authentication, a tenant interacts with the cloud database through the SecureDBaaS client. SecureDBaaS identify which tables are involved and retrieve their metadata from the cloud database. Through the master key the metadata is decrypted and their information is used for translate the original plain SQL into a query that operates on the encrypted database.

Concurrent execution of SQL statements is used by multiple independent (and possibly

geographically distributed) clients is one of the most important benefits of SecureDBaaS. It guarantees consistency among encrypts tenant data and encrypts metadata.

3.4 Algorithm

RSA (Rivest-Shamir-Adleman) widely accepted and implemented general-purpose approach for public-key encryption.

Plain text data is encrypts in block having a binary value less than some number. Both sender and receiver must know the value of n. The sender knows the values of e, and only the receiver knows the values of d. This is a public-key encryption algorithm with a public key of PU= {e, n} and a private key of PR={d, n}.

Encryption and decryption are the following form, for some plain text data block M and cipher block C.

$C = M^e \bmod n$

$M = C^d \bmod n = (M^e)^d \bmod n$

For this algorithm for be satisfactory for public-key encryption, the following requirement must be met.

- It is possible for find values of e, d, n such that $M^e \bmod n = M$ for all $M<n$.
- It is relatively easy to calculate $M^e \bmod n$ & $C^d \bmod n$ for all values of $M<n$.
- It is infeasible for determine d given e and n.

Encryption by Bob with Alice's public key:

Plain text : $M<n$

Cipher text : $C= M^e \bmod n$

Decryption by Alice with Alice's Public key:

Cipher text : $M<n$

Plain text : $C= M^e \bmod n$

4 CONCLUSIONS

SecureDBaaS guarantees confidentiality of data, stored in public cloud databases. It does not consider any intermediate proxy, because a proxy contains a single point of failure and a bottleneck that limiting cloud database availability and scalability. SecureDBaaS support concurrent SQL operation (including statements modifying the database structure) on encrypts data issued by heterogeneous and possibly geographically dispersed clients.

5 FUTURE ENHANCEMENTS

In this future work identifies consistency issues related for concurrent execution of queries over encrypts data and for propose viable solutions for different usage contexts, including data modification for the database structure, and data re-encryption. Data, policy, and credential inconsistency problems that can emerge as transactional database systems are deployed for the cloud. Data re-encryption context is for investigated that arise when clients re-encrypt data stored in the cloud database. This occurs when it is required for change encryption keys, or for use a different encryption algorithm for guarantee confidentiality. A re-encryption command that modifies the encryption key that is used for encrypt customer data stored in the table. The client first reads the current metadata (MR [T]) associated with the encrypted customer data for retrieve all the information related for their encryption policy, current encryption keys. Then, it updates the metadata (MW [T]) according for the new encryption policy, by changing the encryption keys. Hence, the client needs for read all the data, for decrypt them with the old encryption keys, for encrypt them with the new encryption keys and for write new data for the encrypt table. Decryption and encryption operation have for be performed locally by a trusted client because the client never exposes data for the untrusted cloud database.

Re-encryption and data read The database may return data that are not accessible by the client, if a data read command is executed concurrently for a re-encryption command. The case in which a data read command requires a set of data whose encryption key is being modified by a concurrent re-encryption command.

Re-encryption and data write Inconsistent data may be written if the data write command and a re-encryption command are executed concurrently. The case in which a data write command stores a set of data whose encryption key is being modified by a concurrent re-encryption command.

REFERENCES

[1] D. Agrawal, A.E. Abbadi, F.Emekci, and A.Metwally,
"Database Management as a Service: Challenges and opportunities," Proc. 25th IEEE Int'l Conf. Data Eng., Mar.-Apr. 2009

[2] M. Armbrust et al., "A View of Cloud Computing," Comm. Of the ACM, vol. 53,no.4, pp. 50-58, 2010.

[3] 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.

[4] A.J. Feldman, W.P. Zeller, M.J. Freedman, and E.W. Felten,"SPORC:Group Collaboration Using Untrusted Cloud Re-Sources," Proc. Ninth USENIX Conf. Operating Systems Design and Implementation, Oct. 2010.

[5] L. Ferretti, M.Colajanni, and M.Marchetti," Suppoting Security and Consistsency for CloudDatabase," Proc. Fourth Int'I Symp.Cyberspace Safety and Security, Dec.2012.

[6] V. Ganapathy, D. Thomas, T. Feder, H. Garcia-Molina, and R. Motwani," Distributing Data for Secure Database Services," Proc.Fourth ACM Int'I Workshop Privacy and Anonymity in the Information Soc., Mar. 2011.

[7] H. Hacigumus, B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over Encrypts Data in the Database-

Service-Provider Model," Proc. ACM SIGMOD Int'I Conf. Management Data, June 2002.
[8] H. Hacigumus, B. Iyer, and S. Mehrotra,"Providing Database as a Service," Proc. 18th IEEE Int'l Conf. Data Eng., Feb.2002.
[9] J. Li,M .Krohn, D. Mazieres, and D. Shasha, "Secure Untrusted Data Reposiforry (SUNDR)," Proc. Sixth USENIX Conf. Operating Systems Design and Implementation, Oct. 2004.
[10] J. Li and E. Omiecinski, "Efficiency and Security Trade-Off in Supporting Range Queries on Encrypts Databases," Proc. 19th Ann. IFIP WG 11.3 Working Conf. Data and Applications Security, Aug. 2005.
[11] P. Mahajan, S. Setsty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish, "Depot: Cloud Sforrage with Minimial Trust," ACM Trans. Computer Systems, vol.29, no.4, article 12, 2011.
[12] 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.
[13] R.A. Popa, C.M.S. Redfield,N. Zeldovich, and H. Balakrishnan,"CryptDB: Protecting Confidentiality with Encrypts Query Processing," Proc.23rd ACM Symp. Operating System Principles, Oct. 2011.
[14] B. White, J. Lepreau, L. Sforller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar," An integrated Experimental Environment for Distributed System and Networks," Proc. Fifth USENIX Conf. Operating System Design and Implementation, Dec. 2002.