TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 1550 - 1558

Deduplication - Encrypted Single Instance Data Storage in Community Cloud

Dr.K. Karuppasamy, I. Mettildha Mary, M. Priyadharsini, S.S. Sugantha Mallika
Prof& Head, Department of Computer Science and Engineering, RVS College of Engineering and Technology.
Department of Information Technology, Sri Ramakrishna Engineering College, Coimbatore, India.
Department of Information Technology, Sri Ramakrishna Engineering College, Coimbatore, India.
Department of Information Technology, Sri Ramakrishna Engineering College, Coimbatore, India.

Article Info
Volume 82
Page Number: 1550 - 1558
Publication Issue:
January-February 2020

Article History
Article Received: *14 March 2019*
Revised: *27 May 2019*
Accepted: *16 October 2019*
Publication: *07 January 2020*

Abstract:
Subtracting the cost of storage by saving only one replica of similar data is unprecedented with the drastic rise in the magnitude of data that is maintained in the social cloud. For guaranteeing data confidentiality, they are generally encrypted prior to outsourcing. Conventional encryption will invariably provide many different ciphers that are made from the same plain text by different users' secret keys, which can interfere with data exclusion. Combined encryption makes it possible to opt out by simply encrypting the same plain text in the same cipher text. An auxiliary issue is the means of robustly and efficiently managing large number of convex keys. A number of minimization schemes have been proposed to overcome the combined management problem. But, key management servers need to be introduced or communication is necessary among data owners. In this technical work, a new client-end exclusion protocol called KeyDup is designed without using an autonomous key management server using the Advanced Encryption Standard (AES) approach. Users have interaction only with the Cloud Service Provider (CSP) during the times of uploading and downloading the data. Security analysis shows that the KeyDup ensures data confidentiality, simultaneously protecting the integrity of key security and privacy. A rigorous and comprehensive performance comparison reveals that this program makes the best transfer in terms of cost of storage, communication and computational overhead.

Keywords:*Deduplication, Convergent Encryption (CE), Advanced Encryption Standards (AES), Cloud Service Provider (CSP).*

I. INTRODUCTION

Cloud computing includes shared pools of customizablecomputing resources and high-level services that can be delivered quickly with minimal management effort, often via the Internet. Cloud computing as a public utility relies on sharing resources to achieve synchronization and economy. It’s known as cloud computing as the information that is being accessed is saved in "the cloud" and a user is not required to be in a particular location to obtain access to it internal failure, and pay-per-utilize[3]. The most essential and well known cloud administration is information stockpiling administration. Cloud clients transfer personal or classified information to the Cloud Service Provider (CSP) server farm, allowing them to keep this information. Since interruptions and attacks on CSPs are inevitable, it is reasonable to expect that cloud clients cannot fully rely on CSP. Furthermore, loss of control over their own information triggers big data security risks, especially information security leaks. Because of the quick improvement of information mining and different examination innovations, the

January - February 2020
ISSN: 0193 - 4120 Page No. 1550 - 1558

security issue gets to be distinctly genuine. Subsequently, a great practice is to just outsource encoded information to the cloud with a specific end goal to guarantee information security and client protection. Be that as it may, the same or diverse clients may transfer copied information in scrambled frame to CSP, particularly for situations where information is shared among numerous clients. Despite the fact that Cloud storage space is immense, information duplication extraordinarily squanders organize assets, devours a great deal of vitality, and entangles information administration[14]. The advancement of various administrations additionally makes it pressing to convey productive asset administration instruments. Thus, deduplication gets to be distinctly basic for big data stockpiling and handling in the cloud.

II. PROCESS INVOLVED IN DEDUPLICATING DATA IN CLOUD

Due to evolution of the Big Data period, stored data is growing exponentially. If you continue to use the traditional storage path, you have to consistently improve the storage devices. As an alternative, an increasing number of users are likely to outsource their storage to cloud, like Amazon Web Services (AWS) for saving cost. With ever-rising data and users, along with many backups and few other factors, files or modules are being copied more and more in the cloud. In order to improve warehouse storage performance, the Sample Deduction function is adopted where you can delete duplicate data onto the Cloud page[3]. Take an instancewhere ‘M’ users are outsourcing the replicas of same data of ‘N’ TB to CSP.

In the case of data exclusion, just one duplicate is stored in the cloud, and subsequent events are specified to reduce the storage copy again from ‘MN’ to ‘N’ TB.

But, for ensuring the security of the outsourced information, the owners generally encrypt them before they are outsourced to the CSP. Thereafter, arises the issue, on the way that CSP can be excluded when these same data replicas get encrypted by different cipher texts by different users.

Step 1: Upload Data and Generation of Convergent Key

Convergent encryption (CE) encrypts the data copy with an integrated key acquired by calculating the cryptographic hash value of the duplicated data so that it can produce the same cipher[6].

It gives you the confidence to feel excluded while ensuring data confidentiality. Specifically, users perform the encryption of their data copies with the help of the associated accumulation keys and outsourced encrypted data to the CSP.It is necessary thatthe keys are held together such that they can retrieve the data at a later point of time.However, as a data copy is associated with a convex key, the number of interconnected keys sees a linear increase with the number of data replicas.

Step 2: Uploading Data in Cloud

In standard file storage systems, like the Google File System (GFS) [4,10] and the Hadoop Distribution File System (HDFS), the data files are generally delimited into simple volumes, simplifying subtraction management, and consolidating key storage. Also makes radical [5]. Suppose the ‘N’ is partitioned into blocks, with each one of 4 KBsize, convex key is a hash value of SHA-256. All the owners have to store an overall size of 8N GB of convergent keys. This overhead on storage is still a heavy burden for the user.

Step 3: Description of Keys

Every owner who has the data ciphers all of his data replicas employing the associated convergent keys and encodes these convergent keys making use of his master key. Both the encrypted data replicas and convergent keys are saved in the cloud, and users have metadata about their master keys and outsourced data saved in local storage.Even though the cloud can consume ciphered data, the storage of encrypted convergent keys linearly increases as the number of users.

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 1550 - 1558

Step 4: Managing Data in Cloud

Numerous efforts have been aimed towards reducing the convergent key management problem from O(m) to O(1), which means that the expense incurred in storing the convergent keys is independent of the number of owners using the Ramp Secret Distribution Scheme (RSSS) [7,8] Management Cloud Service Providers (KM -CSPs) for distribution through. In order to retrieve the data duplicates, a user needs to access at least a minimal number of key servers by means of authentication and receive confidential shares to remake the convergent keys. The paradigm involved in the approach security is that the number of combined KM-CSPs does not exceed the predetermined threshold; Combined KM-CSPs cannot predict a convergent key. a session-key-based convergent key management program and advanced variant is proposed. You need a trusted gateway to verify the copy and upload the new data sets to the Cloud Storage Server (CSS) from time to time.

III. EXISTING SYSTEM

Data reduction has gained growing significance in storage systems owing to the exponential increase of digital information globally, which has paved way for the big data revolution. One of the major problems faced in large-scale data minimization is the means of increasingly identifying and reducing the redundant activity incurring limited overheads. As per this system, it DARE[16], which is a least-overhead deduplication-sensitive resemblance detection and delimination approach is presented and itis efficient in exploiting the availableinformation on duplicate-adjacency for extremely effectivedetection of resemblance in data deduplication depending on backup/archiving storage systems. The important concept of DARE is to use an approach, known as Duplicate-Adjacency based Resemblance Detection (DupAdj)[16], by taking any two data segments to be identical (i.e., candidates considered for delta compression) when their corresponding neighboring data segments are replicated in a deduplication system, and then the efficiency of resemblance detection is increased further using a modified super-feature technique.

Disadvantages of existing system.

- Deduplication systems are secure.
- No efficient deduplication and security access.
- Does not possess high reliability.

IV. PROPOSED SYSTEM

We consider both the process of file upload and download in KeyDup. For conserving the bandwidth used in upload, users just perform the uploading of new data, which is not present in the cloud storage. Once a file upload request T(F) is received with a user ID, the CSP first verifies that the (F) is stored. Otherwise, it carries out block-level copy verification and just the individual modules need to be uploaded Before outsourcing data storage to CSP, the user first performs the encryption of the data using the convergent keys, ensuring that the same data replica results in the same ciphertext. For a huge count of convergent keys, we use the Identity Based Broadcast Encryption (IBBE) key rather than the primary secret key of the user.

Let S refer to a pool of identities of the users who share the same convergent key (which is the same data replica). The convergent key is encrypted using kB and transmitted to the CSP. All the users present in S have the ability to retrieve this convergent key that utilize their individual keys. As to those duplicate modules, the user operates the Owner Testimonial (PoW) protocol with the CSP to assert their rights. The proprietorship of these modules will get renewed by means of the interactions happening between the user and the CSP.

Advantages of Proposed System

- Minimizes the space required for Storage and prevents copied files.
- Has Single Ownership for every file.
- Confidentiality maintained of files during Upload/Download.

Minimizes the space required for Storage and retrieves the files with accuracy.

January - February 2020
ISSN: 0193 - 4120 Page No. 1550 - 1558

V. SYSTEM ARCHITECTURE

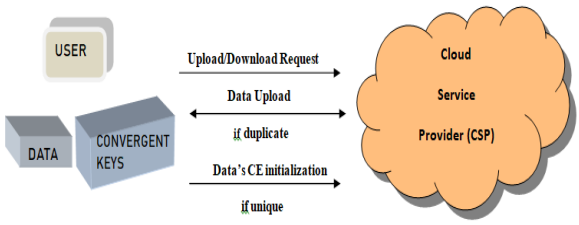

Fig.1: Proposed System Architecture

KeyDup has two processes, one for file upload and the other for download of file. To save the bandwidth used for upload, users just uploads new data, which is not present in the cloud. Once an upload request T (F) for a file is received with a user ID, the CSP first verifies whether F is stored. Otherwise, it carries out block-level copy verification and just the uploading of the individual modules have to be done. Prior to outsourcing the data storage to the CSP, the user first performs the encryption of the data employing the combined keys that guarantees that the same data replica results in the same encrypted text. For a huge number of converged keys, we use the Advanced Encryption Standards (AES) key in place of the primary secret key of the user. Let S refer to a group of users' identities who share the same key (ie the same replica of data).

In particular, the encrypt algorithm of AES will generate an AES key kB and a header Hdr to encapsulate kB. The convergent key is encrypted using kB and transmitted to the CSP. All users in S have the ability to retrieve this convergent key utilizing their own keys. As with those duplicate modules, the user operates the Owner Certificate (Pow) protocol with the CSP to assert its proprietorship. The ownership of these modules (ie S and HDR) will be renewed through interactions between the user and the CSP. If F is a copy, all its modules will be copied and the ownership of every block must get updated. File download involves one single round of communication happening between the user and the CSP. Once a file download request is received from a user, the CSP verifies whether the user is authorized. If it is not so, then the download stops. Otherwise, the CSP provides encrypted modules and encrypted convergent keys with the AES headers and receiver sets of all the volumes that make up this file. The user can eventually retrieve the entire file. (Fig 1)

VI. TECHNIQUES USED IN PROPOSED SYSTEM

A. Convergent Encryption (CE)

Convergent encryption [15], presented by Dussour et al. is extensively employed in the cloud data storage. It is actually a cryptosystem, which generates similar cipher text files from the same simple text files, regardless of an encryption key. To perform the encryption of a data replica (a file or a block) employing concentric encryption, as a primary step, the user computes a hash value that is cryptographically strong from the data replica, and afterwards makes use of this particular hash value in the form of an integrated key for encrypting the data copy[6]. The user can get a tag for the data replica that, in turn can be used to locate the copy. In this manner, the same data replica will get ciphered with the same key, producing the same cipher text and tag. The cyber text and tag is then delivered to the server and the end-user holds the key together. Now, the server can now subtract in cipher text, checking whether it is already saved or not. It is to be noted that both of the convergent key and the tag are separately obtained, and the tag cannot be utilized to compromise data confidentiality by omitting the combined key. An integrated encryption approach can be officially specified as the tetrad of the four methods (keyGen, Encryption, Decrypt, TagGen)[12] given below :

- KeyGen(M) → K refers to the key generation algorithm, which performs the mapping of a data replica M onto a convergent key K;
- Encrypt(K,M) → C refers to the symmetric encryption algorithm, which uses both the convergent key K and the data copy M in the form of inputs and then generates an output of a ciphertext C;
- Decrypt(K,C) → M refers to the decryption algorithm, which considers both the

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 1550 - 1558

ciphertextC and the convergent key K in the form of inputs and in the next step, produces the actual data replica M as the output;

- TagGen(M) → T (M) refers to the tag generation algorithm, which performs the mapping of the actual data copy M and produces an output result of a tag T (M).

Steps Involved in SHA – 256

1. Padding of bits - If M is the message that needs hashing, and l refers to its length in bits where $l < 2^{64}$, then as a prime step, the padded message M' is created, which is message M with a correct padding included, in such a way that M' has a length l', which is a multiple of 512.
2. Append length – M' gets parsed into N blocks with each 512 bits size, M^1 to M^N, and every block is divided as 16 input blocks having a 32 bits size, M_0 to M_{15}.
3. Divide the input into 512 bit blocks - The initial hash value H^0having length 256 bits (8 input blocks comprising of 32 bits) is made by considering the first 32 bits of the fractional portions of the square roots taken of the first eight prime numbers.
4. Initialize chaining variables - Message block M^i is the first block of $W^i$and the next subsequent three blocks are M's versions.
5. Process blocks - The input blocks belonging to the message schedule Ware sent, one after another, to a function indicated below in the form of a graph. The graph a hash $\omega^i(t)$ as inputs and a message schedule input block $W^i(t)$, and produces an output of a hash $\omega^i(t+1)$.

B. Proof of Ownership (PoW)

If two or multiple users have the same file, just one replica will get saved in the cloud. Once a user uploads an existing file, he demonstrates to the user the file's ownership and receives information related to storage, such as a copy pointer. The user transmitting a hash value of the data copy to the server directly for the purpose of checking is found to be a viable solution. But, because the data is in ciphered form, the cloud server is not capable to calculate the hash value of the data replica and store multiple hash values with the data replicas, particularly if the data size is so huge that the concept of ownership (PoW) [11] was first proposed, without actually sending the file. To the server that the file has a copy Helps the client to prove. It is basically an interactive protocol used between a proverb (user) and a verifier (server). The verifier initially gets a small value (M) from the data replica M. In order to assert the proprietorship of the data replica M, the proper is assumed to send μ′ to the validator. If μ = μ(M), the validator will consider that the proper indeed includes M where μ is the time invariant.

In this manner, bandwidth efficiency is achieved and validation is attained between the user and the server. For the sake of simplicity, the notations of POWF and POWB is also used to refer PoW for a file F and block B, correspondingly.

C. Advanced Encryption Standards (AES)

AES makes use of the same secret key that, in turn, is used for the both encryption and decryption process. Dissimilar to AES 128 bit encryption and decryption, if a strong AES 256 bit key is required, Java cryptography extension (JCE) is required. We follow advanced encryption standards for encrypting the convergent keys and they are sent to the Cloud Service Provider (CSP). The private key generator PKG is one essential dominative unit used in the AES. MSK makes use of its primary secret key; PKG can create an encryption key for every new member to decrypt the messages with the ID. One interesting feature of the AES program is that the data holder does not maintain any sensitive data. The encryption of messages can be done using a public key and a group of recipients' names[13]. After this, all the identities in S are capable of decrypting the data. Provided the security parameter S(k) and maximal size m of the target set, an AES program can be properly defined as a tuple made up of

January - February 2020
ISSN: 0193 - 4120 Page No. 1550 - 1558

algorithms (Setup, Extract, Encrypt, Decrypt):Setup(M). The maximum size of a group of receivers for an encryption is taken as input of s (k) and m, releasing a primary secret key MSK along with a public key PK. Extract (MSK, ID). The primary secret key takes the MSK and user ID as input. The designing system creates a user private key Encryption (S, PK). The public key PK and the group of entered identities outputs S = {ID1,IDs} with s ≤ m, along with a pair (Hdr,K). When the message M{0, 1} message is broadcast to the users present in S, the broadcaster (Hdr,M) ← Encrypt(S, PK), performs the encryption CM of M in the presence of the symmetric key M and encrypt (Hdr,S,CM).

Hdr will be referred as the header (which actually encapsulates the advanced encryption) of broadcast ciphertext, K in the form of the message encryption key and CM in the form of the broadcast body.

Steps Involved in AES

1. Obtain agroup of round keys out of the cipher key.
2. Start the state array using the block data (un-encrypted text).
3. Append the initial round key to the beginning state array.
4. Conduct 9 iterations of state manipulation.
5. Carry out the 10thand last iteration of state manipulation.
6. Output the finalized state array out to be the encrypted message (ciphered text).

VII. SECURITY ANALYSIS AND PERFORMANCE EVALUATION

A. Security Analysis

The newly introduced approaches realize a secure mechanism for data deduplication process. CSP has no access to the un-encrypted text at any point of time since the file encrypted by the Data holder. The CSP evaluates the Hash value of the encrypted text. If CSP and Users function without any interference, then it is assured that there is no compromise of data stored in the cloud. The data gets ciphered before it reaches the CSP. Therefore, CSP won‘t have the knowledge of the original data existing in ‘m’. This is guaranteed since the user won‘t have the original data shared with the CSP. Also, CSP will possess solely the data, which is being transferred to CSP. This is to ensure that just E (m) data gets stored. During the event of any compromise on CSP Storage, the actual data cannot be touched by the attacker as only the Data Owner will have access for reading it once more utilizing the CSP interface attached to his device.

B. Data Encryption and Decryption Efficiency

In this test, the amount of time consumed by different AES encryptions standards is computed. It was observed that the higher the bit size the higher is the time taken for the file’s encryption. After the computation of the number of outputs, which is predicted to be output from various AES standard it was demonstrated that a number of 2^128 diverse combinatorial outputs can be obtained, when AES 128 bit encryption is utilized. Employing Birthday attack, a kind of brute force attack will use up 2^64 combinations to defeat this. Attaining this level of distinctiveness and speed has pushed AES to be the foremost preference mainly for attaining lesser time for encryption - decryption in comparison with the rest. This will help in huge reduction of the amount of time consumed during the while process if big data files should be used. During the functional testing carried out on the newly introduced system the below transformation of user‘s file ‘m’ into E (m) is illustrated. The functional test involving the Data Client making a request for data access privileges and transmitting the Data Client secret random key Kr is shown. The Data Client inputs the key in CSP portal and can then have access to the file. (Fig 2)

January - February 2020
ISSN: 0193 - 4120 Page No. 1550 - 1558

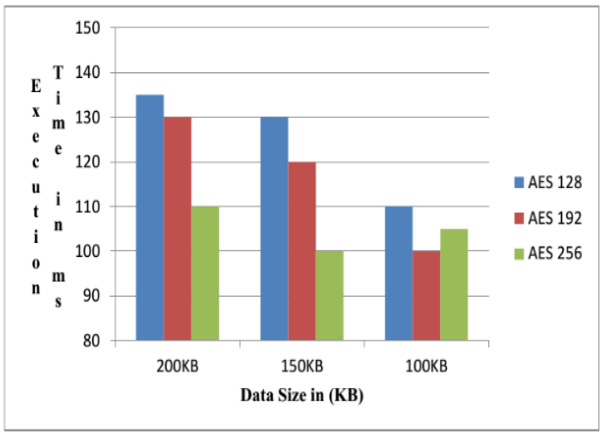

Fig. 2: Performanance of different AESAlgorithms

C. Functional Test

We set up three users in the system of community cloud for testing to show that the proposed approach performs well with the right functional implementation.

User desires to enroll with system IP, name and few other information that can be saved in databases and it can be maintained by the admin. Since the system is designed for community cloud, the specific user of certain organizations having the IP of the system.

Fig 3 shows the evidence of ownership. The admin provides the ownership of every user. So that duplicates could be avoided. The admin provides the individual ownership for every user. Deduplication necessitates that in case of two or multiple users having the ownership of the same file, just one single copy needs to be maintained in the cloud.

If the users perform a file upload, which existed, he will assert the file ownership to the user and gets the knowledge related to the storage, like the pointer of the copy.

SHOWING ALL USERS

USER ID	IP ADDRESS	USERNAME	MAIL ID	PHONE NUMBER
1	192.168.1.1	priya	priya@gmail.com	78945613230
2	192.168.1.2	aishu	aishu@gmail.com	9876543210
3	192.168.1.3	jo	jo@gmail.com	6586798532

Fig. 3: Verify Proof of Ownership

Fig.4 shows File level verification for checking deduplication. Uploading the file in a database which can be secured by convergent encryption process, now it generates hash key and with the hash value we check for duplicated data by uploading files.

In the case of File level deduplication, the hash value is created for a file and then compared with the hash value belonging to other files that are saved in the database. If the hash value does not match the file is stored else it refers that file is already exists.

VIEW FILES

USER ID	TITLE	HASH VALUE
1	hello	09420da33cc25f9a29990d82ce6b2c4a442c99aaf984cb470e6a504c513b5412
2	welcome	38d71dae699ffdfb8decd3909e0d966b1e00f514e38ae69fb1e25f8b07e1e560
3	welcome	38d9e62c5e110de0c8b899cb7730a7b62d0d68f383d00dd8841874337bcd5f1c
4	document	24d8ce6e68242eb80621a67520e1aaf516ca384f19ab39562702de54ae4e617a

Fig. 4: File Level Verification

January - February 2020
ISSN: 0193 - 4120 Page No. 1550 - 1558

Fig 5 shows Block level deduplication. The file is splited into several blocks and the hash value is generating for every block. It compares with hash value of other file blocks stored in the database. If any duplicate block presents, it removes the duplicated blocks and stored in the database. So that the accuracy of block level deduplication is very high also the blocks are divided by each 4KB of files.

Fig. 5: Block Level Verification

VIII. CONCLUSION

KeyD is a secure client-side deduplication program for managing convergent keys effectively. Data deduplication in this work is accomplished through the interactive activities happening between the data owner and the cloud service provider (CSP), with no role played by other trustworthy third parties or core management cloud service providers. Security evaluation reveals that this KeyDup assures the confidentiality imparted by our backup information and the security achieved through the use of convergent keys, while protecting user's privacy. The test results demonstrate that our program does not trade-off safety performance. As to the work intended for the future, attempts would be made to look for means to safeguard the identity rights of the data holders not taken into consideration in this project.

IX. REFERENCES

[1] Amazon Web Services, [Online]. Available: https://aws.amazon.com/cn/.

[2] A Survey and Classification of Storage Dedulplication System, ACM Computing Surveys, Vol. 47, No. 1, Article 11, Publication date: May 2014.

[3] J.R. Douceur, A. Adya, W.J. Bolosky, D. Simon, and M. Theimer, Reclaiming Space from Duplicate Files in a ServerlessDistributedFile System[C]. Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on. IEEE, 2002: 617-624.

[4] Zghair, A.N., Sharma, R., Alfaham, M., Sharma, A.K. Upregulation of BRCA1, ERBB2 and TP53 marker genes expression in breast cancer patients(2018) International Journal of Pharmaceutical Research, 10 (2), pp. 147-154.

[5] S. Ghemawat, H. Gobioff, and S. Leung, The Google File System [M]. SOSP '03 Proceedings of the nineteenth ACM symposium on Operating systems principles, 2003, 37(5): 29-43.

January - February 2020
ISSN: 0193 - 4120 Page No. 1550 - 1558

[6] D. Borthakur, HDFS architecture guide [J]. Hadoop Apache Project, 2008, 53.

[7] J. Li, X. Chen, M. Li, J. Li, P.P.C. Lee, and W. Lou, Secure Deduplication with Efficient and Reliable Convergent Key Management[J]. IEEEtransactions on parallel and distributed systems, 2014, 25(6): 1615-1625.

[8] G.R. Blakley and C.A. Meadows, Security of Ramp Schemes[C].Crypto. 1984, 84: 242-268.

[9] A.D. Santis and B. Masucci, Multiple Ramp Schemes[J]. IEEE Transactions on Information Theory, 1999, 45(5): 1720-1728.

[10] M. Wen, K. Ota, H. Li, J. Lei, C. Gu, and Z. Su, Secure Data Deduplication with Reliable Key Management for Dynamic Updates in Cpss[J]. IEEE transactions on computational social systems, 2015,2(4): 137-147.

[11] W. Leesakul, P. Townend, and J. Xu, Dynamic Data Deduplication in Cloud Storage[C]. IEEE, International Symposium on ServiceOriented System Engineering. IEEE, 2014:320-325.

[12] F. Rashid, A. Miri, and I. Woungang, A secure data deduplication framework for cloud environments[C]. Tenth International Conferenceon Privacy, Security and Trust. IEEE Computer Society, 2012:81-87.

[13] M. Bellare, S. Keelveedhi, and T. Ristenpart, Message-Locked Encryption and Secure Deduplication[C]. Annual International Conferenceon the Theory and Applications of Cryptographic Techniques.Springer, Berlin, Heidelberg, 2013: 296-312.

[14] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, Message-Locked Encryption for Lock-Dependent Messages [M]. Advances in Cryptology CCRYPTO 2013. Springer, Berlin, Heidelberg, 2013: 374-391.

[15] M.W. Storer, K. Greenan, D.D.E. Long, and E.L. Miller, Secure Data Deduplication[C]. Proceedings of the 4th ACM internationalworkshop on Storage security and survivability. ACM, 2008:

[16] P. Anderson and L. Zhang, Fast and Secure Laptop Backups with Encrypted De-duplication[C]. LISA. 2010.

[17] Khirodsankar das, sudiptachoudhury, k. Chanreila l. Nonglait (2017) zootherapy among the ethnic groups of north eastern region of india-a critical review. Journal of Critical Reviews, 4 (2), 1-9. doi:10.22159/jcr.2017v4i2.14698

[18] W. Xia, H. Jiang, D. Feng and L. Tian, "DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduction with Low Overheads," InIEEE Transactions on Computers, vol. 65, no. 6, pp. 1692-1705, 1 June 2016.

[19] Kaur H, Saini S, Peer S, Singh J. "Current Therapies and Novel Targets in Treatment of Breast Cancer." Systematic Reviews in Pharmacy 1.1 (2010), 40-49. Print. doi:10.4103/0975-8453.59511