Distributed, Concurrent, and Independent Access To Encrypted Cloud Databases

Stuti Dixit
M.Tech Scholer
Dr. A.P.J. Abdul Kalam Technical University, U.P, India.
stuti.dixit04@gmail.com

***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. 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. The efficacy of the proposed architecture is evaluated through theoretical analyses and extensive experimental results based on a prototype implementation subject to the TPC-C standard benchmark for different numbers of clients and network latencies.**

Keywords: Cloud Storage, Security, DBaaS, confidentiality

I. INTRODUCTION

The goal of this technology to stored data on cloud in an encrypted form and only authorized person can access data with the help of key which is also called a master key and the possibility of executing concurrent operations on encrypted data. We store the data of owner on cloud. Data Owner is not ensure about his data, so we store his data on cloud by encrypting data. This encryption of data takes place at client side and as secure DBaaS concept, metadata of that data also created. This encrypted data is stored at the cloud along with its encrypted metadata. Privileged user and multifactor access control, data classification and discovery, transparent data encryption, secure configuration management, and data masking is the key of security. In Cloud Databases customers can deploy reliable data security solutions that require no changes to existing applications, it saves time and money. Cloud Databases provide powerful preventive and detective security controls include database activity monitoring and blocking.This project proposes SecureDBaaS. Here all databases are encrypted and stored in the cloud. **It allows multiple and only authorized users can access their own databases concurrently and alone**. Each user use a privet key to encrypt a data and same key is use to decrypt data, So by that it make more secure user data on cloud. There is RSA algorithm plays an important role because to encrypt and decrypt plan text (i.e. user data) by using RSA algorithm and information square measure encrypted exploitation AES technique so overhead on the network will be reduced.SecureDBaaS discard any type of intermediate proxy server, so a user can achieve availability, scalability and elasticity of DBaaS. SecureDBaaS maintain the concurrency as well as confidentiality. Database-as-a-service (**DBaaS**) is very impressive because of two reasons.First, due to it the cost (i.e. economical, energy) incurred by users are much lower when they are paying for a share of a service compare to running everything themselves.

Second, the costs for both software licensing and administrative costs of a well-designed DBaaS will be proportional to actual usage. DBaaS can largely reduce operational costs and perform well.

II. RELATED WORK

SecureDBaaS with it encrypted metadata provides several original features.

- It works without any intermediate server and provides the same availability, elasticity, and scalability of the original cloud DBaaS. Response times are affected by cryptographic overheads that for most SQL operations are masked by network latencies.
- It guarantees security of data and its confidentiality by allowing a cloud database server to execute concurrent SQL operations over encrypted data. These SQL operations allow not only read/write, but also modifications to the database structure.
- This architecture does not require a trusted broker or a trusted proxy because tenant data and metadata stored by the cloud database are always encrypted.
- It is compatible with the most popular relational database servers, and it is applicable to different DBMS implementations because all adopted solutions are database doubter.
- Multiple clients, possibly geographically distributed, can access concurrently and independently to a cloud database service.

SecureDBaaS relates more closely to works using encryption to protect data managed by untrusted databases. In such a case, a main issue to address is that cryptographic techniques cannot be applied to standard DBaaS because DBMS can only execute SQL operations over plaintext data. Some DBMS engines offer the possibility of encrypting data at the file system level through the so-called Transparent Data Encryption feature. This feature makes it possible to build a trusted DBMS over untrusted storage. However, the DBMS is trusted and decrypts data before their use. Other solutions, allow the execution of operations over encrypted data.These approaches preserve data confidentiality in scenarios where the DBMS is not trusted; however, they require a modified DBMS engine and are not compatible with DBMS software (both commercial and open source) used by cloud provider.

The cloud storage system is a kind of distributed storage system that consists of the many freelance storage servers. Knowledge hardiness may be a major demand for storage systems.

C. Gentry [4] proposed a fully homomorphic encryption scheme -- i.e., a scheme that allows one to evaluate circuits over encrypted data without being able to decrypt.

III. ARCHITECTURE DESIGN

SecureDBaaS is designed to allow multiple and independent clients to connect directly to the untrusted cloud DBaaS without any intermediate server. Figure 1 describes the overall architecture. We 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.

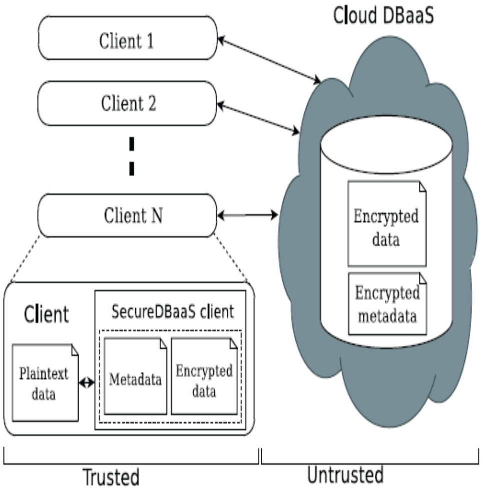

Figure 1. SecureDBaaS architecture

A. Secure DBaaS: A SecureDBaaS allow multiple and independent clients to connect directly to untrusted cloud. Assume an organization obtain database as a service from untrusted cloud provider. A secure DBaaS manage database related information, such as encrypted database and encrypted metadata. An encryption of database in cloud database prevents the violation of confidentiality by untrusted cloud provider.

SecureDBaaS stores a metadata in cloud and allow SecureDBaaS client to retrieve necessary metadata, which is required to extract data from cloud database. Assume that data stored in cloud database is relational database. Encrypted data is stored through secure table in cloud database. Encryption operation is done at SecureDBaas client.

B. Metadata Storage Table: A SecureDBaaS generate a metadata which include all the information need to access the data from encrypted database. Secure DBaaS stores metadata in metadata storage table that is placed in cloud database. This is flexible approach but come with two issues efficiency of data access and confidentiality.

To provide efficiency of data access Secure DBaaS use two metadata.

1. **Database Metadata:** This metadata associated to entire database. This metadata has a only one example for each database in a cloud.
2. **Table metadata:** This is related with secure table. That is this metastable include all the information about encryption and decryption of secure table.

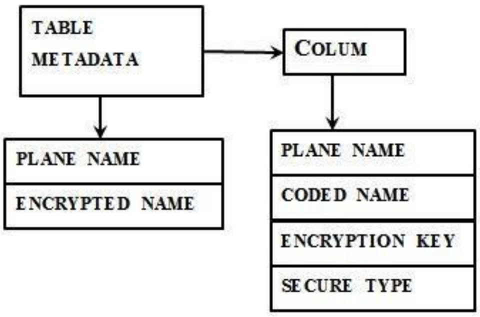

Figure 2: Table Metadata Structure

Database and table metadata are encrypted by using the same encryption key before it has been stored at cloud database. This encryption key is called a master key. Only trusted clients know this key. If you want to decrypt the metadata, it requires that same key (i.e. master key at cloud database. Each client can recover metadata through an associated ID.The ID is work as primary key of metadata table. Through this mechanism each clients are allowed to access metadata independently, which is an important feature in concurrent environments. In addition, SecureDBaaS clients can use caching policies to reduce the bandwidth overhead.

C. Secure DBaaS Client: Suppose that a connection is a resident of cloud and gets cloud database administration from an untrusted DBaaS cloud administration supplier. The resident then arranges one or more machines and introduces a SecureDBaaS customer on each of them. Suppose this customer is a client to extract with the cloud database to get an administration. The data oversaw by SecureDBaaS customer incorporates plaintext information, encoded information, metadata, and scrambled metadata. Plaintext information is the data about information place absent and handle remotely in cloud database. A confined DBaaS customer encoded the information before it has been place absent in cloud remotely. It delivers an arrangement of metadata that include data needed to encode and decoded the information. After store of information it will repair the metadata also in cloud metadata stockpiling table. It recovers the grateful metadata from metadata stockpiling table to get to the cloud database.

A. Advantages of Proposed System:

- There are no theoretical and practical limits to extend our solution to other platforms and to contain 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) over encrypted data.
- The planned architecture does not require modifications to the cloud database, and it is directly applicable to existing cloud DBaaS, such as the experimented PostgreSQL Plus Cloud Database, Windows Azure and Xeround.
- It provides the same availability, flexibility, and scalability of the original cloud DBaaS because it does not require any intermediate server.

IV. FLOWCHART

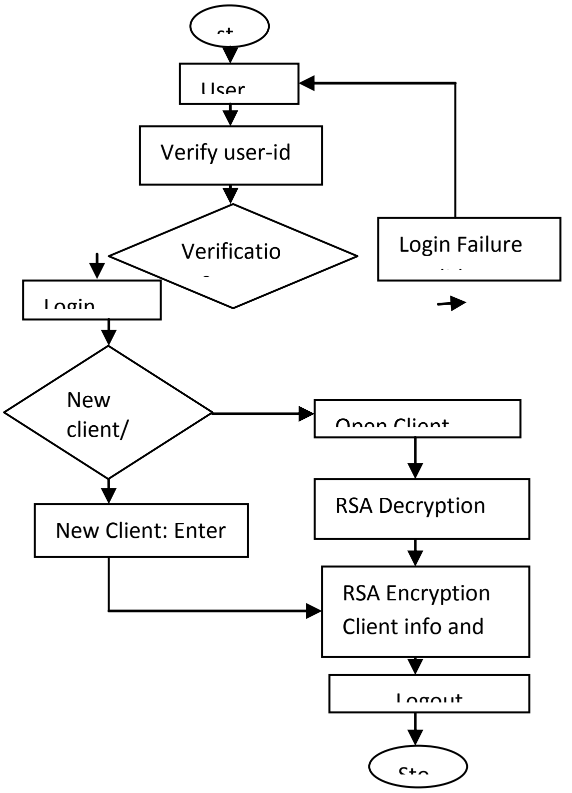

Figure 3: RSA encryption/decryption to access data

-

B. Methodology

A) Setup Phase: So here explain how to initialize a secure DBaaS architecture from a cloud database service acquired by a tenant from a cloud provider. Here suppose that the DBA creates the metadata storage table that at the beginning contains immediately the database metadata, and not the table metadata.

A) Metadata Module: In this module, we develop Meta data. So our system does not require a trusted proxy because tenant data and metadata stored by the cloud database which are always encrypted. In this module, we design such as Tenant data, data structures, and metadata must be encrypted before exiting from the client.

C) Sequential SQL Operations: The first connection between the client and the cloud DBaaS is for authentication purposes. Secure DBaaS relies on ordinary authentication and authorization mechanisms from the original DBMS server. After the authentication, a user interacts with the cloud database through the Secure DBaaS client.

D) Concurrent SQL Operations: The support to parallel 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

V. DATA FLOW DIAGRAM

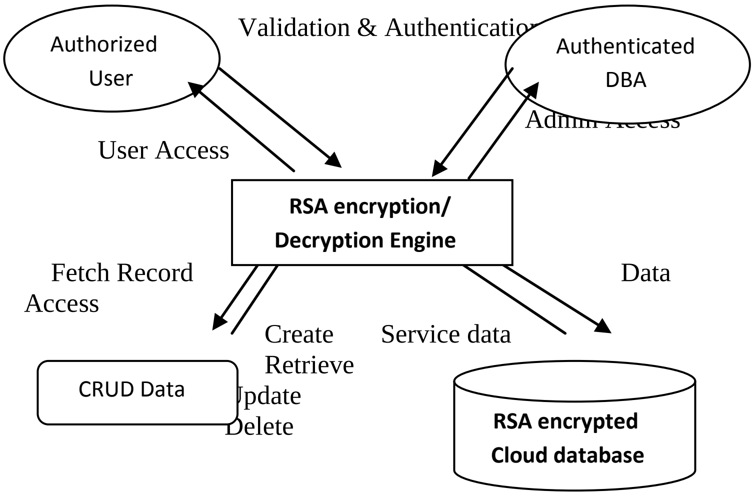

Figure 4: Data flow diagram for accessing data from cloud database

VI. USECASE DIAGRAM

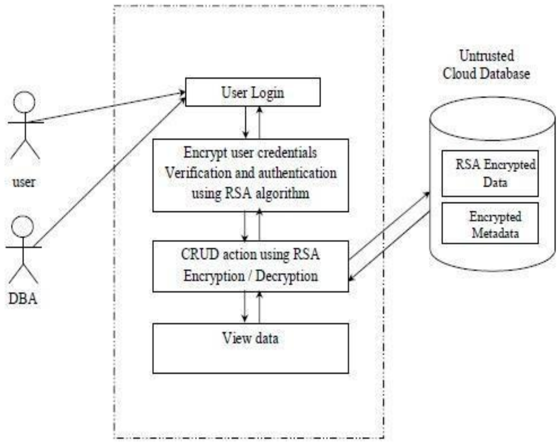
Untrusted
Cloud Database
user
DBA
User Login
Encrypt user credentials Verification and authentication using RSA algorithm
CRUD action using RSA Encryption / Decryption
View data
RSA Encrypted Data
Encrypted Metadata

Figure 5: Usecase diagram to access data

VII. CONCLUSIONS

We plan a modern architecture that guarantees confidentiality of data stored in public cloud databases. Unlike state-of-the-art approaches, our solution does not rely on an intermediate proxy that we consider a single point of failure and a bottleneck limiting availability and scalability of typical cloud database services. A large part of the research includes solutions to support concurrent SQL operations (including statements modifying the database structure) on encrypted data issued by heterogeneous and possibly geographically dispersed clients. This technology provide you to store data whit full security and easy to store. To store data on cloud only an ID is require for encrypt and decrypt data which is called master key, through this it possible to full security of your data.

REFERENCE

[1] Luca Ferretti, Michele Colajanni, and Mirco Marchetti, “Distributed, Concurrent, and Independent Access to Encrypted Cloud Databases”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 2, FEBRUARY 2014.

[2] https://en.wikipedia.org

[3] https://en.wikipedia.org/wiki/Cloud_database[4] http//: www.cloudsolutions.com.

[4] http://www.schandpublishing.com