# DISTRIBUTED, CONCURRENT, AND INDEPENDENT ACCESS TO ENCRYPTED CLOUD DATABASES

Shaik Roshan[1], Ramakrishna Reddy[2]
[1]*M.Tech student, CSE, Kottam college of Engineering,*
[2]*Assistant Professor, CSE , Kottam college of Engineering,*
*Chinnatekuru(V),Kurnool,Andhra Pradesh, India*

***Abstract-*** **Cloud computing can address this problem by providing data storage mechanism to access the data at anywhere. This is one of the storage device used to access their data at any where through networks which is called cloud provider. For this service user worry about the security and privacy issue under this cloud computing for their personal data. 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.**

***Index Terms-*** **Cloud, security, confidentiality, SecureDBaaS, database**

## I. INTRODUCTION

Today user may spend lots of time with a computer to collect lots of data over the network and store it where it as portable to the user. During the roaming time user may need the data from their PC (Personal Computer) it is very difficult to take it as a portable one with large datasets. So they may problem occurred while their roaming time. For this reason, storing an enough data in the network can solve this problem. Cloud storage is used to avoid this problem. Cloud storage refers to storing a large amount of data which in the form of pay-per-use scheme which is referred to cloud computing. It is used to off-site storage scheme maintained by a third party, i.e. cloud provider [1]. It is the most popular one to store the data in geographical environment with infinite computing resources and access the data where, the user need without worry about the data loss. Hence it provides greater availability, scalability, and reliability to the users. This survey shows the features are provided by the cloud provider as a service of Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Database as a Service (DBaaS).

Cloud services: (i) Software as a Service (SaaS: This provides a service to the user by offering different software to the different user over internet. A distinct instance of service which runs in the cloud, here one or more user can utilize the service. Here no charges are detected from the user for the service or software license. In some cases, charges may detect for the maintenance of the service [2].

(ii) Platform as a Service (PaaS): This provides a service to the user for the layer of software platform. It provides a storage mechanism for the various applications and consumptions. User can have an independence to build their personal applications that provide infrastructure for the user. It offers predefined components of combined OS and the application server, e.g. LAMP platforms [2].

(iii) Infrastructure as a Service (IaaS): This provides a service to the user for the basic storage and processor infrastructure as a service over the network. It provides a service to the computer infrastructure for the servers, network administrators, data center, etc. to handle the workload of this service through IaaS. For this service user need to pay charges, when they use this service over the network. In this mechanism cloud computing provide a service over the internet, hardware and software in data centers as a service. The data center of hardware and software is called as Cloud.

(iv) Database as a Service (DBaaS): This provides a service to the user for their data. It does not require modifications to the database, hence it is controlled by the cloud provider. Cloud provider to manage and direct the database and aim to avail the instant services to the data users. Here organizations pay for the database service for getting the service from the service provider. For the organization with fewer amounts of resources limited hardware and time-bound projects, DBaas solve this problem; it is in the bases of pay-per-usage manner. DBaaS is a successful paradigm where the data and the storage devices are located in cloud infrastructure and use the data in any where by the user [3].

Extreme superficial facts sire be reachable solitarily by upright parties become absent-minded complete quite a distance add up internet, intermediaries and cloud providers; other than above parties data must be encrypted. Another levels of complicatedness exists in choice these goals depending on cloud facilitate type.Up are additional solutions ensuring monasticism for the storage as a promote but covertness cannot be set in the database as a service (DBaaS) prototype and is soothe an open research area.

**Disadvantages:** Cannot give out despotic encryption deceit because of their excessive computational complexity.

## II. RELATED WORKS

Ryan K L Ko et.al [4] 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.

Muhammad Rizwan Asghar et.al [5] discusses the problems of enforcing security policies in cloud environment. With the high growth of data in cloud they where problem arises due to untrused person access of the data. To ensure the security is immature, they didn't ensure for the safe data in cloud environments. Security problem is a great issue; here we enforce the security for the owner's data. Providing high security they may high expensive for the users.

L Ferretti et al [6] studied the problem of data leakage of the legitimate user in cloud environment by the cloud provider; they didn't give better security to the user for their personal data or internal data. Main problem arise because of no encrypted data were found, and also it provide the security for the frond-end database only and not controlled the backend database, so the malicious attackers may gain the data access to the outsourced data.

A.J. Feldman et al [7] find the issues of leaking data in server side and study the risk of privacy problem. Due to centralization of information attackers may easily hack the data through cloud computing. Access control under this cloud provider is not a strong one; user data may loss at any time because all a user is not always in the online to check the status of the data. So it is easy to hack the data in anytime by the attackers and also they may modify their data at any time so it is risky one.

Ferretti, Luca, et al [8] study two problems; which are (i) Bandwidth problem due to increase no .of database size because of encrypted data. (ii) Re-encrypted data access, the performance of re-encrypted data may take a lot of time for processing the data when it has a large number of rows. The response time for processing the data may take a lot of time to decrypt the data and the data where not a secure and also not confidential one.

Different approaches guarantee some confidentiality (e.g., [9], [10]) by distributing data among different providers and by taking advantage of secret sharing [14].

A step forward is proposed in [12], that makes it possible to execute range queries on data and to be robust against collusive providers. Secure DBaaS differs from these solutions as it does not require the use of multiple cloud providers, and makes use of SQL-aware encryption algorithms to support the execution of most common SQL operations on encrypted data.

Some DBMS engines offer the possibility of encrypting data at the filesystem level through the so-called Transparent Data Encryption feature [13], [14]. This feature makes it possible to build a trusted DBMS over untrusted storage.

However, the DBMS is trusted and decrypts data before their use. Hence, this approach is not applicable to the DBaaS context considered by

SecureDBaas, because we assume that the cloud provider is untrusted.

Other solutions, such as [15], 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 providers. On the other hand, SecureDBaaS is compatible with standard DBMS engines, and allows tenants to build secure cloud databases by leveraging cloud DBaaS services already available.

## III. PROPOSED ARCHITECTURE DESIGN

SecureDBaaS is designed to allow multiple and independent clients to connect directly to the untrusted cloud DBaaS without any intermediate server. Fig. 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.

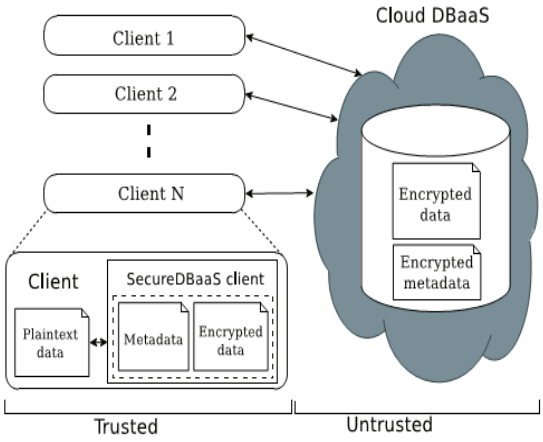



Fig. 1. SecureDBaaS architecture.

The system mainly focuses on following-

- Cloud database
- Metadata Management
- Encryption algorithm

Cloud database: We assume that tenant data are saved in a relational database. 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. We distinguish the strategies for encrypting the database structures and the tenant data.

Metadata Management: Metadata generated by SecureDBaaS contain all the information that is necessary to manage SQL statements over the encrypted database in a way transparent to the user. Metadata management strategies represent an original idea because SecureDBaaS is the first architecture storing all metadata in the untrusted cloud database together with the encrypted tenant data.

Encryption algorithm: Choosing the encryption algorithms used to encrypt and decrypt all the data stored in the database table.

1. Creation of database-In this module client creates its database and store data in the form or columns and rows. After creation of Database the client also creates its metadata which will help for later communication instead of whole database.

2. Selection of encryption and decryption algorithm - In this module we select the encryption algorithm to encrypt and decrypt the created database and its metadata. It will provide security to whole data of client which is to be uploaded on the cloud.

3. Cloud Database-Cloud Database is the service provider, which provides services to the tenants. All the encrypted data from data owner is uploaded on cloud which provides concurrent access to cloud DB to the geographically deployed clients. Cloud DB contains encrypted database and its encrypted metadata.

4. Application -This module contains the application of system to the cloud. How we will Apply these all on cloud this module explains it. We use master key to access cloud data after data is uploaded on data. First we will get encrypted data if our key is correct then by using random decryption keys we will get the final output in the form of plaintext data. Input is taken from user in the form of sql querry.

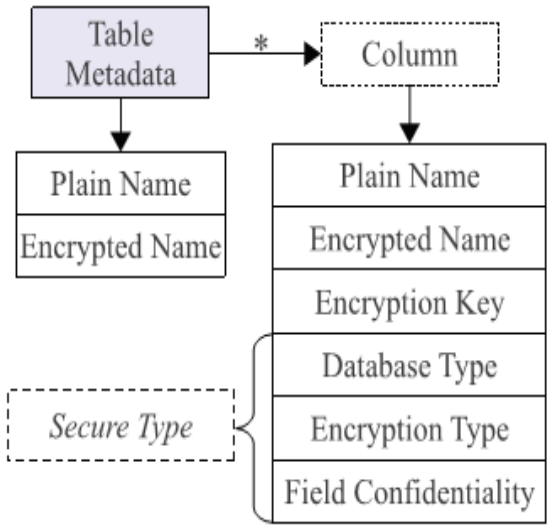
Firstly client will create Database then, will enter rows into the database. After that the metadata of database is created. Then selected encryption algorithm is applied to the database and its metadata. final output gives the encrypted data with all its information and key used.

### A. IMPLEMENTATION

**Data Management:** Cloud database acts as service provider for tenants. The cloud is created first for the system. All information or data store in the relational

database. So for creating tables and column we have to access it with SQL query only.

**Metadata management:** Metadata generated by SecureDBaaS contain all the information that is necessary to manage SQL statements over the encrypted database in a way transparent to the user. Metadata management strategies represent an original idea because SecureDBaaS is the first architecture storing all metadata in the untrusted cloud database together with the encrypted tenant data.




The structure of a table metadata is represented in Fig. 2.

SecureDBaaS uses two types of metadata.

• Database metadata are related to the whole database. There is only one instance of this metadata type for each database.

• Table metadata are associated with one secure table. Each table metadata contains all information that is necessary to encrypt and decrypt data of the associated secure table.

This design choice makes it possible to identify which metadata type is required to execute any SQL statement so that a SecureDBaaS client needs to fetch only the metadata related to the secure table/s that is/are involved in the SQL statement.

Table metadata contain the name of the related secure table and the unencrypted name of the related plaintext table. Moreover, table metadata include column metadata for each column of the related secure table. Each column metadata contain the following information.

• Plain name: the name of the corresponding column of the plaintext table.

• Coded name: the name of the column of the secure table. This is the only information that links a column to the corresponding plaintext column because column names of secure tables are randomly generated.

• Secure type: the secure type of the column. This allows a SecureDBaaS client to be informed about the data type and the encryption policies associated with a column. • Encryption key: the key used to encrypt and decrypt all the data stored in the column.

*Metadata Storage Table*

| *ID* | *Encrypted Metadata* | *Control Structure* |
|---|---|---|
| MAC('.'+Db) | Enc(Db metadata) | MAC(Db metadata) |
| MAC(T1) | Enc(T1 metadata) | MAC(T1 metadata) |
| MAC(T2) | Enc(T2 metadata) | MAC(T2 metadata) |
| | | |

Fig.4. Organization of database metadata and table metadata in the metadata storage table.

SecureDBaaS stores metadata in the metadata storage table that is located in the untrusted cloud as the database. This is an original choice that augmnts flexibility, but opens two novel issues in terms ofefficient 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. The structure of the metadata storage table is shown in Fig.4 This table uses one row for the database metadata, and one row for each table metadata.

## IV. SIMULATION RESULTS

Fig. 5


Fig. 6


Fig. 7


Fig.8

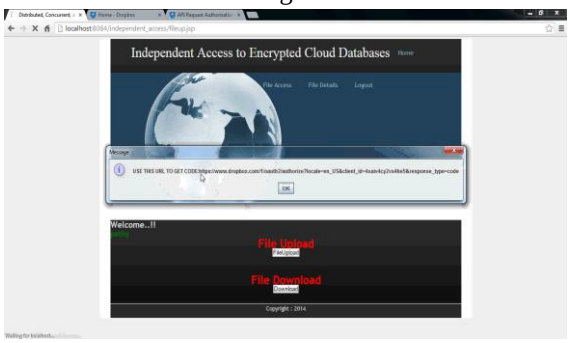



Fig. 9

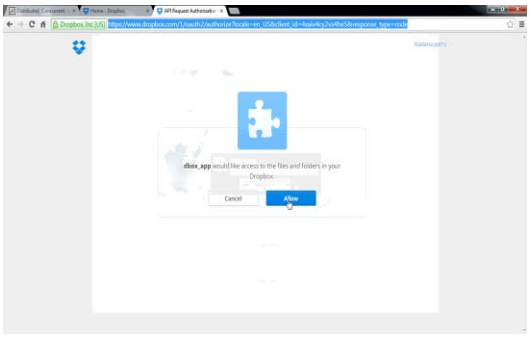


Fig. 10


Fig. 11 Secret key generation

Fig. 12 Files are encrypted format in cloud

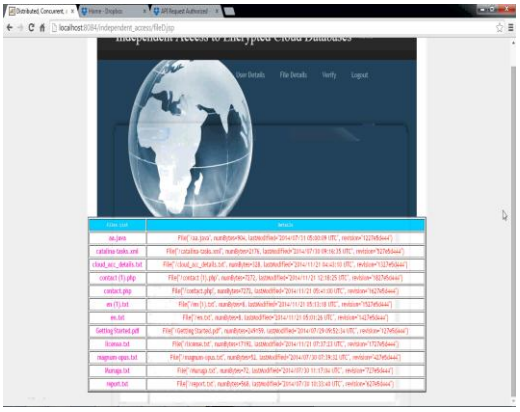


Fig.13 Admin view file details

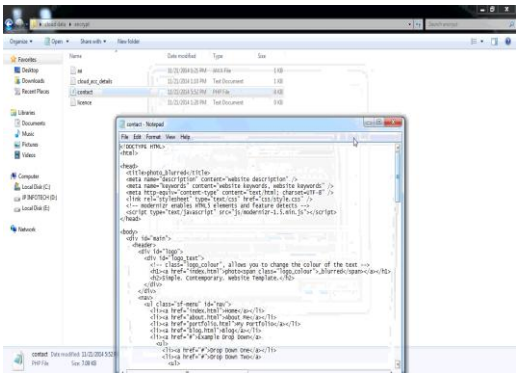


Fig.14 File after download with access

## V. CONCLUSION

In this paper, we have discussed concurrent and independent access to encrypted cloud databases, proposes an innovative architecture that guarantees confidentiality of data stored in public cloud databases. The proposed system will not require modifications to the cloud database, and it will be immediately applicable to existing cloud DBaaS. Resolve problem of single point failure and a bottleneck limiting availability and scalability of cloud database services.

## REFERENCES

[1] Ferretti, Luca, Michele Colajanni, and Mirco Marchetti. "Distributed, concurrent, and independent access to encrypted cloud databases." (2014): 1-1.

[2] Ashalatha, r., and m. Vaidehi. "The significance of data security in cloud: a survey on challenges and solutions on data security".

[3] Arora, Indu, and Anu Gupta. "Cloud Databases: A Paradigm Shift in Databases." International J. of Computer Science Issues 9.4 (2012): 77-83.

[4] 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.

[5] Muhammad Rizwan Asghar, Mihaela Ion, Bruno Crispo, "ESPOON Enforcing Encrypted Security Policies in Outsourced Environment", 2011 Sixth International Conference on Availability, Reliability and Security.

[6] Luca Ferretti, Michele Colajanni, and Mirco Marchetti, "Access control enforcement on query-aware encrypted cloud databases" IEEE 2013.

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

[8] Ferretti, Luca, et al. "Security and confidentiality solutions for public cloud database services." SECURWARE 2013, The Seventh International Conference on Emerging Security Information, Systems and Technologies. 2013.

[9] 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.

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

[11] A. Shamir, "How to Share a Secret," Comm. of the ACM, vol. 22, no. 11, pp. 612-613, 1979.

[12] M. Hadavi, E. Damiani, R. Jalili, S. Cimato, and Z. Ganjei, "AS5: A Secure Searchable Secret Sharing Scheme for Privacy Preserving Database Outsourcing," Proc. Fifth Int'l Workshop Autonomous and Spontaneous Security, Sept. 2013.

[13] "Oracle Advanced Security," Oracle Corporation, http://www.oracle.com/technetwork/database/options/advanced-security,Apr. 2013.

[14] G. Cattaneo, L. Catuogno, A.D. Sorbo, and P. Persiano, "The Design and Implementation of a Transparent Cryptographic File System For Unix," Proc. FREENIX Track: 2001 USENIX Ann. Technical Conf., Apr. 2001.

[15] E. Damiani, S.D.C. Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati, "Balancing Confidentiality and Efficiency in UntrustedRelational Dbmss," Proc. Tenth ACM Conf. Computer and Comm. Security, Oct. 2003.

**BIODATA**

**Author**

**Shaik Roshan** presently pursuing his M.Tech in CSE, Kottam College of Engineering, Chinnatekuru (V),Kurnool,Andhra Pradesh, India.

**Co-Author**

**Ramakrishna Reddy** received M.Tech. Presently working as Assistant Professor in Kottam College of Engineering, Chinnatekuru (V) ,Kurnool,Andhra Pradesh, India.