ISSN 2347 - 398

International Journal of Emerging Trends in Engineering Research (IJETER), Vol. 3 No.6, Pages : 60- 67 (2015)
Special Issue of NCTET 2K15 - Held on June 13, 2015 in SV College of Engineering, Tirupati
http://warse.org/IJETER/static/pdf/Issue/NCTET2015sp13.pdf

Complex Queries Perform Concurrent Operations on Encrypted Stored Data in un-Trusted Cloud Contexts

A. Revathi [1], N. Harish [2]
[1] Associate professor, [2] M.Tech Student, Dept. of CSE, SVCE, Tirupati.
[1] arevathi20@gmail.com,
[2] harish.nagichetty@gmail.com,

Abstract:

Cloud tenants are suspecting that how cloud will provide security, confidentiality and availability of data, if they store critical information in un-trusted third party context. Cloud providers should come forward to give assurance to protect the confidentiality of clients' data by encrypting it and make it available in all perspectives. By rigorous analysis, we came to know that there are several stored procedures available for storing the data securely into the cloud. But, because of lack of knowledge many of them are failing to maintain data confidentially in database. We have newly designed an integrated approach which maintains secrecy and perform concurrent operations on stored cipher text without losing their confidentiality. It enables the geographically distributed clients to directly connect to perform individual independent operations concurrently on encrypted database and also users can update the existing table structure. To evaluate the performance of proposed functionality we have integrated database as a service with the help of TPC-C standard to test the network latencies to connect different no of clients.

Keywords: cloud providers, confidentiality, database, cloud storage.

I INTRODUCTION:

Maintaining the confidentiality is a critical job for cloud providers because millions of users are accessing information like banking, medical records, insurance and social network are sensitive information which is stored and retrieved through some interface maintain by several organizations hosted in the cloud. In this particular situation due to some reasons like scarcity of memory in a database or database crashes reason, if cloud providers or users move on to copy the important information in the infrastructure of un-trusted third parties means security and confidentiality is a big question for that storage. It clearly shows that cloud requires an effective data management with control access where the data might be stored in trusted and un-trusted context. Cloud allow the users those how have authorized privileges they may access the original plain text. The user doesn't have enough privileges and also in any un-trusted context data must be in encrypted form. To implement these things we have to face some difficulties in different stages on the basis of cloud services.

The architecture design was motivated by a threefold goal: 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; to eliminate any intermediate server between the cloud client and the cloud provider. The possibility of combining availability, elasticity, and scalability of a typical cloud DBaaS with data confidentiality is demonstrated through a prototype of SecureDBaaS that supports the execution of concurrent and

ISSN 2347 - 398

International Journal of Emerging Trends in Engineering Research (IJETER), Vol. 3 No.6, Pages : 60- 67 (2015)
Special Issue of NCTET 2K15 - Held on June 13, 2015 in SV College of Engineering, Tirupati
http://warse.org/IJETER/static/pdf/Issue/NCTET2015sp13.pdf

independent operations to the remote encrypted database from many geographically distributed clients as in any unencrypted DBaaS setup. To achieve these goals, SecureDBaaS integrates existing cryptographic schemes, isolation mechanisms, and novel strategies for management of encrypted metadata on the un-sstrusted cloud database. This paper contains a theoretical discussion about solutions for data consistency issues due to concurrent and independent client accesses to encrypted data. In this context, we cannot apply fully homomorphic encryption schemes because of their excessive computational complexity.

II SYSTEM OVERVIEW

Our system primarily focuses on following things:

- Cloud database as a Service
- Metadata Organization
- Encryption algorithms

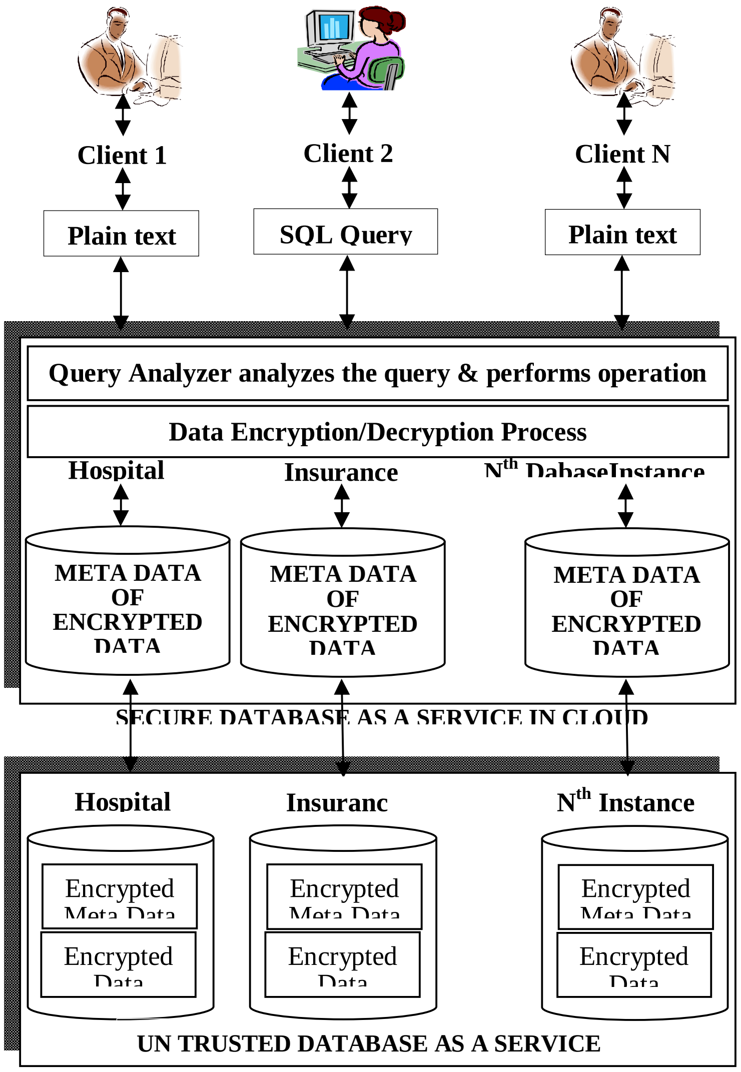

Client 1
Client 2
Client N
Plain text
SQL Query
Plain text
Query Analyzer analyzes the query & performs operation
Data Encryption/Decryption Process
Hospital
Insurance
Nth DabaseInstance
META DATA OF ENCRYPTED DATA
META DATA OF ENCRYPTED DATA
META DATA OF ENCRYPTED DATA
SECURE DATABASE AS A SERVICE IN CLOUD
Hospital
Insuranc
Nth Instance
Encrypted Meta Data
Encrypted Data
Encrypted Meta Data
Encrypted Data
Encrypted Meta Data
Encrypted Data
UN TRUSTED DATABASE AS A SERVICE

Fig.1: SecureDBaaS Architecture

Cloud database as 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 Organization: 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.

ISSN 2347 - 398

International Journal of Emerging Trends in Engineering Research (IJETER), Vol. 3 No.6, Pages : 60- 67 (2015)
Special Issue of NCTET 2K15 - Held on June 13, 2015 in SV College of Engineering, Tirupati
http://warse.org/IJETER/static/pdf/Issue/NCTET2015sp13.pdf

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 algorithms: Choosing the encryption algorithms used to encrypt and decrypt all the data stored in the database table. Fig. 1 describes the overall architecture. We assume that a tenant organization acquires a cloud database service from an un-trusted 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. SecureDBaaS is designed to allow multiple and independent clients to connect directly to the untrusted cloud DBaaS without any intermediate server.

III SYSTEM DESIGN

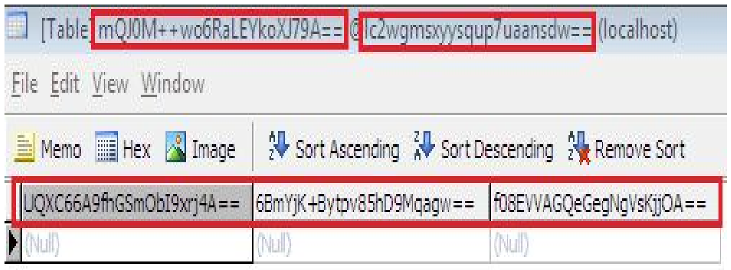
3.1 Cloud database: We assume that tenant data are saved in a relational database. We have to preserve the confidentiality of the stored data and even of the database structure because table and column names may yield information about saved data. We distinguish the strategies for encrypting the database structures and the tenant data.

3.2 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 un-trusted cloud database together with the encrypted tenant data.

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

The table defined in the **Trusted Cloud Database**:	The table defined in **un-Trusted Cloud Database**:
Employee (Database Schema) | |--**emp (Table Name)** |----**eid, ename, salary (attributes)** **Eid** | **Ename** | **Salary**	mQJ0M++wo6RaLEYKoXJ79A== **(Database Schema)** | |-- Ic2wgmsxyysqup7uaansdw== **(Table Name)** |---- UQXC66A9fhGSmObI9xrj4A==, 6BmYjK+Bytpv85hD9Mqagw==, f08EVVAGQeGegNgVsKjjOA== **(attributes)**

Table 1: Define structure of trusted and un-trusted table from encrypted database


[Table] mQJ0M++wo6RaLEYkoXJ79A== @ Ic2wgmsxyysqup7uaansdw== (localhost)
File Edit View Window
Memo Hex Image Sort Ascending Sort Descending Remove Sort

UQXC66A9fhGSmObI9xrj4A==	6BmYjK+Bytpv85hD9Mqagw==	f08EVVAGQeGegNgVsKjjOA==
(Null)	(Null)	(Null)

Fig. 2.1: Structure of table un-trusted

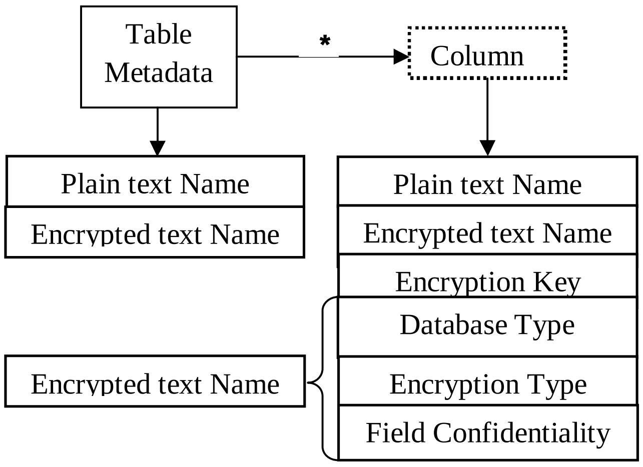


Fig.2.2: Structure of table metadata.

ISSN 2347 - 398

International Journal of Emerging Trends in Engineering Research (IJETER), Vol. 3 No.6, Pages : 60- 67 (2015)
Special Issue of NCTET 2K15 - Held on June 13, 2015 in SV College of Engineering, Tirupati
http://warse.org/IJETER/static/pdf/Issue/NCTET2015sp13.pdf

IV IMPLEMENTATION

4.1 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.

4.2 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.

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.

This design choice 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, it allows multiple clients to access independently metadata related to different secure tables. Database metadata contain the encryption keys that are used for the secure types. A different encryption key is associated with all the possible combinations of data type and encryption type. Hence, the database metadata represent a key ring and do not contain any information about tenant data.

The structure of a table metadata is represented in Fig. 3. 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.

SecureDBaaS stores metadata in the metadata storage table that is located in the untrusted cloud as

ISSN 2347 - 398

International Journal of Emerging Trends in Engineering Research (IJETER), Vol. 3 No.6, Pages : 60- 67 (2015)
Special Issue of NCTET 2K15 - Held on June 13, 2015 in SV College of Engineering, Tirupati
http://warse.org/IJETER/static/pdf/Issue/NCTET2015sp13.pdf

the database. This is an original choice that augmnts 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. 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.

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

Metadata Storage Table

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

Fig. 3: Organization of database metadata and table metadata in the metadata storage table.

4.3 Algorithms:

Encryption algorithms are applied to encrypt the database. There are various encryption algorithms 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.

5. Experimental Results:

We demonstrate the applicability of SecureDBaaS to different cloud DBaaS solutions by implementing and handling encrypted database operations on emulated and real cloud infrastructures. The present version of the SecureDBaaS prototype supports PostgreSQL, MySql, and SQL Server relational databases. As a first result, we can observe that porting SecureDBaaS to different DBMS required minor changes related to the database connector and minimal modifications of codebase.

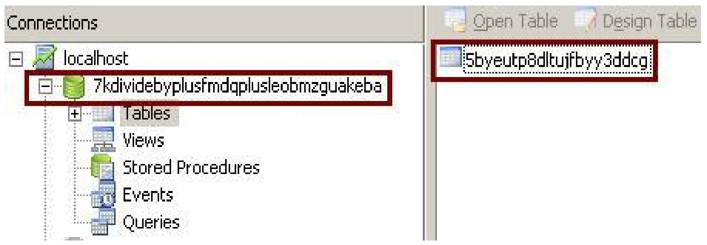
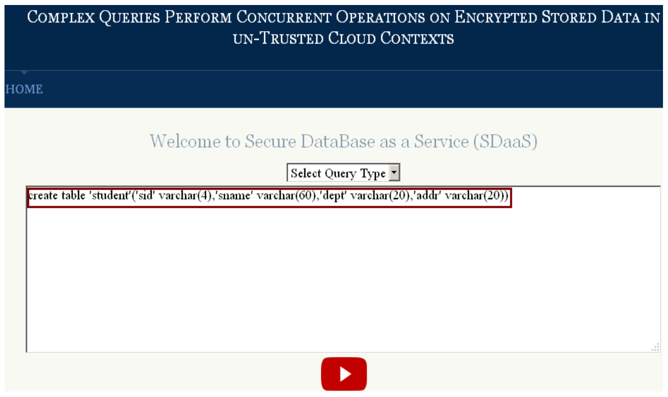


Fig.4.1: database definition in un-trusted cloud database


Complex Queries Perform Concurrent Operations on Encrypted Stored Data in un-Trusted Cloud Contexts
HOME
Welcome to Secure DataBase as a Service (SDaaS)
Select Query Type

```
create table 'student'('sid' varchar(4),'sname' varchar(60),'dept' varchar(20),'addr' varchar(20))
```

Fig.4.2: Interface for creating a table structure

International Journal of Emerging Trends in Engineering Research (IJETER), Vol. 3 No.6, Pages : 60- 67 (2015)
Special Issue of NCTET 2K15 - Held on June 13, 2015 in SV College of Engineering, Tirupati
http://warse.org/IJETER/static/pdf/Issue/NCTET2015sp13.pdf

create table 'student'('sid' varchar(4),'sname' varchar(60),'dept' varchar(20),'addr' varchar(20));

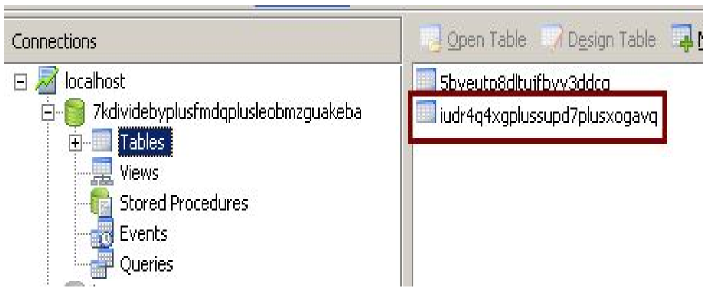
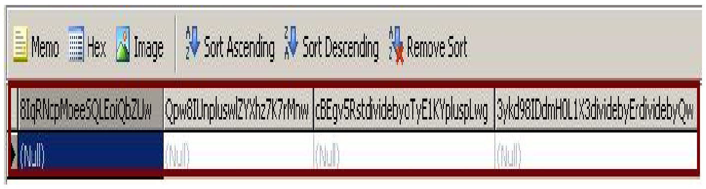
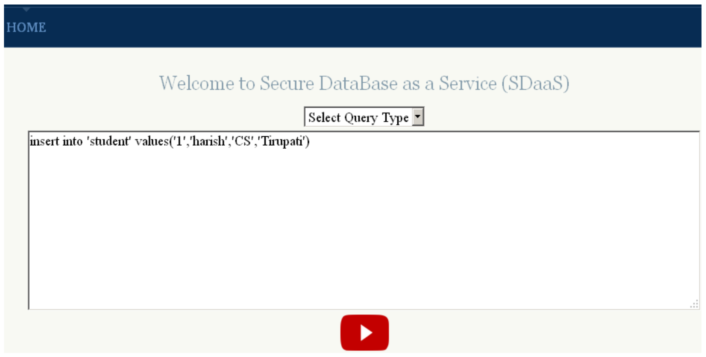


Fig.4.3: New Table with Encrypted Name in MYSQL



8IqRNcpMoeeSQLEoiQbZUw	Qpw8IUnpluswlZYXhz7K7rMnw	cBEgv5RstdividebyoTyE1KYpluspLwg	3ykd98IDdmH0L1X3dividebyErdividebyQw
(Null)	(Null)	(Null)	(Null)

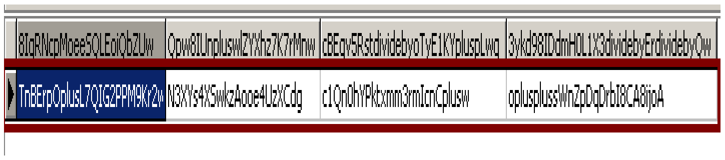
Fig.4.4: new table create with Encrypted Columns



```
insert into 'student' values('1','harish','CS','Tirupati')
```

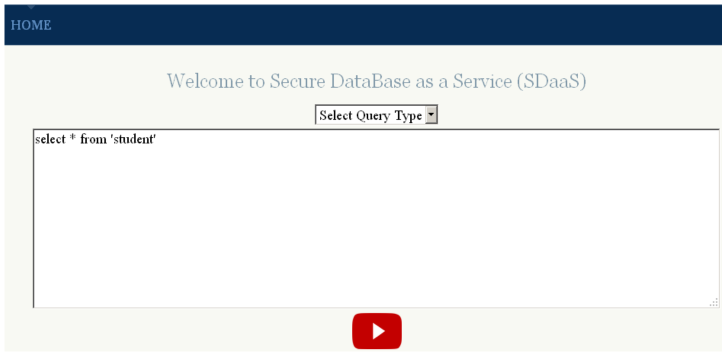
Fig.4.5: Data stored into the database through insert query

insert into 'student' values('1','harish','CS','Tirupati');



8IqRNcpMoeeSQLEoiQbZUw	Qpw8IUnpluswlZYXhz7K7rMnw	cBEgv5RstdividebyoTyE1KYpluspLwg	3ykd98IDdmH0L1X3dividebyErdividebyQw
TnBErpOplusL7QIG2PPM9Kr2w	N3XYs4XSwkzAooe4UzXCdg	c1Qn0hYPktxmm3rmIcnCplusw	opluspluss WhZpDqDrbI8CA8ijoA

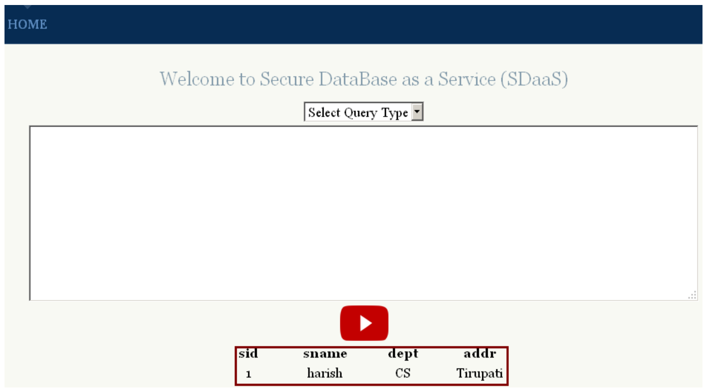
Fig.4.6: Plain text encrypted automatically and stored in the table



```
select * from 'student'
```

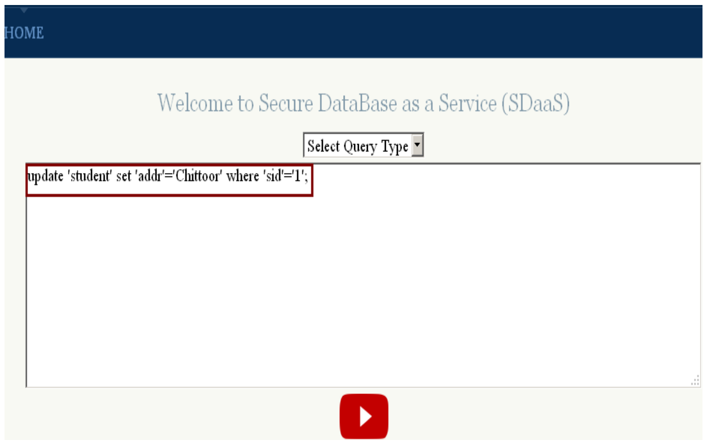
Fig.4.7: query posed to view the stored details records from a encrypted table

select * from 'student';



sid	sname	dept	addr
1	harish	CS	Tirupati

Fig.4.8: Data automatically decrypted while retrieving from Encrypted table in encrypted database



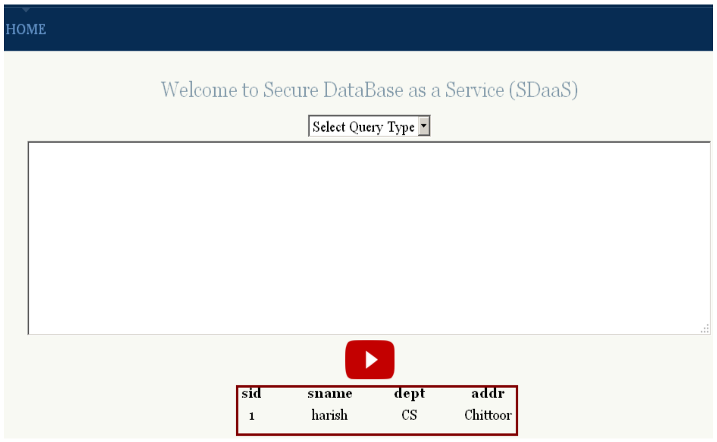
```
update 'student' set 'addr'='Chittoor' where 'sid'='1';
```

Fig.4.9: update already existing data in encrypted table

update 'student' set 'addr'='Chittoor' where 'sid'='1';

ISSN 2347 - 398

International Journal of Emerging Trends in Engineering Research (IJETER), Vol. 3 No.6, Pages : 60- 67 (2015)
Special Issue of NCTET 2K15 - Held on June 13, 2015 in SV College of Engineering, Tirupati
http://warse.org/IJETER/static/pdf/Issue/NCTET2015sp13.pdf



sid	sname	dept	addr
1	harish	CS	Chittoor

Fig.4.10: After updating data from an encrypted table

In scenarios characterized by a static database structure, SecureDBaaS 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.
SecureDBaaS is the first architecture that allows concurrent and consistent accesses even when there are operations that can modify the database structure. In such cases, we have to guarantee the consistency of data and metadata through isolation levels, such as the snapshot isolation [21], that we demonstrate can work for most usage scenarios.

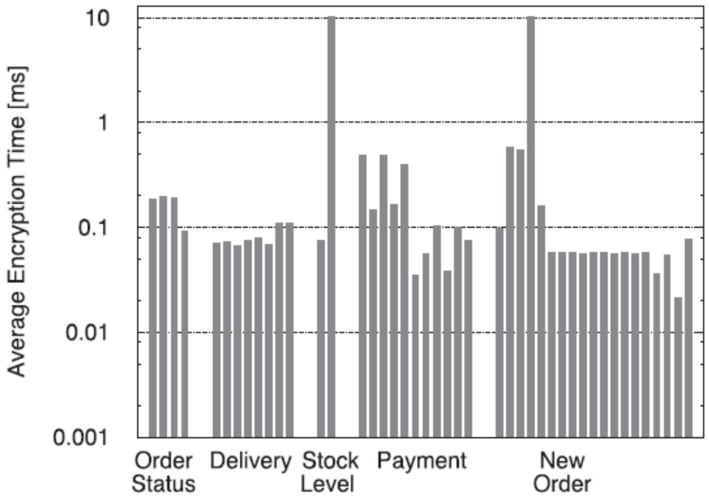


Fig. 5: Encryption times of TPC-C benchmark operations grouped by the transaction class.

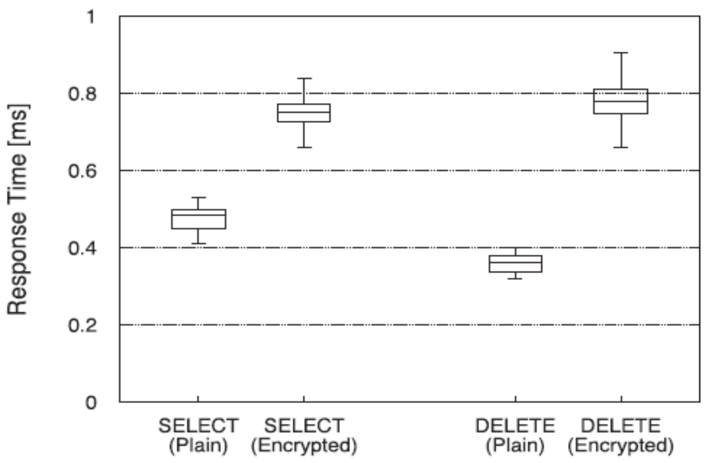


Fig. 6: Plain versus encrypted SELECT and DELETE operations

CONCLUSION:

Our solution provides a strong protection to an individual user's confidentiality, even though if they store the sensitive information in any un-trusted third party context. For security and privacy reasons, our system automatically encrypts the data before storing into the cloud and it retrieves by decrypts the data based on user posed SQL Query basis from un-trusted cloud context. By doing this it doesn't show any impact on performance degrading/varying, if n no of authorized users can access the distributed information from different geographical location. It allows only trusted users to search the stored information from relational tables, store the data into tables, retrieve and update and delete the already existing information from database smoothly without having any inconvenience.

REFERENCES

[1] M. Armbrust et al., "A View of Cloud Computing," Comm. of the ACM, vol. 53, no. 4, pp. 50-58, 2010.

[2] W. Jansen and T. Grance, "Guidelines on Security and Privacy in Public Cloud Computing," Technical Report Special Publication 800-144, NIST, 2011.

[3] 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.

[4] J. Li, M. Krohn, D. Mazie res, and D. Shasha, "Secure Untrusted Data Repository (SUNDR)," Proc. Sixth USENIX Conf. Opearting Systems Design and Implementation, Oct. 2004.

[5] P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish, "Depot: Cloud Storage with Minimal Trust," ACM Trans. Computer Systems, vol. 29, no. 4, article 12, 2011.

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

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.