ISSN No: - 2456 - 2165

Preserving Data Confidentiality and Consistency at Encrypted Cloud Databases

Indhupriya D
ME-Student
P.S.V college of Engineering and technology
Krishnagiri, India.

B.Sakthivel
HOD/CSE
P.S.V College of engineering and technology
Krishnagiri, india.

***Abstract*—Cloud Computing provide computation, software, data access, and storage resources without requiring cloud users to know the location and other details of the computing infrastructure. It is a technology that uses the internet and central remote servers to maintain data and applications. The main aim of this paper is to integrate cloud database service with data confidentiality and the possibility of executing concurrent operations on encrypted data i.e. to provide Secure Database as a Service (DBaaS). Here geographically distributed clients to connect directly to an encrypted cloud database, and to execute concurrent and independent SQL operations including those modifying the database structure. Data confidentiality is achieved by encrypting the data using Advance Encryption Standard (AES) algorithm. 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.**

Keywords: *Cloud ,DBaaS, Confidentiality,Encryption.*

I. INTRODUCTION

Database as a service provides its customers seamless mechanisms to create, store, and access their databases at the host site. In the Database-As-a-Service (DAS) model, clients store their database contents at servers belonging to potentially untrusted service providers. To maintain data confidentiality, clients need to outsource their data to servers in encrypted form. At the same time, clients must still be able to execute queries over encrypted data. [2], [3]. 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.

SecureDBaaS is the 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 [1]. The SecureDBaaS architecture is tailored to cloud platforms and does not introduce any intermediary proxy or server between the client and the cloud provider. Eliminating any trusted intermediate server allows SecureDBaaS to achieve the same availability, reliability, and elasticity levels of a cloud DBaaS. To achieve SecureDBaaS encryption is the only effective solutions. In the Database-Service provider model, user's data resides on the premises of Database-Service provider. Most corporation view data as a very valuable asset. The service provider would need to provide sufficient security measures to guard data privacy i.e. to obtain SecureDBaaS [4]. Encryption of the storage data is the straight forward solution to achieve data confidentiality were the users data are encrypted and stored in the cloud database in a secure form. So that the confidentiality of the data stored in the cloud can be preserved from the hacker who theft the customer data and from the cloud provider too.

Executing concurrent SQL operations such as modification, alter, delete etc., on encrypted data stored in the cloud is an another scenario: a native context characterized by a single client, and realistic contexts where the database services are assessed by concurrent client. 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 SecureDBaaS. 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. Secure DBaaS allows clients to issue concurrent SQL commands to the encrypted cloud database without introducing any new consistency issues with respect to unencrypted databases. After metadata retrieval, a plaintext SQL command is translated into one SQL command operating on encrypted tenant data. As metadata do not change, a client can read them once and cache them for further uses, thus improving performance.

The overall conclusions of this paper is to demonstrate the applicability of preserving confidentiality to cloud database services in terms of feasibility and performance. The remaining part of the paper discuss about: section 2 gives comparison of existing systems with our proposed system, section 3 explain about the architecture and the operation of our system, section 4 deals with the conclusion and future works of the paper.

ISSN No: - 2456 - 2165

II. RELATIVE WORK

Luca Ferretti, Michele Colajanni, and Mirco Marchetti proposed a "Distributed, Concurrent and Independent Access to Encrypted Cloud DataBase". This paper guarantees data confidentiality by allowing a cloud database server to execute concurrent SQL. The architecture design was motivated by a three goals: to allow multiple, independent, and geographically distributed clients to execute concurrent operations on encrypted data, including SQL statements that modify the database structure; to preserve data confidentiality and consistency at the client and cloud level;

Jum Li, Edward R. Omiecinski proposed an "Efficiency and Security Trade-Off in Supporting Range Queries on Encrypted Databases". This paper discusses concerned about protecting sensitive information of data and queries from adversaries in the DAS model. Data and queries need to de encrypted, while the database service provider should be able to efficiently answer queries based on encrypted data. [4]

Luca Ferretti, Michele Colajanni, and Mirco Marchetti proposed a "Supporting Security and Consistency for Cloud Database". This 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 cryptographic algorithms that allow the execution of standard SQL queries on encrypted data. [6]

Divyakant Agrawal, Amr El Abbadi, and Fatih Emekci Ahmed Metwally proposed a "Database Management as a Service: Challenges and Opportunities". This paper presents scalable secure and privacy preserving algorithms for data outsourcing. Instead of encryption, they use distribution on multiple data provider sites and information theoretically proven secret-sharing algorithms as the basis for privacy preserving outsourcing. The research is timely due to the ever increasing private and public data being generated. [5]

PAPER NAME	ISSUES
Distributed, Concurrent and Independent Access to Encrypted Cloud DataBases	Fully homomorphic encryption is not applicable
Efficiency and Security Trade-Off in Supporting Rage Queries on Encrypted Databases	Expensive protocol between client and database provider is Needed.
Supporting Security and Consistency for Cloud Database	Intermediate proxies are required for certain Architecture.
Database Management as a Service: Challenges and Opportunities	Data confidentiality is not preserved

III. PROPOSED MODEL

A. System Model

The architecture proposed in this paper guarantees data confidentiality by encrypting the user data using advanced encryption standard algorithm together with the ability to execute concurrent SQL operations [10].

SecureDBaaS is designed to allow multiple and independent clients to connect directly to the untrusted cloudDBaaS without any intermediate server. Here N number of clients can assess the database provided by the client. User can create tables in database provided by the cloud. Each plaintext table is transformed into a secure table by encrypting the data. The encrypted data and metadata are stored in the cloud database. Now the user is a Secure DBaaS client.

ISSN No: - 2456 - 2165

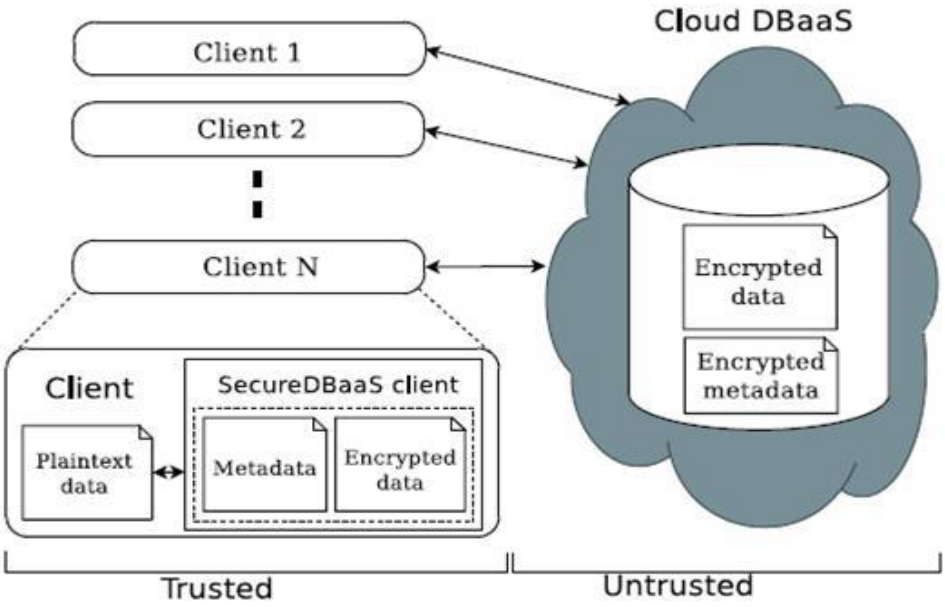

Fig 1. System Architecture for Preserving Confidentiality of DataBases as a Service (DBaaS)

The information managed by SecureDBaaS includes plaintext data, encrypted data, metadata, and encrypted metadata. Metadata generated by Secure DBaaS contain all the information that is necessary to manage SQL statements over the encrypted database in a way transparent to the user [10].

SecureDBaaS clients can retrieve the necessary metadata from the untrusted database through SQL statements, so that multiple instances of the SecureDBaaS client can access to the untrusted cloud database independently with the guarantee of the same availability and scalability properties of typical cloud DBaaS.

This paper is further splitter in to three modules

- Registration
- Data management
- SQL operations & File storage.

B. AES Algorithm

AES (Advanced Encryption Standard) is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits. It uses 10, 12, or 14 rounds. The key size, which can be 128, 192, or 256 bits, depends on the number of rounds. For example, if the key size used is 128 then the number of rounds is 10 whereas it is 12 and 14 for 192 and 256 bits respectively. At present the most common key size likely to be used is the 128 bit key [7].To provide security, AES uses four types of transformations: substitution, permutation, mixing, and keyadding. High level description for AES algorithm:

- Given a plaintext X, initialize state to be X and perform an operation Addroundkey, which x-ors the Round key with state.
- For each of the first r − 1 rounds, perform a substitution operation called SubBytes on state using an S-box; perform a permutation Shift Rows on state; perform an operation Mix Columns on state; and perform AddRoundKey.
- Perform SubBytes; perform ShiftRows; and perform AddRoundKey.
- Define the cipher text Y to be state.

C. Data Management

Users have an initial level Registration Process at the web end. The users provide their own personal information for this process. The server in turn stores the information in its database. User account details will be maintained in Bank. This is to buy Database as a Service through Internet Banking. User can access database services provided by the cloud. User can choose database and specify time period of the database. Cloud provides cost of the database service. After successful

ISSN No: - 2456 - 2165

payment of bank transaction, user can access database service from cloud.

We have to preserve the confidentiality of the stored data and even of the database structure because table and column names may yield information about saved data [12]. Encrypted tenant data are stored through secure tables into the cloud database. To allow transparent execution of SQL statements, each plaintext table is transformed into a secure table because the cloud database is untrusted. The name of a secure table is generated by encrypting the name of the corresponding plaintext table. Table names are encrypted by means of the same encryption algorithm and an encryption key that is known to all the SecureDBaaS clients.

Metadata generated by Secure DBaaS contain all the information that is necessary to manage SQL statements over the encrypted database in a way transparent to the user [8].User can create tables in database provided by the cloud. Each plaintext table is transformed into a secure table. Now the user is a Secure DBaaS client. After successful creation of the table, now user can insert data into a table. Table data will be encrypted by means of the Advanced Encryption Standard (AES) algorithm.

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 SecureDBaaS stores metadata in the metadata storage table that is located in the untrusted cloud as the database. This is an original choice that augments 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.

This mechanism has the further benefit of allowing clients to access each metadata independently, which is an important feature in concurrent environments.

C. SQL Operations & File Storage

User performs SQL operations in Database. User can alter table data in cloud database. Modification of Table data will be encoded and encrypted in the Cloud. User doesn't need a table in database; he/she can perform delete operation. Table deleted by the user will be removed from cloud database. 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 SecureDBaaS 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 [9]. Secure DBaaS allows clients to issue concurrent SQL commands to the encrypted cloud database without introducing any new consistency issues with respect to unencrypted databases [8]. After metadata retrieval, a plaintext SQL command is translated into one SQL command operating on encrypted tenant data. As metadata do not change, a client can read them once and cache them for further uses, thus improving performance. User store files with limited storage in cloud. If the user needs extra storage, user can request cloud to provide storage [11]. User can specify space requirements and he/she buy storage space. After successful bank transaction, requested storage will be allocated to the user. Now user can upload data to cloud. At the time of uploading, data will be encoded using Base64 algorithm and encrypted using Data Encryption Standard (DES) algorithm. Cloud users data is plotted as a Graph. Graph shows each user storage space in cloud.

IV. CONCLUSION AND FUTURE WORK

This paper proposes a 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 Advanced Encryption Standard (AES) algorithms and allow the execution of standard SQL queries on encrypted data stored in the cloud. This paper allows direct, independent and concurrent access 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.

In particular, concurrent read and write operations that do not modify the structure of the encrypted database cause negligible overhead. Dynamic scenarios characterized by (possibly) concurrent modifications of the database structure are supported, but at the price of high computational costs. These performance results open the space to future improvements that we are investigating.

V. ACKNOWLEDGMENT

The authors would like to thank Prof. Lorenzo Alvisi of the University of Texas at Austin for his constructive comments on preliminary versions of this paper.

REFERENCES

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

[3]. 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.
[4]. J. Li and E. Omiecinski, “Efficiency and Security Trade-Off in Supporting Range Queries on Encrypted Databases,” Proc. 19th Ann. IFIP WG 11.3 Working Conf. Data and Applications Security, Aug. 2005.
[5]. 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.
[6]. 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.
[7]. The AES Algorithm from en.wikipedia.org/wiki/AES.
[8]. H. Berenson, P. Bernstein, J. Gray, J. Melton, E. O’Neil, and P. O’Neil, “A Critique of Ansi Sql Isolation Levels,” Proc. ACM SIGMOD, June 1995.
[9]. 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.
[10]. H. Hacigu¨mu¨ s¸, B. Iyer, C. Li, and S. Mehrotra, “Executing SQL over Encrypted Data in the Database-Service-Provider Model,” Proc. ACM SIGMOD Int’l Conf. Management Data, June 2002.
[11]. M. Armbrust et al., “A View of Cloud Computing,” Comm. of the ACM, vol. 53, no. 4, pp. 50-58, 2010.
[12]. W. Jansen and T. Grance, “Guidelines on Security and Privacy in Public Cloud Computing,” Technical Report Special Publication 800-144, NIST, 2011.