Deduplication Mechanism for Storing Data over Cloud

[1]Shaikh Basirat Tazin, [2]Prof S.D. Pingle

[1]M.E. Student, [2]Associate Professor
CSE Department,
P.E.S. College of Engineering, Aurangabad, India

***Abstract*—Since cloud computing has been driving attention of large number of users for storing their important data, data compression technique managing data stored over cloud had become huge necessity. Task of data deduplication technique is to remove duplicate copies of data on a server and keep only one physical copy. Duplicate copies over cloud are identified by generating tags from content of data copy and matching them with tags of files already residing over cloud. Duplicate copies will correspond to same tags generated using MD5 hashing technique. To maintain security of data, it is encrypted using linear congruentiality algorithm before uploading over cloud which employs convergent encryption concept. To encrypt data, convergent keys are derived from content of data using SHA-512 hashing technique and data is encrypted using this key according to convergent encryption concept which requires key derived from content of data.**

***Index Terms*—Data deduplication, key generation, tag generation, authorized duplicate check, convergent encryption.**

I. INTRODUCTION

Due to heavy increase in usage of Internet, and fast development of storing and processing data technologies, computing resources have become powerful and easily available. Cloud computing provides you access to a networked storage and computing resources in a virtualized environment. Peoples are becoming more interested in big data for utilizing them in their own various application fields. Data tends to be in enormous amount when dealing with big data, where cloud computing can prove to be a useful solution for management of data. Main functions performed by cloud computing are efficient management of distributed data, offering services to users which they are requesting for, and solving complex problems.

Cloud computing manages data residing over it by compressing amount of data. And for compressing amount of data, technique used is data de-duplication which deals with removing redundant copies of data. It stores only single copy of data, and for other copies of data reference is passed to that single copy. It is a simple storage optimization technique which is used by various cloud storage providers such as Dropbox, Bitcasa [7], JustCloud, Mozy [8], Amazon S3 (Simple Storage Service) and Google Drive, etc. With data deduplication technique bandwidth needed to transfer same content multiple times is reduced. When data is outsourced to cloud server, security and privacy issues arises since it is susceptible to vulnerable attacks.

To overcome these issues, before outsourcing data to cloud it is encrypted. However, when different users will encrypt data by using their own keys, cipher text produced will be different thereby making deduplication impossible. Therefore, data is encrypted with help of convergent key which is derived from data itself and this technique of encryption is referred as convergent encryption. Data deduplication with convergent encryption provides secure and optimized storage.

II. PRIMITIVES

Data de-duplication

To efficiently manage data, well- known data compression technique used is data de-duplication which deals with removing redundant data. Using this technique, only single copy of data is stored on cloud server and other copies of similar data are passed reference to that unique copy. Data de-duplication can be performed at file level where redundant copies of similar files are removed or block level where redundant blocks of data that occur in dissimilar files are removed. De-duplication can be performed at client side. When de-duplication is performed at client side, bandwidth for transmitting data to server is reduced.

Convergent Encryption

Convergent encryption is a technique that produces similar cipher text from similar plain text. It performs this operation with help of convergent key which derived by calculating cryptographic hash value from data copy. After encryption, user stores key with him/her and sends cipher text to cloud. By using this technique, users will have same encryption keys and cipher text for similar data.

Proof of Ownership

De-duplication of data is performed by calculating corresponding hash values with help of which similarity between data copies is determined. If file is already present at cloud, the instead of uploading again that file pointer to ownership of file is updated and user can download that corresponding file. When an encryption is done at client side, an attacker may somehow retrieve hash values and access corresponding file. To defend from such an attack PoW is used to assure whether user owns a file or not.

Hybrid Cloud Architecture

This architecture consists of three components, namely, public cloud, private cloud and user. Architecture is illustrated in figure 1 below.

- ***Public Cloud:*** Usually, public cloud deals with storage of data. Users store data on public cloud which they want to access later. Public cloud uses S-CSP (Storage Cloud Service Provider) for storing data and stores only unique copy of a particular data. User need to interact with public cloud while storing and retrieving data.
- ***Private Cloud:*** Private cloud acts as an interface between public cloud and user. User who wants to upload data on public cloud first need to request for token from private cloud where private keys for users are stored. Only authenticated users are issued token for accessing public cloud.

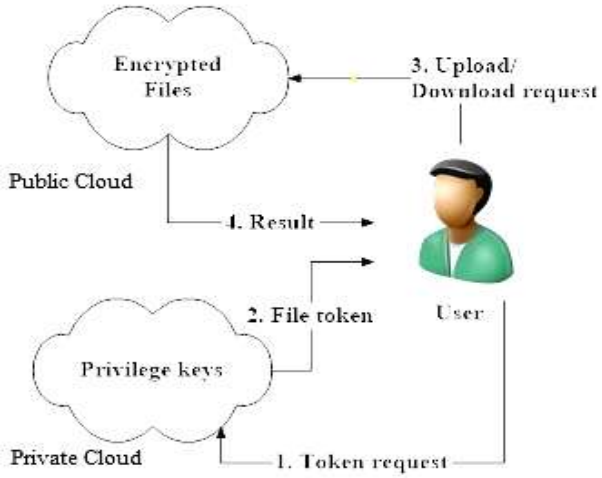

Fig 1: Hybrid cloud architecture.

- ***Users:*** User is an entity who wants to upload data to public cloud. User should upload only unique data and if data is already present then it should not uploaded to save bandwidth. In storage systems, performing deduplication of data, users are assigned privileges to predefine type of access they can perform.

III. Related Work

Due to increase in use of cloud computing, data deduplication had gained much importance. Data deduplication is mostly performed with convergent encryption and together they are used in used in various systems such as Bitcasa, Backup system patent filed by Stac Electronics, Farsite, [8]etc.

Dropbox:

Dropbox [6] offers cloud storage for users to store storage, data amount will increase. To manage this data, de-duplication is needed. Dropbox performs de-duplication by considering fixed-size blocks. Each fixed-size block is of 4MB size. Hashing is done using SHA 256 technique. Calculated hash values of files and mapping of file and its corresponding block is managed by control server. And blocks representing unique blocks of data are stored in storage server in Amazon S3. Control server receives all calculated hash values sent by client and returns back only unique values to client. As different users are accessing this cloud Clients retrieves this values and sends corresponding blocks to storage server.

Bitcasa:

In his interview [7], Bitcasa CEO Tony Gauda had explained the use of convergent encryption in Bitcasa cloud storage provider for performing deduplication. In his interview he mentioned use of AES-256 hash, SHA 256 hashing for data to be stored. Encryption is performed on client side.

Encrypted de-duplication with fast and secure laptop ba-ups:

In this paper [1], a community of laptop users are considered, which are using backup scheme to back up their data. This scheme emphasizes on increasing speed of performing back up and storage required for storing backup data. Algorithm this scheme uses is based on convergent encryption technique, in which data de-duplication is performed by using data common between different users. File to be backed up is first searched in list of backup storage. If file is not present, then file is backed up otherwise index is returned indicating location of file already backed up. If data to be backed up is confidential, then personnel can perform encryption on client-end also. Main disadvantage of this scheme is that direct backing up data to cloud can be very costly.

DupLESS: Server aided encryption for de-duplicated storage:

In this paper [2], message locked encryption scheme is used which is based on convergent encryption technique. It consists of keys derived from messages which can be obtained from key server and is shared amongst different users. Users need to perform authentication to key server and do not supply any information about data to it. Data is stored on storage server where unique copies of data are maintained. Users interact with storage server very less number of times.

A secure data de-duplication scheme for cloud storage:

This scheme [6] considers that data are of two types popular and unpopular depending upon number of users accessing them. Popular data is considered as less sensitive and unpopular data is considered as sensitive data. Multi- layered cryptosystem is used in this scheme, with two cryptosystems, namely, convergent and threshold cryptosystem. The unpopular data is sensitive and is used by less number of users. Therefore, it needs security and is protected by two layers. On the other hand, popular data is less sensitive and used by many users. Therefore, it needs weaker security and is protected by only one layer. Security to be applied depends on layers that has been deployed depending upon how sensitive data is.

Secure data de-duplication:

In this paper [5], existing data is divided into small chunks from which keys for encrypting these chunks are generated. Two models are implemented in this paper, one is authenticated model which has similar design to convergent encryption technique that has been deployed in Farsite system and other one is anonymous model which mainly focuses on hiding details of authors and readers. In both models, client first divides a file to be uploaded into set of chunks using content based chunking. After chunks of data are formed, data is encrypted using convergent encryption technique at client side.

Enhanced De-key Approach to Reduce Data De-dupli- cation in Cloud Storage:

In this paper [3], main problem dealt is management of convergent keys. Dekey scheme is applied in this paper in which key management is not needed. In this scheme, convergent keys derived from messages are distributed over different server and client side deduplication is implemented with POW. Ramp Secret Sharing scheme is implemented in this paper.

A Hybrid Cloud Approach for Secure Authorized Deduplication:

In this paper [13], deduplication of data is performed in hybrid cloud architecture, where there are two clouds public and private cloud. Here public cloud is used for storage of data and private cloud deals with managing privileges of users. User need to receive token from private cloud where authenticity of user is checked before issuing token and with help of this token user uploads file over public cloud or runs POW if file already exists on public cloud. Convergent encryption is performed using 256-bit AES encryption to perform deduplication of data. Convergent key needed to encrypt data is derived using SHA-256 hash. Tags are generated to check duplicate files over cloud using SHA-1 hash.

IV. PROPOSED SYSTEM DESCRIPTION

Problem Statement:

To design an authorized deduplication scheme that will perform data de-duplication in cloud computing with twin clouds: public and private cloud and to use linear congruentiality algorithm for encryption of data and generate tags associated to it and to associate right of ownership of uploaded file along with access control rights in list.

Proposed Scheme:

An authorized deduplication scheme is proposed in this paper, where hybrid cloud architecture with public and private cloud is used. Public cloud deals with storing data. Private cloud deals with maintaining privileges for files and users and issuing private keys. Users first need to interact with private cloud by providing files and his privileges as input. Private cloud receives request, matches privileges and generates tag which is used for performing duplicate check. After performing duplicate check at public cloud, if that file is not already present at public cloud then token is generated for that corresponding user with help of which he can upload that file at public cloud. If file is already present on public cloud, then ownership list is updated and pointer is returned to user with help of which he can download corresponding file. The architectural diagram for authorized deduplication system is shown in figure 2.

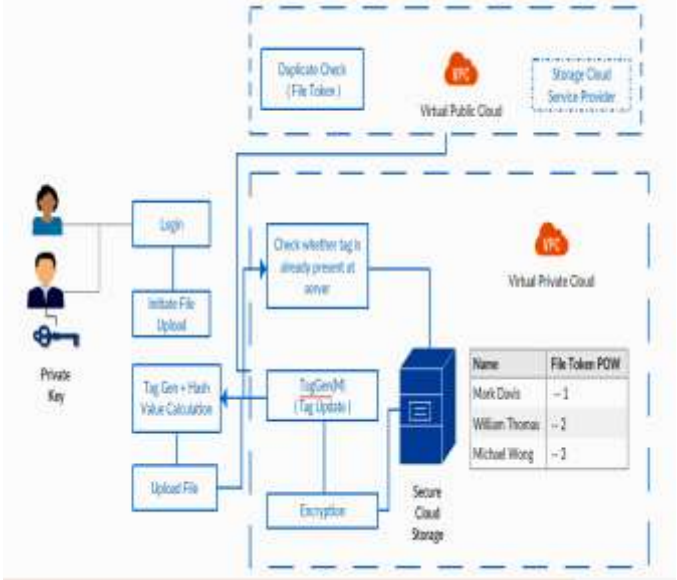

Fig: Architecture of authorized deduplication system.

The processes involved are described as follows:

- ***Access control list:*** Different users will have different privileges and these privileges will be stored using access control list. This access control list will have few attributes that will identify user by his credentials and it will also store access right of that user.
- ***Tag and Key Generation:*** From content of data copy, hash values are derived using MD5 and SHA-512 hashing technique. Hash value derived using MD5 technique will be used as tag which will be useful in detecting duplicate copies of file stored over cloud. Hash value generated using SHA-512 hashing technique will be used as convergent key needed to encrypt file before uploading over cloud.
- ***Encryption and Decryption:*** To perform encryption and decryption, Linear Congruentiality Algorithm is used which uses layered approach for performing encryption and decryption. This algorithm uses symmetric encryption technique in which key is embedded within itself and output is transmitted as bitmap file.

V. IMPLEMENTATION

A secure data deduplication scheme is implemented in this paper, where convergent encryption concept is implemented with help of linear congruentiality algorithm. Tags for detecting duplicate files are derived using MD5 hashing technique. Key is generated by computing cryptographic hash value using SHA-512 technique.

Encryption and Decryption:

Encryption and decryption process consists of two layers for performing its operation [10]. These processes layer wise are described below:

- ***Encryption:*** Encryption consists of two layers, namely, mapping layer and core encoding layer.
  Layer-1: This layer is considered as mapping layer since it maps each of its character by another character which is also present in same set. It confuses attacker by jumbling characters. It considers two sets: one is repeated character set and other is non-repeated. Repeated characters are those whose probability of occurrence is maximum while non-repeated characters are those which occur very often. Each character is replaced by another character in similar set and thus performing first layer of encryption. No key is used in this layer. Number character will also be replaced, to cause a mismatch.
  Layer-2: This layer is considered as core-encoding layer since it performs encoding of characters at this stage by using bitwise logic and ASCII format. In this layer, each character obtained in first-layer is converted to an ASCII character. Starting from first character of message obtained in layer-1 each character is XORed with negated ASCII character of key starting from first character onwards. Each and every character is encoded using this process. Since key is of a small length, it is repeatedly applied to the message. This can be formulated as in Eq. 1.

$$char\_new = char\_old \wedge (\sim key[i]) \quad (1)$$

- ***Decryption:*** Decryption process also consists of two layers, namely, mapping layer and core decoding layer.
  Layer-1-This layer is referred character-restructuring layer since it restructures characters in and it forms groups of bits from bitmap fields to form ASCII characters. Each 8-bit data is considered and its ASCII value is found. Then character representing that ASCII value is identified.
  Layer-2: This layer is referred as core-decoding layer since it decodes each character. While performing encryption we applied XOR logic, by using same logic twice we can retrace original character. Hence, by using same algorithm we can perform decryption and bitwise logic is also used here.

So, with help of this algorithm we performed symmetric encryption since we used same key for encryption and decryption. Furthermore, in encryption and decryption process layered approach except one layer other layers are dependent on keys.

Tag Generation: Tags are generated using MD5 [11] hashing technique. MD5 is a message digesting algorithm which takes input of any length and produces output in form of digest with length of 128 bits. This algorithm processes input by dividing them into blocks of 512 bits. These blocks are then divided into 16 sub-blocks each comprising of 32 bits. Algorithm is mentioned below:

1. Input bits number is checked.
2. Adding bits to message input so as resultant data length is equal to multiple of 512.
3. Add 64 bits length message input to step 2output and output obtained is represented as *m.*
4. $m\ to\ b$ blocks is divided (512 bit each).
5. b blocks to x(16 blocks) where each has 32 bits.
6. Four rounds are performed in algorithm where there are 16 steps in each round.
7. Four 32 bits shift register having hex. Values are represented as:

$$reg\ a = [7\ 6\ 5\ 4\ 3\ 2\ 1\ 0]32 - bits\ [a] = [d]$$
$$reg\ b = [f\ e\ d\ c\ 8\ a\ 9\ 7]32 - bits\ [b] = [c]$$
$$reg\ c = [8\ 9\ a\ b\ c\ d\ e\ f]32 - bits\ [c] = [d]$$
$$reg\ d = [\ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7]32 - bits\ [d] = [a]$$

8. Temporarily storing *a, b, c* and *d* values in *aa, bb, cc and dd* respectively.

9. Every round performed deals with implementation of functions f, g, and i . Single step function operation is shown in Eq. 2.

$$a = b + ((a + f(b, c, d) + x_i[k] + t[i] \lll S \quad (2)$$

where $x_i\ [k]$ is 32 bit k[th] word of x_i,
<<< S is left circular shift of S bits
In each four round end, adding output to first round input.

10. 128 bits output is obtained.

Key Generation: Keys are generated using SHA-512 hashing technique. SHA-512 and SHA-256 follows similar structure for generating hashes, difference is only that they perform their operation on blocks of different sizes. SHA-512 operates on eight 64-bit words. In SHA-512 hashing technique, 1024-bit size message block are formed and these blocks are processed by compression function of SHA-512. It generates a 512-bit intermediate hash value. Then this intermediate hash value will be encrypted by taking message block as key. Overall implementation of SHA-512 resides on compression function and message schedule of SHA-512. Message which needs to be hashed is

- added with its length so that result will be multiple of 1024 bits long
- then parsed into 1024-bit *N* message blocks.

These N message blocks of 1024-bit size are dealt one at a time: It begins implementation by considering a fixed initial hash value $H^{(0)}$, sequentially compute

$$H^{(i)} = H^{(i-1)} + C_{M^{(i)}}\left(H^{(i-1)}\right) \quad (3)$$

where C - *compression function,* + - word-wise $mod\ 2^{64}$ addition. $H^{(N)}$ is hash of M.

Compression functions used in SHA-512 are mentioned below:

$$Ch(x, y, z) = (x \wedge y) \oplus (\sim x \wedge z) \quad (4)$$
$$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \quad (5)$$
$$\Sigma_0(x) = S^{28}(x) \oplus S^{34}(x) \oplus S^{39}(x) \quad (6)$$
$$\Sigma_1(x) = S^{14}(x) \oplus S^{18}(x) \oplus S^{41}(x) \quad (7)$$
$$\sigma_0(x) = S^{1}(x) \oplus S^{8}(x) \oplus R^{7}(x) \quad (8)$$
$$\sigma_1(x) = S^{19}(x) \oplus S^{61}(x) \oplus R^{6}(x) \quad (9)$$

where $\oplus$ is bitwise XOR, $\wedge$ is bitwise AND, ~is bitwise complement , + is mod 2^{64} addition, R^n is right shift by n bits, S^n is right rotation by n bits.

VI. RESULT

Time required for encryption, tag generation, key generation and check duplication is calculated which are illustrated in figure 3 below. It can be seen from distribution chart in figure 3, time required for tag generation, key generation, duplication check and encryption increase as file size increases. Here, tags are generated for performing duplicate check for files that reside over cloud to check whether file being uploaded is duplicate or not. Time required for tag generation is very less, because MD5 has a very fast response time [10] and it generates tag very fast. Keys are generated using SHA-512, to encrypt contents of file that needs to be uploaded. SHA-512 hashing technique is used for generation of key since to accomplish convergent encryption concept wherein data must be encrypted using convergent key which must be derived from data itself. SHA-512 considers 512-bits block while processing data. SHA-512 and SHA-256 follows similar structure for generating hashes, difference is only that they perform their operation on blocks of different sizes. For encryption, linear congruentiality algorithm is used, it requires very less time for encryption since its working involves only performing basic arithmetic operations and it easily accepts key of variable length so convergent key derived using SHA-512 can be easily used for encrypting data.

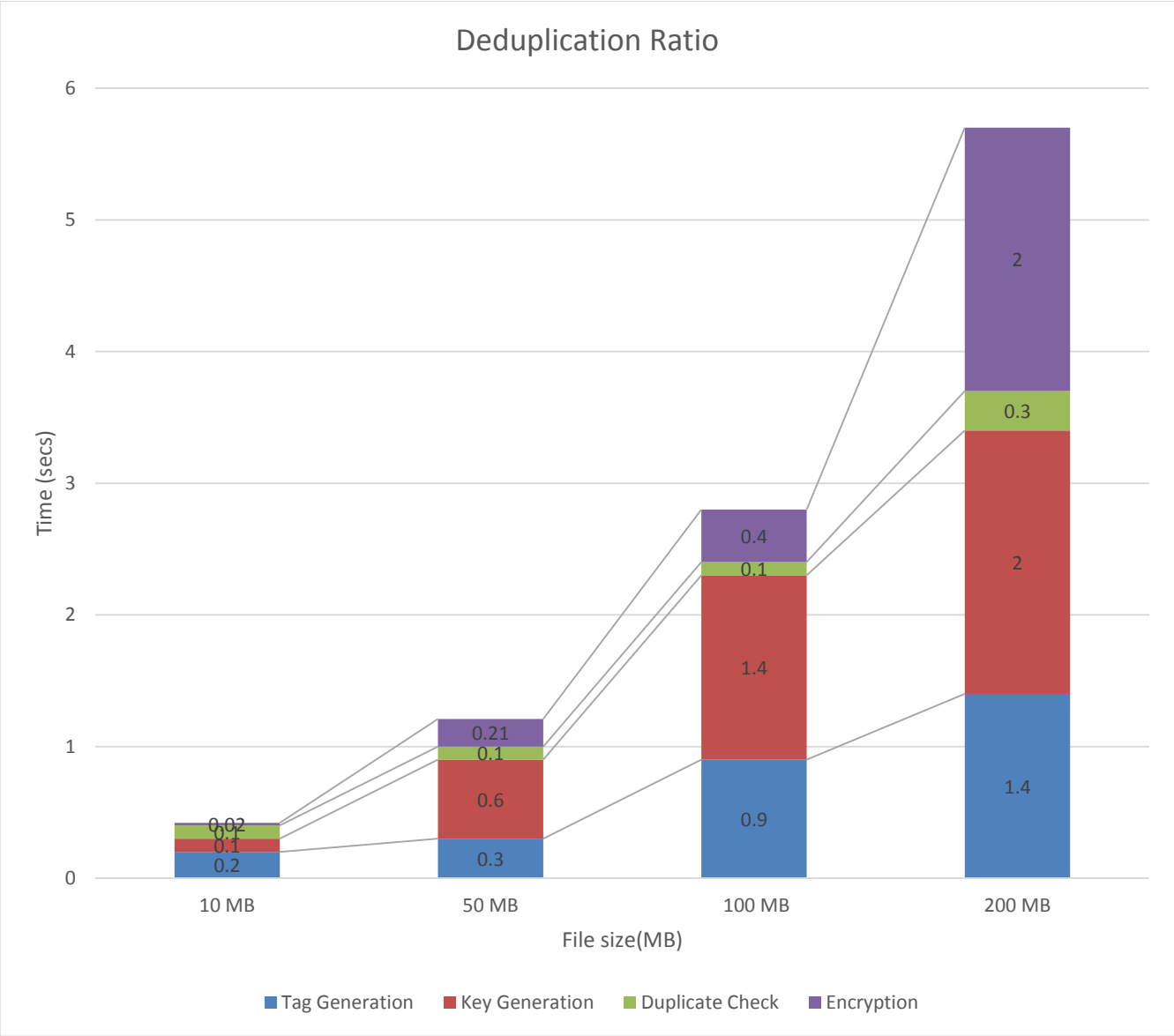

Fig 3: Deduplication ratio for files of varying sizes.

VII. Conclusion

In this paper, an authorized data deduplication scheme is implemented in which deduplication is performed by using linear congruentiality algorithm. Data is secured by assigning differential privileges of users for performing duplicate check maintained with help of access control list. User performs authorized duplicate check to check for file in cloud storage after which user can upload or download file from cloud depending on whether file exists on server or not and accordingly owners of file are updated. This framework is applicable for both large scale and small scale organization that require access control mechanism along with privacy preservation.

References

[1] P. Anderson and L. Zhang, "Fast and secure laptop backups with encrypted de-duplication", In Proc. of USENIX LISA, 2010.

[2] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Dupless: Server-aided encryption for deduplicated storage", In USENIX Security Symposium, 2013.

[3] Mr. Antony Xavier, Dr. V. Sai, Dr. S. P. Rajagopalan, "Enhanced De-key Approach to Reduce Data De-Duplication in Cloud Storage", p. 5316-5320, 2016.

[4] M.W. Storer, K. Greenan, D.D.E. Long, and E.L. Miller, "Secure Data Deduplication", p. 1-10, 2008.

[5] D. Kim, S. Song, B. Choi, "Data deduplication for data optimization for storage and network systems." p. 42-44, 2017.

[6] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data deduplication scheme for cloud storage," Tech. Rep. IBM Research, Zurich, ZUR 1308-022, 2013.

[7] https://techcrunch.com/2011/09/18/bitcasa-explains-encryption

[8] https://en.wikipedia.org/wiki/Convergent_encryption

[9] Wanzhong Sun, Hongpeng Guo, Huilei He, Zibin Dai, "Design and Optimized Implementation of the SHA-2(256, 384, 512) Hash Algorithms ", 2007

[10] Anak Agung Putri Ratna, Anak Agung Putri Ratna, Anak Agung Putri Ratna and Muhammad Salman, "Analysis and Comparison of MD5 and SHA-1 Algorithm Implementation in Simple-O Authentication based Security System",International Conference on QiR, 2013.

[11] Rohit Rastogi, Shashank Mittal, Shashank Shekhar, "Linear Algorithm for Imbricate Cryptography Using Pseudo Random Number Generator", 2nd International Conference on Computing for Sustainable Global Development, 2015.

[12] Adviti Chauhan, Jyoti Gupta, "A Novel Technique of Cloud Security Based on Hybrid Encryption by Blowfish and MD5 ", 4th IEEE International Conference onSignal Processing, Computing and Control, 2017.

[13] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P.C. Lee, and Wenjing Lou, "A Hybrid Cloud Approach for Secure Authorized Deduplication", Ieee Transactions On Parallel And Distributed Systems, Vol. 26, No. 5, May 2015.