iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** V **Month of publication:** May 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Trust DB-Providing Confidentiality and Trust in Cloud Databases

M.K.S Karthikeya[1], Arif Mohammed Abdul[2]
[1]P.G. Student, [2]Assistant Professor
Department of Computer Science, GITAM University, Hyderabad, Telengana, India

Abstract-Cloud computing provides a high usage of resources and on demand with applications without the use of hardware and software. Cloud computing is a highly resource provisioning and promising technology such that it helps in managing data as per requirements. Data is encrypted and stored in the databases. When a particular data is needed by the user then queries are used to run on the encrypted data such that the level of confidentiality is high which provides guarantee to users. Hence a system with Crypt DB is used such that it provides the level of security high during the process of getting the data. ***Keywords: CryptDB, Cloud, Confidentiality, Decryption, Encryption, Key, Queries***

I. INTRODUCTION

A database query is the intermediate tutelage for DBMS to get the data from the physical stored medium. Getting the exact information is a skill in the query processing. Many organizations cannot work properly if their database is slow and they need security in which the information should not go out who would fraud it. Sensitive information is present when it is lost it leads to damage. Other than loss of data the number of factors that arise are 1) Unavailability of data: In certain cases the user has to check the availability of data and its handling in authorized way.2) Physical access: In this, the data is steal and the intruders may get entire access to the data.

So in order to avoid such problems cryptography is introduced in which confidentiality, authentication, non- repudation and access control are observed. These cryptographic concepts are applied on the Database Management Systems to change the readable formats to unreadable formats(Confidentiality), proving the identity of the user (Authentication), Mechanism that sender sent message and receiver could not blame that message is not received (Non-repudation) and preventing the unauthorized use of resources(Access control). On implementing the techniques the online applications can have a minimum amount of security. But to provide security and confidentiality to the database, the data is encrypted and on the encrypted data the SQL operations are performed by the SQL database which is nothing but Crypt Database. On encrypting the data in the database, the data is accessed by the user by performing operations on the encrypted data. The main challenge is that in order to do encryption organizations are most concerned about key management which is also the main barrier of database encryption. The other challenge is that to minimize the leakage of Meta data in the database server when a user gives an input to the server. A solution is provided by using the concept of secure computing technology which greatly reduces the risk of information leakage. The secure computing technology is also known as secure in theory in which data is obtained by manipulating the operations based on our proprietary algorithms and takes a less amount of time to sort thousands of records in the database.

II. CONTENT DIAGRAM

The above diagram represents the architecture of Trust DB. The architecture consists of a client severs, interface servers and database servers. Cryptographic operations are performed on the database server to encrypt and decrypt the data at the database. Interface server acts as an interface to provide access in between the user and the database server through which sequence of operations are generated by the user. When user gives an input in the middle the interface server gets the input and sends it to database server. On doing so the database server accepts the input through the interface server and query is run on the database and so the query is decrypted at the interface server.

©IJRASET 2015: All Rights are Reserved

Volume 3 Issue V, May 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

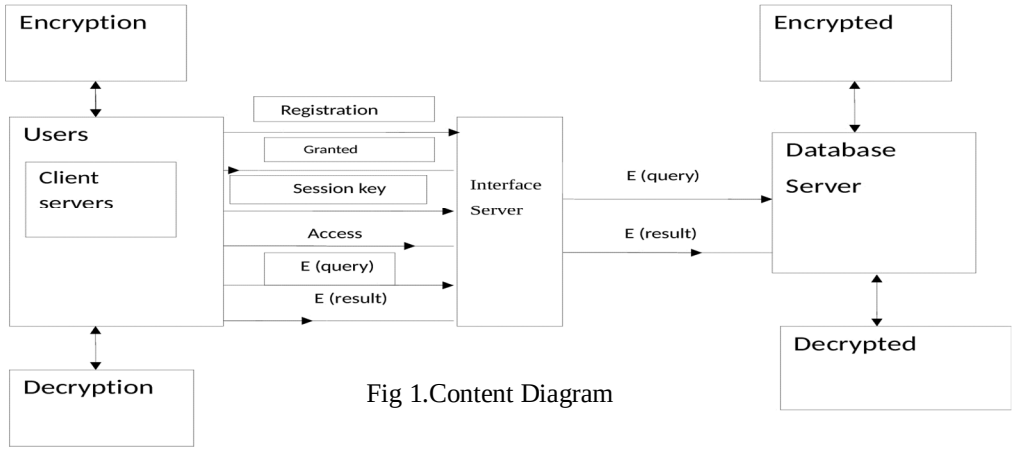


Fig 1.Content Diagram

The technique used here is 2 way encryption that is low level encryption and high level encryption. Trust DB uses these level of encryptions in such a way that user gets the exact requirement which executes to transform data with the help of user given inputs. Here a session key is generated in such a way that user can login with the help of session key so that a level of confidentiality is high than the previous one in the presence and also in the absence of the user.

III. KEY MANAGEMENT

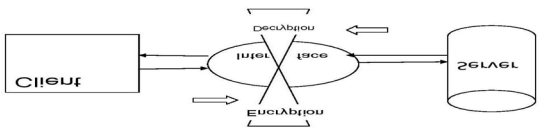
Encryption is a process of conversion of data in unreadable format called cipher text. Decryption is a process of making unreadable data to readable data to its original form. According to kickoff's principle security rely upon the secrecy of the scheme but not on the obfuscation of the code. Cryptography scheme is assumed to be publically known where as the secret piece of information such as key is responsible for the secrecy of the scheme.

IV. RUNNING QUERIES ON THE ENCRYPTED DATA

The need for this concept is that even when the users store their data and if the organization gets compromised with the attackers then it may lead to data loss. To avoid that data is encrypted and then it is stored in the database on which the queries are made to run. Searchable Symmetric Encryption schemes are used for finding the exact matches corresponding to query. Here search pattern is used to get the feature set for n consecutive queries. Here a binary function π is used to get the matrix π [u, j] =1 else π [I, j] =0. On searching the pattern then we access the pattern along with data items. The access pattern helps to define the accessing mechanisms to construct number of data availability between distinct elements of data such that it may get relative sensibility of data. When we get the data, then a similarity pattern is used such that it represents the multi component feature set for n consecutive queries and i[j] represents the ith feature for jth component. To get the exact output, partition is done such that individual relational operations help to get the data at the client side. The purpose of partitioning the data is that to guarantee the client side operations that are to be implemented effectively and we use sql queries for splitting the query and individual computations are performed.

V. ENCRYPTION AND DECRYPTION TECHNIQUES

Encryption and decryption are used to provide security to data. Here when user sends the input then at the interface server the input gets encrypted at then it is stored in the server.



©IJRASET 2015: All Rights are Reserved

Here when input is observed then it undergoes a layered of encryptions with algorithms like Blowfish, AES and RSA. When a data is entered in the database then it should through a sequence of encryption operations like Blowfish, AES and RSA. To decrypt the data it should go through the operations like RSA, AES and Blowfish to get the data.

A. Blowfish Algorithm

It uses some symmetric pair of encryption and decryption techniques to encrypt and decrypt the data except changing of keys and divide the message in fixed length of blocks during encryption and decryption.

Encryption:

Encrypt (plain text)

Data= (“blowfish key”).get Bytes ()

Cipher (Cipher.Encrypt_mode, key);

Hasil=Cipher (plaintext)

return new. Encode (hasil);

Decryption:

Decrypt (String cypher)

Cipher (Cipher.Decrypt_Mode, key);

Hasil=Cipher (cypher)

return new (hasil);

B. AES Algorithm

AES works by repeating the same defined steps multiple times.AES is a secret key encryption algorithm.AES operates on fixed number of bytes. In this data is protected by iterative, symmetric key block cipher text that can be used to encrypt and decrypt data in blocks. Iterative ciphers are implemented such that they are performed based upon the performance of input data. The working of AES is defined as follows

AES encryption:

Encrypt (String strToEncrypt)

{

Cipher=Cipher.(“AES”);

Cipher (Encrypt);

SetEncryptedString (cipher.doFinal (strToEncrypt))

}

AES decryption:

©IJRASET 2015: All Rights are Reserved

```
Decrypt (String strToDecrypt)

{

Cipher=Cipher (“AES”);

Cipher (Encrypt);

SetDecryptedString (cipher.doFinal (strToDecrypt));

}
```

C. RSA Algorithm

A randomly generated pair of keys is used such that the data is encrypted with public key and decryption is done with private key such that it achieves confidentiality and encrypting with private key and decrypting with public key also helps to achieve authentication and integrity checking. The plain text is typically hashed with a symmetric key and it will represent a hash value. The process of encryption and decryption is done as follows:

Decrypting for message M i.e.

D (E (M)) =M

Publicly of revealing E does not reveal an easy way to compute D. As such only the user decrypt messages which were encrypted with E.

Deciphering a message M and then deciphering it results in M. That is E (D (M)) =M

Encryption in RSA: c=m^e (mod n)

Decryption in RSA: m=c^d (mod n)

The functioning of RSA follows here:

Encryption:

```
Encrypt (message)
{
return (message).modpow (e, n)
}
Decryption:
Decrypt (message)
{
return (message).modpow (d, n)
}
```

VI. CONCLUSION

Data integrity a feature of sound DBS is considered such that a message is used as an authentication code. A notification like message is provided in the interface server such that it helps to check the recorded list of logged in users in such a way that user can have a review of his logged in history. Leakage of meta data is avoided in such a way that it leaves no traces of data.

VII. FUTURE WORK

The review of files accessed by the user and the level of authentication can be improved. User is unaware about the queries inserted

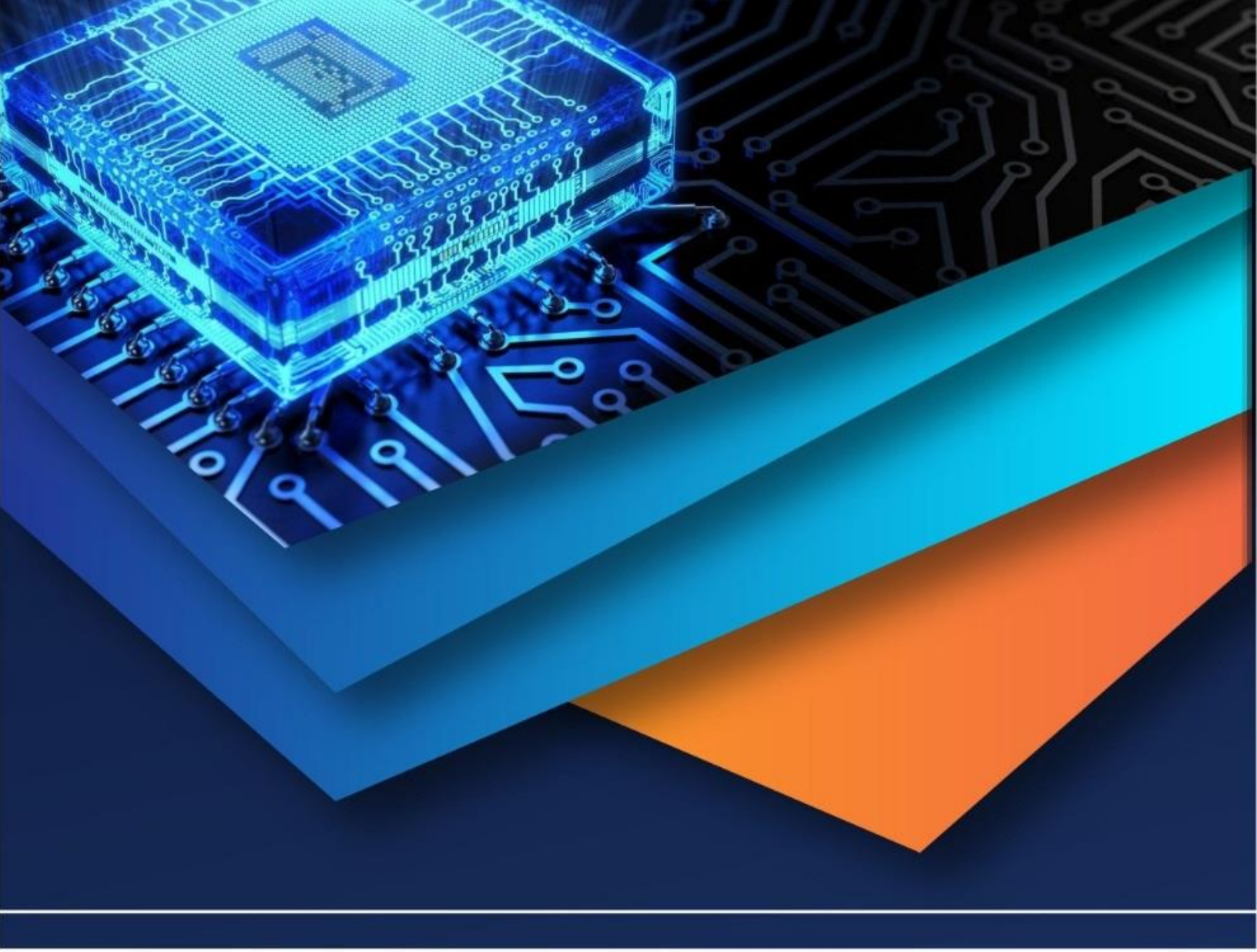
©IJRASET 2015: All Rights are Reserved

and data. A history can also be maintained such that the user can review his previous operations.

REFERENCES

[1] Blowfish- Kevin Allison Keith Feldman Ethan Mick
[2] Security Issues in Querying Encrypted Data Murat Kantar coglu, Chris Clifton
[3] Executing SQL over Encrypted Data in the Database Service Provider Model –Hakan Hacıg¨um¨us¸Bala Iyer, Chen Li Sharad Mehrotra
[4] Efficient Similarity Search over Encrypted Data Mehmet Kuzu, Mohammad Saiful Islam, Murat Kantarcioglu
[5]CryptDB: Protecting Confidentiality with Encrypted Query Processing by Raluca Ada Popa, Catherine M. S. Redfield, Nickolai Zeldovich, and Hari Balakrishnan MIT CSAIL
[6] Client and Data Confidentiality in Cloud Computing Using Fragmentation Method Thota Reshma Kishore, D.Akhila Devi, S.Prathyusha, D.Bhagyasri, Bhuma Naresh
[7]http://courses.cs.vt.edu/~cs5204/fall00/protection/rsa.html
[8]http://en.wikipedia.org/wiki/Blowfish_ (cipher)
[9]http://aesencryption.net/
[10]http://javapapers.com/java/java-symmetric-aes-encryption

©IJRASET 2015: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)