Survey on Concurrent Access on Encrypted Cloud Databases

[1]Anupama P L,[2]Anver S R
[1]M.Tech Student,[2]Associate Professor
[1]Department of Computer Science and Engineering,
[1]LBSITW, Poojappura, Trivandrum, Kerala, India

***Abstract*— In earlier days our personal data is stored in our system and need to carry the data with us wherever required. But it is difficult to carry if our dataset is very large. Cloud computing provide a mechanism for storing our data and accessing the data anywhere everywhere and anytime. Even-though the best security standards are implemented in cloud, security of confidential files on external service providers is still an open issue. We can solve this problem by encrypting the data before storing it in the cloud. Here we focus the enhancement of security in cloud databases. In this paper the various architectures for security enhancement that are evaluated in the previous works have been discussed. From this survey we can infer the privacy of user data and can analyze the performance of independent and concurrent access on encrypted data.**

***Index Terms*—Cloud, security, confidentiality, proxy, DBaaS.**

I. INTRODUCTION

Confidentiality violation [1] is one of the most serious concerns while placing user's sensitive data in the remote machine, owned and managed by an untrusted third party. Either maliciously or accidently cloud provider can tamper user's data. Unauthorised disclosure of user's critical data by the service provider is a serious issue. Encryption is the effective method for securing the data before it is stored at the provider.

Database as a Service (DBaaS) [4] guarantee availability and scalability, but it may lead to many concerns about data confidentiality. Integrating encryption with SQL operations is the current approach although it is characterized by many open issues. Earlier proposals based on some trusted intermediate proxy servers limit the availability and scalability of original cloud database services. An alternative architecture that avoids any intermediary component is the existing proposal and achieves availability and scalability comparable to that of unencrypted cloud database services. Moreover, this proposal guarantees data consistency in scenarios in which geographically distributed independent clients concurrently execute SQL queries, and access the database.

The remaining part of this paper is prepared by the following: The literature survey discussed in the 2nd section. The future enhancement about the current study discussed in the 3rd section.

II. LITERATURE SURVEY

With a massive growth in user data in cloud, user requires changing data storage while their roaming, privacy and security for their personal data, better transferring data, better broadband facilities, etc... And cloud computing led to the emergence of cloud databases. It is of vital importance to guarantee users that their data are being correctly stored and maintains, as users no longer have their data locally.

Ryan K L Ko et.al [2] studied the problems and challenges of the trusted cloud, where the unauthorized user can access the entire data without disturbing the actual user. An unauthorized person may do the two things which is accessing the data and putting duplicate data because cloud storage provides a geographical database. It is not a trusted one to store the data of the users.

For this problem they proposed a Trust Cloud framework, to achieve a trusted cloud to the user, to provide a service by making use of detective controls in cloud environment. Detecting process has the accountability access with the cloud. Here user is the responsible person for their data, hence user must tell the accountability with the technical and policy based services. By providing the accountability through user it may solve the problem from the untrusted one. Hence this approach provides privacy, security, accountability and auditability.

Hacigumus H., Iyer B., C.Mehrotra, S proposed a proxy-based architectures in [4]. The most popular solutions for the confidentiality of data outsourced to untrusted database propose a proxy-based architecture, which is represented in Figure 1.

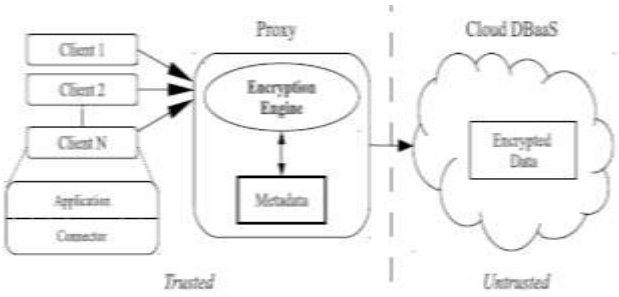

Fig. 1 Proxy Based Architecture

Clients can access the database by issuing unmodified SQL queries to the proxy through a standard database connector. Encryption engine here is the module responsible of applying encryption strategies on customer data. The proxy executes and manages all metadata. The cloud database stores only encrypted customer data. That means the cloud provider access neither plaintext data nor metadata. Metadata are required to decrypt the encrypted customer data.

These proxy-based architectures have some drawbacks. The proxy is a bottleneck and a single-point-of-failure that limits availability, scalability and elasticity of the cloud DBaaS. Proxy cannot be outsourced to the cloud and has to be deployed and maintained locally. So the proxy is treated as a trusted entity. Moreover, proxy based architectures cannot scale trivially by increasing the number of proxies. Such a naive solution would imply the replication of metadata among all the proxies. But this would require synchronization algorithms and protocols to guarantee consistency among all the proxies.

A different approach proposed by Damiani E., De Capitani di Vimercati [6] is shown in Figure 2.

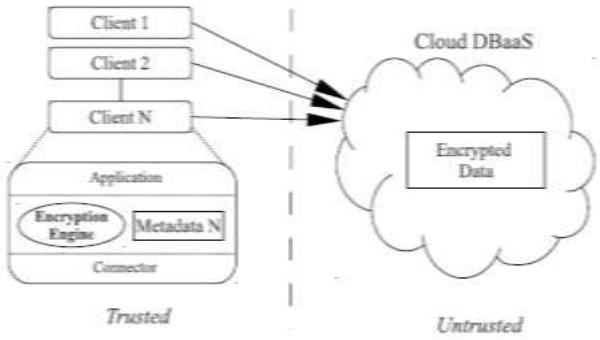

Fig. 2 Proxy less Architecture with metadata in client side

Here, the architecture does not use an intermediate proxy. The metadata are stored in the clients. Since clients connect directly to the cloud database, this architecture achieves availability, scalability and elasticity comparable to those of the original DBaaS. However, each client has its own encryption engine and manages a local copy of metadata.

This solution can represent a sub-case of the proxy-based architecture, in which a different proxy is deployed within each client. As a consequence, a similar architecture for cloud accesses would suffer from the same consistency issues of proxy-based architectures. Guaranteeing metadata consistency in the face of concurrent query execution would require novel synchronization algorithms and protocols among all the clients.

The novel proxy-less architecture proposed by L. Ferretti, M. Colajanni, and M. Marchetti [5] represented in Figure 3.

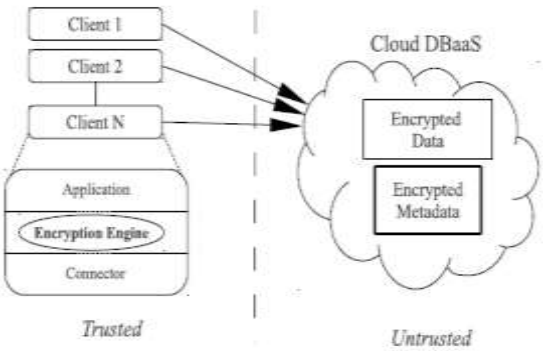

Fig. 3 Proxy less Architecture with metadata in cloud as encrypted

In this architecture the main idea is to move metadata to the cloud database and the encryption engine is executed by each client. Since metadata are not shared among clients there is no need of synchronization mechanisms. Client machines execute a client software component that allows the user to connect and issue queries directly to the cloud DBaaS. This component retrieves

the necessary metadata from the untrusted cloud database through SQL statements and makes them available to the encryption engine. Multiple clients can access the untrusted cloud database independently, with the guarantee of the same level of availability, scalability and elasticity of cloud-based services.

This architecture overcomes the main drawbacks of proxy based solutions; however it introduces new issues with respect to metadata security and data consistency. Previous proposals solve metadata security issues by storing and managing them on trusted components. Since they do not take into account the concurrent management of metadata by multiple components. They do not address any consistency issues related to data and metadata.

Luca Ferretti, Michele Colajanni, and Mirco Marchetti proposed another architecture [3] shown in figure 4.

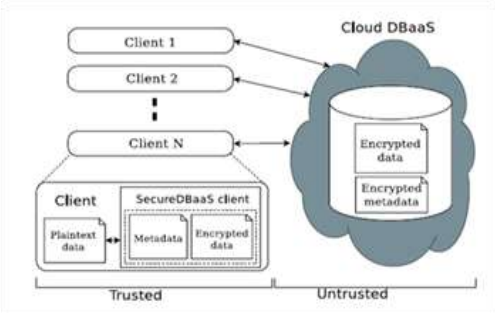

Fig. 4 SecureDBaaS Architecture

This architecture guarantees security of metadata when at rest, in motion and in use by encrypting metadata stored in the cloud. Only clients that know the encryption key can decrypt metadata. Therefore, only these clients can access data that are stored in an encrypted form in the cloud DBaaS. The plaintext database is transformed into an encrypted database by translating each plaintext table into a corresponding encrypted table. Each encrypted table is associated with a set of metadata that contains the information required to encrypt and decrypt the data belonging to that table. Metadata associated with different tables are independent.

Assume that a tenant organization acquires a cloud database service from an untrusted DBaaS provider. The tenant then deploys one or more machines (Client 1 through N) and installs a SecureDBaaS client on each of them. This client allows a user to connect to the cloud DBaaS to administer it, to read and write data, and even to create and modify the database tables after creation. SecureDBaaS is designed to allow multiple and independent clients to connect directly to the untrusted cloud DBaaS without any intermediate server.

It guarantees data confidentiality by allowing a cloud database server to execute concurrent SQL operations over encrypted data. SecureDBaaS stores metadata in the metadata storage table, which is located in the untrusted cloud as a database. It opens two novel issues in terms of efficient data retrieval and data confidentiality. Metadata confidentiality is guaranteed through encryption. 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. This mechanism has the further benefit of allowing clients to access each metadata independently.

Table 1 Summary of Previous Works

Year	***Proposed work***	***Advantage***	***Disadvantage***
2002 [9]	• The cloud computing allows to access information from cloud through an intermediate proxy	• Cloud database store only encrypted customer data • Cloud server neither access plain text nor metadata	• Bottleneck • Single point of failure • Limitations in availability reliability and scalability • Need synchronization algorithm • Not support the concurrent access
2005 [5]	• Client has its own encryption engine and manages metadata	• Does not need an intermediate proxy • Achieve availability scalability & elasticity	• Different proxy within each client • Metadata consistency issue • Need synchronization algorithm
2007 [8]	• The data owner not allows to access confidential data. • Cloud server not allowed decrypting the data.	• Encrypt the data for confidentiality	• The data resources are not physically under the full control of the owner

2012 [5]	• Client has its own encryption engine • Cloud store encrypted metadata	• Does not need an intermediate proxy • Achieve availability scalability & elasticity • Metadata not shared • No synchronization algorithm needed	• Metadata security and data consistency issue • Less confidentiality • Excessive computational complexity
2014 [3][7]	• Cloud store encrypted metadata • Installs a SecureDBaaS client on each client	• Multiple and independent clients can execute queries on the cloud database concurrently • No intermediate proxy • Guarantee data confidentiality • Allowing the execution of SQL queries over encrypted data	• Encryption based on one master key • High computational cost • Response time are affected by cryptographic over head

III. FUTURE ENHANCEMENT

In the current architecture the encryption is based on one master key. Information leakage may occur if the key loss or a client machine compromise. To solve this problem multiple keys are used instead of a master key for encrypting the data. Access control policies can also be incorporated along with this. As a result the problems due to the confidentiality violations can be solved to an extent.

IV. CONCLUSION

All data outsourced to the cloud provider are encrypted through cryptographic algorithms that allow the execution of standard SQL queries on encrypted data. The existing solution allows independent and concurrent access to the cloud databases. It does not rely on a trusted proxy that represents a single point of failure and a system bottleneck, and that also limits the availability and scalability of cloud database services. Internal data leakage is a serious issue in cloud database. Although many techniques exist, confidentiality is still a challenging issue.

V. ACKNOWLEDGMENT

I am thankful to my guide Mr. Anver S R, Associate Professor of Computer Science and Engineering, for his guidance and encouragement for the paper work. I also acknowledge with grateful thanks the authors of the references and other literatures referred to in this survey.

REFERENCES

[1] Kuyoro S.O, Ibikunle F,Awodele O, "Cloud Computing Security Issues and Challenges", International Journal on Computer Networks March 2011, vol 3,issue (5), pages 247-255

[2] Ryan K L Ko, Peter Jagadpramana, Miranda Mowbray, Siani Pearson, Markus Kirchberg , Qianhui Liang , Bu Sung Lee, "TrustCloud: A Framework for Accountability and Trust in Cloud Computing" 2011 IEEE World Congress on Services.

[3] Luca Ferretti, Michele Colajanni, and Mirco Marchetti, "Distributed, Concurrent,and Independent Access to Encrypted Cloud Databases", IEEE transanctions on parallel and distributed systems VOL. 25, NO. 2, FEBRUARY 2014, pp 437-446

[4] Hacigumu¸s, H., Iyer, B., Mehrotra,S.: "Providing database as a service" In: Proceedings of the 18th International Conference on Data Engineering, pp. 29–38 (2002)

[5] L.Ferretti, M. Colajanni, and M. Marchetti, "Supporting Security and consistency for Cloud Database," Proc. Fourth Int'l Symp. Cyberspace Safety and Security, Dec. 2012

[6] Damiani, E., De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S.,Samarati, P.: "Metadata management in outsourced encrypted databases" Secure Data Management (2005) 16–32

[7] Ferretti, Luca, Michele Colajanni, and Mirco Marchetti. "Access control enforcement on query-aware encrypted cloud databases." Cloud Computing Technology and Science (CloudCom), 5th International Conference on Vol. 2. IEEE, 2013.

[8] Bethencourt, J.; Sahai, A.; Waters, B. (2007): "CiphertextPolicy Attribute-Based Encryption". Security and Privacy, IEEE Symposium: pp 321 – 334

[9] R. Buyya, C. S. D.Pratiba Yeo, and S. Venugopa, "Marketoriented cloud computing:Vision, hype, and reality for Delivering it services as computing utilities", Proc. the 10th IEEE International Conference on High Performance Computing and Communications (HPCC 08), 2002, pp. 5-13.