Integrating Encrypted Cloud Database Services using Query Processing

Jadhav Sonali S.
PG.Student
M B E Society's College of Engineering, Ambajogai, Maharashtra, India

B. M. Patil
Assistant Professor
PG Dept, M B E Society's College of Engineering, Ambajogai, Maharashtra, India

ABSTRACT

In today's environment the various vital information should need to be stored in more secured manner. In the cloud computing, original plain data must be accessible only by trusted parties that do not include internet and cloud providers or intermediaries. Storing this confidential information in cloud must provide guarantee of availability of data and security. there are too many solutions are provided to handle data, but still confidentiality problem is at risk. For that reason in this work proposed a new novel architecture SecureDBaaS which provides confidentiality and as well as allows concurrent execution of operations on encrypted data with distributed policy also. SecureDBaaS architecture retrieves the necessary information or metadata through SQL processing. This architecture has advantage that eliminates the intermediate server between client and cloud database also modifies the database structure. It guarantees for data confidentiality by performing SQL operations over encrypted cloud databases. This intended result of the proposed architecture is evaluated through comparison of AES n DES algorithm, where AES is better than DES is proved by studying comparison results.

Keywords

SecureDBaaS, cloud, security, DBaaS

1. INTRODUCTION

Cloud computing has been increased for providing services over the internet. These services must be potential to fulfill all the requirements of the customers. An organization acquires these cloud services for the model of computing, storage and model for communication over the internet. These services must provide scalability, availability and elasticity properties of databases. Cloud computing growing rapidly due to interest in recent years for handling large amount of data with its elasticity, flexibility properties. Whereas privacy and security policies monitors the organizations information system and standards, procedures and controls guidelines for preserving confidentiality ,integrity and availability of cloud database system. Organizations must need a particular security management services and control management over the cloud computing. For ensuring privacy and security requirements for cloud computing data should be in encrypted format.

There are too many security issues are available to protect clouds from outside threats are similar to those who already facing big data centers. In the cloud, however, this responsibility is divided among potentially many parties, including the cloud user, the cloud vendor, and any third-party vendors that users rely on for security-sensitive software or configurations. A cloud database is a one that typically executes on a cloud computing platform. For the storage and management of structured data database as a service of cloud based approach is used. On the other hand in cloud based approach DBaaS is an oriented toward self-service and easy management provides a flexible, scalable and on demand platform. Users may purchase access to a database service maintained by a cloud database provider.

In a DBaaS application owners do not need maintain and install the database instead the service provider takes care of the responsibility for maintaining and installing database and they pay according to their use All the data needs to be providing data confidentiality because database containing highly valuable information. Users wishing to protect their sensitive information among untrusted proxies or third parties; so plaintext data should be visible only to trusted parties excluding cloud provider and internet. Whereas in untrusted context data must be in encrypted format for preventing unauthorized access to sensitive information. Here proposed an architecture named as a SecureDBaaS for allowing multiple, independent and distributed users to perform simultaneous operations on encrypted data. Reason behind of proposing SecureDBaaS is to allow concurrent and independent execution of operation on encrypted data through SQL statements with database qualities such as elasticity, scalability and ease of availability. The main focus is using SQL statements to modify the database structure. In earlier because of intermediate proxies, sometimes failure occurs that results in bottleneck that limits the qualities of database service. So intermediate proxies are eliminated to increase the data confidentiality level of database system. This architecture is solution for geographically distributed users who want to access its database concurrently and independently.

SecureDBaaS uses a various isolation techniques, cryptographic techniques for managing encrypted metadata on the untrusted cloud databases. To use SecureDBaaS architecture here it should achieve also the reliability and availability and elasticity properties of cloud DBaaS. The best quality of the SecureDBaaS is that it is immediately applicable to any DBMS because it does not require changes to the cloud database services. Here SecureDBaaS is a new architecture is designed where concurrent and independent n distributed clients can access the information through cloud database by sql statements. In a new approach in this work two types of encryptions are defined, one is DES and another is AES. by comparison determined result is AES is better than DES in all way like security, availability and scalability. The overall conclusion of paper is very crucial because first time it demonstrates the availability and applicability of encryption cloud DBaaS with respect to performance and overheads.

2. LITERATURE SURVEY

Unlimited numbers of computing resources are available on demand, quickly for the cloud computing users. One main aspect in the cloud service is the able to pay for use of computing resources on a short-time basis as needed and release them as it is no longer needed. An internet service provider provides the cloud services where quality of service is dependent on the cost, so a customer doesn't jump on lowest cost service. Security is the most important objections to cloud computing. Many of the security issues are evolved during protecting clouds from cloud users, vendors and third-party vendors. most of the security concerns is to protect cloud from cloud provider. Various techniques like the standard defense and user level encryption are effective in the cloud [1].Continuous checking of secured information requires maintaining awareness of threats ,security controls and vulnerabilities to support risk management. For fulfilling the operation of monitoring the organization is dependent on cloud provider. An analysis of system security controls and security features are used for vulnerabilities identification to protect cloud environment. Cloud computing is emerging technology as technology advances there need to improve performance and other quality services from public cloud and including privacy and security [2].DBaaS provides various features related to security and database services. Working in this field varies with time as well as new techniques to improve the performance of remote database services.

Data confidentiality is the important aspect of cloud database services and concurrent/simultaneous execution of operations with distributed manner. Also new techniques arrived which allows distributed access to database with platform independent properly. SecureDBaaS provides data confidentiality by executing sql operations on encrypted data which allows concurrent read, write and modification to the database structure. It maintains databases properly elasticity, scalability and availability of cloud database because it does not need any intermediate server. It always removed a trusted proxy because tenant and metadata are always in encrypted format. Its use for relational database which are very applicable to different database management system implementation.[3].A number of group of trusted clients outsources an arbitrary computational service to a remote provider, which they do not fully trust and that may be cause to attacks. Here presented a novel protocol that guarantees atomic operations to all clients when the provider is correct and fork-linearizable semantics when it is faulty; this means that all clients which observe each other's operations are consistent, in the sense that their own operations, plus those operations whose effects they see, have occurred atomically in same sequence.. This protocol generalized previous approaches that provided guarantees for outsourced storage services [4].This article defines the design, implementation, and evaluation of Depot, a cloud storage system that minimizes assumptions of trust. Depot tolerates malicious behavior by number of clients or servers, yet it provides safety of guarantees to correct clients. It provides the guarantees by using two-layer architecture. Second, Depot implements some protocols that uses this consistent of updates to provide consistency, staleness, durability, and recovery properties. Here our evaluation suggests that the costs of these guarantees are modest and that Depot may tolerate faults and maintain good availability, latency, overhead, and staleness even when significant faults occur [5]In this cloud environment an inserting secure important information in untrusted third parties, which causes risks of the confidentiality of data. Where it takes care of guarantees confidentiality of the Database as a service (DBaaS) which remains a problem. Therefore to resolve the Confidential and Concurrent to SecureDBaaS is determined because there is initial resolution to produce availability, accessibility, reliability and security which not exposing unencrypted information to the cloud provider. It additionally permits multiple, freelance and regionally distributed clients to execute synchronic operations on encrypted and preserve information confidentiality at the consumer and cloud level. It removes intermediate server between the cloud consumer and the cloud provider. To realize that Confidential concurrent to Secure DBaaS integrates existing cryptographic schemes, isolation mechanisms and management of encrypted information on the untrusted cloud information.[6].here proposed a fully homomorphism encryption scheme - i.e., a scheme that allows one to evaluate circuits over encrypted data without being able to decrypt. So solution comes in three steps.

First, construct an encryption scheme which permits evaluation of arbitrary circuits , that suffices to construct an encryption scheme that may evaluate its own decryption circuit ;in this work called a scheme that can evaluate its (augmented) decryption circuit boots trappable. Next, here given description of a public key encryption scheme using ideal lattices that is almost boots trappable. Lattice-based cryptosystems specifically have decryption algorithms with low complexity, often dominated by an inner product computation. Also, ideal lattices provide both additive and multiplicative homeomorphisms, as needed to evaluate general circuits. There by obtain a boots trappable encryption scheme, without reducing the depth that the scheme can evaluate. Here accomplished this by enabling the encrypted to start the decryption process, leaving less work for the decrypted, much like the server leaves less work for the decrypted in a cryptosystem [7]. The technological aspects of developing database as a service lead to new research challenges. The service provider always would need to provide sufficient security measures to protect the privacy of data. Here proposed data encryption as the solution to this problem. Second key challenge is that of performance. Since the interaction between the database service provider and users takes place in a different medium, the network, than it does in traditional databases, there are potential overheads introduced by this architecture. Data privacy can be achieved by using a suitable encryption algorithm. here proposed, implemented, and evaluated different encryption schemes [6]

Here in this work proposed a fully homomorphism encryption scheme – i.e., a scheme that allows one to evaluate circuits over encrypted data without being able to decrypt. so solution comes in three steps. First, here provided a general result – that, to construct an encryption scheme that permits evaluation of arbitrary circuits , it suffices to construct an encryption scheme that can evaluate (slightly augmented versions of) its own decryption circuit ; in this work called a scheme that can evaluate its (augmented) decryption circuit boots trappable . Here determined a public key encryption using ideal lattices it is almost boots trappable. Lattice-based cryptosystems have decryption algorithms with low circuit complexity, which is most of dominated by an inner product computation that is in NC1.Unfortunately, so initial scheme is not quite boots trappable – i.e., the depth that the scheme can correctly evaluate can be logarithmic in the lattice dimension, just like the depth of the decryption circuit, but the latter is greater than the former. In the final step, have shown that how to modify the scheme to reduce the depth of the decryption circuit, and thereby obtain a boots trappable encryption scheme, without reducing the depth. Abstractly, it accomplished by enabling the encrypted to start the process of the decryption process and leaving less work for the decrypted, much like the server leaves most of less work for the decrypted cryptosystem [7].

CryptDB executes SQL queries over encrypted database over sql aware-encryption schemes. One solution to overcome the damage done by server comprises is to encrypt sensitive secure data as in sundr, Sporc &depot and run all computations on clients. Another approach is to use homomorphic encryption in which server performs arbitrary functions on encrypted data where only clients can see decrypted data. Earlier approach for performing computations on encrypted data are either too slow or do not provide adequate confidentiality. On another hand, encryption of data with efficient and strong cryptosystem, such as AES would prevent the DBMS server from executing many sql queries. Goal of cryptDB is to provide data confidentiality by execution of sql queries on encrypted data. So new approach for DBMS server is execution of query processing on encrypted databases as it would on an unencrypted database by enabling it to compute certain functions over the data item based on encrypted data. The DBMS server returns (encrypted) query results, which the proxy decrypts and returns to the application. A proxy server performs operation between DBMS server and application server. SUNDR uses cryptography to provide privacy and integrity in a file system on top of an untrusted file server. Using a SUNDR-like model, SPORC and Depot show how to build low-latency applications, running mostly on the clients, without having to trust a server. However, existing server-side applications that involve separate database and application servers cannot be used with these systems unless they are rewritten as distributed client-side applications to work with SPORC or Depot. Many applications are not amenable to such a structure. CryptDB provides confidentiality guarantees for user data even if the adversary gains complete control over the application and database servers. [8].

In untrusted DBMSs data confidentiality is preserved through various encryption techniques, which performs sql operations on encrypted data which is compatible to common DBMS engines. These solutions based on trusted and intermediate proxy that interacts between each client and untrusted DBMS server. If any modification occurs at sql operation by its client then significant overhead causes on both server and proxy. [10][11] Introduces generalization & optimization that increases a subset of sql operators,[8][9] applicable to multitier web application, but they causes some drawbacks. So the proxy is trusted and its function is not being provided to an untrusted cloud service provider. Hence the implementation and management of proxy can be done by cloud tenant. Fast growing in internet and networking technologies have evolved software as a service for enterprise computing. DBaaS provides a services like create, delete, store, modify and retrieve data from anywhere in the world by accessing the internet. it specifies issues which related to data privacy. There are two main issues; in which first on e is data owner must ensure that data must be protected from outsiders' thefts. Second one is secure data must be protected from the service provider also. Here main focus is on another challenge is that develop techniques to execute sql queries over encrypted data. So goal is to process the query as possible at the site of service provider without decrypting data. [9].

Here concerns about securing sensitive information of data and queries from adversaries in the Database as a service model. Queries and data need to be encrypted, where the database service provider should be able to efficiently provide answers queries based on encrypted [10].In DBaaS with the presence of untrusted service providers, clients store their database at servers. To provide data confidentiality, clients must store their data to server in encrypted format. At the same time clients still need to execute a sql queries over encrypted data.One of well known technology in cryptography is homomorphic encryption in which arithmetic operations are performed on encrypted data [11].In the database as a service, storing data with third party resulted in concern with data privacy. Services provided on data encryption can cause large overhead in query processing [13]. Here new approach is defined that includes division of data into no of pieces where it can easily reconsrtuctable .This scheme enables the robust key construction key management for cryptographic system [14].Oracle database provides data redaction and encryption capabilities to protect sensitive application data. Transparent data encryption prevents unauthorized access to sensitive information at the application layer in the operating system. It also supports multitenant option with unparalleled performance.TDE secure sensitive data against unauthorized access from outside of database system by encryption. It directly inspects the elements of database.TDE also provide security against theft, loss and improper decommissioning of database storage files and backups [16], [17].Traditional progress in computing environment becomes more protective in providing secrecy of information. Here preventing access from unauthorized users to outsourced data is provided by encryption[18].in today's computing environment in cloud databases main concern is about data confidentiality but as well as there is also need of high availability n scalability. Existing methodology proposed number of trusted intermediate servers, but limited in scalability and availability of database servers. Here proposed an alternative solution for that is avoiding intermediary component, moreover this guarantees that data consistency where independent clients can execute concurrently sql queries and modifications in structure database [19].TPC is corporate to determine transaction processing benchmarks to provide objective, verifiable performance data to companies [20].The properties like elasticity, availability and scalability of trusted proxy becomes the reason of system bottleneck. By adapting cloud services properties of databases are more applicable to SecureDBaaS without connections to any intermediate proxies clients directly connected to cloud database. By using intermediate server it is not suitable to cloud based phase in the proxy based approach, where multiple clients in distributed locations need an access to data concurrently in DBMS. On another hand SecureDBaaS supports execution of concurrent and independent sql operation on databases. SecureDBaaS is also managed in a way that where multiple clients may access cloud databases without the need of intermediate servers.securedbaas clients may directly connect to the cloud DBaaS and perform operations like read, write and modify [21].

3. ARCHITECTURE

SecureDBaaS is managed in a way that where multiple clients can access cloud databases without the need of intermediate servers.securedbaas clients can directly connect to the cloud DBaaS and perform operations like read, write and modify. As shown in Fig.1 the information managed by SecureDBaaS includes plaintext information, encrypted information, metadata, and encrypted data. Plaintext information include data that a tenant needs to store and method remotely within the cloud DBaaS. To prevent associate untrusted cloud supplier from violating confidentiality of tenant information hold on in plain type, SecureDBaaS adopts multiple science techniques to remodel plaintext information into encrypted tenant information and encrypted tenant data structures as a result of even the names of the tables and of their columns should be encrypted. SecureDBaaS shoppers produce additionally a group of data consisting of knowledge required to cipher and decipher information also as different administration data. Even data are encrypted and hold on within the cloud DBaaS. SecureDBaaS moves off from existing architectures that store simply tenant information within the cloud info, and save metadata within the consumer machine [9] or split databetween the cloud info and a sure proxy [8]. When considering eventualities wherever multiple purchasers will access the same info at the same time, these previous solutions area unit quite inefficient. as an example, saving data on the clients would need heavy mechanisms for data synchronization, and therefore the sensible impossibility of permitting multiple purchasers to access cloud info services severally. Solutions supported a sure proxy area unit additional feasible, however they introduce a system bottleneck that reduces handiness, elasticity, and measurability of cloud database services.

SecureDBaaS proposes a distinct approach wherever all data and data square measure hold on within the cloud information base. SecureDBaaS shoppers will retrieve the required data from the untrusted information through SQL statements, so that multiple instances of the SecureDBaaS shopper will access to the untrusted cloud information severally with the guarantee of an equivalent availableness and quantifiability properties of typical cloud DBaaS.

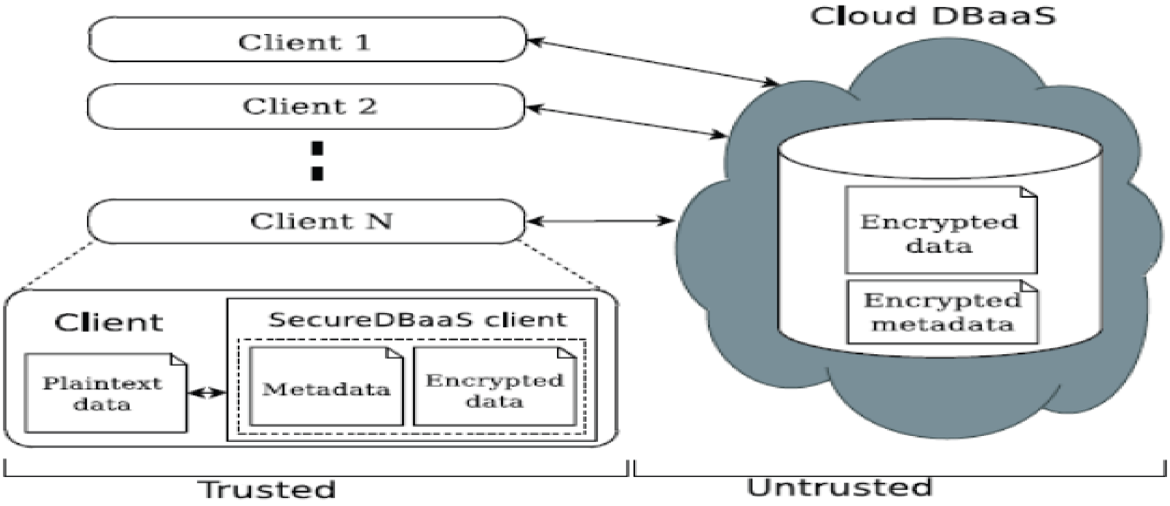

Fig.1.SecureDBaaS Architecture

3.1 DES

DES was designed with key length of 56 bits, which is vulnerable to exhaustive search. DES is a block cipher, where it has a 64-bit block size and a 56-bit key. A DES consists of 16 rounds of substitution and permutations. In each round, first data and key bits are shifted, permutated, XORed, and sent through, 8s-boxes, a set of lookup tables that are essential to the DES algorithm. Decryption is same as a process, which performed in reverse. To decrypt, first a algorithm is used, then needed to reverse that order of subkeys where The L and R blocks are 32 bits each, which yields a block size of 64 bits. The "S-boxes defines the hash function "f" that takes a 32-bit data block and one of the 48-bit sub keys as input and produces 32 bits of output. A DES is a 64-bit key in size, but out of the 64 bits only 8 bits are used only for parity checking, so the effective key size is 56 bits.

3.2 AES

It is 128-bit blocks and it accepts keys of size 128, 192 and 256 bits. It has no academic weakness worse than exhaustive key search. It should be as fast as 3DES. To provide more security to AES, it uses Substitution permutation, mixing and key adding each round type of transformation of AES except the last uses the four transformations. The resistance of AES towards differential and linear cryptanalysis comes from a better "avalanche effect" (a bit flip at some point quickly propagates to the complete internal state) and specially crafted, bigger "S-boxes" (a *S-box* is a small lookup table used within the algorithm, and is an easy way to add non-linearity; in DES, S-boxes have 6-bit inputs and 4-bit outputs; in AES, S-boxes have 8-bit inputs and 8-bit outputs).

AES is always a better encryption standard than other. i.e. it is more advanced as compared to DES. the encryption key of a DES is 56 bits and having a maximum of 256 combinations, while key size of AES is 128, 192 or 259 bits long, while each containing 2128, 2192 and 2256 combinations, thus it is difficult to crack. Secondly, the block size of DES is 64 bits long. Finally, before going to the encryption steps DES uses the Feistel network where it divides the block into two halves. on the other hand, AES uses permutation-substitution, which contains a series of substitution and permutation steps to create the encrypted block, so making it difficult to break the code.

3.3 Management of data and metadata

Management of data in the manner to secure stored data as well as structure of database which includes columns name is also encrypted. Here each plaintext tables are transformed into more secure table because of untrusted cloud database. These secure tables include data type which is also stored in encrypted format. SecureDBaaS generate metadata which consists of all the information which is needed for management of sql statements over encrypted database.

4. EXPERIMENTAL RESULTS

Here in this work demonstrated the applicability of SecureDBaaS to different cloud DBaaS solutions by handling and implementing encrypted database operations on real and emulated cloud infrastructures. The present version of the SecureDBaaS prototype supports PostgreSQL, MySql, and SQL Server relational databases. a user interacts with the cloud database through the SecureDBaaS client.

SecureDBaaS analyzes the original operation to identify which tables are involved and to retrieve their metadata from the cloud database. For accessing cloud database it needs a cloud services to upload the secure information as well as access that information through cloud service. In this work jelastic layershift cloud hosting services is used..

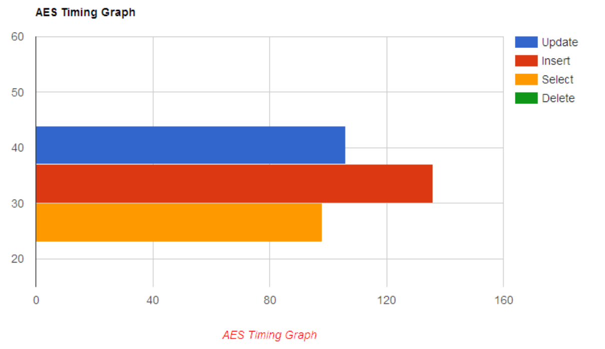

Fig.2. AES Timing Graph

It is a platform specific as a service designed for hosting purpose to deploy and make easily available to its users. It is unique and it doesn’t need any code modifications requirements. Phpmyadmin is a free and open source software tool used to handle at the administration of MySql using web browser. It is developed for creating, deleting, modifying tables, fields and database as well as executing sql statements also manages user’s access permissions

Table1.Execution Timing of queries in AES

Types of queries	Execution time in ms
Update	105ms
Insert	138ms
Select	98ms
Delete	0ms

The performance of encrypted sql operations are evaluated through measurement of response times of queries. Here sql queries respectively update, insert, select and delete are executed and their respective response times are calculated. By observing this can determine that insert query requires maximum time to execute because of it needs to encrypt all columns.

Here Fig.2 shows the required execution time of select, update, insert and delete sql queries has given for AES encryption.

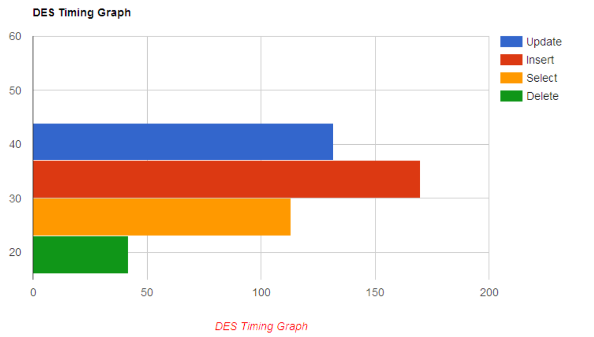

Fig.3. DES Timing Graph

Table2. Execution Timing of queries in DES

Types of queries	Execution time in ms
Update	131ms
Insert	170ms
Select	112ms
Delete	42ms

As shown in Fig.3. Response time of update, insert, select and delete command has given. This can see that DES response time is maximum as compare to response time of given queries. Here in this work compared the response times of sql queries based on encryption policy. These comparison defines the response time of encrypted queries in AES requires minimum time as compared to response time of encrypted queries in DES so determined result is AES is more secure n gives high performance. So concluded comparative results are seen by these graphs, AES is more secure than DES in all the way of encryption and maintaining confidentiality.

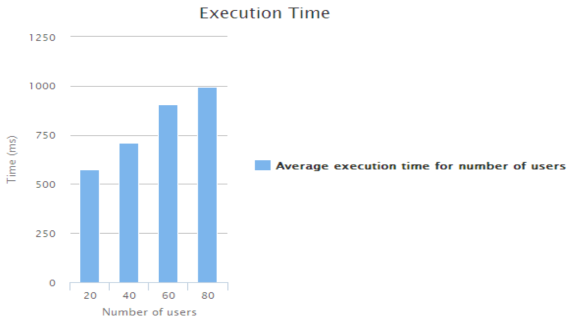

Fig.4. Network latency

Another experiment is oriented to determine the impact of concurrency and network latency from geographically distributed clients on the cloud databases. Here in this set values of 20 to 80 users and their execution time has given n finally calculated their latency. Overall result shows that as a number of users increases their respective execution time is also increases.

Fig.4. shows the average execution time for number of users. Where increasing number of concurrent clients can increases the delay in their execution time. As per experimental results this can see the latency of number of users at a given time. This result is important because it confirms that SecureDBaaS is a valid and practical solution for guaranteeing data confidentiality in real cloud database services.

5. CONCLUSION

Here proposed a novel architecture that provides confidentiality of data stored in public cloud databases. In this work so solution does not depends on an intermediate proxy that while it cause bottleneck and single point of failure which limiting scalability and availability of cloud database services .A large content of the research involves solutions to support concurrent SQL operations over encrypted data issued by heterogeneous and possibly geographically dispersed clients. The proposed architecture does not need modifications to the cloud database, and it is immediately applicable to existing cloud DBaaS. There are no practical and theoretical limits to extend so solution to another platform and to include new algorithms of encryption. In particular, here specified that concurrent read and write operations do not need to modify the structure of the encrypted database which causes 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 in this work are investigating. Here in this work compared the response times of sql queries based on encryption policy. Particularly these comparison defines the response time of encrypted queries in AES requires minimum time as compared to response time of encrypted queries in DES so determined result is AES is more secure and gives high performance. So concluded comparative results are seen by these graphs, AES is more secure than DES in all the way of encryption and maintaining confidentiality.

6. 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/Aug2006.

[12] 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.

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

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

[15] 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.

[16] E. Damiani, S.D.C. Vimercati, S. Jajodia, S. Paraboschi, and P.Samarati, “Balancing Confidentiality and Efficiency in Untrusted Relational Dbmss,” Proc. Tenth ACM Conf. Computer and Comm.Security, Oct. 2003.

[17] 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.

[18] “Transaction Processing Performance Council,” TPC-C, http://www.tpc.org,Apr.2013.

[19] Xeround: The Cloud Database,” Xeround, http://xeround.com,Apr. 2013.

[20] “Postgres Plus Cloud Database,” EnterpriseDB, http://enterprisedb.com/cloud-database, Apr. 2013.

[21] 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