ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

SDDA: An Implementation of Secure Data Deduplication Approach for Cryptographic Data Storage

Akanksha Upadhyay[1], Pooja Meena[2], Chetan Agrawal[3]

PG Scholar, Computer Science & Engineering, RITS, India[1]

Asst. Prof., Computer Science & Engineering, RITS, India[2]

HOD, Computer Science & Engineering, RITS, India[3]

Abstract: Perhaps, today's need to secure end user data on storage server with full of security where user can retrieved or manage data without any vulneraries. So that this issues of data security and maintained data without storing multiple copies of same data is essential to overcome. Therefore, in this thesis work, we proposed a mechanism named "Secure Data Deduplication Approach i.e. SDDA" using cryptographic technique which ensure user authenticity and maintained data deduplication on storage server for the same data. For this perform we implement cryptographic algorithm as MD5 for hash Key generation and SHA1 Algorithm. For Data Encryption we use AES Algorithm where key generated by SHA1 is pass to AES algorithm. Additionally the small chunks of the encrypted data are prepared. These chunks of data are further used with the markle hash tree algorithm using SHA1 algorithm. Firstly, data will be check using MD5 generated hash key. After this, individual block of data is verified by the available data using markele has tree if the data is available then data is duplicate and need not to upload, otherwise the data is uploaded to server. According to the experimental results the proposed SDDA method provides better security approach and facility to store data on server with scalability and availability. The performance of the system is measured in terms of encryption, decryption time and memory respectively.

Keywords: Cloud Storage, Data De-duplication, Data Privacy, AES, Cloud Server, Cloud Security, MD5, Merkle Hash tree.

I. INTRODUCTION

Cloud computing technology not only has contributed to various applications and influenced the operation of original networks, but also allowed the Internet to exist in different corners with different devices. Nevertheless, with the appearance of numerous devices and data, data access and nodes management and control of cloud network become the significant issues to be emphasized because the efficiency of control methods enormously affects the performance and quality of cloud network. With the rapidly increasing amounts of data produced worldwide, networked and multi-user storage systems are becoming very popular. However, concerns over data security still prevent many users from migrating data to remote storage. As the world moves to digital storage for archival purposes, there is an increasing demand for systems that can provide secure data storage in a cost-effective manner. By identifying common chunks of data both within and between files and storing them only once de-duplication can yield cost savings by increasing the utility of a given amount of storage. Besides these powerful advantages of cloud Storage, however, many people and companies is still feel hesitant to store their data in cloud. The reason behind this hesitancy is the fear of people and companies regarding loss of control on their data because there are some incidents of data loss and data leakage which make people to think about it. In this paper work, we developed a security structure for data privacy preservation of cloud data storage to make accessible to data file in secure manner in large public cloud environment.

II. PROPOSED WORK

The cloud is one of technology which is frequently accessed now in these days. Now only for computational ability its now in these days also be used for storing data. This section provides the detailed understanding about the proposed solution and their formulation. Thus the components of the proposed system and their functional importance are described in this section.

A. Methodology

The proposed work is aimed to provide the secure data deduplication approach for cloud storage issues. In this context a model is reported in this section which deals with the data to identify the duplicate data to manage previously reside data storage. The proposed data model is demonstrated in figure 1 and 2.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

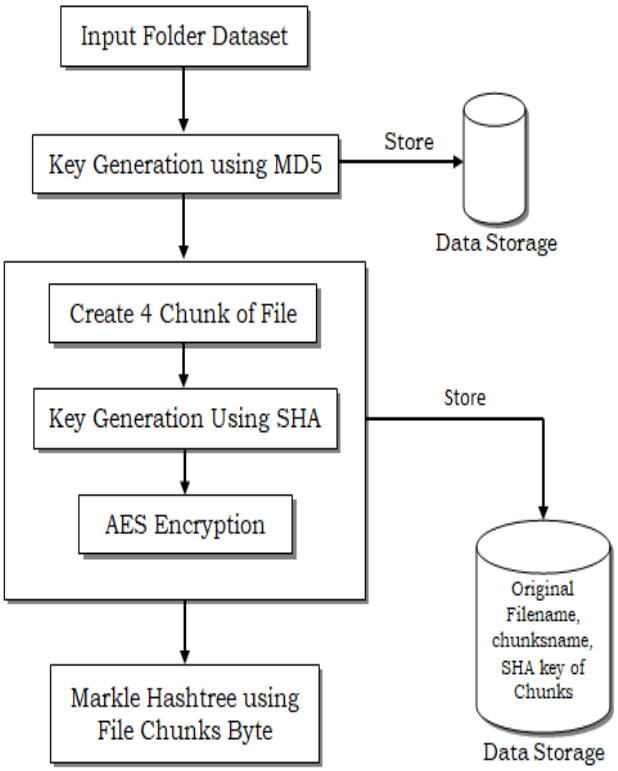

Figure 1 File Encryption and Upload

The working of the proposed Secure Data Deduplication Approach is given and explained in details in this section. To understand the core concept of the proposed methodology by using description of both diagram:

The given flow diagram 1and 2 depicts Secure Data Deduplication Approach. Figure 1 Data upload with encryption and 2 shows Data Deduplication check process. Here, firstly, we take an input dataset which is contain different folders of subject domain name. Dataset 4 different folders contain different text file. Following table 1 shows folder information of dataset.

Table 1 Dataset Information

Folder Name	**Data File Name**
Artificial Intelligence	AI1, AI2
Computer Graphics	CG1, CG2
Data Mining	DM1, DM2, DM2, DM4
Networking	NW1, NW2, NW3

After input dataset, we prepare inverted index using MD5 algorithm. This MD5 hash algorithm generate hash key for each file. For indexing we store this hash key with their file name in database storage. After Indexing of data, on input data file we divide file in to 4 block i.e. chunks. This chunks contain plain text of file. Now we apply SHA1 algorithm on this chunks file and generate different SHA key of these 4 files. Finally, this file have been uploaded before encryption process. For encrypting this chunks, we apply AES symmetric key algorithm where we input this SHA key to AES algorithm. After this process, we store file name, chunks name and SHA key of each chunks file in database storage. Furthermore, we generate markele hash tree using file chunks byte. This overall process demonstrate the data encryption and data upload on the server.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

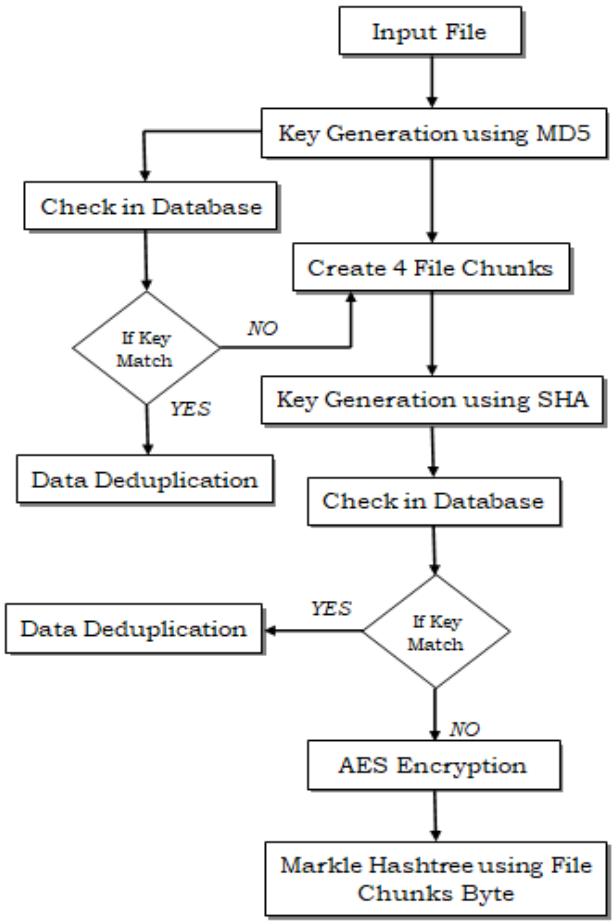

Figure 2 Data Deduplication Process

Thereafter, figure 2 shows the process to check data deduplication for the new input text file. Here we also generate MD5 hash key of input text file and check duplicity of this file i.e. this file already exist or not. Therefore, we check in database of hash key of newly input text file to already exist hash key in database storage. If this both key is matched then data deduplication found and system will reject of that file to process further. If both key doesn't match then file will process further as create four chunks of file and apply SHA1 algorithm on this chunks for producing SHA hash key. Each block of data is mounted on the tree using the binary manner. Here we also check data duplicity by means of SHA key. Hence, check in database of SHA key of new input file to already stored SHA key of existing file. If the SHA key match then data deduplication found. If this hash key doesn't match then we encrypt this file using AES algorithm by giving SHA key for encryption. Finally generate markle hash tree using chunks of file byte. This overall process producing efficient output of secure data deduplication technique in terms of encryption decryption time and memory.

B. Proposed Algorithm

This section introduces the summarized steps of the proposed algorithm of SDDA. Hence Table 2 depicts the file encryption and upload process and table 3 shows data deduplication check process.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

Table 2 File Upload and Encryption Process

Input: Input Folder (F), Database Storage (S)

Output: Upload Data and Generate Markle Tree

Process:

1. $R = readInputFolder\ (F)$
2. $fileList\ [] = getFiles\ (R)$
3. $\boldsymbol{for}(i = 0; i < filelist.length; i++)$
4. $Key_{MD5} = invertedIndex.MD5\ (fileList\ [])$
5. $S = fileHashDetail(Key_{MD5}, fileName\ [])$
6. $\boldsymbol{endfor}$
7. $\boldsymbol{for}\ (j = 0; j < filelist.length; j++)$
8. $chunks = fileList.split\ \left(\frac{filesize}{4}\right)$
9. $ChunksName_{1,2,3,4} = chunksname\ (filename,$ *Chunks1,2,3,4*
10. $Key_{SHA} = Key_{SHA}\ (ChunksName_{1,2,3,4})$
11. $Encryption = AES.encrypt\ (Chunks_{1,2,3,4}, Key_{SHA})$
12. $S = insert.data\ (fileName, ChunksName_{1,2,3,4}, Key_{SHA})$
13. $MarkleData = hashtree.Data\ (chunks)$
14. $\boldsymbol{endfor}$
15. $MarkleHashTree = MarkleHashTree\ (MarkleData)$
16. $Return\ Markle\ Hash\ Tree$

Table 3 Data Deduplication Check Process

Input: Input text file (F'), Database Storage (S)

Output: Check Deduplicate Data and

Process:

1. $R = readInputFolder\ (F')$
2. $Key_{MD5} = invertedIndex.MD5\ (R)$
3. $\boldsymbol{if}\ (S.exist[Key_{MD5}])$
4. $found\ Data\ Deduplication$
5. $\boldsymbol{else}$
6. $deduplication\ not\ Found$
7. $chunks = R.split\ \left(R, \frac{fileSize}{4}\right)$
8. $chunksName = chunksname\ (R,\ chunks_{1,2,3,4})$
9. $Key_{SHA} = Key_{SHA}\ (ChunksName_{1,2,3,4})$
10. $\boldsymbol{if}\ (S.exist\ (Key_{SHA}))$
11. $Duplicity\ found$
12. $\boldsymbol{else}$
13. $Encryption = AES.encrypt\ (Chunks_{1,2,3,4}, Key_{SHA})$
14. $chunks = R.split\ \left(R, \frac{fileSize}{4}\right)$
15. $chunksName = chunksName\ (R,\ Chunks_{1,2,3,4})$
16. $S = insert.data\ \left(R,\ ChunksName_{1,2,3,4}, Key_{SHA}\right)$
17. $MarkleData = hashtree.Data\ (chunks)$

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

18.	***endif***
19.	***endif***
20.	$MarkleHashTree = MarkleHashTree\ (MarkleData)$
21.	*Return Markle Hash Tree*

III. RESULT ANALYSIS

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

A. 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 3 and the table 4.

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

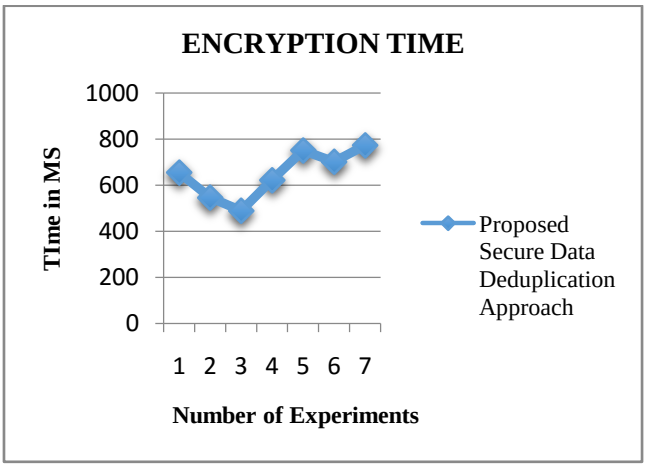

Figure 3 Encryption Time

In order to show the performance of implemented Secure Data Deduplication Approach, encryption execution time is reported in figure 3 and table 4. In this diagram the X-axis shows the different experiments on which we run different files as an input and the Y-axis shows the amount of time consumed for encrypting the input text data file. Additionally, 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, the encryption

Table 4 Encryption Time

Number of Experiments	Proposed Secure Data Deduplication Approach
1	655
2	546
3	489
4	622
5	751
6	702
7	773

B. 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 4 and table 5 shows the obtained performance of the system in terms of millisecond. To show the performance of secure data deduplication approach blue line shows the performance of proposed algorithm.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

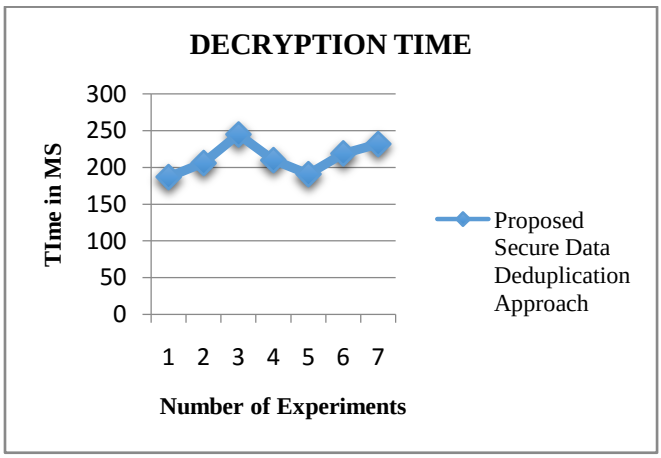

Figure 4 Decryption Time

In given figure 4, 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 results the encryption time is higher than the decryption time in the system, but the decryption time of the proposed algorithm is much adaptable and if user wants to access his data then user can download by selecting particular file.

Table 5 Decryption Time

Number of Experiments	Proposed Secure Data Deduplication Approach
1	187
2	206
3	245
4	210
5	191
6	219
7	232

C. 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.

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

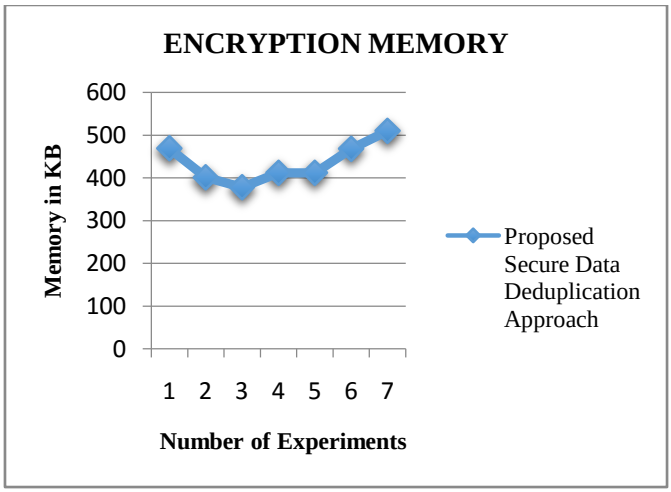

Figure 5.3 Encryption Memory

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

The figure 5 and the table 6 show the encryption memory consumption of the proposed approach. In this diagram the amount of main memory consumed is given in Y-axis and the number of experiments is reported in X-axis. According to the obtained performance the proposed algorithm consumes fewer resources as we seen during the execution of algorithm.

Table 6 Memory Consumption

Number of Experiments	Proposed Secure Data Deduplication Approach
1	469
2	401
3	378
4	412
5	412
6	468
7	510

D. 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 6 and table 7 shows the amount of main memory consumed during the data recovery process.

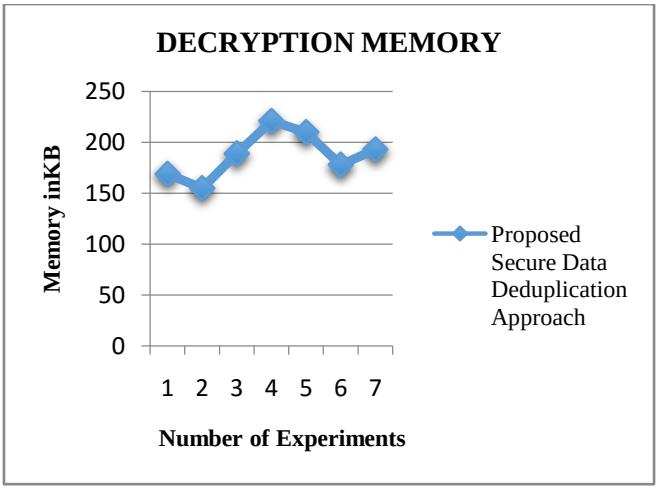

Figure 5.4 Decryption Memory

In this graph, X-axis depicts the different experiments on same input used for decryption and Y axis shows the amount of main memory consumed during decrypting data file. According to the obtained results the amount of used main memory is less than of encryption memory and consumes less space for decipher the encrypted file.

Table 7 Decryption Memory

Number of Experiments	Proposed Secure Data Deduplication Approach
1	169
2	155
3	189
4	221
5	210
6	178
7	193

E. 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 7 and table 8.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

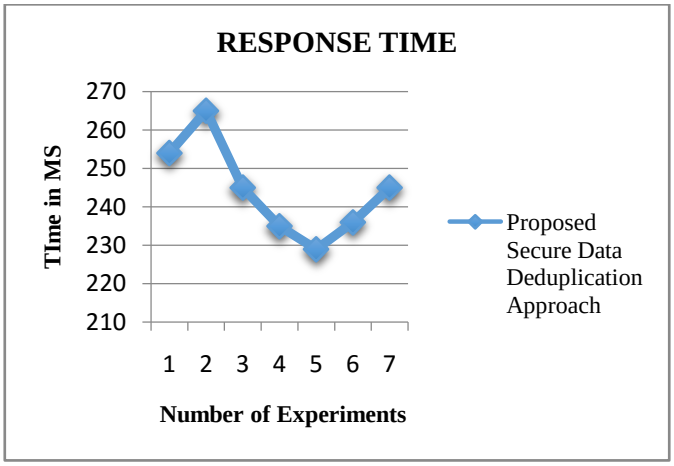

Figure 7 Response Time

X-axis of this diagram contains the amount of experiments performed 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 Response Time

Number of Experiments	Proposed Secure Data Deduplication Approach
1	254
2	265
3	245
4	235
5	229
6	236
7	245

IV. CONCLUSION AND FUTURE WORK

The proposed work is intended to provide a secure data de-duplication over cryptographic cloud for end user data privacy and authenticity. This section provides the summary of the performed for cloud oriented data services for the security concerns and the future extension of the work is also suggested.

A. Conclusion

Nowadays, the explosive growth of digital contents continues to raise the demand for new storage and network capacities, along with an increasing need for more cost effective use of storage and network bandwidth for data transfer. Indeed, although data is outsourced in its encrypted form, there is no guarantee of data privacy when an honest but curious CSP handles the management of confidential data while these data reside in the cloud. Cloud Server provider are able to maximize data storage space by incorporating data deduplication into cloud storage. Although data deduplication removes data redundancy and data replication by storing only a single copy of previously duplicated data, it also introduces major data privacy and security issues for the user.

In this paper work, we developed "*Secure Data Deduplicaion Approach*" i.e. *SDDA*, which securely implement user data privacy and their authenticity while managing cloud server storage space after checking data deduplication. The main aim of the proposed work is to provide a secure and efficient data storage on cloud server that eases of use for data owner. Reducing data storage overhead and processing cost is a mandatory requirement of any organization, while analysis of data and information is always the most important tasks in all the organizations for decision making in the end.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 6, June 2018

B. Future Work

Therefore in near future the proposed work considering following directions are:

✓ This scheme can be further enhanced by uploading very large data and can do compression on those data. Also, enhance to measure the Quality of Service (QoS).
✓ We also extend our work to implement a privacy preserving public auditing of the uploaded data in the cloud such that the third party external auditor (TPA) should not learn about the content of the data. For this purpose, we wish to allow the TPA to only audit the data for correctness without learning about it.
✓ We also plan to use artificial intelligence (AI) techniques (such as neural networks) to identify the duplicates in an efficient and fast manner. This will allow to make more customized and user friendly methods for identifying the duplicates.

REFERENCES

[1] Jiang, Tao, Xiaofeng Chen, Qianhong Wu, Jianfeng Ma, Willy Susilo, and Wenjing Lou. "Secure and efficient cloud data deduplication with randomized tag." IEEE Transactions on Information Forensics and Security 12, no. 3 (2017): pp. 532-543.
[2] I. Foster, Z. Yong, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," in Grid Computing Environments Workshop, 2008. GCE '08, 2008, pp. 1-10
[3] Armbrust, Michael, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee et al. Above the clouds: A berkeley view of cloud computing. Vol. 4. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.
[4] Sookhak, Mehdi, et al. "Remote data auditing in cloud computing environments: a survey, taxonomy, and open issues." ACM Computing Surveys (CSUR) 47.4 (2015): 65.
[5] Bhagwat, Deepavali, KaveEshghi, Darrell DE Long, and Mark Lillibridge. "Extreme binning: Scalable, parallel deduplication for chunk-based file backup." In Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS'09. IEEE International Symposium on, pp. 1-9. IEEE, 2009.
[6] Li, Jin, Yan Kit Li, Xiaofeng Chen, Patrick PC Lee, and Wenjing Lou. "A hybrid cloud approach for secure authorized deduplication." IEEE Transactions on Parallel and Distributed Systems 26, no. 5 (2015): pp. 1206-1216.
[7] Li, Jin, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick PC Lee, and Wenjing Lou. "Secure deduplication with efficient and reliable convergent key management." IEEE transactions on parallel and distributed systems 25, no. 6 (2014): pp. 1615-1625.
[8] Liu, Jian, N. Asokan, and Benny Pinkas. "Secure deduplication of encrypted data without additional independent servers." In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 874-885. ACM, 2015.