Cloud Data Security using Homomorphic Encryption

Sohit Simon Mecwan
Department of Computer Science and Engineering
Jagad Guru Dattatray College of Technology,
Indore, MP, India

Khusboo Sawant
Department of Computer Science and Engineering
Jagad Guru Dattatray College of Technology,
Indore, MP, India

ABSTRACT
The security is primary need of any data storage system. In this work the cloud data storage and their security is the key area of investigation. The cloud data is always mobile by nature because the cloud infrastructure providers are host or move their data from one storage to another for reducing maintains cost of server. In this context the security and privacy of data is key concern or service provider and data owner. In order to keep secure data on cloud the cloud service providers to encrypt the data over server in cryptographic format. The cryptography is effective manner of security because it requires some key to recover the original data. But the key management policies and key generation techniques are key factor of deciding the security depth of any cryptographic cloud. The proposed work favors the Homomorphic key cryptographic technique for securing the data on cloud. Therefore the Paillier algorithm is proposed for implementing with the cloud. The Paillier algorithm is an asymmetric key encryption technique which first generates the pubic key and private key for securing the data. But the technique is not much suitable for the text data cryptography therefore the data is converted in two equivalent numerical value and the cryptographic operation is performed. After encryption the data is preserved on file on cloud storage. The security of the proposed cryptographic technique is demonstrated using the three applications i.e. file upload, download and the sharing of files among the cloud server users. The implementation of the proposed secure cloud hosting is performed on JAVA technology. After implementation the performance of the algorithm is computed in terms of time consumption, memory consumption, and server response time. The results demonstrate the proposed technique is acceptable for data hosting on cloud.

Keywords
Cloud Computing, Homomorphic Encryption, Security, Cloud Storage, Cryptography, Cipher-text, and plaintext

1. INTRODUCTION
Cloud is a huge infrastructure for providing the computational and storage service to their clients. According to the new generation requirements and new devices the need of storage is also increases much rapidly. The users are usages the cloud storage for storing their personal and confidential data. Therefore the data owner is worried about the data and its confidentiality. On the other hand for managing the data service provides are also outsource the data to other servers and they are responsible for any security thread and leakage therefore they are also worried for data and their availability.

In this context the cloud server providers are offering the data storage is secured using the cryptographic techniques. The different kinds of security techniques such as identity based encryption, attribute based encryption techniques, symmetric key encryption techniques are offered for securing and preserving the data on cloud. But due to different issues such as attribute management, authority overhead, key generation issues all these techniques are not much adoptable for securing the data on cloud. In this presented work the Homomorphic encryption technique is proposed for securing the data and their advantage and disadvantage is tried to measure. Therefore the secure cloud storage is proposed for host the user data on cloud storage, download and search options on the cryptographic data.

2. PROPOSED WORK
The cloud is one of the technology which is frequently accessed now in this days. Now only for computational ability its now in these days also be used for storing data. In this presented work the cloud data storage and security is primary area of investigation. This chapter includes the system using which the proposed security system is demonstrated

2.1 System Overview
The security is need of the remote data hosting and communication. Cloud service providers are continuously receiving data from their client to host on server but form reducing the effort of data management they preserve this data on others hosting providers. Therefore Cloud servers are securing the data on their own end by cryptographic techniques. The cryptographic techniques are frequently used techniques data security due to low cost of implementation and maintenance. The cryptographic techniques which are used with for security is need to be updated day by day because various attacks can be applied for braking them. In addition of that the encryption algorithms are not much secure due to their key management techniques. In this context the proposed work is motivated for securing the data using the Homomorphic cryptographic technique.

The proposed technique is usages the Paillier algorithm for securing the data on cloud. Therefore the user first encrypt the data using the Paillier algorithm and store it on the server. in addition of that for searching data on cloud using this cryptographic system a secure search system is also implemented which usages the SHA1 algorithm for comparing the user keywords and the data base keywords. Finally the data is searched and shared between users to make utilizable. The shared data can be downloaded using the other users end for demonstrating the security and sharing mechanism of the proposed system. This section provides the formal overview of the proposed work and the next section includes the detailed methodology of the proposed cryptographic security on cloud.

2.2 Methodology
The proposed methodology is described in this section which includes the different component of the model which is used to process the data one by one. The proposed technique is described in three major modules: namely upload, download and data sharing.

2.2.1 File Upload

The model is divided into three major modules with the application. In this section the process of file upload process is described. Figure 2.1 shows how the file is securely uploaded on the cloud server.

Input file: the file which is needed to upload on cloud is selected from local computer. That file is which is secured on the cloud server. In this presented work the text file is used for demonstration.

Read file: in this phase system read the input text file and the content is extracted. Therefore the file data is used here for processing the system.

Extract term frequency: in this phase the text mining approach is used for finding the essential keywords. In this context the data is used for finding the word frequency the word frequency of the available words is computed using the following formula:

$$\text{word frequency} = \frac{\text{word counnt}}{\text{total words in file}}$$

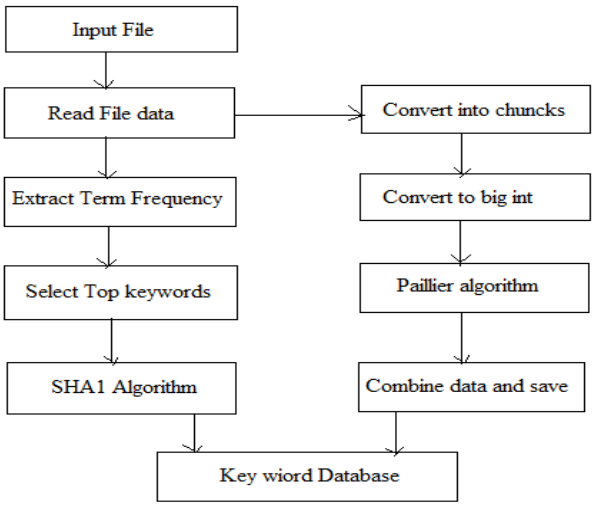

Figure 1: File Upload Process

The word frequency is obtained here between 0-1 which works as the probability of keyword occurrence.

Select top key words: now the word frequency is used for selecting the available keywords in the file. These keywords are used for searching data from the database for sharing and download.

SHA1 algorithm: SHA1 algorithm is a hash generation algorithm that provides the 160 bit hash for each input text bock. All the keywords selected from the text file is used with the SHA1 algorithm.

Keyword database: the SHA1 hash codes are stored in the database for utilizing the keywords for making search on database for finding the file. The keyword data base is prepared in the following structure as given in table 1.

Table 1: keyword database

File name	SHA1 hash	User name

Convert to chucks: the input text file is used in this phase where the entire text contents are sub divided into the small chunks. These chunks are further used for encryption and storage of text file.

Convert to big integer: the available chucks are available in text format which is not appropriate with the Paillier algorithm. Therefore all the chunks of the file is converted into big integer format.

Paillier algorithm: the Paillier algorithm is an asymmetric key encryption technique which usages the concept of public and private key. The algorithm is first generate both the pairs of key and encrypt the chuck data which is converted into big integer.

Combine and save file: the small chucks of the data is now combined in a file and stored in the cloud server.

2.2.2 File Download

After uploading the file into the server, user need to search the file and download when required to use. This process of data search and download of file is demonstrated in figure 2.

User keywords: for making search for the file user provide the keywords which is available in database. These keywords can be similar which is used to recognize the file from the file storage.

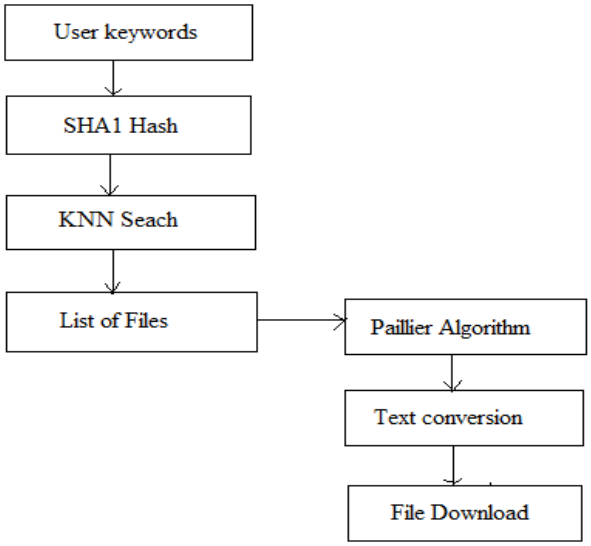

Figure 2: Data Download Process

SHA1: The user keyword which is used to make file search is used with the SHA1 hash generation algorithm and for each user keyword the SHA1 hash code is computed.

KNN: the KNN is also termed as the k-nearest neighbor. That accepts two initial parameter, for searching the data relevant form the data base. The first input is data base sequence and the sequence is user keyword sequence. Similarly the system accepts the user keyword hash codes as the query input and the database which is previously converted into the hash code. The KNN algorithm finds the relevant outcomes form the data according to the supplied keywords.

List of files: the KNN results the list of files which contains the user keywords by comparing the user key words hash codes and the database hash codes. Among the user select a file which is required to download.

Paillier algorithm: the selected file by user is a encrypted file using the Paillier algorithm. Therefore to decode the data Paillier algorithm is implemented in decoding model. The file data is again converted into small chunks and produced to the Paillier algorithm. Using this process all the data is decrypted and their equivalent numerical data is prepared.

Text conversion: the data is in numerical format is now converted into the text format and new decrypted file is organized. That file is downloaded to the end users file system

2.2.3 File Sharing

The additional operation is also demonstrated in the proposed secure cloud hosting server which involves the process of file sharing.

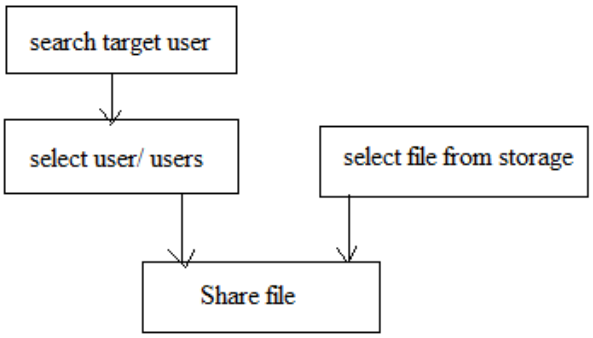

Figure 3 Data Sharing

The file sharing application of the cloud storage is demonstrated using figure 3. In this diagram the user first search the user target user or list of users who are wants the files. After selecting the appropriate user the file owner select the uploaded file form the file list. This file is shared between the uses by simple tagging of file. Now the n number of selected users can utilized this file according to their use.

2.3 This Proposed Algorithm

The proposed system involve two major task in the demonstrated cryptosystem namely encryption and decryption. The encryption and decryption algorithms are described in this section as:

Table 2: Encryption Algorithm

Input: file to encrypt F Output: encrypted file E
Process: 1. $R = readfile(F)$ 2. $CH_n = R.splitFile(R)$ 3. $[K_p, K_{pu}] =$ Paillier. gennrateKeyPair 4. $for(i = 1; i \leq n; i++)$ a. $C_i = convertToNumber(CH_i)$ b. $EC_i =$ Paillier. encrypt(C_i, K_p, K_{pu}) c. $E.appendData(EC_i)$ 5. $end\ for$ 6. Return E

Table 3 includes the process of decryption of the proposed cloud storage system.

Table 3 Decryption Algorithm

Input: encrypted file E Output decrypted file F
Process: 1. $R = readFile(E)$ 2. $EC_n = splitFile(R)$ 3. $for(i = 1; i \leq n; i++)$ a. $C_i =$ Paillier. decrypt(EC_i, K_p, K_{pu}) b. $D_i =$ convertTotext(C_i) c. F. append(D_i) 4. end for 5. return F

3. RESULT ANALYSIS

The experimental evaluation and the system performance is computed and demonstrated in this chapter. Therefore some essential performance parameters are obtained and listed with their obtained observations.

3.1 Encryption Time

The amount of time required to perform encryption using the selected algorithm is termed as the encryption time of the system. The encryption time of the proposed system is demonstrated using figure 4

$$\text{Time consumption} = \text{End Time} - \text{Start Time}$$

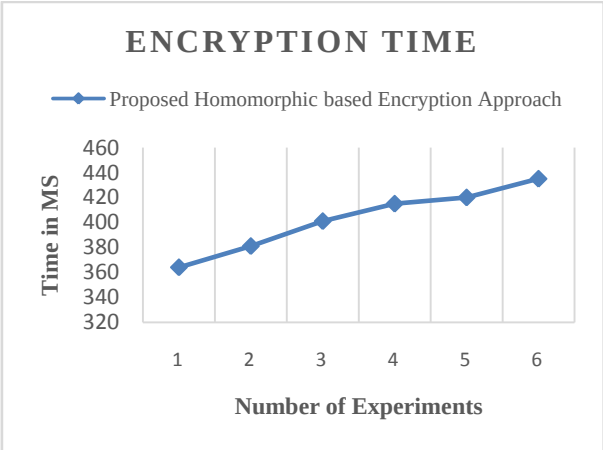

Figure 4: Encryption Time

In order to show the performance of implemented systems the encryption execution time is reported in figure 4 and table 4. In this diagram the X axis shows the different file size on which different experiment values performed and the Y axis shows the amount of time consumed for encrypting the input text file. Additionally the performance of proposed system is given using blue line. According to the given results the proposed system consumes less time for file uploading. Additionally the results shows the amount of time consumed is depends on the amount of data provided for execution. Moreover, while using proposed data security, enhance the security for cloud storage publicly.

Table 4: Encryption Time

Number of Experiments	Proposed Homomorphic based Encryption Approach
1	364
2	381
3	401
4	415
5	420
6	435

3.2 Decryption Time

The amount of time required to recover (Decrypt) the original data from the cipher text is known as the decryption time of the algorithms. The figure 5 and table 5 shows the obtained performance of the system in terms of millisecond. To show the performance of security system the blue line shows the performance of proposed algorithm.

$$\text{Time consumption} = \text{End Time} - \text{Start Time}$$

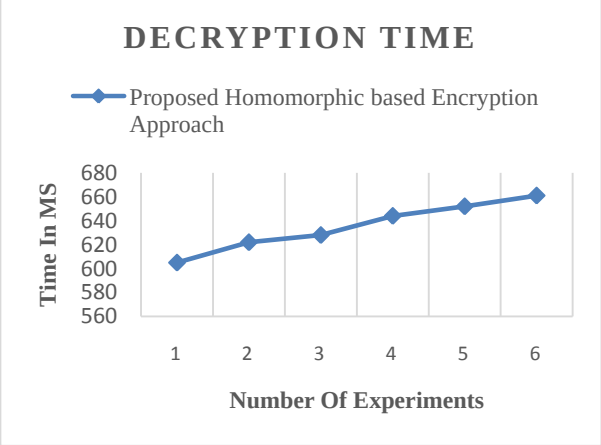

Figure 5: Decryption Time

In given figure 5, X-axis shows the different numbers of experiments are performed and the Y-axis shows the amount of time consumed for decryption process. According to the generated observations the encryption time is higher than the decryption time in the system, but the decryption time of the proposed algorithm is much adaptable.

Table 5: Decryption Time

Number of Experiments	Proposed Homomorphic based Encryption Approach
1	605
2	622
3	628
4	644
5	652
6	661

3.3 Encryption Memory

The amount of main memory required to execute the algorithm with the input amount of data is known as the encryption memory. The total memory consumption of the algorithm is computed using the following formula.

$$\text{Consumed Memory} = \text{Total Memory} - \text{Free Memory}$$

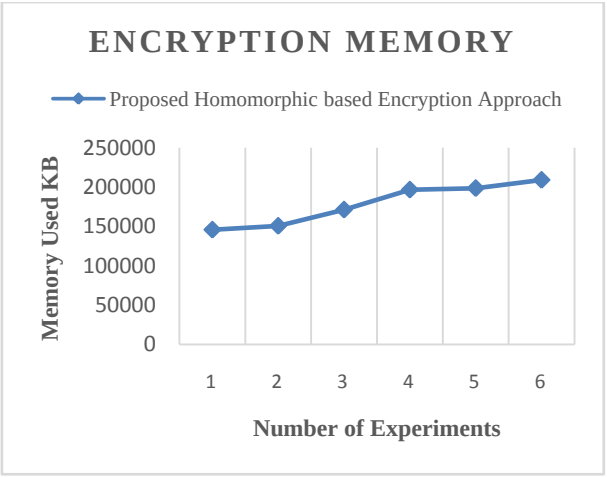

Figure 6: Encryption Memory

The figure 6 and the table 6 show the encryption memory consumption of the system. In this diagram the amount of main memory consumed is given in Y axis and the number of experiments are reported in X axis. According to the obtained results the proposed algorithm consumes fewer resources as we seen during the execution of algorithm.

Table 6: Encryption Memory

Number of Experiments	Proposed Homomorphic based Encryption Approach
1	145775
2	150625
3	171265
4	196631
5	198513
6	208996

3.4 Decryption Memory

The amount of main memory required to recover the original file from the cipher text is known as the decryption memory consumption. The figure 7 and table 7 shows the Amount of main memory consumed during the data recovery

$$\text{Time consumption} = \text{End Time} - \text{Start Time}$$

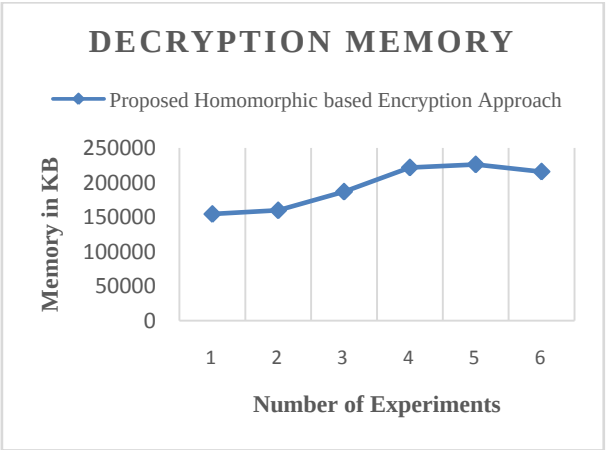

Figure 7: Decryption Memory

In this diagram the X axis shows the different file size used for decryption and the Y axis shows the amount of main memory consumed during the decryption. According to the

obtained results the amount of main memory used is less than of encryption memory and consume less space of proposed algorithm.

Table 7: Decryption Memory

Number of Experiments	Proposed Homomorphic based Encryption Approach
1	154330
2	159629
3	186620
4	221542
5	225961
6	215621

3.5 Server Response Time

The amount of time required to produce the outcome after making the request from the server is termed as the server response time. The response time not included the encryption or decryption activity during these measurements. The computed response time for proposed cryptographic technique is shown in figure 8 and table 8

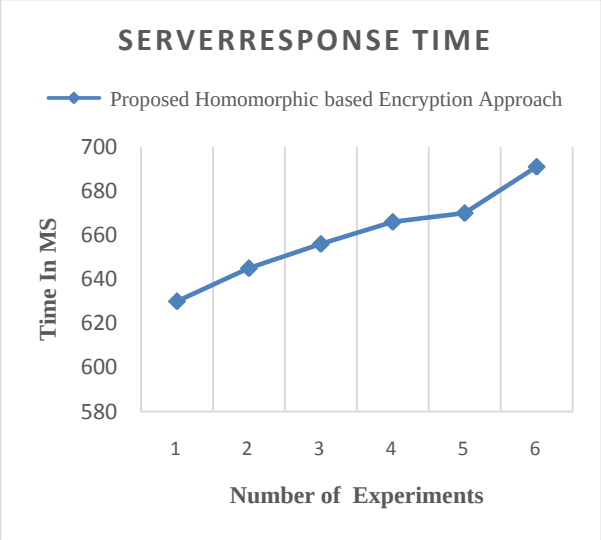

Figure 8: Server Response Time

X axis of this diagram contains the amount of experiments performed using the system and the Y axis shows the amount of time required for generating the response through the server. This can also term as the communication overhead for the system. According to the computed results the response time is not depends on the amount of file size or other parameters. That is directly depends on the amount of work load on the target server where the data is stored or the application is hosted.

Table 8: Encryption Time

Number of Experiments	Proposed Homomorphic based Encryption Approach
1	630
2	645
3	656
4	666
5	670
6	691

4. CONCLUSION

The proposed work is intended to provide study of the Homomorphic cryptographic technique in cloud environment. This chapter provides the summary of the work conduced and the experiments performed. In addition of that the future work is also included in this section.

4.1 Conclusion

The cloud is become more usable technology now in these days, not only the big organizations even the mobile users are utilizing the service of cloud for storing their mobile data on cloud. Cloud computing now in these days not only supports the efficient computing experience even it also offers the scalable storage. The data on cloud is hosted and accessed by the number of users therefore the cloud service provider's move their data on the third party storage providers to reduce the data management issues. Therefore the security and privacy is a significant concern of cloud storage. In this context the cloud service provides are offers their clients to manage their data on cloud using the cryptographic security. In study it is found that not all the cryptographic security is suitable for the data the data can be leaked or recovered by hackers in different cryptographic secure techniques too.

Therefore the proposed technique is proposed to incorporate the Homomorphic cryptographic technique securing data on cloud. The proposed technique demonstrates the file encryption, decryption, sharing and search of data on cryptographic cloud. The entire system includes the implementation of SHA1 hash key generation and search process for the secure data search on the cryptographic data. In addition of that for preserving data on cloud storage the Paillier algorithm is used for encryption and decryption process. Basically the Paillier algorithm is not suitable for the text data encryption therefore the entire data is first converted into a small text strings and then the data is converted into relevant numerical values this numerical values are used to encrypt and store the data during the decryption the numerical data is recovered first and then converted into the text strings. The proposed method is therefore highly secure for data encryption in cloud.

The technique is implemented using the JAVA technology and using JSP for web application deployment. After implementation the performance of the algorithm is computed in terms of time and space complexity. The table 9 contains the performance summary of the proposed cryptographic cloud.

Table 9: Performance Summary

S. no	Parameters	Remark
1	Encryption time	The encryption time is acceptable for secure data storage service and increase with the amount of data of to be encrypt
2	Decryption time	The decryption time is less than the decryption time
3	Encryption memory	The memory consumption is acceptable for encryption process
4	Decryption memory	The memory consumption of decryption algorithm is less than the encryption process
5	Response time	Efficient response time even the server implement the cryptographic scenarios

The proposed technique is secure and efficient for cloud data storage, search and sharing purpose thus the proposed system is acceptable for data hosting.

4.2 Future Work

The proposed work is a secure and effective technique of data security over the cloud servers but still the some improvements are remaining to enhance the cloud data security. Therefore the following area of work is proposed for future extension of the work.

- ✓ The proposed work implements the Paillier cryptosystem which is asymmetric technique of data security but the technique is not much efficient for large amount of data therefore in near future the efficient is key area of work
- ✓ The proposed work need to modify more for text based cryptography because the Paillier algorithm is suitable for numerical data encryption.

5. REFERENCES

[1] Ding, Manish M Potey, C. A. Dhote, and Deepak H. Sharma, "Homomorphic Encryption for Security of Cloud Data", Procedia Computer Science 79 (2016): pp. 175-181.

[2] Sookhak, Mehdi, et al. "Dynamic remote data auditing for securing big data storage in cloud computing." Information Sciences 380 (2017): 101-116.

[3] Aarti P Pimpalkar and H.A. Hingoliwala, 'A Secure Cloud Storage System with Secure Data Forwarding', "International Journal of Scientific & Engineering Research", Volume 4, Issue 6, June-2013, page no-3002-3010

[4] Jinesh varia," AWS Cloud Security Best Practices", "White Paper", November 2013

[5] Luit Infotech Private Limited, "What is Cloud Computing", available online at: http://www.luitinfotech.com/kc/what-is-cloud-computing.pdf

[6] B. Grobauer, T. Walloschek, and E. Stöcker, "Understanding Cloud Computing Vulnerabilities". 2011 IEEE Security and Privacy, pp. 50-57.

[7] S. Zhang, S. F. Zhang, X. B. Chen, and X. Z. Huo, "Cloud Computing Research and Development Trend," In Proceedings of the 2010 Second International Conference on Future Networks (ICFN '10). IEEE Computer Society, Washington, DC, USA, pp. 93-970

[8] J. F. Yang and Z. B. Chen, "Cloud Computing Research and Security Issues," 2010 IEEE International Conference on Computational Intelligence and Software Engineering (CiSE), Wuhan pp. 1-3, Dec. 2010.

[9] Kaufman, Lori M., "Data security in the world of cloud computing", IEEE Security & Privacy 7, no. 4 (2009).

[10] Pradip Lamsal, "Understanding Trust and Security", Department of Computer Science University of Helsinki, Finland, 20th of October 2001

[11] V. Abricksen, "A Survey on Cloud Computing and Cloud Security Issues", International Journal of Engineering Research and Applications (IJERA), , International Conference on Humming Bird (01st March 2014).

[12] Mohammad Asadullah and R. K. Choudhary, "Data Outsourcing Security Issues and Introduction of DOSaaS in Cloud Computing", International Journal of Computer Applications (IJCA), PP. 40-45, Volume 85 – No 18, January 2014.

[13] Protect Data Privacy, http://www-01.ibm.com/software/data/optim/protect-data-privacy/.

[14] Data Encryption, Post note, Online: http://www.parliament.uk/documents/post/postpn270.pdf [Accessed on 25 October 17].