ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Distributed, Concurrent and Independent Access to Encrypted Cloud Databases

K.Bharati Reddy
M.Tech Student,
Department of CSE,
Aurora's Scientific Technological & Research Academy, Hyderabad.

P.Vaishali
Sr. Assistant Professor,
Department of CSE,
Aurora's Scientific Technological & Research Academy, Hyderabad.

Abstract:

In the present era of computing data in a cloud will be placed anywhere because of the critical nature of the applications and it is very important to maintain secure clouds and the major security challenge with clouds is that the owner of the data may not have control of where the data is placed because if one wants to exploit the benefits of using cloud computing then this requirement imposes clear data management choices such as the original plain data that must be accessible only by trusted parties that do not include cloud providers or intermediaries and Internet.In case of any un-trusted context then the data must be encrypted by which we satisfies these goals that has different levels of complexity depending on the type of cloud services.In this paper we propose a Secure DBaaS as the first solution that allows cloud tenants to take full advantage of DBaaS qualities such as availability then reliability and elastic scalability without exposing unencrypted data to the cloud provider and the architecture design was motivated by the goal to allow multiple or independent and geographically distributed clients to execute concurrent operations on the encrypted data including SQL statements that modify the database structure.

Keywords:

Cloud, security, confidentiality, SecureDBaaS, database.

I. INTRODUCTION:

In the present era of computers our main aim of writing this paper for implementing the system is to integrate the cloud database services with data confidentiality and the possibility of executing concurrent operations on encrypted data so we use cloud for uploading owner's data where a data owner is the one who has uploaded his data on cloud and he or she is not ensured about his or her data so we have to store the data on the cloud by encrypting it.

Generally this encryption of data takes place at client side and metadata of that data also created that is secureD-BaaS concept where the encrypted data is stored at the cloud along with its encrypted metadata and then the authorized clients can access the data by using only metadata.We consider this to be the first solution that supports geographically distributed clients to connect directly to an encrypted cloud database and to execute concurrent and independent operations including those modifying the database structure.Our proposed system includes the advantage of eliminating intermediate proxies that limit the elasticity; availability and scalability properties that are intrinsic in the cloud-based solutions where a secureD-BaaS provides several original features that differentiate it from previous work in the field of security for remote database services.

II. EXISTING SYSTEM:

Our existing system mainly focuses on following:

A. Cloud database:

We consider that the tenant data is saved in a relational database and 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 and we distinguish the strategies for encrypting the database structures and the tenant data.

B. Metadata Management:

The metadata that is generated by SecureDBaaS contains all the information that is necessary to manage the SQL statements over the encrypted database in a way for transparent to the user as the metadata management strategies represent an original idea because the SecureDBaaS is the first architecture storing all metadata in the un-trusted cloud database together with the encrypted tenant data.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

C. Encryption algorithm:

The encryption algorithm chooses and used to encrypt and decrypt all the data that is stored in the database table.

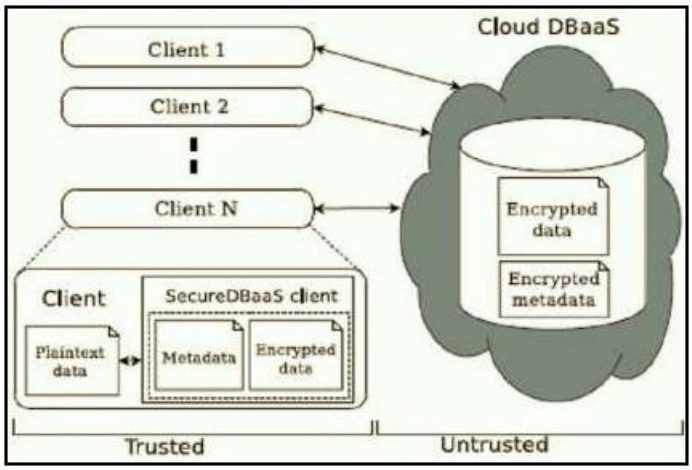

Fig.1. SecureDBaaS Architecture.

In the above figure Fig 1describes the overall architecture and we assume that a tenant organization acquires a cloud database service from an un-trusted DBaaS provider where the tenant then deploys one or more machines say 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 then to read and write the data and even to create and modify the database tables after creation where the SecureDBaaS is designed to allow multiple and independent clients to connect directly to the un-trusted cloud DBaaS without any intermediate server in the system.

III.PROPOSED SYSTEM|:

A.Cloud database:

Let us assume that tenant data are saved in a relational database and 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 as we distinguish the strategies for encrypting the database structures and the tenant data.

B. Metadata Management:

Metadata that is generated by the SecureDBaaS contains all the information that is necessary to manage all the available SQL statements over the encrypted database in a way transparent to the user and the metadata management strategies represent an original idea because SecureDBaaS is the first architecture storing all the metadata in the un-trusted cloud database together with the encrypted tenant data in the system.

C. Encryption algorithm:

In order to choose the encryption algorithms used to encrypt and decrypt all the available data stored in the database table.

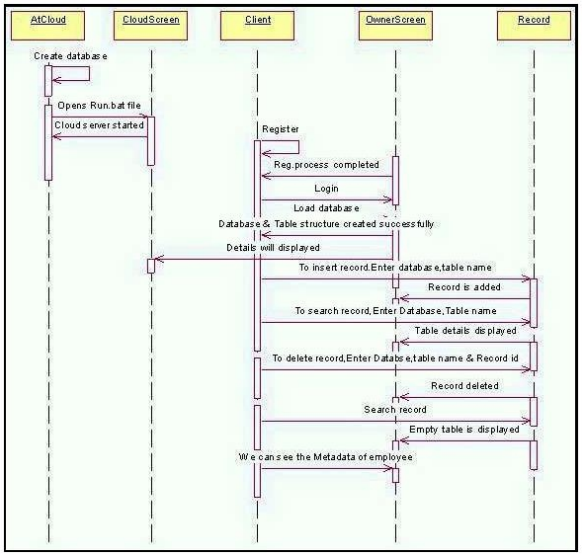

Fig.2. Sequence diagram of the proposed system

The above figure Fig 2 describes the system design that has the modules and its components.

A. Creation of database:

In this module a client creates a specific database and stores the data in the form of columns and rows then later creates the database along with the metadata which will help for later communication instead of providing the whole database to a end user.

B. Selection of Encryption and Decryption algorithm:

In this module we select the encryption algorithm to encrypt and decrypt the created database and its metadata by providing the security to whole data of client which is to be uploaded on the cloud.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

C.Cloud Database:

A cloud database is the service provider which provides services to the tenants as all the encrypted data from data owner is uploaded on cloud which provides the concurraccess to cloud DB to the geographically deployed clients and the cloud database contains encrypted database and its encrypted metadata.

D. Application

In this module contains the application of system to the cloud and then we will apply these data and encryption techniques all on the cloud as the module will explain it and we use the master key to access the cloud data after data is uploaded on data.In this process first we will get encrypted data if our key is correct then by using random decryption keys and we will get the final output in the form of plaintext data as the input is taken from user in the form of sql query as the primary process is creation of database by a client and will enter rows into the database along with the metadata of database which is created based on the selected encryption algorithm which is applied to the database and its metadata and the final output gives us with the encrypted data with all its information and key used.

IV. IMPLEMENTATION:
A. Data Management:

Whenever a cloud database acts as service provider for tenants then the cloud is created first for the system and all the information or data stored in the relational database for creating tables and column we have to access it with SQL query only is available to the end user.

B. Metadata Management:

Metadata generated by SecureDBaaS contains all the information that is necessary to manage SQL statements over the encrypted database in a way which is transparent to the user and the metadata management strategies represent an original idea because SecureDBaaS is the first architecture storing all metadata in the un-trusted cloud database together with the encrypted tenant data and the SecureDBaaS uses two types of metadata.

•In a database metadata are related to the whole database and there is only one instance of this metadata type for each database.

•In a table metadata are associated with one secure table where each table metadata contains all information that is necessary to encrypt and decrypt data of the associated secure table.

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

Field	Type	Null	Key	Default	Extra
weqUeqqUPfbSXBAfJwfYQXjAeqeq	varchar(500)	YES		NULL	
deqzPvKUneqcMeqPrPnSWaRugeqeq	varchar(500)	YES		NULL	
iBcseqfeqeqvkZeqNLvbDfJeqwgeqeq	varchar(500)	YES		NULL	
gvbleqaqWiOeqyeqOkOElxyUAeqeq	varchar(500)	YES		NULL	
cskDzeqceqotNYTkIMSeqequIgeqeq	varchar(500)	YES		NULL	

Fig.3. Data View

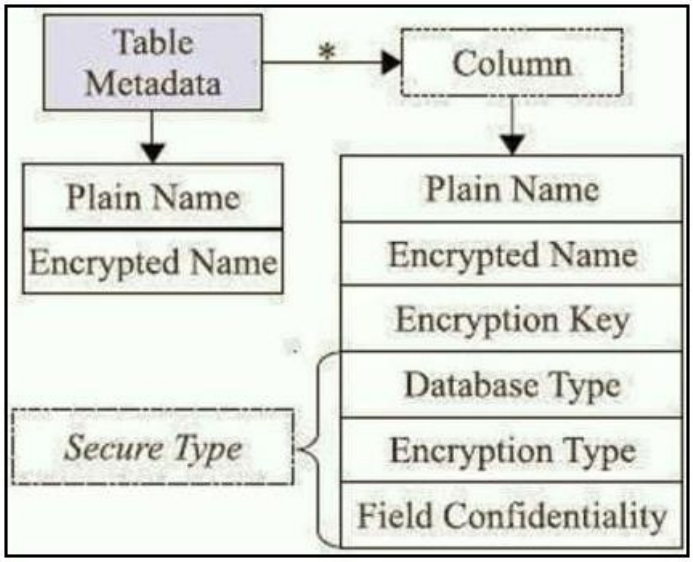

Fig.4. Structure of Table Metadata

This choice of design minimizes the amount of metadata that each SecureDBaaS client has to fetch from the un-trusted cloud database thus reducing bandwidth consumption and processing time. Moreover as it allows multiple clients to access independently metadata related to different secure tables that are available. As the database metadata contain the encryption keys that are used for the secure types and a different encryption key is associated with all the possible combinations of data type and encryption type as the database metadata represents a key ring and do not contain any information about tenant data.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In the above figure Fig 4 the structure of a table metadata is represented and the table metadata contains the name of the related secure table and the unencrypted name of the related plaintext table and the table metadata includes the column metadata for each column of the related secure table where each column metadata contains 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 and 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 which allows a SecureDBaaS client to be informed about the data type and the encryption policies associated with a column in the table.
•Encryption key: the key used to encrypt and decrypt all the data stored in the column.

The SecureDBaaS stores metadata in the metadata storage table that is located in the untrusted cloud as the database which is an original choice that augments flexibility but opens two novel issues in terms of efficient data retrieval and data confidentiality.In order to allow SecureDBaaS clients to manipulate metadata through the SQL statements we save the database and table metadata in a tabular form and the metadata confidentiality is guaranteed through encryption.

Both the database and the table metadata are encrypted through the same encryption key before being saved where the encryption key is called a master key and 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. Where each of the metadata can be retrieved by clients through an associated ID which is the primary key of the metadata storage table and the ID is computed by applying a Message Authentication Code (MAC) function to the name of the object database or table that describes the corresponding row. The deterministic MAC function allows clients to retrieve the metadata of a given table by the knowing based on its plaintext name where this mechanism has the further benefit of allowing clients to access each metadata independently which is an important feature in concurrent environments and in addition to SecureDBaaS clients can use caching policies to reduce the bandwidth overhead.

Most of the encryption algorithms are applied to encrypt the database and there are various encryption algorithms such as symmetric and asymmetric but we will apply symmetric algorithm which proved key distribution only once to all tenants there will be no different private key related to every user.

V. CONCLUSION:

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

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]Kevin Hamlen, Murat Kantarcioglu, Latifur Khan, Bhavani Thuraisingham, “Security Issues for Cloud Computing”, International Journal of Information Security and Privacy, 4(2), 39-51, April-June 2010.
[3]Auditor Bhavna Makhija,VinitKumar Gupta,Indrajit Rajput, “Enhanced Data Security in Cloud Computing with Third Party”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 2, February 2013.
[4]M. Armbrust et al., “A View of Cloud Computing”, Comm. of the ACM, vol. 53, no. 4, pp. 50-58, 2010.
[5]L. Ferretti, M. Colajanni, and M. Marchetti, “Supporting Security and Consistency for Cloud Database”,Proc. Fourth Intl Symp. Cyberspace Safety and Security, Dec. 2012.
[6]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 IdealLattices, Proc.41st Ann. ACM Symp. Theory of Computing May 2009.