ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Implementing Secure Database as a Service (SDBaaS)

Arpita R. Malpe[1], Prof P. A. Tijare[2], Prof G.S.Thakare[3]

M.E.2nd Year, Department of Information Technology, SIPNA COET, Amravati, India[1]

Associate Professor, Department of Information Technology, SIPNA COET, Amravati, India[2]

Assistant Professor, Department of Information Technology, SIPNA COET, Amravati, India[3]

ABSTRACT: Current cloud computing systems create serious limitation to protective users' information confidentiality. Since users' sensitive information is conferred in unencrypted forms to remote machines in hand and operated by third party service providers, the risks of unauthorized revealing of the users' sensitive data by service providers could also be quite high. There how to defend the confidentiality of users' information from service providers, and ensures service providers cannot collect users' confidential data whereas the data is processed and hold on in cloud computing systems. It integrates cloud database offerings with data confidentiality. Within the Secure Database-As-a-Service (SDBaaS) model, clients save their database contents at servers possessing to doubtlessly untrusted service providers. To stay data confidentiality, clients stores their data to servers in encrypted form. At the same time, clients got to still be capable of execute queries over encrypted data.

KEYWORDS: Cloud, confidentiality, Secure DBaaS, database.

I. INTRODUCTION

Cloud computing is a form of internet-based fully research that offers shared computer running resources and data to pc methods and others, their, the units on contact for. It's a type for allowing common, on-demand get access to a discussed share of configurable processing methods (e.g., laptop, hosts, storage, offers and services), which may be rapidly provisioned and presented with minimal administration effort. Cloud processing and storage provide consumers and enterprises with varied abilities to search and program their data in alternative party data services that may be based a substantial ways from the consumer–ranging in range from across a city to the planet over. Cloud processing utilizes discussing of options to get coherence and economy of scale.

Current cloud processing structures present significant concern to defensive user's data confidentiality. Because that user sensitive document emerges in unencrypted bureaucracy to rural models possessed and run by way of third party service providers, the dangers of unauthorized disclosure of the user's sensitive details by providers might be quite high. There are numerous methods for defending user' data from outside opponents, but presently number strong way is available for protecting users' sensitive data from company in cloud computing.

Based on SaaS, DBaaS actions database management system (DBMS) from a conventional client-server structure whereby the data owner is chargeable for coping with DBMS and responding to user's queries to a third party service provider whereby data administration isn't handled with the support of the reality owner. Data owner outsource their information to data service companies. Database as a service (DBaaS) gives a considerable array of blessings including data outsourcing, multi-tenancy, and helpful resource sharing and so on.

The Secure DBaaS structure is personalized to cloud systems and doesn't put any intermediary proxy or broker machine involving the customer and the cloud support provider. Throwing down any depended on advanced machine enables in Secure DBaaS to reap exactly the same accessibility, stability, and strength levels of a cloud DBaaS. Secure DBaaS that assists the delivery of concurrent and separate procedures to the distant protected repository from several geographically designated customers as in nearly any unencrypted DBaaS setup.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

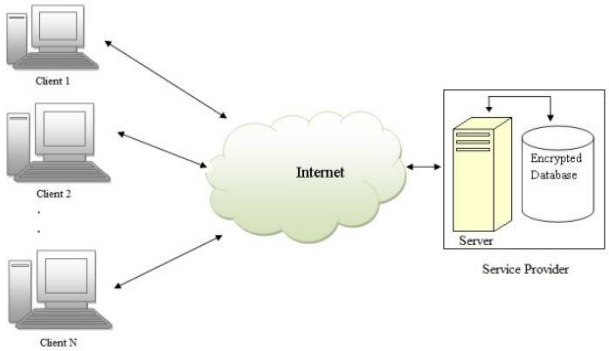

Figure 1.1: Secure Database as a Service Overview

As described in figure 1.2 for cloud computing the clients are related to the cloud through the internet .On the cloud aspect there may be a cloud provider issuer to which the server is related to the encrypted database via using the encryption method. Customers can store their records in database.

II. .RELATED WORK

As in Luca Ferretti, Michele Colajanni, and Mirco Marchetti [1] describes an structure that degrades the risk for almost any advanced part, by reaching availability and scalability by unencrypted cloud database services. Advantages are assures file consistency in scenarios throughout which independent clients simultaneously performs on SQL requests, and the structure from the database may be modified. Decreased isolation leads to the amounts for multiple variation system have not been characterized before notwithstanding being executed in different products and services and the disadvantage are concurrent improvements from the database structure are generally supported but at the cost of larger overhead in addition to stricter exchange remote location level.

M. Armbrust et al [2], has produced with revolutionary ideas for new Internet services no longer require the considerable outlays in hardware to deploy their company or the human expense to use it. As cloud computing grows, designers must take it into account. Moreover: 1. Applications Computer software needs to equally scale down rapidly along with scale up, which is a new requirement. Such application also needs a pay-for-use licensing design to match that is requirement of Cloud Computing. 2. Infrastructure software wants to keep aware that it is no longer operating on simple metal but on VMs. Moreover, billing wants to create in from the start. 3. Hardware system must be developed at the scale of a container (at least a dozen racks), that will be may be the minimum buy size.

Wayne Jansen and Timothy Grance [3] describes the change to an outsourced, public cloud processing environment is in lots of ways a exercise in risk management. Risk management entails identifying and assessing risk, and taking steps to cut back it to an acceptable level. Assessing and managing risk in cloud computing system requires constant checking of the security state of the system, and may prove challenging, because substantial amounts of the processing environment are underneath the control of the cloud providers and likely beyond the organization's purview. Throughout the system lifecycle, dangers which can be discovered should be cautiously balanced contrary to the security and privacy controls accessible and the estimated advantages from their utilization. Way too many controls can be inefficient and ineffective.

A.J. Feldman, William P. Zeller, Michael J. Freedman, and Edward W. Felten [4] has proposed SPORC is a simple framework for developing a wide selection of collaborative application with untrusted servers. In SPORC, a server observes just encrypted knowledge and can not deviate from right execution without being detected. It enables concurrent, low-latency editing of shared state, enables disconnected operation, and helps dynamic access control on even yet in the presence of concurrency. SPORC's flexibility through two model applications: a causally-consistent key-value store and a browser-based collaborative text editor. SPORC shows the complementary great things about operational transformation (OT) and fork consistency. The former enables SPORC clients to perform concurrent operations without locking and to solve any resulting issues automatically. The latter prevents a misbehaving server from equivocating in regards to obtain of operations unless it is consenting to fork clients into disjoint sets.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Significantly, unlike prior systems, SPORC may instantly cure such harmful forks by leveraging OT's conflict resolution mechanism. Operational Transformation (OT) describes a framework for executing lock-free concurrent operations that equally keeps causal consistency and converges to a common shared state. It does therefore by transforming operations so they can be used commutatively by distinct clients, resulting in exactly the same final state. While OT begun with decentralized applications applying pair wise reconciliation.

P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish [5] have described in a model Depot where cloud storage system minimizes trust assumptions. Thus it put up with buggy or malicious behaves by many clients or servers, yet it provides protection and liveness guarantees to correct clients. They also provide these guarantees using a two-layer architecture. First, Depot guarantees that the changes observed by appropriate nodes are consistently ordered under Fork-Join- Causal consistency (FJC). FJC is a small weakening of causal consistency which can be equally secure and stay despite faulty nodes. Second, Depot uses protocols that make use of this consistent ordering of update to implements other appealing reliability, staleness, durability, and recovery properties.

As in H. Hacigu¨mu¨ s¸, B. Iyer, and S. Mehrotra [6] NetDB2, a database design on-line provides a useful mechanism for organizations to obtain data management such as a service. Database as an e-mail finder support makes the advantages of additional overhead of much gets right of access to data, an infrastructure to assure records privacy, and program layout. Records privateers may be achieved in distinctive stages via utilizing security methods with equally computer software and hardware levels. NetDB2 variation also functions create/eliminate records, views, triggers, perform data types, SQL queries, produce and call customer defined functions and located methods, providing and deleting indexing, and so forth. A few DBMS engine gives the chance of encrypting knowledge at the remote system level through the alleged translucent data protection feature. That function causes it to be sensitive to gather a trusted DBMS over untrusted cloud. The DBMS is relied on and decrypts knowledge sooner than their use.

R. A. Popa, Catherine M. S. Redfield, N. Zeldovich, and H. Balakrishnan [7] in CryptDB is a system that assigns a designing in which proposes confidentiality for application using Database Management System (DBMSes). CryptDB leverages the typical structure of database-backed applications, consisting of a DBMS server and a different application server the latter works the application code and issues DBMS queries on behalf of more than one user. It's approach would be to accomplish queries over encrypted data, and the important thing understanding that makes it practical is that SQL works on the well-defined set of operators, each of which are able to support efficiently over encrypted data. CryptDB handles two threats. The first risk is really a curious Database Administrator (DBA) who attempts to learn private data (e.g., health records, financial statements, personal information) by snooping on the DBMS server; here, CryptDB stops the DBA from learning individual data. The second risk can be an adversary that gains complete control of applications and DBMS servers. In that case, CryptDB can't offer any guarantees for users that are signed into the application all through an attack, but can still assure the confidentiality of logged-out users data. Online applications are at risk of theft of sensitive data because adversaries can use software bugs to get access to private data, and because curious or malicious administrators might capture and leak data. CryptDB is really a process that provides practical and provable confidentiality in the face area of these attacks for applications backed by SQL databases. It functions executing SQL queries over encrypted data using a collection of efficient SQL-aware encryption schemes. CryptDB may also chain encryption keys to user passwords, therefore that the data item could be decrypted only by using the password of among the users with access to that data. As a result, a database administrator never gets access to decrypted data, and even when all servers are compromised, an adversary cannot decrypt the information of any user who is perhaps not signed in.

Hakan Hacıg¨um¨us¸, Bala Iyer, Chen Li, Sharad Mehrotra [8] has described a quick innovations in networking and Internet technologies have fueled the emergence of the "software as a service" model for enterprise computing. Successful types of commercially practical software solutions contain rent-a-spreadsheet, digital send solutions, general storage solutions, disaster protection services. "Database as a Service" model provides clients power to create, store, modify, and retrieve data from any place in the world, provided that they've access to the Internet. It introduces several problems, an essential concern being data privacy. It's in this situation that people particularly handle the problem of data privacy. There are two principal solitude issues. First, the owner of the data must rest assured that the data saved on the service-provider site is protected against data thefts from outsiders. Second, data must be protected even from the service providers, if the providers themselves cannot be trusted. It focus on the second challenge. It also performs

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

SQL queries over protected data. To process the maximum amount of the query at the service provider's site, and never having to decrypt the data. Decryption and the remaining of the query control are done at the client site.

Jun Li and Edward R. Omiecinski [9] proposed the database-as-a-service (DAS) model is a recently emerging computing paradigm, where in actuality the DBMS operations are outsourced. It is desirable to store data on database servers in secured form to reduce security and privacy dangers since the server may not be fully trusted. But that frequently suggests this one has to lose efficiency and effectiveness for security. These strategies change from one another in how a list of feature prices is created. Random one-to-one mapping and order-preserving are two examples. A prefix-preserving security system to generate the index. Certainly, it locates a easy trade-off between effectiveness and security. The security issues and effectiveness of the strategies for promoting selection queries on secured numeric data are implemented.

E. Mykletun and G. Tsudik [10] as proposed in the paper Database-As-a-Service (DAS) model where clients keeps their database contents in servers residing to untrusted service providers. To preserve data confidentiality, clients have to outsource their data to servers in encrypted form. Simultaneously, clients must however manage to perform queries over encrypted data. One prominent and fairly effective method for executing SQL-style range queries over encrypted data involves partitioning (or bucketization) of encrypted attributes. One cryptographic tools that applies to support encrypted aggregation is homomorphic encryption; it assist arithmetic operations on encrypted data. One technique predicated on a specific homomorphic encryption function was recently planned in the context of the DAS model.

Vignesh Ganapathy, Dilys Thomas, Tomas Feder, Hector Garcia-Molina, Rajeev Motwani [11] proposes introducing in database solutions has arise privacy concerns for the clients storing data with third party database service providers. A distributed architecture for secure database services is proposed as a solution to this problem wherever data was stored at numerous sites. The distributed architecture offers privacy as well as fault tolerance to the client. It works on two algorithms for (1) distributing data: our results contain hardness of approximation results and hence a heuristic greedy hill climbing algorithm for the distribution problem (2) partitioning the query at the customer to queries for the many sites is performed with a bottom up state based algorithm. Finally the results at the sites are incorporated to obtain the obvious answer at the client.

Mohammad Ali Hadavi1, Ernesto Damiani2, Rasool Jalili1, Stelvio Cimato2, and Zeinab Ganjei [12] proposed a model of A Secure Searchable Secret Sharing Scheme (AS5) tolerates statistical attacks based on opponent's knowledge about outsourced data distribution. In AS5 data shares generates uniformly across a domain to prevent information leakage about the outsourced data. In a searchable secret sharing scheme in which the ordering relation between values is preserved in their corresponding shares, while the distribution of shares is different from the original data distribution. Searchable secret sharing scheme to be secure against adversaries powered by a priori knowledge of outsourced data to tolerate statistical analysis on data shares.

Ernesto Damiani, S.De Capitani di Vimercati, Sushil Jajodia, Pierangela Samarati [13] has describes a straightforward however robust single-server solution for remote querying of encrypted databases on untrusted servers. Our method is on the foundation of the usage of indexing information attached to the encrypted databases which is to be used by the server to select the data to be returned in a effect to a query without the necessity of disclosing the database content. The indexes trade of between efficiency needs in query execution and protection are required because of possible inference attacks exploiting indexing information.

Luca Ferretti, Michele Colajanni, and Mirco Marchetti [14] propose a novel alternative that guarantees confidentiality of data stored in to cloud database that are untrusted by definition. All data outsourced to the cloud providers are encrypted by cryptographic algorithms that permit the execution of typical SQL queries on encrypted data. It also permits direct, independent and concurrent use of the cloud database and that helps even changes to the database structure. It generally does not count on a trusted proxy that shows a single position of failure and a system bottleneck, and that restricts the access and scalability of cloud database services. Concurrent read and write operations that do not modify the design of the encrypted database are supported with little overhead. More active circumstances indicated by (concurrent) alterations of the repository design are supported but at the buying price of larger expense and stricter transaction isolation levels.

Jinyuan Li, Maxwell Krohn, David Mazi`eres, and Dennis Shasha [15] SUNDR is a network file system created to store knowledge securely on untrusted servers. SUNDR enables client discover any attempts at unauthorized

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

file alteration by malicious server operators or users. SUNDR's protocol achieves a property called fork consistency, which guarantees that clients can discover any reliability or consistency failures so long as they see each other's file modified. SUNDR's protocol enables clients discover unauthorized attempts to change files, also by opponents in control of the server. When the server behaves properly, a fetch reflects exactly the authorized modification that occurred before it.

Amazon Elastic Compute Cloud (Amazon EC2) [16] provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. We can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

III.PROPOSED ARCHITECTURE

A. *Description of the proposed architecture:*

Figure 3.1 System Architecture user login using user id and password if not user will register. After user login first user generates key or it can use from existing key. The key is downloaded to the local system. User may upload file or it can execute DDL command, DML command or DAL command. DDL command contains the create, alter and drop commands. DML commands contains insert, update and delete command. DAL command contains the select command. When user gives input to the query parser. Parser parses the query and encrypts the query which is table name and the column name depending on the type of query user uses with the secret key by using AES encryption. The encrypted data is wrapped into JSON format and is send to the server through SSL/HTTPS channel [17]. The user can also upload file by encrypting the contents of the file with the secret key. Integrity validation is done by checking the integrity of the selected file using same secret key which it has been encrypted the contents. The user selected file and the file stored in database both are compared with MD5 algorithm, therefore integrity is checked. The encrypted data wrapped in JSON format are processed to the server. Server side parses the encrypted data and form new encrypted query. New encrypted query are processed and stored in the database.

The tenant web server parses the JSON format query are unwrapped and the encrypted data are processed to the Java Web Service forms the new encrypted query are processed to the database which are further storing and retrieval of the encrypted data.

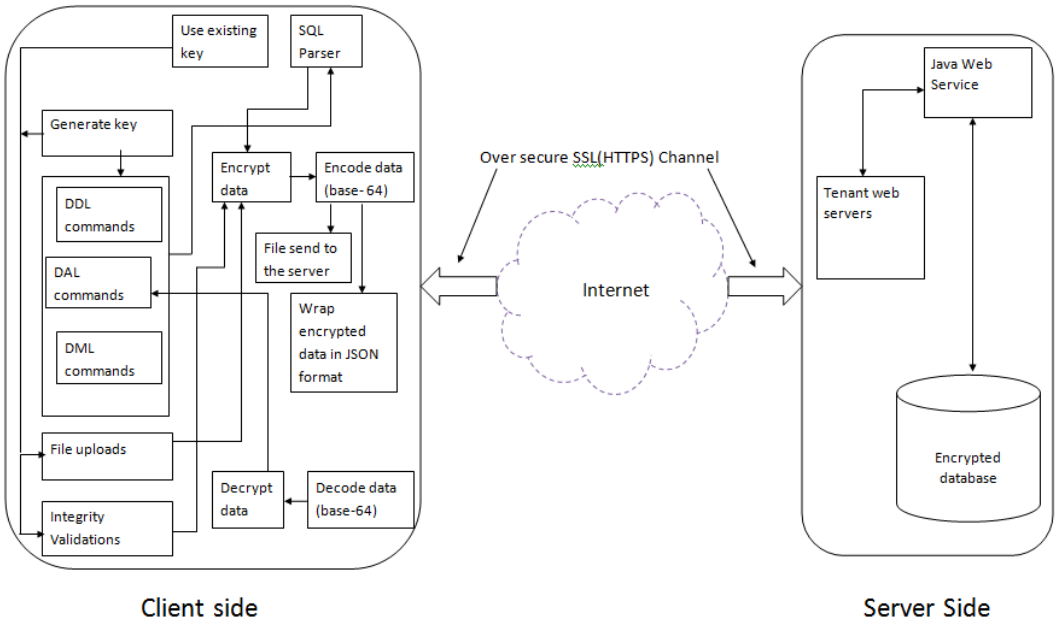

Figure 3.1: System Architecture

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

B. *AES Algorithm:*

In [18] the important thing length used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, known as the plaintext, into the final output, called the cipher text. The wide varieties of cycles of repetition are as follows:

- 10 cycles of repetition for 128-bit keys.
- 12 cycles of repetition for 192-bit keys.
- 14 cycles of repetition for 256-bit keys.

Excessive-level description of the algorithm

- Key Expansions—round keys are derived from the cipher key the use of Rijndael's key time table. AES requires a separate 128-bit round key block for each spherical plus one more.
- Initial round.
- AddRoundKey—each byte of the nation is blended with a block of the add round key the use of bitwise xor.
- Rounds

1. SubBytes—a non-linear substitution step where every byte is changed with some other according to a lookup table.
2. ShiftRows—a transposition steps wherein the remaining 3 rows of the kingdom are shifted cyclically a sure quantity of steps.
3. MixColumns—a mixing operation which operates at the columns of the nation, combining the four bytes in each column
4. Add round Key.

1) The SubBytes Step

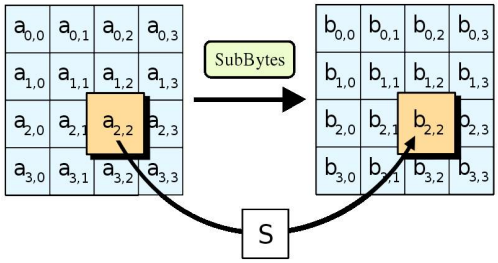

Figure 3.2: SubBytes Step

Inside the Sub Bytes step, every byte inside the state is changed with its entry in a hard and fast eight-bit look-up table, S; bij = S(aij).

2) The ShiftRows step

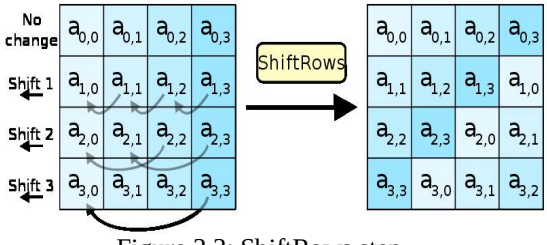

Figure 3.3: ShiftRows step

In the ShiftRows step, bytes in each row of the state are shifted to the left. The number of places each byte is shifted are different for each row.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

3) The MixColumns step

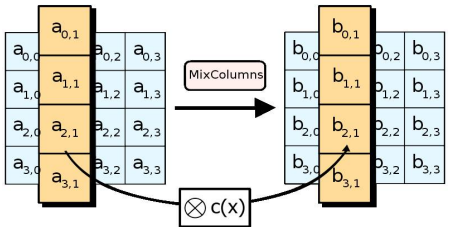

Figure 3.4: MixColumns step

In the MixColumns step, each column of the state is multiplied with a fixed polynomial

4) The AddRoundKey step

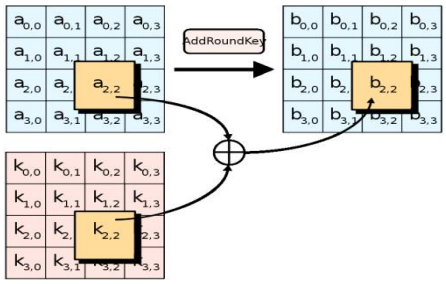

Figure 3.5: AddRoundKey step

In the AddRoundKey step, each byte of the state are combined with byte of the round subkey using the XOR operation (⊕).

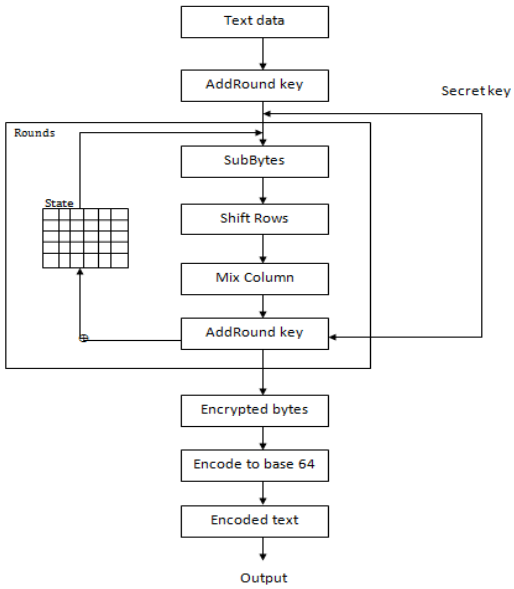

Figure 3.6: AES encryption

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

In this the figures shows that the table name user has generated the secret key. For AES encryption the table name goes through the AES rounds and finally the encrypted bytes is obtained which are encoded to base 64 and sends to the server. Base64 is a group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation.

The AES Decryption is the reverse process of encryption in this received encrypted data is decoded to base-64 processed through the AES round along with the secret key it had used during the encryption. The decrypted bytes is then transform from bytes to string and the decrypted text is obtained which is plaintext data.

IV. RESULTS

We implemented the proposed system for securing user data from service provider for securing the user data. It has encrypted using AES algorithm the encrypted data and the plaintext data processing response time and the key generation time is shown.

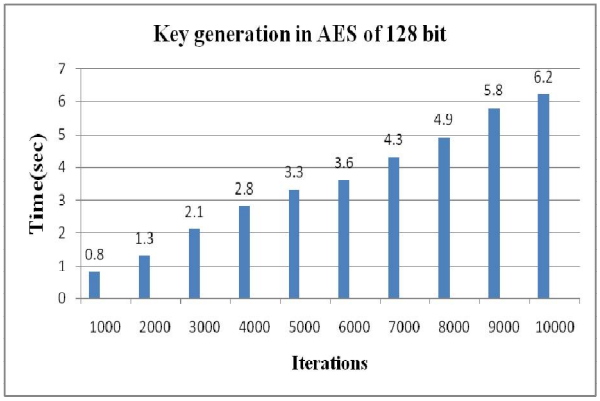

Figure 4.1: Result graph for key generation in AES of 128 bit.

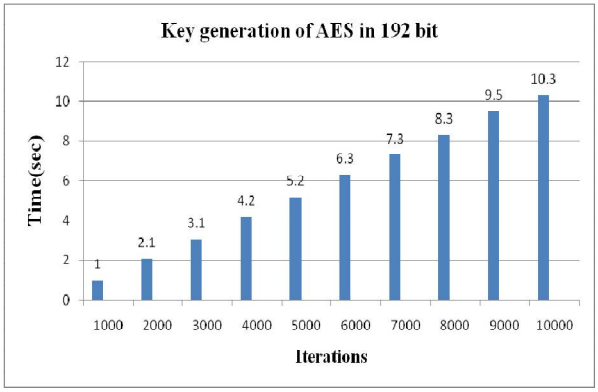

Figure 4.2: Results for key generation of AES of 192 bit

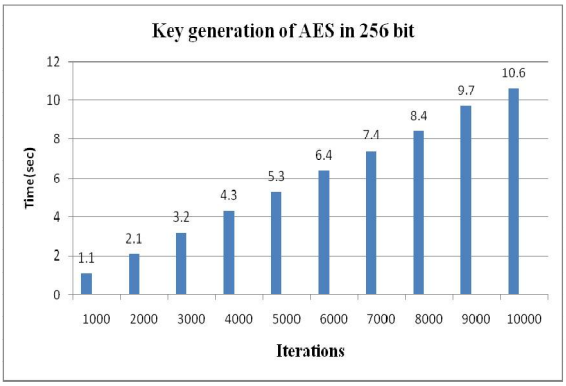

Figure 4.3: Results for key generation of AES of 256 bit.

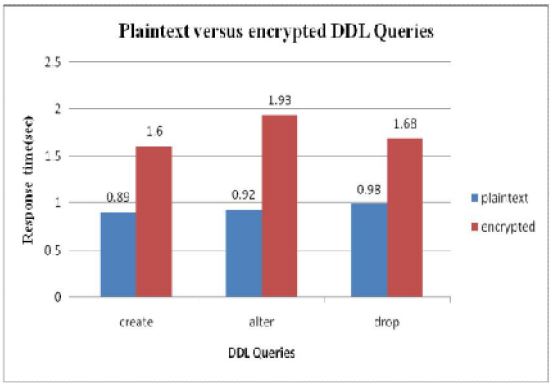

Figure 4.4: Results for Plaintext versus encrypted DDL Queries

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

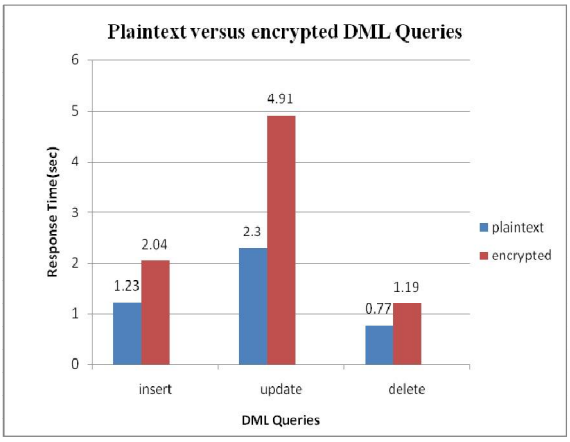

Figure 4.5: Results for Plaintext versus Encrypted DML Queries

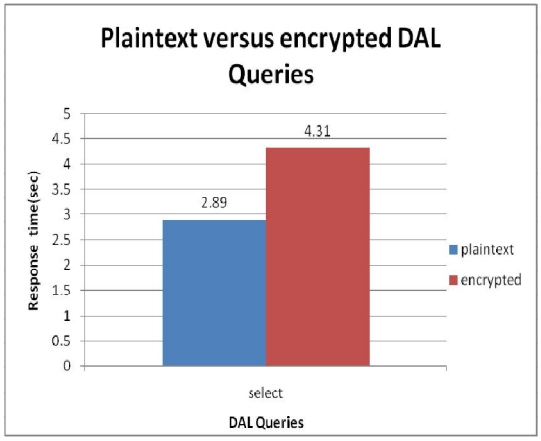

Figure 4.6: Results for Plaintext versus Encrypted DAL Queries

V. CONCLUSION

In the proposed work its miles modern structure that guarantees confidentiality of data stored in public cloud databases. In contrast to present day strategies the proposed structure does not require adjustments to the cloud database, and it's far without delay relevant to present cloud DBaaS. An architecture that guarantees confidentiality of information stored in public cloud databases as well as maintaining integrity of data.

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] M. Armbrust et al., “A View of Cloud Computing”, Comm. of the ACM, vol. 53, no. 4, pp. 50-58, 2010.
[3] W. Jansen and T. Grance, “Guidelines on Security and Privacy in Public Cloud Computing”, Technical Report Special Publication 800-144, NIST, 2011.
[4] 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.
[5] P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin , “Depot: Cloud Storage with Minimal Trust,”, and M. Walfish, 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] 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.
[8] 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.
[9] 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.
[10] 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.
[11] Ganapathy, D. Thomas, T. Feder, H. Garcia-Molina, and R. Motwani, “Distributing Data for Secure Database Service”, Proc. Fourth ACM Int’l Workshop Privacy and Anonymity in the Information Soc., Mar. 2011.
[12] M. Hadavi, E. Damiani, R. Jalili, S. Cimato, and Z. Ganjei, “AS5: A Secure Searchable Secret Sharing Scheme for Privacy Preserving Database Outsourcing”, Proc. Fifth Int’l Workshop Autonomous and Spontaneous Security, Sept. 2013.
[13] 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.
[14] 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.
[15] J. Li, M. Krohn, D. Mazie`res, and D. Shasha,“ Secure Untrusted Data Repository (SUNDR)”, Proc. Sixth USENIX Conf. Operating Systems Design and Implementation, Oct. 2004.
[16] “Amazon Elastic Compute Cloud (Amazon Ec2),” Amazon Web Services (AWS), http://aws.amazon.com/ec2, Apr. 2013.
[17] https://en.wikipedia.org/wiki/HTTPS.
[18] https://en.wikipedia.org/wiki/Advanced_Encryption_Standard.