International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 18
December 2016

Cloud database security Using Secure DBaaS for confidentiality Data

N PAVAN KUMAR[1] & N V SAILAJA[2]
[1]M.Tech,Dept of CSE VNR Vignan Jyothi Institute of Engineering And Technology Hyderabad, Mail Id:- pavan99482@gmail.com
[2]Assistant Professor,Dept of CSE VNR Vignan Jyothi Institute of Engineering And Technology Hyderabad, Mail Id:- sailaja__nv@vnrvjiet.in

Abstract

Placing critical data in the hands of a cloud provider should come with the guarantee of security and availability for data at rest, in motion, and in use. Several alternatives exist for storage services, while data confidentiality solutions for the database as a service paradigm are still immature. 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. This requirement imposes clear data management choices: original plain data must be accessible only by trusted parties that do not include cloud providers, intermediaries, and Internet; in any untrusted context, data must be encrypted. Satisfying these goals has different levels of complexity depending on the type of cloud service.

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.

1. INTRODUCTION

Cloud based mostly services have become common as they specialise in high accessibility and quantifiability at low value. whereas providing high accessibility and quantifiability, inserting essential knowledge to cloud poses several security problems. For avoiding these security problems the info area unit keep within the cloud information in associate encrypted format. The encrypted cloud information permits the execution of SQL operations by choosing the cryptography schemes that support SQL operators. Encrypted cloud information permits differing types of accesses like distributed, concurrent, and freelance. one in all the design that supports these 3 types of access is Secure DBaaS.

The Secure DBaaS design supports multiple and freelance shoppers to execute synchronal SQL operations on encrypted knowledge. Knowledge consistency ought to be maintained by investment concurrency management mechanisms utilized in

database management system engines. This survey explains the assorted concurrency management protocols that may be utilized in the encrypted cloud information. The applications want 1SR if knowledge is replicated. Hence, to ensure the deserves of cloud, it\'s essential to produce high quantifiability, accessibility, low value and knowledge with sturdy consistency, that is ready to dynamically adapt to system conditions. Self optimizing one copy serializability (SO-1SR) is that the concurrency management protocol that dynamically optimizes all stages of dealing execution on replicated knowledge within the cloud information. Current DBMSs supported by cloud suppliers permits relaxed consistency guarantees that successively increase the look complexness ofrequests.

The second concurrency controllingprotocol is that the pic isolation (SI) that provides redoubled concurrency in cloud atmosphere in comparison to 1SR. Transactions area unit browse from the pic, reads area unit never blocked attributable to write locks that successively will increase concurrency. SI doesn't permit several of the inconsistencies, SI permits dealing inversions. To avoid dealing inversions sturdy consistency guarantee is needed, i.e. sturdy SI (SSI). The third concurrency management protocol is that the session consistency (SC). Session consistency could be a completely different type of ultimate consistency. The system provides browse your writes consistency within every session. Session consistency is at a coffee value whereas considering interval and dealing value. The value based mostly concurrency management within the cloud is that the C three i.e. cost-based adaptive concurrency management in cloud. C3 dynamically switch between sturdy consistency level and weak consistency level of transactions during a cloud information in line with the value at runtime. it\'s designed on the highest of 1SR and SSI.

2. RELATED WORK

2.1 Existing System:

Original plain data must be accessible only by trusted parties that do not include cloud providers, intermediaries, and Internet; in any untrusted context, data must be encrypted. Satisfying these goals has different levels of complexity depending on the type of cloud service. There are several solutions ensuring confidentiality for the storage as a service paradigm, while guaranteeing confidentiality in the database as a service (DBaaS) paradigm is still an open research area.

2.2 Proposed System:

We propose a novel architecture that integrates cloud database services with data confidentiality and the possibility of executing concurrent operations on encrypted data.This is the first solution supporting geographically distributed clients to connect directly to an encrypted cloud database, and to execute concurrent and independent operations including those modifying the database structure. The proposed architecture

has the further advantage of eliminating intermediate proxies that limit the elasticity, availability, and scalability properties that are intrinsic in cloud-based solutions. Secure DBaaS provides several original features that differentiate it from previous work in the field of security for remote database services.

Advantages of proposed system:

- The proposed architecture does not require modifications to the cloud database, and it is immediately applicable to existing cloud DBaaS, such as the experimented PostgreSQL Plus Cloud Database, Windows Azure and Xeround .
- here are no theoretical and practical limits to extend our solution to other platforms and to include new encryption algorithm.
- It guarantees data confidentiality by allowing a cloud database server to execute concurrent SQL operations (not only read/write, but also modifications to the database structure) over encrypted data.
- It provides the same availability, elasticity, and scalability of the original cloud DBaaS because it does not require any intermediate server.

System Architecture:

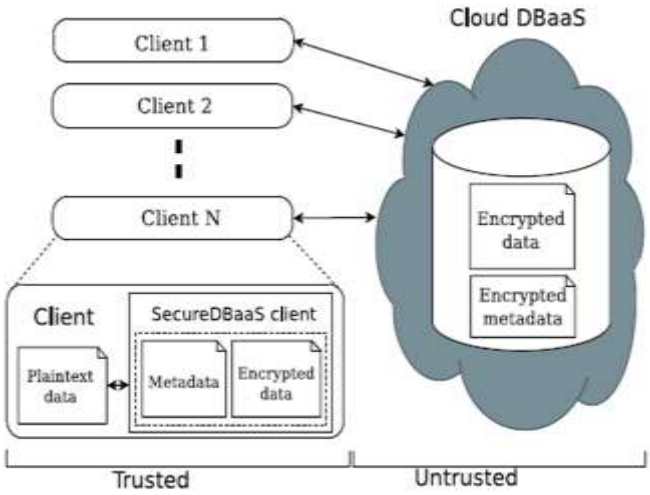

Fig 1: Architecture of Untrusted secure Encrypted Cloud Database

The System has to reach different platforms and to incorporate new encoding algorithmic rule with untrusted cloud information and trustworthy proxy has to be removed. The virtual machine image on cloud uses cloud information severally. this will be achieved by victimization consumer application and cloud knowledgebase with RSA encoding engine for top security and AES encoding for normal data to quick access. The encrypted knowledge is keep into the untrusted cloud information with encrypted data. Clouds don\'t would like any trustworthy proxy for authentication and cloud information is genuine as untrusted.

3. IMPLEMENTATION

3.1 Structure Phase:

We describe how to initialize Secure DBaaS architecture from a cloud database service acquired by a tenant from a cloud provider. We assume that the DBA creates the metadata storage table that at the beginning contains just the

database metadata, and not the table metadata.

The DBA populates the database metadata through the Secure DBaaS client by using randomly generated encryption keys for any combinations of data types and encryption types, and stores them in the metadata storage table after encryption through the master key. Then, the DBA distributes the master key to the legitimate users. User access control policies are administrated by the DBA through some standard data control language as in any unencrypted database. In the following steps, the DBA creates the tables of the encrypted database.

3.2 Meta information Module:

In this module, we develop Meta data. So our system does not require a trusted broker or a trusted proxy because tenant data and metadata stored by the cloud database are always encrypted. In this module, we design such as Tenant data, data structures, and metadata must be encrypted before exiting from the client. The information managed by SecureDBaaS includes plaintext data, encrypted data, metadata, and encrypted metadata. Plaintext data consist of information that a tenant wants to store and process remotely in the cloud DBaaS. SecureDBaaS clients produce also a set of metadata consisting of information required to encrypt and decrypt data as well as other administration information. Even metadata are encrypted and stored in the cloud DBaaS.

3.3 Subsequent SQL Operations:

The first connection of the client with the cloud DBaaS is for authentication purposes. Secure DBaaS relies on standard authentication and authorization mechanisms pro-vided by the original DBMS server. After the authentication, a user interacts with the cloud database through the Secure DBaaS client. Secure DBaaS 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 Secure DBaaS client can issue to the cloud database. Translated operations are then executed by the cloud database over the encrypted tenant data. As there is a one-to-one correspondence between plaintext tables and encrypted tables, it is possible to prevent a trusted database user from accessing or modifying some tenant data by granting limited privileges on some tables. User privileges can be managed directly by the untrusted and encrypted cloud database. The results of the translated query that includes encrypted tenant data and metadata are received by the Secure DBaaS client, decrypted, and delivered to the user. The complexity of the translation process depends on the type of SQL statement.

3.4 Coexisting SQL Operations:

The support to concurrent execution of SQL statements issued by multiple independent (and possibly geographically distributed) clients is one of the most important benefits of Secure DBaaS with respect to state-of-the-art solutions. Our architecture must guarantee consistency among encrypted tenant data and encrypted metadata because corrupted or out-of-date metadata would prevent clients from decoding encrypted tenant data resulting in permanent data losses. A thorough analysis of the possible issues and solutions related to concurrent SQL operations on encrypted tenant data. Here, we remark the importance of distinguishing two classes of statements that are supported by Secure DBaaS: SQL operations not causing modifications to the database structure, such as read, write, and update; operations involving alterations of the database structure through creation, removal, and modification of database tables (data definition layer operators).

4. IMPLEMENTATION ALGORITHM

KeyGen:

Using KeyPair Class system can randomly generate a Public Key (pk) and Private Key (sk) and Key size is 256 bits.

KeyPair kp = kpg.genKeyPair();

PublicKey pk = kp.getPublic();

PrivateKey sk = kp.getPrivate();

Encrypt (pk, message):

This method can take inputs Public Key **pk** and message (Ex: Patient details and Report). Using of a public-key **pk** we can encrypt a **message**, it outputs a cipher text C.

Encrypt (pk, message) = Cipher text C.

Decrypt (sk,C):

This method can take inputs Private Key **sk** and Cipher text **C**. Using of a Private Key **sk** we can decrypt a Cipher text **C**, it outputs a message **m**.

Decrypt (sk,C)= Message m.

5. EXPERIMENTAL WORK

Fig 2: File Upload and Download Module.

Fig 3: Cloud Data.

6. CONCLUSION

The paper proposes a novel solution that guarantees confidentiality of data saved into cloud databases that are untrusted by definition. All data outsourced to the cloud provider are encrypted through RSA and AES cryptographic algorithms that allow the execution of standard SQL queries on encrypted data. This is one of the solution that allows direct, independent and concurrentaccess to the cloud database and that supports even changes to the database structure. It does not rely on a trusted proxy that represents a single point of failure and a system bottleneck, and that limits the availability and scalability of cloud database services. Concurrent read and write operations that do not modify the structure of the encrypted database are supported. There are various encryption decryption techniques available and are having their limitations. The architectural design in this paper uses RSA algorithm which is highly secure for data, but RSA encryption may increase overheads, therefore to decrease the overhead in the network. Very important data are encrypted using RSA and remaining data are encrypted using AES. Specifically, simultaneous read and compose operations that don't adjust the structure of the encoded database cause unimportant overhead. Dynamic situations described by simultaneous adjustments of the database structure are upheld, however at the cost of high computational expenses. These execution effects open the space to future changes are exploring.

6. REFERENCES

[1]L. Ferretti, F. Pierazzi, M. Colajanni, and M. Marchetti, "Security and confidentiality solutions for public cloud database services," in SECURWARE2013, The Seventh International Conference on Emerging Security Information, Systems and Technologies, 2013, pp. 36–42.

[2]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.

[3] H. Hacigumu s, B. Iyer, and S. Mehrotra, Providing Database as a Service, Proc. 18th IEEE Intl Conf. Data Eng., Feb. 2002. [7] C. Gentry, Fully Homomorphic Encryption Using Ideal Lattices, Proc.41st Ann. ACM Symp. Theory of Computing May 2009.

[4] 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

[5] H. Hacigu¨mu¨ s„ B. Iyer, and S. Mehrotra, "Providing Database as a Service," Proc. 18th IEEE Int'l Conf. Data Eng., Feb. 2002.

[6] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices,"Proc. 41st Ann. ACM Symp. Theory of Computing, May, 2009.

[7] H. Hacigu¨mu¨ s¸, B. Iyer, and S. Mehrotra, "Providing Database as a Service," Proc. 18th IEEE Int'l Conf. Data Eng., Feb. 2002.

[8] 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.

[9] Efficient Method to Secure Web applications and Databases against SQL Injection Attacks, Zeinab Raveshi, Sonali R. Idate IJARCSSE Volume 3, Issue 5, May 2013 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering.

[10] Secured Data Storage in Google Cloud S.SABARI VASAN, I.GOLDA SELIA, International Journal of Computer Trends and Technology (IJCTT) - volume4 Issue4 –April 2013.

[11] Handling Confidential Data on the Untrusted Cloud: An Agent-based Approach Ernesto Damiani, Francesco Pagano CLOUD COMPUTING 2010:International Conference on Cloud Computing, GRIDs, and Virtualization.