Indian Journal of Science and Technology, Vol 9(13), DOI: 10.17485/ijst/2016/v9i13/86153, April 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Implementing Encrypted Database for Concurrent Access in Cloud Environment

Jayshri C. Wagh[1]* and Sonali Mhatre[2]

[1]Computer Engineering Department, Bharati Vidyapeeth College of Engineering, Mumbai University, Navi Mumbai - 400614, Maharashtra, India; jayshriwagh16@gmail.com
[2]Information Technology Department, Bharati Vidyapeeth College of Engineering, Mumbai University, Navi Mumbai - 400614, Maharashtra, India; sonalinmhatre@gmail.com

Abstract

Background/Objectives: Database as a Service (DBaaS) model is used to manage databases in public cloud environment. Placing critical data on the cloud environment outside the organization that should have guarantee that data is secure, confidential and available at any time to legitimate user. User should be able to retrieve cloud database efficiently. **Methods**: Architecture proposed in this work provides data confidentiality for cloud databases. It is designed to allow multiple and independent clients to connect to the cloud without intermediate server. Data is encrypted before upload to the cloud. Multiple cryptography techniques are used to convert plaintext into encrypted data. Data will not be exposed to the cloud provider and any other public user who are not registered to access the database. Encrypted query submission model is used to secure the query values and maintain confidentiality. **Findings:** By using adaptive encryption system does not require the choice of encryption scheme must be adopted for each database column and SQL operation at design time. Encryption schemes are able to perform most of the query operations on encrypted cloud database. In this system set of queries are not decided at design time. It works even when set of query will change dynamically. This is not possible with the existing system. **Application/Improvements:** Using adaptive encryption scheme most of the operations are performed on encrypted cloud database. Execution time of various SQL operations is improved as compared to the previous system.

Keywords: Adaptive, Confidentiality, Encryption, Public Cloud, Security

1. Introduction

Cloud computing is a recent trend in IT that moves computing and data away from desktop and portable PCs into large data canters. Actual cloud infrastructure is combination of hardware and systems software in data centres that provide these services to the user over the internet.

In today's life everybody is spending more time with computing devices to collect their data from various sources over network and store it at the place where it as portable for the user. During roaming time user may require their data which is stored at a computer and it is very difficult to carry that computer with whole data while roaming time. So it becomes a problem to the user while using data at roaming time. So to store data in cloud storage environment or in network can solve the problem. Cloud storage refers to storing a very large amount of data which is referred to cloud computing basis on pay per use scheme. So it provides highest availability, scalability, and reliability to the users. User does not worry about their data lost or any other problem related to data. There are various cloud service models which provides various services to the user on pay per use basis. 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).

*Author for correspondence

1.1 Cloud Services

1.1.1 Software as a Service (SaaS)

It provides the capability to consumer to use the provider's application over internet. Here various deployed readymade applications are there which can be used by the user so various users can use this application on pay per use basis. Here does not requires to pay for software license or service. Sometimes they need to pay for the maintenance of the service[1].

1.1.2 Platform as a Service (PaaS)

Capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created applications using programming languages and tools supported by the provider. Here in this service model user can develop software by using providers given platform like user can build software by using any programming language and any database server which is provided by service provider. Users are independent to build and deploy their software[1].

1.1.3 Infrastructure as a Service (IaaS)

Capability provided to the consumer is to rent processing, storage, networks, and other fundamental computing resourcesover the network. Here in this type of service my provider provides various computing resources, storage devices, various servers, large data centers. The data centre of hardware and software called as Cloud. The IaaS again classify into three types depends on type of service it provides. i) Database as a Service ii) Storage as a Service iii) Computing as a service. These types of IaaS are depends on the type of service provided by it. Among these types our interest is Database as a Service.

1.2 Database as a Service (DBaaS)

This is the one of the sub-service model of Infrastructure as a service where tenant organization can create their database and deploy into the cloud environment. Cloud provider provides service to the tenants and apply charges from the tenants depends on use. The database which is created by tenant not requires modifications to the cloud database hence it is controlled by the cloud provider. Here tenant organization has to pay for the database service for getting the service from the cloud service provider. A DBaaS show in Figure 1 is successful paradigm where the data and the storage devices are located in cloud infrastructure and user can use data from anywhere[3].

In this model all tenant data is stored in cloud environment means in the hands of third party so user have to worry about the security, confidentiality and privacy problems from the cloud database as a service provider. Cloud provider provides a security to the frontend resource not

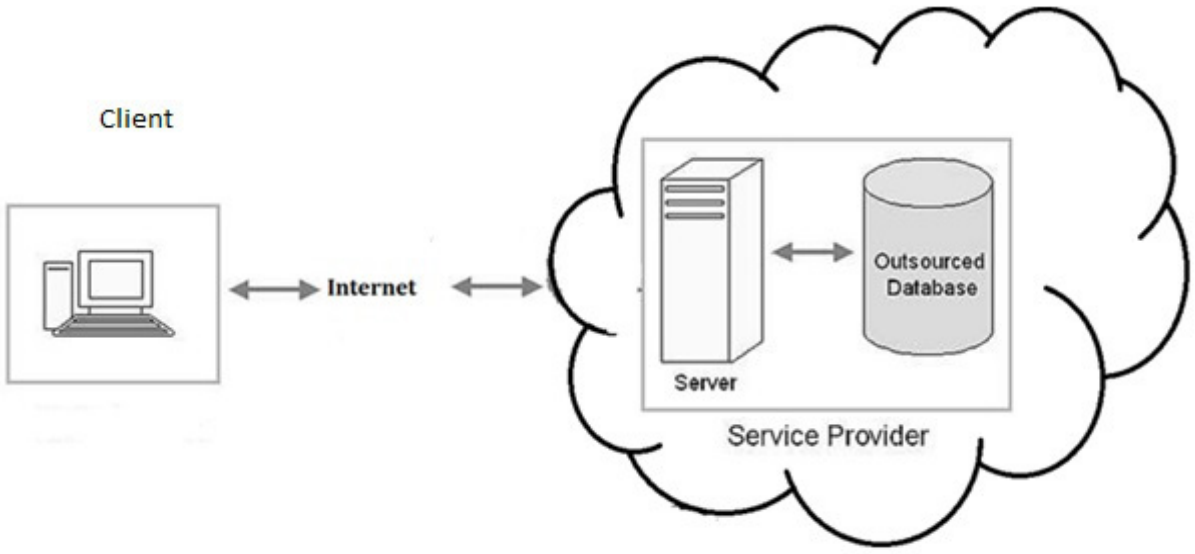

Figure 1. Database as a Service Architecture.

to backend resources, so it is very easy to attacker to hack data from backend. So tenant has to compromise the data integrity and confidentiality. Where leakage details of data might be at the tenants cloud resources and cloud service provider are the responsible for that issue[2]. Thus user must provide a security from the cloud provider between the attackers and the forgoing cloud resources by encrypting their own data at the tenant side only. Encryption is a process of converting the data in other format to protect data that is managed by untrusted server. So here in cloud environment overall data which is stored by using database as a service model is in encrypted form to maintain the confidentiality of the tenant data.

2. Existing System

The system which is proposed in[4] call it as existing system which allow multiple client to connect the untrusted cloud DBaaS without using intermediate or proxy server. Figure 2 describes the overall architecture of existing system. It assumes that a user uses a cloud database service from DBaaS provider. Then users/tenants use multiple machines and deploy SecureDBaaS client on each machine. The tenant organization allows a legitimate user to connect to cloud DBaaS to read, write data from the cloud environment. They have used the same security model which is defined in systems[5,6], where the client are trusted, network is untrusted, and the cloud provider is honest but they are curious about what happening in cloud environment, that is, whether cloud service operations are executed correctly or not, because of this user/client information confidentiality is at risk. For these reasons, clients overall data must be encrypted before exiting from the client. The information managed by client is plain data, encrypted data, metadata and encrypted metadata. Plaintext data consist of information that a client wants to store and process in the cloud DBaaS. To avoid confidentiality violation from untrusted cloud provider system uses multiple encryption schemes to transform plaintext data into encrypted data. In this system each and every details of the database is stored in the encrypted form. In this system client produce a set of metadata consisting of information required to encrypt and decrypt data. Even metadata are encrypted and stored in the cloud DBaaS.

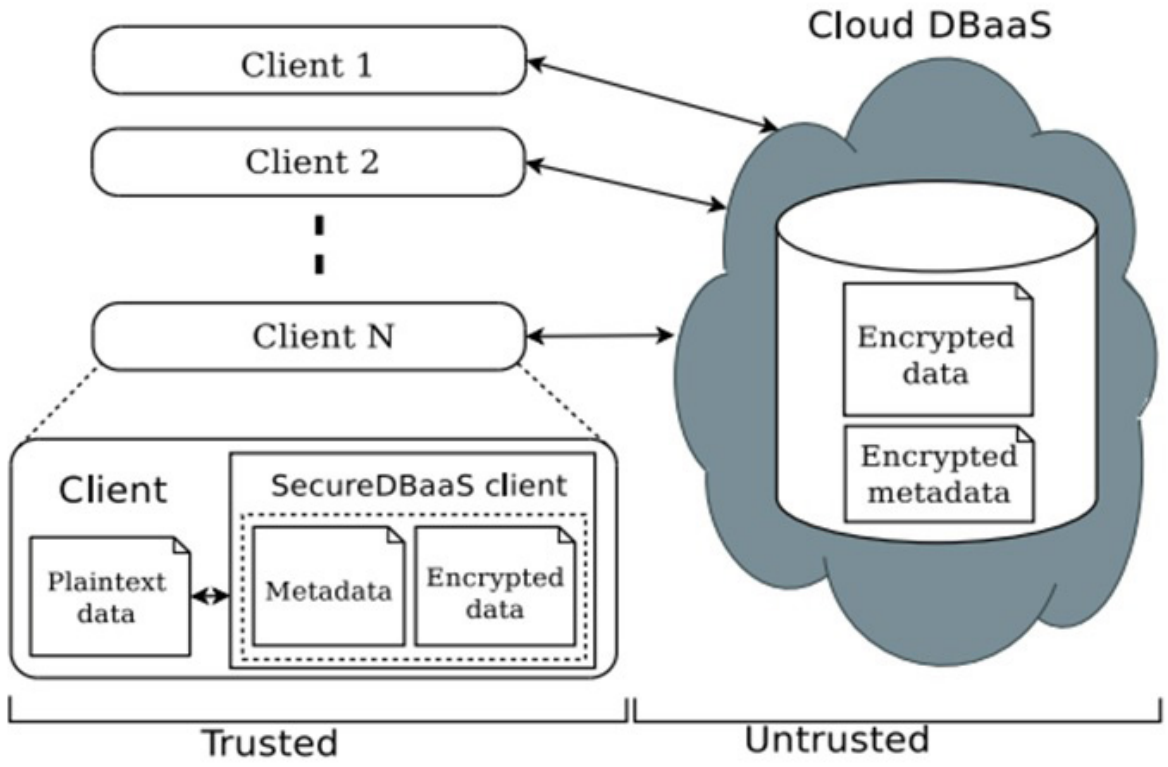

Figure 2. Secure DBaaS Architecture.

There are some limitations of the existing systems which require the choice of encryption technique or scheme must be applied for each attribute of the database table and also SQL operations at design time. These prototype works when the set of queries can be statically defined at design time, if workload changes after the database design so existing prototype will not work dynamically. Some system can perform access control mechanism without the intervention of cloud provider but do not allow execution of SQL operations on encrypted data.

3. Proposed System

We proposed architecture which is extension of existing system[4] describe the overall structure shown in Figure 3. We assume that a tenant organization acquires a cloud database service from an untrusted DBaaS provider[4]. The proposed system supports adaptive encryption in public cloud environment DBaaS where multiple clients are distributed over various geographical locations and can perform concurrent and independent operations on encrypted cloud database. It does not use any intermediate servers[5, 6] between the clients and the cloud database. It guarantees the same level of scalability and availability. All data and metadata stored in the cloud database are encrypted. In this system details of the user and what he wants to store on the cloud both stores in encrypted form. As per the requirement, the database designer will store user data using onion structure (adaptive encryption scheme[5]), because using normal encryption tenant are not able to perform all SQL operations.The adaptive encryption schemes which are already purposed for other application not for cloud environment, encrypts each plain column into one or more encrypted columns, and each value is encapsulated into different layers of encryption. The outer layers have higher confidentiality but it supports less computation capabilities with respect to the inner layers. In adaptive encryption scheme we have used deterministic encryption, order preserving encryption and random encryption. Below various adaptive encryption schemes are defined.

3.1 Adaptive Encryption Schemes

Adaptive encryption schemes[5] guarantee higher data confidentiality and allow the cloud database provider to execute SQL operations on encrypted data. Various encryption schemes listed under adaptive encryption also known as SQL query aware encryption. Here each algorithm supports a specific subset of SQL operators;

- Random (Rand): This is the very first encryption scheme in that it does not reveal any details about the plain data. It is the most secure encryption. It does not support any SQL operator and used only for data retrieval.
- Deterministic (Det): This is second encryption scheme which encrypts data deterministically. It supports the equality operator. Here equality of plaintext is preserved.
- Order Preserving Encryption (Ope): It is third encryption scheme which preserves the order of original unencrypted numerical data in the encrypted values. It supports the comparison operators i.e. =, <, ≤, >, ≥.
- Search (Search): It's forth scheme which supports equality check on full strings.
- Plain: It does not encrypt data; it stores plain data. So support all SQL operations.

To store data in database field confidentiality[4] is the major term to share key among tables or database. There are three field confidentiality which are explained below.

3.2 Field Confidentiality

Allows a user to define explicitly which columns of which table should share the encryption key if any. Those are listed below.

Column (COL) in this every column is encrypted using single encryption key which is not shared by other column. It is the default confidentiality that can be used when SQL statements executed on one column.

Multicolumn (MCOL) in these two or more columns is encrypted through the same key. It can be used for the column on which join operations or other operations can performed and where multiple columns are involved.

Database (DBC) here in this there is a single key is used to encrypt the database data. It means when overall database is involved then it is recommended.

The choice of the field confidentiality makes it possible to execute SQL statements over encrypted data. In our system we have used COL field confidentiality, where each column has a separate encryption key to encrypt and decrypt the column value.

In proposed system each plain column of the table is encrypted in one or more columns which depend on the

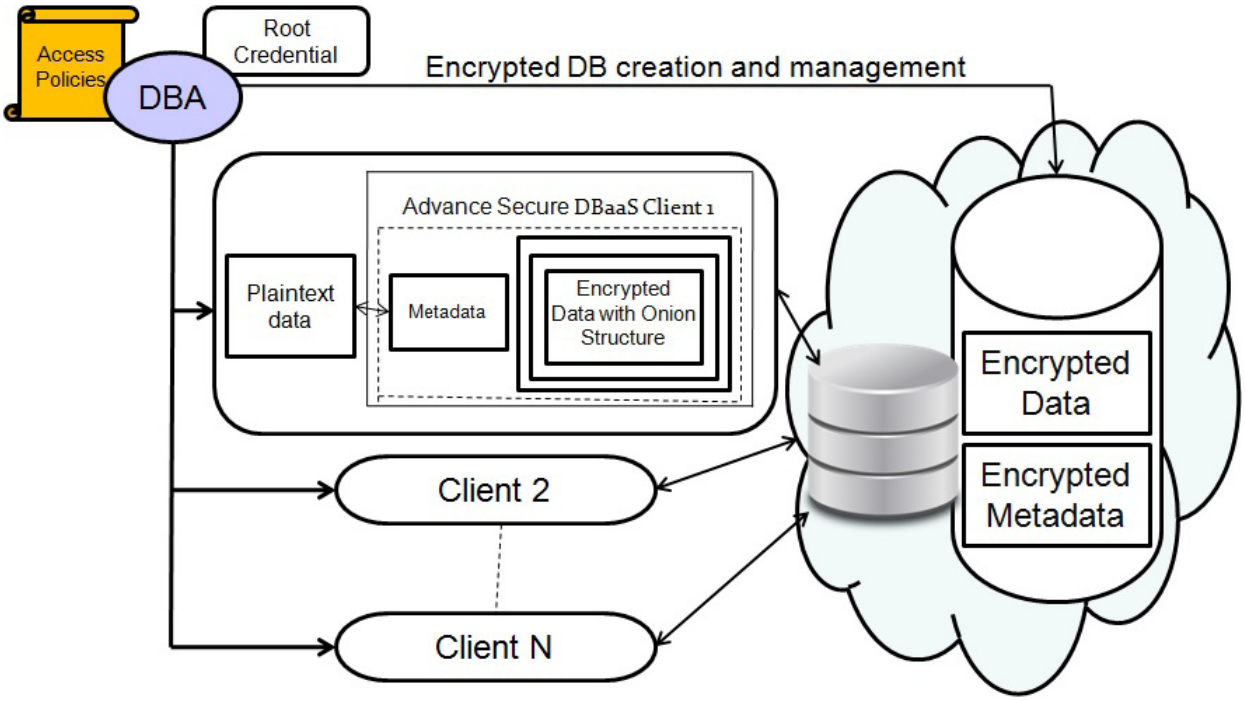

Figure 3. System Architecture.

need of database designer. It is encrypted into multiple columns then it will provide higher confidentiality. If numbers of encrypted layer are more then it is very difficult to perform all SQL operation over that column. In this client can issue SQL operations like Select, update, insert, delete to encrypted cloud database through an interface.Every user has assign privilege according to access policy he can perform SQL operation on the encrypted cloud database. Every user can store a file on the cloud in encrypted form. Same file can be access or download by users using same access policies.

The DBA shown in Figure 3 is the only subject that owns root credentials for the DBA client, and that no internal nor external attackers are able to access, steal or crack the credentials. The DBA manages user accounts, and enforces the tenant access control policies[7]. These policies represent the set of rules adopted by the tenant organization to define which user can access to which tenant data. The importance of data isolation through access control policies should be clear: the tenant users must access all and only authorized data where authorizations are specified as if the database was maintained by the tenant.

The innovation of the proposed models and schemes is to enforce access control mechanisms on cloud databases while allowing the execution of SQL operations on encrypted data stored in the cloud that are accessible by any cloud client. At the best of our knowledge, no existing proposal is able to satisfy both requirements. For example, there are encryption schemes that enforce access control mechanisms for cloud storage services[8], and other solutions that support concurrent accesses from independent clients[9]. Using query-aware encryption algorithms[5] allow a user to obtain all and only the requested data from the database, but that proposal is based on a trusted proxy that intercepts all operations between the tenant clients and the encrypted database, executes data re-encryption, and implements access control policies as in a privately managed infrastructure.

4. Implementation Details

Our implementation supports adaptive encryption methods[5] for public cloud database, where concurrent clients can issue SQL operations on encrypted database in cloud environment. This architecture maintains five types of information. i) Original plain data is the client information ii) Encrypted data which stored in the cloud database iii) Plain metadata stores details about encryption and

decryption necessary to execute SQL query on encrypted data iv) Encrypted metadata is encrypted version of plain metadata v)Master key is the encryption key of the metadata that is distributed to client.

4.1 Encrypted Database Management

In this section we describe the main operations which are involved in encrypted database management. It mainly consists of Database creation, SQL command implementation and Adaptive layer removal.

4.1.1 Database Creation

It is the initial set up phase where database administrator generates a special key known as master key which is used to initialize the overall architecture metadata. This master key is then distributed to legitimate client. At the time of table creation every column (attribute) of table is inserted as a new row of the metadata table. In table creation admin adds a column by entering the column name, data type and field confidentiality parameters[4]. In this work we have implemented Column field confidentially. Field confidentiality is most important for system because it include the set of onions to be associated with the column.

4.1.2 SQL Commands Implementation

When a user/client wants to execute any operation on the cloud database, then the user query is received by encryption engine. The encryption engine analyzes the SQL command structure and identifies tables, columns and SQL operators are involved in query. The client requests for the table metadata means encrypted keys[10] which are stored for each involved table, and decrypts the metadata with using master key which is already received from administrator. Then client checks whether the SQL operators are supported by the actual layers of the onions[5] associated with the involved columns. If it not supported then the client issues a request for layer removal in order to support the SQL operators at runtime using the information stored in the table metadata. Client can encrypt the parameters of the SQL operations: columns names, and constant values. The client request new statement called encrypted SQL operation to the cloud database to execute over encrypted data. The encrypted results are decrypted by using details stored in metadata table.

4.1.3 Adaptive Layer Removal

This section describe[5] the process of dynamically removal of external onion layer to guarantee adaptivity of an encrypted cloud database. Consider a example of table registration with columns name of type string and salary of type int , and a client issue a statement to the encrypted cloud database: SELECT * FROM Registration WHERE salary>=10000 and salary<=60000. Encryption engines analyses the received SQL statement and identifies that the operation salary>=10000 and salary<=60000 has to be performed on the encrypted database. Then it reads the metadata of the table registration and checks whether the Onion-Ord attribute associated to the column salary of the registration table because only Onion_Ord supporting the operator <, > If the uppermost layer of Onion-Ord associated to salary is set to Rand, then the client dynamically issues a stored procedure on the cloud database that removes Rand layer of Onion-Ord of the column salary, thus Ope layer exposed. The client can now encrypt the SELECT query and execute the operation (salary>=10000 and salary<=60000) on the Ope layer of Onion-Ord. As each layer has a different encryption key, the data remains encrypted and the cloud provider cannot access plaintext data. So it will original knowledge unexposed.

4.2 System Modules

We have implemented attribute level encryption with adaptive encryption scheme. Here each column of the database is encrypted with one or more encryption layer. In our system major two modules are used.

Client module consists of:

User registration: User will register themselves. All fields of user registration will be encrypted

Data Storage: Stores all types of files on the cloud database in the encrypted form. Every attribute of the file is encrypted.

Data Retrieval: Search file which is in encrypted form. It is decrypted at client side.

Update: Users update their profile like email, contact no.

Admin module consists of:

User Activation: User is activated by entering salary and accepting his request.

Executing SQL Operation: Executes SQL operations like select, delete, insert, update, range queries, aggre-

gate functions, group by and order by on encrypted data in the database.

User Deletion: Can delete user account.

6. Evaluation and Results

We have implemented our prototype by conducting experiments in a LAN, where each machine equipped with Intel Core-2 Duo 2.93 GHz CPU, 4GB RAM, Windows 7 Professional 32-bit Operating System. The LAN machines are connected with 100Mbps Ethernet network. The current prototype supports SQL operation Select, Insert, Update, Delete, Where, Order by, Min, Max, Count, and Group by using encryption schemes like Deterministic, Plain, Order Preserving and Random. This implementation is done using adaptive encryption scheme also called as Onion structure.

In existing system statically encrypted database is used where each column is encrypted at design time through only one encryption scheme with per record key. In our system columns are encrypted with the onions supported by its data type according to DBA requirement. Table1. Shows response time of SQL operations which are performed on Encrypted and Adaptively Encrypted Databases.

In Table 1 response time of both the prototype is measured. This response time has been measured at a particular time instance. Figure 4 shows graph of both systems response time. For Insert operation response time of both systems is good because it receives the keys, encrypt data and store. For update, select and delete operation exiting system performs poorly. Because first it has to receive every key one by one then encrypt record and match the data from table so it takes more time to perform operations. In proposed system all the keys are fetched which are assigned to every attribute of the table and encrypt data then compare so it becomes easy to perform select, update and delete. If number of records is more then also proposed prototype performs well as compare to existing system. For more number of records existing system doesn't work efficiently. It takes more time to execute SQL operation. Basically Adaptive Encryption prototype takes less time to execute various SQL operations. Because for every attribute there is separate key so it becomes very easy to find particular record from the database using key. Here we have implemented double encryption on two attributes i.e. Salary and Email id. Operations performed on this two attributes takes more time to execute as compare to other attributes in the table. For data storage (Upload File) and data retrieval (Download File) the performance of system is depends on the size of the file. If file size is large then both systems takes more time. This is depends on network speed, network delay. Below is the chart (Figure 4) which shows the difference between response times of both the prototypes. X-axis indicates time in millisecond and y-axis indicates various SQL operations.

Table 1. Response Time of Systems

Operation	Adaptive Response Time(ms)	Encryption Response Time(ms)
Insert	83	88
Select	149	312
Update	106	170
Delete	70	175
Upload(txt)	430	453
Download(txt)	180	140
Upload(multi.)	717	796
Download(multi.)	310	303

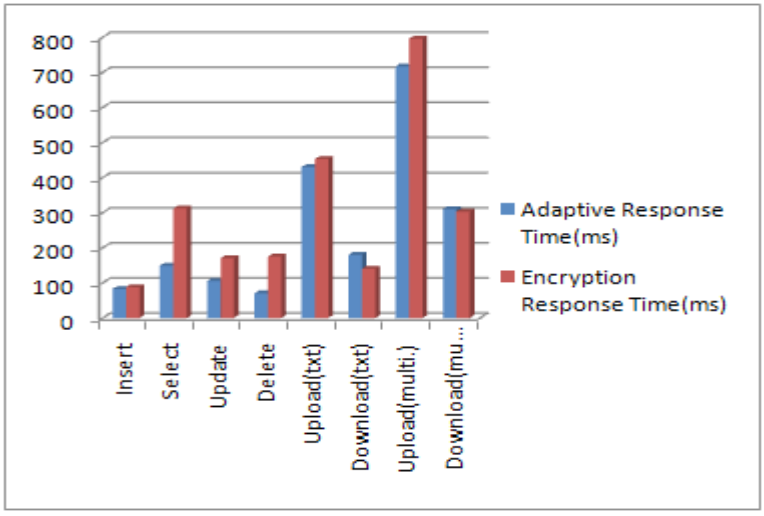

Figure 4. Response Time of Adaptive and Encryption.

7. Conclusion

In this work we have shown that legitimate cloud users can take advantage of DBaaS features and qualities like availability, accessibility, security, reliability and confidentiality while not exposing original knowledge to the cloud service provider. It permits multiple and regionally distributed clients to execute concurrent operations on encrypted data. It eliminates intermediate server between the tenant and cloud provider. Client registration details are stored in cloud database using adaptive encryption scheme. Clients are capable of reading and writing data on cloud database which is stored in encrypted form. The scheme in this work allows a client to encrypt all stored and transmitted data, to enforce standard database access control mechanism. It supports the execution of SQL DML command, range query, aggregate function, order by and group by clause on encrypted data stored in a public cloud environment.

8. References

1. Ashalatha R, Vaidehi M. The significance of data security in cloud: a survey on challenges and solutions on data security. 2012; 1(3):15–8.
2. Indu A, Gupta A. Cloud Databases: A Paradigm Shift in Databases. International J of Computer Science Issues. 2012; 9(4):77–83.
3. Agrawal D, Abbadi AE, Emekci F, Metwally A. Database Management as a Service: Challenges and Opportunities. Proc 25th IEEE Int'l Conf. Data Eng, 2009 Mar-Apr; 29-2:1709–16.
4. Ferretti L, Colajanni M, Marchetti M. Distributed, concurrent,and independent access to encrypted cloud databases. IEEE Trans Parallel Distrib Syst. 2014 Feb; 25(2):437–46.
5. Popa RA, Redfield CMS, Zeldovich N, Balakrishnan H. CryptDB: Protecting Confidentiality with Encrypted Query Processing. Proc 23rd ACM Symp Operating Systems Principles. 2011 Oct. p. 85–108.
6. Hacigumus H. , Iyer B, Li C, Mehrotra S. Executing SQL over Encrypted Data in the Database-Service-Provider Model. Proc ACM SIGMOD Int'l Conf Management Data. 2002 June; p. 216–27.
7. Ferretti L, Colajanni M, Marchetti M. Access control enforcement of query-aware encrypted cloud databases, in Proc. 5th IEEE Int Conf Cloud Comput Technol Sci. 2013 Dec. p. 717–22.
8. Yu S, Wang C, Ren R, Lou W. Achieving secure, scalable, and fine-grained data access control in cloud computing. Proc of the IEEE INFOCOM, 2010 Mar 14-19. p. 1–9.
9. Feldman AJ, Zeller WP, Freedman MJ, Felten EW. Sporc: group collaboration using untrusted cloud resources. Proc of the 9th USENIX conference on Operating Systems Design and Implementation. 2010 Oct. p. 1–14.
10. Mehrnoush Toghian, Matei Ciobanu Morogan. Suggesting a Method to Improve Encryption Key Management in Wireless Sensor Networks. Indian Journal of Science and Technology. 2015 Aug; 8(18): Doi:10.17485/ijst/2015/v8i19/75986