International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 20 (2018) pp. 14799-14805

Secure Data Sharing Scheme using Cryptographic Algorithm for Cloud Storage

Sheenal Malviya[1], Sourabh Dave[2]

Computer Science Department, Medi-capsUniversity, Indore, Madhya Pradesh, India.

Abstract

Data integrity and data confidentiality are two important requirements for open cloud environment. Cloud computing has emerged as a long-dreamt vision of the utility computing paradigm that provides reliable and resilient infrastructure for users to remotely store access data and use on-demand applications and services. In this paper, we design and developed Secure Data Sharing Scheme for dynamic groups in public cloud environment. In this scheme, user is able to share data with others in the group without revealing characteristics privacy to the cloud. Moreover, this approach supports efficient user deletion and fresh user joining. Hence, efficient user deletion can be achieved through a public revocation list without harming security of the other remaining users in user portal. In addition, the storage overhead and the encryption decryption computation time is minimized even multiple groups are requested for the file access. Widespread analyses of our data sharing scheme which show that this is satisfies the desired security requirements along with the secure sharing with other and preserve privacy policy when group sharing is processed that guarantees efficiency as well.

Keywords: Cloud Computing, Homomorphic Encryption, Data Security, Secure Sharing, Encryption, Decryption.

INTRODUCTION

Cloud computing is the long dreamed vision of computing as a utility, where data owners can remotely store their data. The basic service provided by the Cloud is Data Storage. However, it is a difficult task for sharing data in multi-owner manner where group admin and all group members can store and modify data while preserving data and identity privacy from an un-trusted cloud server, due to the frequent change of the membership. Many of the public cloud computing services have appeared for data storage in group applications. Two important problems that arise when sharing group data in public cloud are the privacy and security of group member's data. Cloud service providers are separate administrative entities and users don't have access to the cloud internal operational details. Because of the semi trust nature of cloud service provider, the traditional security technologies cannot be directly applied to the public cloud based group data sharing applications. So that, Data sharing is increasingly important for many users and sometimes an essential requirement, especially for industries and societies used to gain proceeds. Sharing group resource among cloud users is a major problem, still the data privacy leak. Most of the traditional systems are use Group Key Management method for sharing Key Generation and distribution in the group member or users. Sometimes change to user one group to another group, the group key to enable authenticated users to access the files securely and efficiently is still a challenging problem. Here we are proposing a security framework for group data sharing that make accessible to data file in secure manner in public cloud environment

There are a number of different approaches that exist for securing the data on network & among them the cryptography is a widespread and classical approach to secured data. Furthermore the key reason behind use of cryptography for security is their low cost implementation and freedom and springiness to change the security according to needs. Therefore, in this paper key area of work is investigated and design of a secure data sharing techniques for cloud storage.

PROPOSED WORK

This section introduces the functional characteristics of the proposed system. In addition of that the core system design concept and the algorithm steps are explained in detail.

A. 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 units: Admin, Cryptographic Server and User.

The user need to share data over the open cloud without any security for access of data. Therefore, in thus project work we have proposed secure data sharing scheme for the cryptographic cloud. This approach can be made of three entities. The entities are following:

1. Admin Portal
2. User Portal
3. Cryptographic Server (CS) Portal

This entities perform different task according to their responsibility. Firstly,

1. Admin Portal: The admin entity is independent to the CS portal and user portal. The admin entity is responsible for handle activity of all user that means different users are sharing their file and have the access permission of system privilege. In this panel, admin have to right to approved or decline newly registered user request. If admin granted the authorization to new user, the user can access log in to their panel and access privilege i.e. upload, download and share file.

Admin can also view all files as stored by all user and activity of sharing mechanism. There are unlike number of user listed in user portal also view by the admin portal.

2. User Portal:

The users are the clients which accessed services of the cloud storage. A user is the single unit which are for accessing data storage functionality after ensured of security for his sensitive data on public cloud. On user section, different action is perform concurrently from file uploading/ downloading/updating. The overall process of user portal are demonstrated using figure 1.

In this section, to access different functions on user entity, initially user need to login via their secret identifications e.g. (User name, login Id, Password). Once the user is successfully login then it will redirect to user home page. If there are multiple user are already logged in then can create group for sharing of data securely. Hence all group user can share file among all user. Finally, created group will have a access and sharing permission for internally depending upon what right permission need by the user

In the second scene, user input text file for access and sharing purpose to other user. Moreover, users uploaded this input file on the cryptographic server. For this, group user need access permission by the admin to store file o server. If Admin permitted user access request then he can send file encryption request to cryptographic server. *How file will be encrypted by CS?* We will explain in next segment.

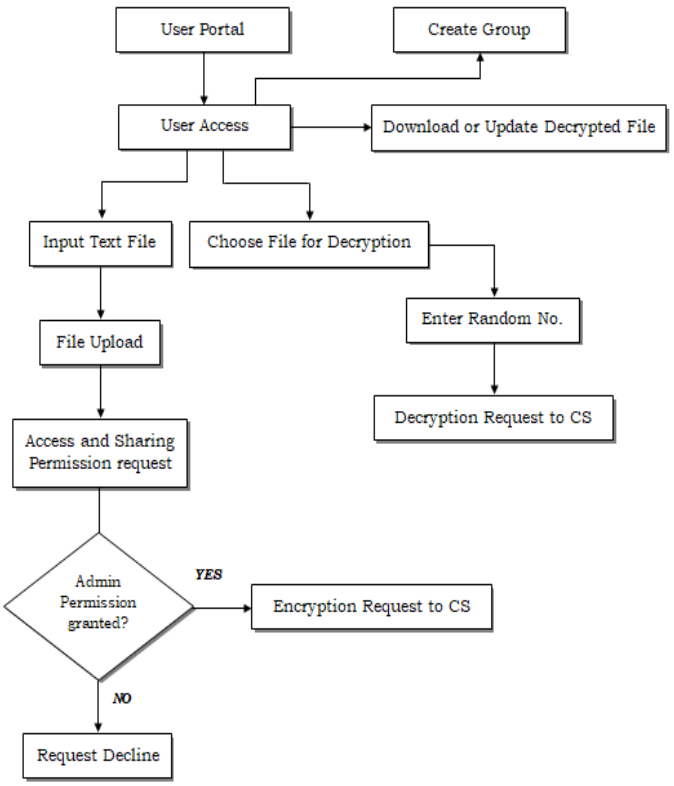

Figure 1. User Portal Activity

Consequently, once file will be encrypted and ready to decrypt and download, this action will processed by user. First the user need to choose file which he want to decrypt and download, then for security assurance of legitimacy of user, already generated random number by the CS at the time of encryption time, the random number will mail to the users email id. After entering this random number, decryption request is send to the CS portal.

In third phase, user want to download file, then he can also download along with edit/view file. In the next phase we explain encryption, decryption and key management process maintained trusted party i.e. cryptographic server (CS) portal.

3. Cryptographic Server Process

The CS is a trusted third party and is responsible for key management process, encryption, decryption, and access rights of users. The CS generates the symmetric key and encrypts the data with the generated key. The cryptographic server maintain all possibility of the user request and sharing types. The CS is assumed to be a secure object in our proposed work. The CS can be maintained by an organization/company or can be owned by a third-party provider. However, the CS maintained by an organization will generate more trust in the system for end user applications. The proposed approach maintain single cryptographic key for each data file of user.

The diagram 2 depicts the overall process of the encryption, decryption, and key management. The user entity and CS portal are both dependent to each other for completion to entire data security and sharing process.

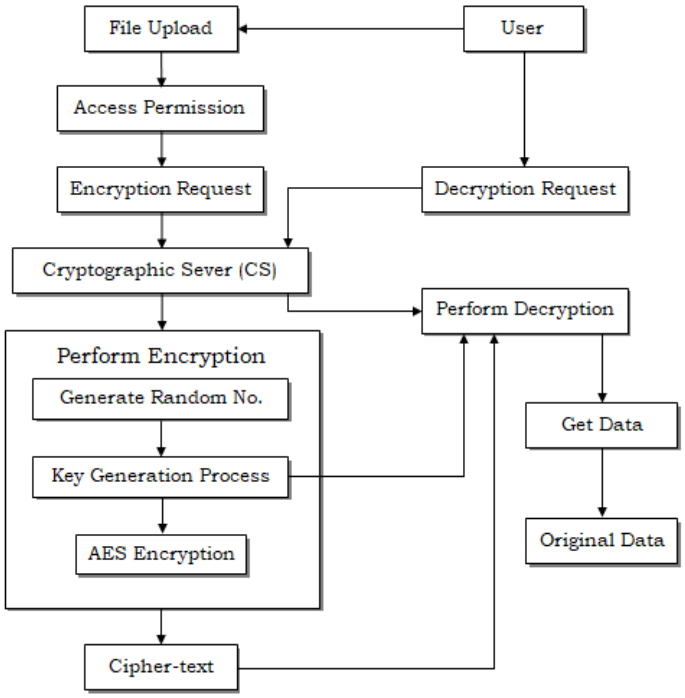

Figure 2. Cryptographic Working Process

In the previous scenario, we have seen how access request arrives at the CS portal for encryption and decryption. Once a file have uploaded by user and similarly access and share permission is executed by CS, perform main task for providing security of the user data by encrypting and decrypting of file.

In first vision of view, for encryption, the CS starts the encryption process and it will used only when user need to decrypt file at the end of decryption. This random number will sent to the group user email id. After this, cryptographic key is generated which is used by encryption algorithm. The overall process of key generation is depicted using figure 3:

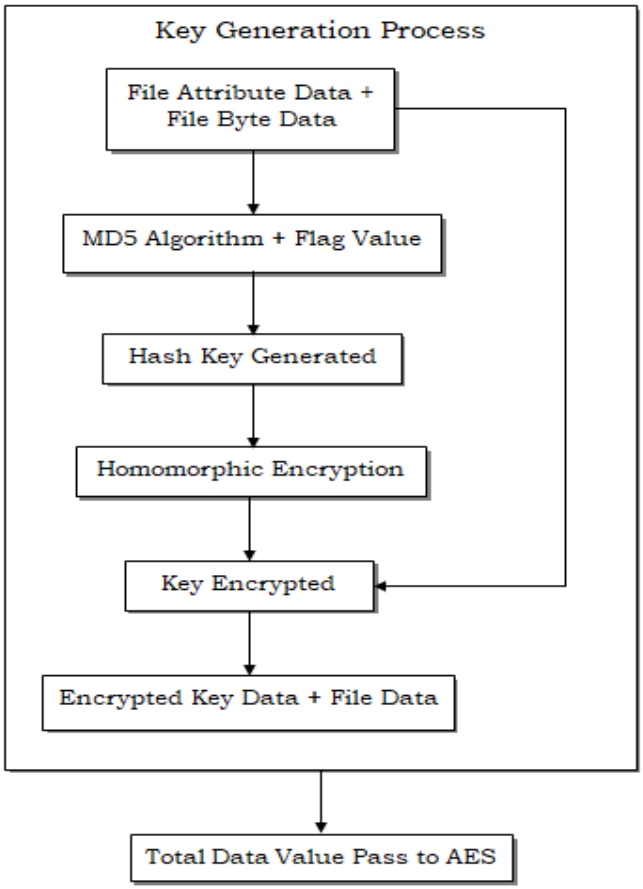

Figure 3. Key Generation Process

In key generation level it is mandatory to highlight the process of key encryption. For the security promises, we also encrypt the key which is concurrently used for file encryption. Firstly, we have generated take the attribute and byte of the input file data.

On this data, we pass to the MD5 hash algorithm and individual flag value. The flag should be 0 or 1 depend on the Key. This process will generate hash key and on this hash value we apply encryption algorithm i.e. paillier algorithm which is frequently known as homomorphic encryption algorithm. This algorithm is positively encrypt the key. For the purpose of key variation, again we pass the attribute and byte data of input file in to encrypted key then we will get total key value of data which is need to data encryption performed by AES algorithm. Hence, we encrypt input data file by using this total key data value. Consequently, it will generate the cipher-text (encrypted text). The overall process is the mixture of key generation and file encryption. Likewise, for decryption of the data file, user send the decryption request to the CS and CS will process this request. The CS will decrypt the data by getting the cipher-text and total key data value which is pass into AES algorithm. After, this process, decrypted data i.e. original text is produced. This data is downloaded from the user portal.

The encryption, decryption and key management is the entire process of the cryptographic server (CS) which is fully maintained in a systematic way. This is ensure secure sharing of data file among the user and fully secure from outside malicious activity.

B. Proposed Algorithm

In order to demonstrate the cryptographic scenario of the proposed Secure Data Sharing Scheme for the cryptographic cloud storage by means of encryption and decryption process is obtained in this section. The proposed cryptographic technique in terms of algorithm is given in table 3.1, 3.2 and table 3.3 of key generation, encryption and decryption respectively.

Table 1: Key Generation Process

Input: File Byte (F_B), Flag (F), Attribute (F_{Attr}) **Output:** key Data value (K_{data})
Process: 1. $sum = [F_B + F_{Attr}]$ 2. $MD5Hash_{Key} = generate.MD5Algo\ (Sum) + F$ 3. $Key_{encrypted} = homomorphic.encrypt\ (MD5Hash_{Key})$ 4. $Sum1 = [Key_{encrypted} + (F_B + F_{Attr})]$ 5. $Key\ Data\ value = produced.DataValue(Sum1)$ 6. Return K_{data}

Table 2: Encryption Process

Input: Input TextT_I, K_{data}, **Output:** Ciphertext (C_t)
Process: 1. $D = InputTextData(T_I)$ 2. $Text_{encryted} = AES.encrypt(D, K_{data})$ 3. Return C_t

Table 3: Decryption Process

Input: Ciphertext (C_t), K_{data} **Output:** Original text T_O
Process: 1. $F = \text{Ciphertext}\ (C_t)$ 2. $\text{Text}_{\text{Decrypted}} = \text{AES.decrypt}\ (F, K_{data})$ 3. Return T_O

RESULT AND DISCUSSION

A. Encryption Time

The amount of time required to perform encryption process using the selected algorithm is termed as the encryption time of the system. The encryption time of the proposed system is demonstrated using figure 5.1 and the table 5.1.

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

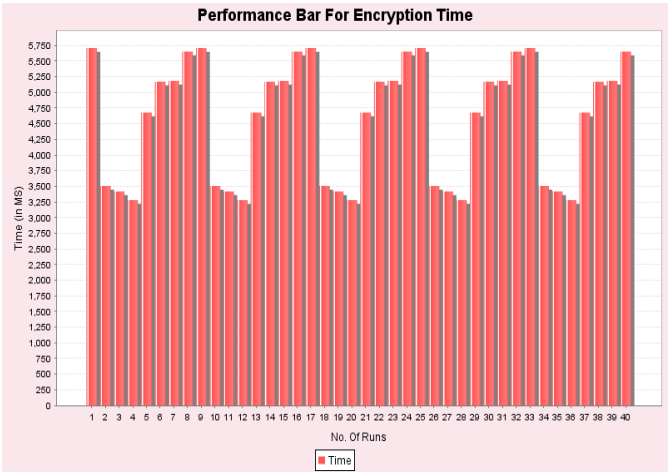

Figure 4. Encryption Time

In order to show the performance of implemented data sharing scheme, encryption execution time is reported in figure 4 and table 4. In this diagram the X axis shows the different experiments where we executes different input files and the Y axis shows the amount of time consumed for encrypting the input text file. Furthermore, 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 respect to the sharing of file among different parties.

Table 4. Encryption Time

Number of Experiments	Proposed Secure Data Sharing Scheme
1	3427
2	3506
3	3419
4	3380
5	3305
6	3345

B. Decryption Time

The amount of time required to recover (Decipher) original data from the encrypted text is known as the decryption time of the algorithm. The figure 5 and table 5 shows the got performance of the system in terms of millisecond. To show the performance of secure data sharing scheme the blue line shows the performance of proposed algorithm.

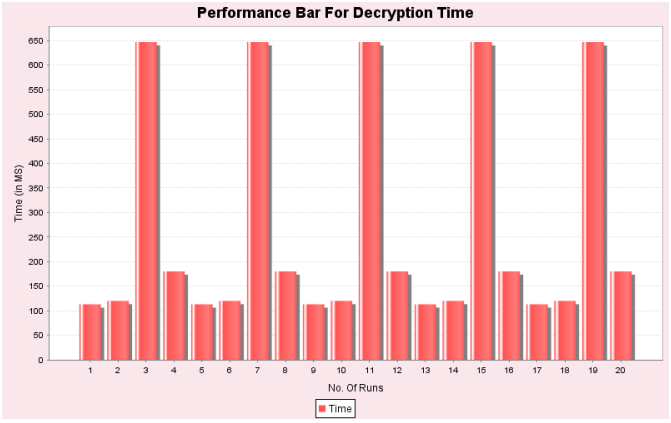

Figure 5. Decryption Time

In given figure 5, X-axis shows the diverse numbers of experimentation are performed and the Y-axis shows the amount of time consumed for decryption process. According to the produced results the encryption time is more than the decryption time in the system, but the decryption time of the proposed algorithm is much adaptable and after secure sharing user can be downloaded in their system.

Table 5. Decryption Time

Number of Experiments	Proposed Secure Data Sharing Scheme
1	113
2	104
3	156
4	189
5	145
6	108

C. File Upload Time

The time, where upload the data to the server form our local computer drive. Moreover, uploading time depend on the speed of our server. Hence, this time can be calculated using given formula:

$$\text{Upload Time} = \text{Time of Submission of Reqest} + \text{Processing of Encryption}$$

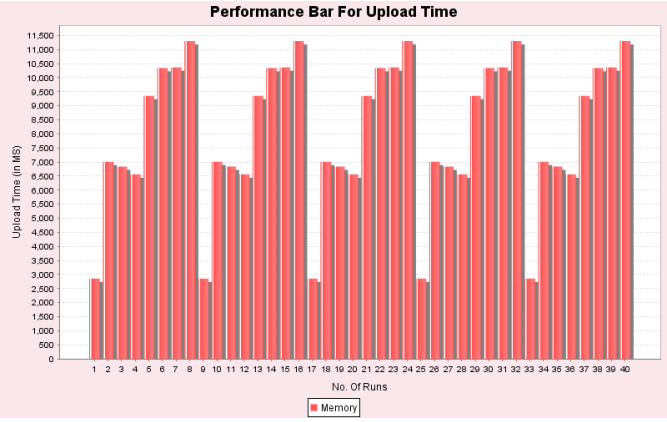

Figure 6. Upload Time

The figure 6 and the table 6 show the encryption memory consumption of the proposed scheme. In this diagram the amount of main memory consumed is given in Y-axis and the number of file executed are reported in X axis. According to the found performance the proposed algorithm consumes less resources as we understood during the execution of algorithm.

Table 6. Upload Time

Number of Experiments	Proposed Secure Data Sharing Scheme
1	6854
2	7012
3	6839
4	6990
5	7156
6	7145

D. File Download Time

Download Time is the total sum at which user request to the CS for decryption of file and processing of decryption algorithm. The downloaded file store to the user local drive. Hence, this time can be computed using following formula:

$$\text{Download Time} = \text{Time of Submission of Reqest} + \text{Processing of Decryption}$$

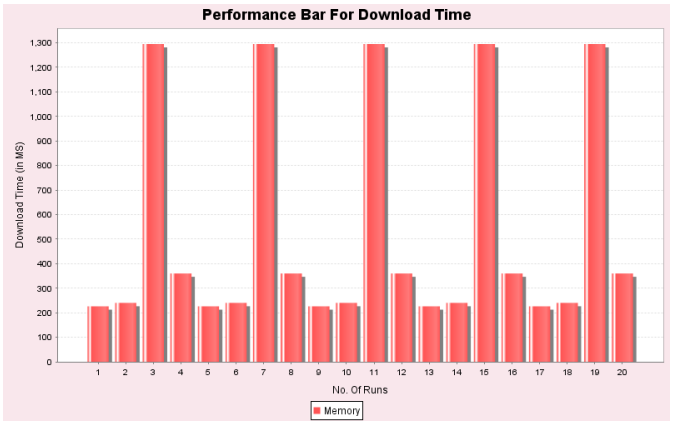

Figure 7. Download Time

-memory consumption during the data recovery process. In this diagram the X-axis represents the different experiments of diverse file size used for decryption and the Y axis shows the total of main memory consumed during the decrypting data file. According to the gained results the amount of main memory used is less than of encryption memory and consume less space of proposed algorithm.

Table 7. Download Time

Number of Experiments	Proposed Secure Data Sharing Scheme
1	552
2	625
3	516
4	698
5	678
6	579

E. Key Generation Time

Key generation is the process of generating keys in cryptography. A key is used to encrypt and decrypt whatever data is being encrypted/decrypted. In our scheme, we are calculating key generation time using following formula:

$$\text{Key Generation Time} = \text{Received Encrypted Key Time} - \text{Start time of File Processing}$$

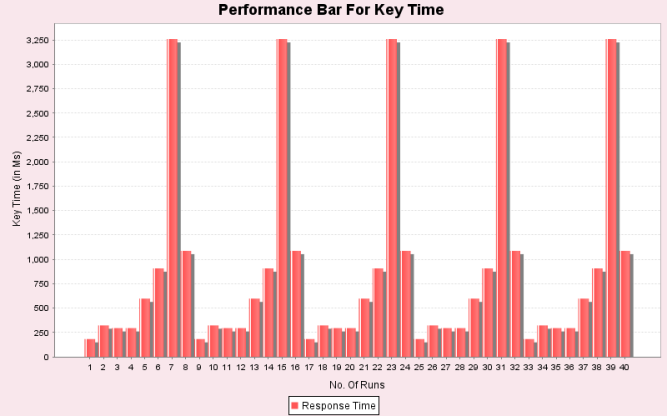

Figure 8. Key Generation Time

X-axis of this graph contains the amount of experiments performed and the Y axis shows the amount of time required for key generation time. The key generation is the total time of file processing and algorithm processing time. According to the computed results the key generation time is totally depends on the amount of file size or other parameters. This is also directly depends on the algorithm processing time. In this graph, blue line shows the key generation time.

Table 8. Key Generation Time

Number of Experiments	Proposed Secure Data Sharing Scheme
1	1025
2	1123
3	1156
4	1225
5	1189
6	1201

CONCLUSION AND FUTURE WORK

The proposed work is intended to provide a dynamically secure group data sharing and access services in a decentralized manner. This chapter provides the summary of the performed for cloud oriented in security concerns and the future extension of the work is also suggested.

A. Conclusion

In this paper, we developed approach, secure data sharing is introduced in the cloud isolated environment. The file is shared between one user to multiple user or/and single to single user i.e. group sharing. In addition to that for thwarting the unauthorized access to the system we make data owner module which is responsible to handle uncommon activities perform by user panel. In this project work, whenever, file is uploaded by means of user panel then firstly it will be process for encryption on the cryptographic server (CS) portal and CS also gives the access (read or write or both) and sharing permission to particular user want to access privileged. CS generate the random number and key before file encryption process. Once, file is encrypted successfully then it will be available for sharing to user which have access permission already. In this process, data are encrypted using AES algorithm and key is encrypted using homomorphic (paillier) algorithm. Hence this, process is also known as cryptographic process.

This approach is implemented using the JAVA environment and using JSP for web application deployment. After implementation the performance of the approach is calculated in using difference performance factors. The table 9 includes the performance summary of the proposed approach.

Table 9. Performance Summery

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 and it is adoptable.
3	File Uploading Time	The file uploading time is the sum of encryption algorithm time and data storing time on server.
4	File Downloading Time	File downloading time is the sum of decryption algorithm processing time and data download time to the user system.
5	Key Generation Time	The key generation time is the time of hash generation time of key followed by key encryption time.

The proposed approach is secure and well-organized of data sharing among different parties of data file in public cloud environment, secure and availability of data, thus the proposed system is acceptable for data hosting.

B. Future Work

The proposed work for group data sharing and their open access from the isolated manner is implemented successfully. Additionally the system performance with the cryptographic implementation of the system is also obtained which is adoptable.

- ✓ This approach can be extend using concept of split key management over cryptographic server.
- ✓ The proposed work need to modified more for text based cryptography because the Paillier algorithm is suitable for numerical data encryption

REFERENCES

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

[2] 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.

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

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

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

[6] 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.

[7] Herdman, R. "Information security and privacy in network environments." The Office of Technology Assessment (OTA) (1994).

[8] Sattarova Feruza Y. and Tao-hoon Kim, "IT Security Review: Privacy, Protection, Access Control, Assurance and System Security", International Journal of Multimedia and Ubiquitous Engineering Vol. 2, No. 2, April, 2007.