Secured Storage for Cloud Based Relational Database Management Systems

[1]S.Muthurajkumar, [2]S. Murugesan, [3]A. Kannan
[1]Madras Institute of Technology Campus, Anna University, Chennai, India
[2,3] College of Engineering Guindy Campus, Anna University, Chennai, India
muthurajkumarss@gmail.com, muruga13@gmail.com, kannan@annauniv.edu

***Abstract*— The Cloud database system is a new trend which anticipated to reshape all developments in computer technology. The data outsourced may be confidential which results in encryption of the documents. The proposed system applies encryption techniques for storage with security. This system considers structured data for storage as it would be easier for them extract information from the raw data. Moreover, data mining would be easier in structured data. The structured data are often sensitive and hence it is to be in a protected form. In order to ensure security, the proposed model provides a mechanism which increases the level of security based on are password protection and are encryption. It is achieved through blind storage which is a mechanism of concealing the access pattern and the storage pattern is also concealed by applying the concept of blind storage over relational databases in cloud.**

***Index Terms*— Cloud Data Storage, Blind Storage, Cloud Database, Cloud Computing**

I. Introduction

A cloud environment consists of three major sub divisions. They are data owner, cloud server and end user. The cloud server is the place where everyone can outsource their data. The outsourced data is stored remotely in the cloud and will be available for the user when requested. The document owner outsources the volume of documents to the cloud server. The documents are stored in the cloud server depending on the mechanism of storage the service provider follows. The user can obtain a document by requesting the desired document. The requested resource can be obtained from the server if the user has required permissions to access the document. Search over encrypted document is one of the recent trends in cloud computing. Programmers with sufficient knowledge and even the service providers can use the insensitive data to derive sensitive data from it. The sensitive data can be either photographs, e-mails, list of files a user has.., etc. This is a violation of privacy of the user. In order to ensure privacy and improve the security of documents outsourced to the cloud, Blind storage mechanism can be used. Blind Storage is the process of concealing the access pattern of user, storage details and location of storage from the cloud environment itself. Blind storage has to be implemented on the existing storage structure. The implementation acts as layer of abstraction over the existing storage mechanism and hides the details of storage to the outside environment. By achieving Blind storage, leakage of data, service providers prying into users' privacy can be avoided. People without proper authenticity and access will not be able to access resources which are out of their scope.

II. Literature Survey

In the past, many researchers [2], [4], [6], [8].[11], [12], [13], [14], [15] have proposed new methods for storage and retrieval data with secured method. Hongwei Lil et. al. [1] proposed a mechanism of to search over encrypted documents and to conceal the access pattern using Blind storage. In mobile cloud environment, data needs to be outsourced to cloud. For security purposes the outsourced data have to be encrypted and stored in the database. To make the search over the data accurate, relevance score and k nearest neighbor techniques are utilized to retrieve data from the server. They have developed efficient indexing mechanism to store the documents in the cloud server. In order to conceal the access pattern of the user, it adopts the mechanism of blind storage. Trapdoor privacy, unlink ability and confidentiality of documents are achieved through blind storage. Secure transmission of data from the client system can be achieved. In addition, efficient retrieval of documents in accordance with relevancy can be obtained by calculating relevance score.

Naveed et. al. [3] proposed a search mechanism over encrypted cloud data revealing less information to the cloud server. The dynamic collection of documents is stored using dynamic symmetric encryption by a client. The keyword searches are carried out on these encrypted documents and because of encryption only a minimal data is revealed to the server. A new mechanism named Blind Storage is employed to provide minimal data leakage to the cloud server. Conventional methods use a linked list representation which involved decryption on the cloud which may lead to leakage of data. Blind storage mechanism is proposed to overcome the limitations in the traditional system. A pseudorandom function is used to generate the location of the sequence of blocks of files. A pseudorandom set is used to represent the files in the cloud server. The set makes the leakage of data to the cloud server minimal. The pseudorandom function is collision resistant and it maps the location of blocks to unique location.

Mohammed Faez et. al. [5] proposed a model to check the integrity and privacy of the data stored in the cloud server.

They discussed various issues such as verification of integrity and preservation of privacy of the data stored in cloud server when a untrusted third party is used to examine the privacy and integrity of the documents outsourced to the cloud. The ability to find any modifying of clients' data and violation of data privacy and integrity results in the damage of the document and clients' data. They check for the changes in the document which may result in obtrusion to data privacy as well as integrity.

Acklyn Murray et. al. [7] surveyed various cloud models and identified the security issues in each model. They have discuss various encryptions such as Cloud Data Encryption, Homomorphic Encryption and Access Control. Cloud service model such as Infrastructure as a Service (IaaS), Software as a Service (SaaS) and Platform as a Service (PaaS) are discussed their vulnerability is checked using the vulnerability checker software. Simulations are done to analyse the degree of vulnerability.

Lo'ai Tawalbeh et. al. [9] proposed a secure and efficient cloud computing framework. Security of data outsourced to the cloud is one of the issues in cloud systems. Data privacy and integrity has to be ensured to manage the data outsourced to the cloud. They have proposed a model to efficiently outsource data to the cloud. The problems with the existing encryption mechanisms such as Rivest Shamir Adleman (RSA) and Data Encryption Standard (DES) are identified. Extensive simulation suggests that Advanced Encryption Standard (AES) encryption is better than the existing security algorithms.

Niteen Surv et. al. [10] proposed a framework for client side AES encryption to improve the security of the existing system.DES algorithm was used for security but too many attacks on the cipher have made it weak and not usable. 3DES encryption is an upgrade of DES algorithm which used three levels of DES encryption. 3DES Encryption is slower in performance compared to other encryption block cipher methods. 3DES has low performance in terms of power usage and result generation and it takes triple more time to encrypt than DES. AES encryption technique is flexible, fast and secure block cipher method. It is supported even in small devices and almost every platform. All algorithms have their own set of disadvantage and overheads. These encryption algorithms have privacy concerns over data encryption and decryption, transmission of data, which may result in leakage of data to unauthorised parties and hackers. Invalid authorization may result in accessibility to cloud data which may result in insecure data transmission and storage.

III. SYSTEM ARCHITECTURE

The blind storage is an effective mechanism to abstract the storage details. The information outsourced to the cloud might be sensitive and the method of blind storage improves the security and acts as a layer of abstraction. The system architecture comprises of seven modules distributed over the client and the server side. The system has two phases: client and server phase. Fig. 1 shows the modules of the architecture and the process flow of proposed system. In the client side, pre-processing processes such as registration of user to the system, authentication of the user and selection and division algorithms are implemented. Hence, in the keys are defines and input is encrypted. Authentication is performed before storage. Key selection is perform based odd and even key. Block size is defined to provide maximum possible load to the storage. In the server side, the process of indexing and storage of blocks over the database system is done. The document is represented as set of blocks. The index selection is used to identify the storage location of the blocks.

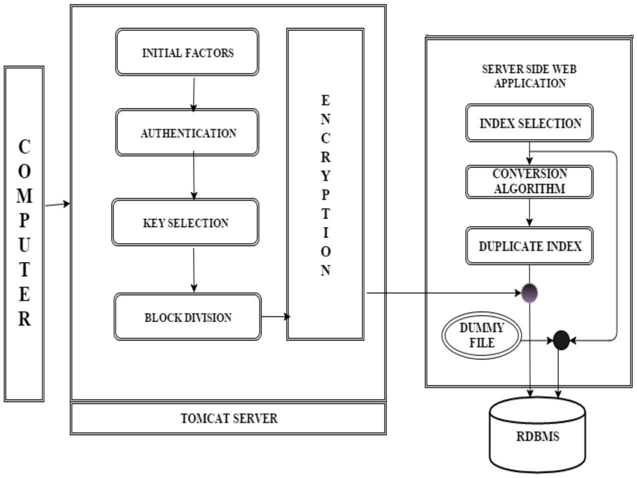

Fig. 1. System Architecture.

The conversion algorithm takes the index as input and generates another set of index. The output of the conversion algorithm provides the list of indexes. The index selection algorithm and conversion algorithm are the two main aspects. The index selection algorithm is used to generate the index in to which the blocks are to be inserted.

IV. SECURED CLOUD DATA STORAGE ALGORITHM USED IN CLIENT SIDE

```
For each user
Step 1: Get the initial factors such as email id, native place, date of birth and user id.
Step 2: Get the password for their profile.
Step 3: Encrypt the initial factors using AES algorithm. AES encrypt (email id, native place, date of birth, user id)
Step 4: Write the encrypted details using output stream OutputStream.WriteTo (input.txt)
Step 5: Encrypt the password using AES algorithm. AES encrypt (password)
Step 6: Write the encrypted details using output stream OutputStream.WriteTo (Password.txt)
Step 7: The user has to authenticate using the password. Authenticate (password, AESdecrypt ( password))
Step 8: Generate the Key based on the initial factors.
    Odd Key: Concat ( "P",substr(DOB,3,5), "R", substr (DOB,1,2), substr(native place,n-3), "A",substr (regid,1,7))
    Even Key: Concat( substr (native place,n-3), "P", substr (DOB,3,5), "R",  substr( DOB,1,2),"A", substr (regid,1,7))
Step 9: Define a factor K.
Step 10: Divide the document into block of size K.
Step 11: Encrypt the block using Odd and Even Key.
End for
```

Algorithm for Index Selection

```
For each encrypted document
do
    Create master_index
    While document
        select a random number master_index(block no/table name/entry no/totalblocks)
    End
    If (master_index)
        Add(dummy files)
    End if
End for
```

Algorithm for Conversion

```
For encrypted document do
If (master_index)
    Conversion(master_index)
    return duplicate_index
End if
If duplicate_index
    Add (original document)
End if
End for
```

V. PERFORMANCE ANALYSIS

The project is implemented in java using Eclipse IDE. Spring framework is used to ease the compatibility options with the cloud setup. Spring framework uses only. Because of the object oriented characteristic of spring, hibernate is preferred for storage. Hibernate is used to connect the hosted SaaS application to the MySQL database. The web application is hosted in Apache Tomcat server. The server is used to simulate the real time server. The application follows the MVC model. The implementation is divided into three category: model, view and controller. Model is used to define the blueprint of the application. The controller controls all the actions of the application. The result is displayed in the view model of the application.

The Table 1. Compares the performance of the existing system with proposed system by plotting the time taken for uploading and downloading a document against file size. The table compares the time taken to upload and download a document before and after encryption.

TABLE I. PERFORMANCE ANALYSIS

File Size (MB)	Time Taken to Upload (ms)		Time Taken to Download (ms)	
	Before Encryption	After Encryption	Before Encryption	After Encryption
2.68	1749	1749	491	491
4.8	2469	2470	314	314
7.6	2577	2580	423	425
10	2835	2840	525	530
13	5850	5858	668	672
14.5	5192	5197	664	669
16	5185	5192	719	727
19	8772	8779	941	949
25.6	10803	10812	1134	1142
28.9	18033	18045	1342	1355

Fig. 2 compares the time taken to upload a document against file size of the existing system with proposed system. The time taken for the whole document to be outsourced is upload time. The difference is calculated for the existing system and proposed work. The data is plotted as a graph comparing the existing system with the proposed system.

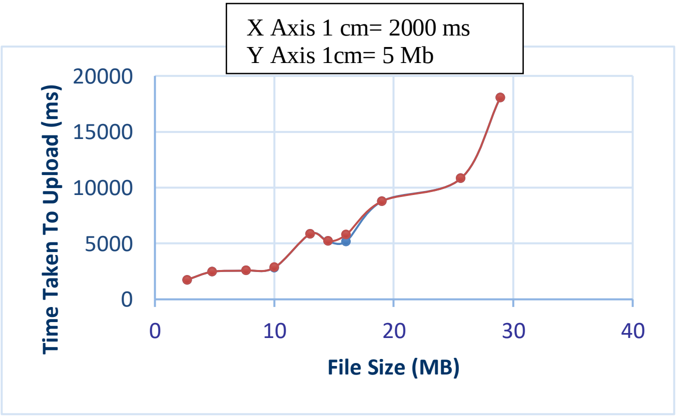

Fig. 2. Upload Performance Analysis

Fig. 3 compares the time taken to download a document against file size of the existing system with proposed system. The time taken for the whole document to be downloaded is download time. The download time is calculated by finding the difference between initial times of request for downloading to the completion of request. The comparison shows that the performance of the proposed system is same as that of the existing system. The Security is improved without any hindrance to the performance.

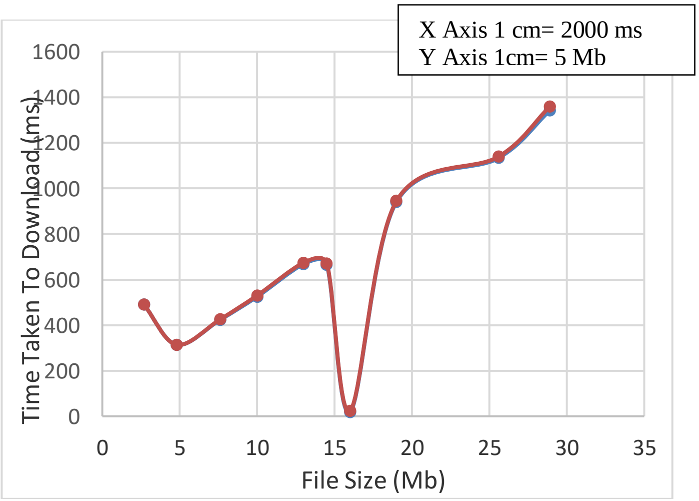

Fig. 3 Memory Analysis for Static Document

VI. Conclusion

In this paper, a new method for applying blind storage to relational database management system is proposed in this work in order to ensure the privacy of data and in the same time storing the data in easier way. The proposed system combines the security features of existing system with the implementation of blind storage. Blind Storage abstracts the storage of documents in database. Blind storage improves the privacy of data outsourced in comparison with other existing system. The present system can be enhanced such that it uses private keys which provide better data privacy and security.

References

[1] Hongwei, Lil., Dongxiao, Liu1., Yuanshun, Dai1., Tom, H., Luan, Xuemin, (Sherman), Shen.: Enabling Efficient Multi-keyword Ranked Search over Encrypted Mobile Cloud Data through Blind Storage. IEEE transactions on Emerging topics in Computing. 3, pp. 127-137, 2015.

[2] Ganapathy, S., Kulothungan, K., Muthurajkumar, S., Vijayalakshmi, M., Yogesh, P., Kannan, A.: Intelligent feature selection and classification techniques for intrusion detection in networks: a survey, EURASIP-Journal of Wireless Communications and Networking - SpringerOpen Journal. 271, pp. 1 – 16, 2013.

[3] Naveed. M., Prabhakaran, M., Gunter, C. A.: Dynamic Searchable Encryption via Blind storage. IEEE Symposium on Security and Privacy. pp. 639-654, 2014.

[4] Muthurajkumar, S., Ganapathy, S., Vijayalakshmi, M., Kannan, A.: Secured Temporal Log Management Techniques for Cloud. Procedia Computer Science. 46, pp. 589–595, 2015.

[5] Mohammed, Faez, Al-Jaberi., Anazida, Zainal.: Data Integrity and Privacy Model in Cloud Computing. International Symposium on Biometrics and Security Technologies. pp. 280-284, 2014.

[6] Muthurajkumar, S., Vijayalakshmi, M., Kannan, A.: Intelligent Temporal Role Based Access Control for Data Storage in Cloud Database. 2014 Sixth International Conference on Advanced Computing(lCoAC), IEEE Digital Library. pp. 184-188, 2015.

[7] Acklyn, Murray., Geremew, Begna., Ebelechukwu, Nwafor., Jeremy, Blackstone., Wayne, Patterson.: Cloud Service Security and Application Vulnerability. Proceedings of the IEEE Southeast Conference. Fort Lauderdale, Florida. pp. 1-8, 2015.

[8] Muthurajkumar, S., Vijayalakshmi, M., Kannan, A.: Energy Efficient and Optimal Cloud Storage Algorithms for Temporal Query Processing in Cloud Databases. Asian Journal of Research in Social Sciences and Humanities. 6, pp. 368-382, 2016.

[9] Lo'ai, Tawalbeh., Raad, S., Al-Qassas, Nour., Darwazeh, S., Yaser, Jararweh., Fahd, AlDosari.: Secure and Efficient Cloud Computing Framework. International Conference on Cloud and Autonomic Computing. pp. 291-295, 2015.

[10] Niteen, Surv., Balu, Wanve., Rahul, Kamble., Sachin, Patil., Jayshree, Katti.: Framework for Client Side AES Encryption Technique in Cloud Computing. IEEE International Advance Computing Conference. pp. 525-528, 2015.

[11] Wang, B., Yu, S., Lou, W., Hou Y. T.: Privacy-Preserving Multi-keyword Fuzzy Search over Encrypted Data in the Cloud. In Proc. IEEE INFOCOM. pp. 2112-2120, 2014.

[12] Muthurajkumar, S., Vijayalakshmi, M., Kannan, A.: Intelligent Trust Based Temporal Data Storage and Retrieval Methods for Cloud Databases. AENSI Journals in Advanced in Natural and Applied Sciences. 9, pp. 123 – 128, 2015.

[13] Wang, Qian., Cong, Wang., Kui, Ren., Wenjing, Lou., Jin, Li.: Enabling Public Auditability and Data Dynamics for Storage Security in Cloud computing. IEEE Transations on Parallel and Distributed Systems. 22 (2011) 847-859, 2011.

[14] Sethukkarasi, R., Sannasi, Ganapathy., Yogesh, Palanichamy., Arputharaj, Kannan.: An Intelligent Neuro Fuzzy Temporal Knowledge Representation Model for Mining Temporal Patterns. Journal of Intelligent and Fuzzy Systems. 26, pp. 1167-1178, 2014.

Muthuswamy, Vijayalakshmi., Arputharaj, Kannan.: Proactive Location-based Context Aware Services using Agents. IJMC. 7, pp. 232-252, 2009.