INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Privacy Preserving Approach Using Encrypted Cloud Database: Result and Analysis

Ms. Reshma Narawade[1], Prof. Vilas Jadhav[2]

[1,2]*Computer Engineering Department, Mumbai University*

Abstract: This Quick growth in communications, storage and processing information allow us to move all data into some remote place. Data management systems activated by automated traditional tasks such as keeping record of business transactions and many more. This data involved mainly of numeric and characteristics and graphical representation. Cloud is not a particular product; it is a simple technique of delivering IT services based on demand, adaptable to rescale as required based pay-as-use model. Cloud computing becomes popular because of its delivery of flexible access to the cloud data storage. The cloud data storage capacity allows users to access and extract the sensitive information whenever they require. The storage and access of data through the third party server or proxy server increases the load of the users this leads to a memory and time complexity. Storing and accessing data through the cloud server or proxy server will arise security issue. This concludes occurs cloud servers are not fully trustable as data is openly available to them. Some existing storage services provide security of stored data, while data confidentiality for the database as a service model is still not fully formed. In this proposed architecture a different architecture that takes part cloud database services with confidentiality of data and the opportunity of performing parallel operations on encrypted data. This architecture supports physically scattered clients which are directly connected to an encrypted cloud database, and to perform concurrent and independent operations such read, write and modify the database structure. All information stored on cloud is in encrypted format. Metadata related to the information is in encrypted format so ease access of information is controlled. This work proposed architecture eliminates intermediate proxy servers that limit the availability, elasticity, and scalability properties cloud-based solutions. Blowfish algorithm technique is used for the encryption and decryption of data. Because of variable key length decryption of the data is difficult. User can store any type of information on cloud and can retrieve in distributed environment. Response time is also increases as omission of intermediate proxy server. Blowfish algorithm is fast in speed as well as not attach are found on the application in which blowfish algorithm is used. Higher security with great response time data is available everywhere.

Keywords: Cloud Computing, DBaaS, Blowfish, Encryption

I. INTRODUCTION

Secure DBaaS provide solution on the use of multiple cloud providers, and makes use of SQL-aware encryption algorithms to support the execution of most common SQL operations on encrypted data. Secure DBaaS relates more closely to works using encryption to protect data managed by untrusted databases [1]. In such a case, a main issue to address is that cryptographic techniques cannot be natively applied to standard DBaaS because DBMS can only execute SQL operations over plaintext data. Some DBMS engines offer the possibility of encrypting data at the file system level through the Transparent Data Encryption feature. This feature makes it possible to build a trusted DBMS over untrusted storage. The DBMS is trusted and decrypts data before their use. Hence, this approach is not applicable to the DBaaS context considered by Secure DBaaS, because we assume that the cloud provider is untrusted. Secure DBaaS solves this problem by letting clients connect directly to the cloud DBaaS, without the need of any intermediate component and without introducing new bottlenecks and single points of failure. A proxy-based architecture requiring that any client operation should pass through one intermediate server is not suitable to cloud-based scenarios, in which multiple clients, typically distributed among different locations, need concurrent access to data stored in the same DBMS. On the other hand, Secure DBaaS supports distributed clients issuing independent and concurrent SQL operations to the same database and possibly to the same data.

II. PROPOSED SYSTEM

Secure DBaaS is solution for storing and accessing data on cloud in secure way. Data is store and retrieve on cloud in encrypted format. Encryption is done at client side only. So untrusted user are unaware about the data.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VIII, August 2017- Available at www.ijraset.com

A. Metadata Management

Your Secure DBaaS generate the metadata that Metadata generated by Secure DBaaS contain all the information that is necessary to manage SQL statements over the encrypted database in a way transparent to the user [2]. Metadata management strategies represent an original idea because Secure DBaaS is the first architecture storing all metadata in the untrusted cloud database together with the encrypted tenant data. Secure DBaaS uses two types of metadata:

1) Database Metadata are related to the whole database. There is only one instance of this metadata type for each database.
2) Table Metadata are associated with one secure table. Each table metadata contains all information that is necessary to encrypt and decrypt data of the associated secure table.

The structure of a table metadata is represented in Fig.1. Table metadata contain the name of the related secure table and the unencrypted name of the related plaintext table. Moreover, table metadata include column metadata for each column of the related secure table. Each column metadata contain the following information.

3) *Plain name:* The name of the corresponding column of the plaintext table.
4) *Coded name:* The name of the column of the secure table. This is the only information that links a column to the corresponding plaintext column because column names of secure tables are randomly generated.
5) *Secure type:* This allows a Secure DBaaS client to be informed about the data type and the encryption policies associated with a column.
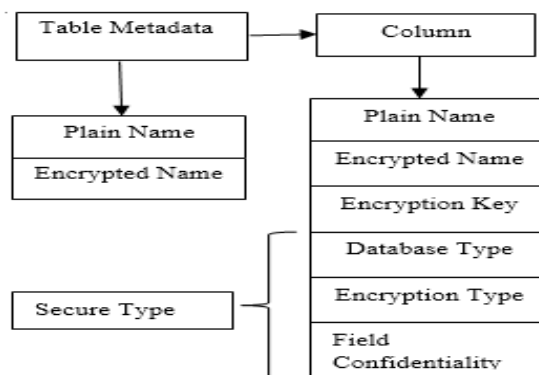6) *Encryption key:* The key used to encrypt and decrypt all the data stored in the column.

Fig 1. Structure of Table Metadata

III. SYSTEM IMPLEMENTATION

A. Key and Data Management

The DBA creates the metadata storage table that at the beginning contains just the database metadata, and not the table metadata. The DBA populates the database metadata through the Secure DBaaS client by using randomly generated encryption keys for any combinations of data types and encryption types, and stores them in the metadata storage table after encryption through the master key [3]. Then, the DBA distributes the master key to the legitimate users. User access control policies are administrated by the DBA through some standard data control language as in any unencrypted database.

Secure DBaaS relies on standard authentication and authorization mechanisms provided by the original DBMS server. After the authentication, a user interacts with the cloud database through the Secure DBaaS client. Secure DBaaS analyze the original operation to identify which tables are involved and to retrieve their metadata from the cloud database. The metadata are decrypted through the master key and their information is used to translate the original plain SQL into a query that operates on the encrypted database. Translated operations contain neither plaintext database table and column names nor plaintext tenant data. Valid SQL operations that the Secure DBaaS client can issue to the cloud database [4]. Translated operations are then executed by the cloud database over the encrypted tenant data. As there is a one to-one correspondence between plaintext tables and encrypted tables, it is possible to prevent a trusted database user from accessing or modifying some tenant data by granting limited privileges on some tables. User privileges can be managed directly by the untrusted and encrypted cloud database. The results of the translated query that includes encrypted tenant data and metadata are received by the Secure DBaaS client, decrypted, and delivered to the user. The complexity of the translation process depends on the type of SQL statement.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VIII, August 2017- Available at www.ijraset.com

B. Blowfish Algorithm

Blowfish algorithm is used for encryption and decryption of data. It generate random key for encryption and stored it into metadata as stated in above description. As blowfish is symmetric key encryption technique, same key is used for encryption and decryption. Blowfish Algorithm is a Feistel Network, iterating a simple encryption function 16 times. The block size is 64 bits, and the key can be any length up to 448 bits.

1) *Advantages of Blowfish Algorithm*
a) It has been repeatedly tested & found to be secure.
b) It is fast due to its taking advantage of built-in instructions on the current microprocessors for basic bit shuffling operations.
c) It was placed in the public domains.
d) No attack found
e) Fast in encryption and decryption speed

Fig 2. Describe the single round of blowfish encryption technique. In the proposed system of blowfish algorithm reduced the rounds of blowfish algorithm and in the algorithm each single round is introduced new modified[6][7]. In the blowfish algorithm there will be 64 bits then the bits are separate into 32bits and there will be four s-boxes. Each s-box contains 32bits. Now design the algorithm like two s-boxes connecting with XOR as like same other two 2 s-boxes connected with XOR and then from the two XOR added then from there get key plain text.
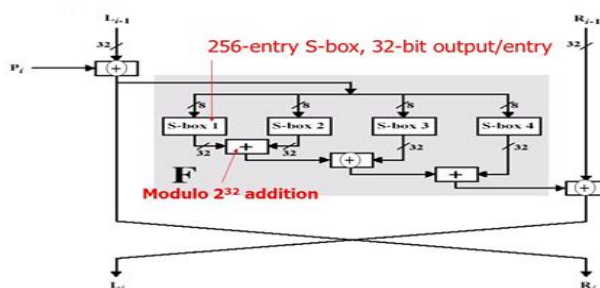

Fig 2. Blowfish single round

IV. RESULT AND ANALYSIS

Result of the implemented system with database is as follows. Working of Admin, Manager and Client module explained by inserting real time data .All data is stored in database in encrypted format.

A. Admin Module

Client need to be register himself on side. Admin gives permission to the client to store data on cloud by accepting request. Admin can manage the client and add them as a manager as well.
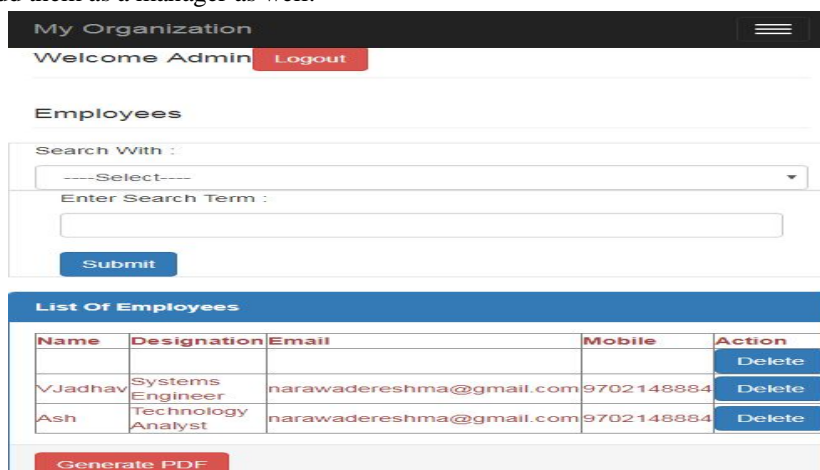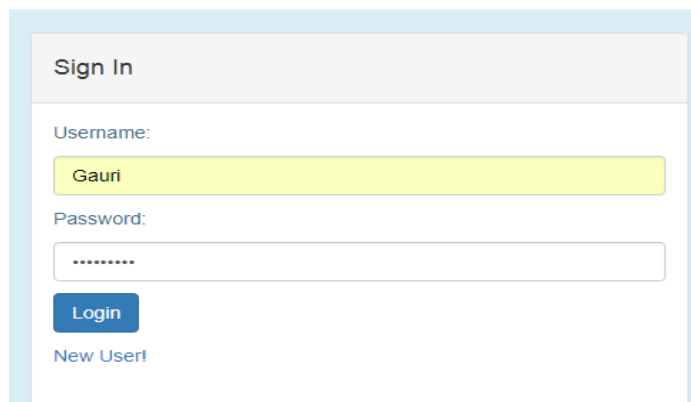

Fig 3. Client List on admin module

Fig 3. Show the list of employees successfully register and can upload data on Secure DBaaS.

Fig 4. Client Login

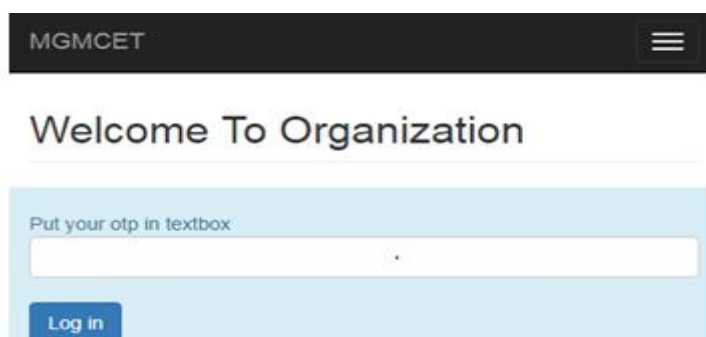
Fig 5. One Time Password window

Fig 4. describe the client login window. Client need to enter OTP(One Time Password) for successful login shown in Fig 5.

Fig 6. Uploaded file by client

Fig 6.show the file uploaded by client.It is in encrypted format.

Table I: Register Client On System

id	MtQ+KIqFA/I=	vSvYTY5zyvSO...	rfYV5E5+39Q...	iMFGFtc+JU8=	P/J2zqJmaMk=	3ET9b1r3aG2...	q2vzXmkK
20041	AFr/IsC5+yE=	hsX9rIJy99A=	zLZQIC8JIV8=	z/y/4YyeOpaPU...	hCz/61Aj7pKXa...	iOQSP173Hju9...	active
20042	mC0u/bWFrg...	0qNJcfGfxI4+tA...	AsT4CBoSuw...	z/y/4YyeOpaPU...	hCz/61Aj7pKXa...	iOQSP173Hju9...	active
20043	WA/ZwG8CpX...	rG2VaqVdjG4=	4nOQueBbwI...	z/y/4YyeOpaPU...	hCz/61Aj7pKXa...	7GX9oMeo37x...	active

Table I. show the database table of registered client. All details are in encrypted format.

TABLE II: Data Uploaded On Database by Client

username	fname	fbytes	fext	fmetadata	fhash
I3+I1yzAB8A=	dv1RQOLIXjUL0...	<Binary data>	.docx	787pIbd0TRq1n...	y31pp1Uaji1L5o...
5xLuEeHtICTSA...	kIqSiTdodbbDR...	<Binary data>	.doc	CTQKmhtQB5i...	JRIZxsEYB+Q3R...
Ot9p7Sy+ILs=	2o/CqeH1V7cf...	<Binary data>	.pdf	CTQKmhtQB5jx...	tey+ykN8NpHL...
Ot9p7Sy+ILs=	yn4RtLyBKxS13...	<Binary data>	.doc	ofbDg2z5mA6...	wQANFhR1Sgt...
Ot9p7Sy+ILs=	QjB/iJq76gqPq...	<Binary data>	.doc	ofbDg2z5mA76...	n4Ss0FVwhHRy...
ZnyL2undNdI=	gkKlJXOwLVsD...	<Binary data>	.jpg	CTQKmhtQB5i...	Lu6VjC1ek+IrtR...
AFr/IsC5+yE=	gkKlJXOwLVsD...	<Binary data>	.jpg	CTQKmhtQB5i...	ydoMwn+JgHl...

Table II. describe the data stored on database.

V. CONCLUSIONS

The The proposed an innovative architecture that guarantees confidentiality of data stored in public cloud databases. Unlike state-of-the-art approaches, the solution does not rely on an intermediate proxy that we consider a single point of failure and a bottleneck limiting availability and scalability of typical cloud database services. The solutions to support concurrent SQL operations including statements modifying the database structure on encrypted data issued by heterogeneous and possibly geographically dispersed clients. The proposed architecture does not require modifications to the cloud database, and it is immediately applicable to existing cloud DBaaS, such as the experimented PostgreSQL, Plus Cloud Database, Windows Azure, and Xeround. There are no theoretical and practical limits to extend our solution to other platforms and to include new encryption algorithms. It is worth observing that experimental results based on the TPC-C standard benchmark show that the performance impact of data encryption on response time becomes negligible because it is masked by network latencies that are typical of cloud scenarios. In particular, concurrent read and write operations that do not modify the structure of the encrypted database cause 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 are investigated.

VI.ACKNOWLEDGMENT

REFERENCES

[1]   Reshma Narawade,V. Jadhav International Journal of Latest Trends in Engineering and Technology of titled "Privacy Preserving Approach Using Encrypted Cloud Database", www.ijltet.com Volume 6,Issue 3,January 2016.

[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

[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]   J. Li, M. Krohn, D. Mazie`res, and D. Shasha, "Secure Untrusted Data Repository (SUNDR)," Proc. Sixth USENIX Conf. Opearting

[6]   Systems Design and Implementation, Oct. 2004.

[7]   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.

[8]   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.

[9]   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.

[10]   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.

[11]   M. Hadavi, L.Ferretti , 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.

[12]   A"Oracle Advanced Security," Oracle Corporation, http://www. oracle.com/technetwork/database/options/advanced-security, Apr. 2013.

[13]   D. Agrawal, A.E. Abbadi, F. Emekci, and A. Metwally, "Database Management as a Service: Challenges and Opportunities," Proc. 25th IEEE Int'l Conf. Data Eng., Mar.-Apr. 2009.

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

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

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

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 ◎ (24*7 Support on Whatsapp)