www.ijaceeonline.com ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 7, July 2019, pp. 7–15

Secure Cloud Data Deduplication Using Tag Equality Testing

S. Muthurajkumar
Assistant Professor, Department of Computer Technology, MIT Campus,
Anna University, Chrompet, Chennai, India
Email: muthurajkumarss@gmail.com

Abstract: *Cloud computing is one of the fast growing paradigms in Computer Science. Cloud storage is a service model in which data is maintained, managed, backed up remotely and made available to users over a network. The storage service is generally pay-per-use basis in which the users pay for the amount of storage used by the user. Cloud storage provides accessibility across a variety of devices, reliability, rapid deployment, strong protection for data backup and lower overall storage costs as a result of not having to purchase, manage and maintain expensive hardware. There are many benefits of using cloud storage however, cloud storage does have the potential for security and compliance concerns that are not associated with traditional storage systems. In this modern technology world, massive amount of data are produced that has become necessary to handle those big data on demand which is a challenging task for current data storage systems. Data De-duplication is a process that eliminates redundant copies of data which in turn reduces storage overhead. Reducing redundant storage can lead to efficient use of the Cloud storage. The main aim of the project is to achieve deduplication on secured data on client side. The proposed method checks for the duplicate file in cloud server before actually uploading the file to cloud server which ignores the storage of redundant data in cloud servers by using Message Locked Encryption (MLE) technique. The method also reduces network bandwidth consumption as the entire file need not be transmitted to the server if there is de-duplication.*

Keyword: *Data Storage; Data Deduplication; Cloud Computing; Message Locked Encryption*

1. INTRODUCTION

A growing computer paradigm for data and services to function in large and scalable data centres in the cloud and accessed from devices that are connected over the internet. Cloud computing comes into picture when the cloud is used for utilities. For example pictures and videos can be stored in the cloud instead of the home computer so that it can be viewed and managed from anywhere. For a company, the invoice system can be deployed in the cloud instead in home invoice system for the ease of access. The cloud computing offers services over the internet. Internet is used to provide services of storing data, computation performing and other tasks. For an organization a lot of its processes can be automated over the cloud for improvising the reliability.

The main characteristics of cloud computing are Infrastructure Sharing, Dynamic Provisioning, Access of Network and Managed Accounting. Infrastructure Sharing is the cloud aims mainly at resource sharing where the computing, networking and storage capabilities and resources are virtualized and they are shared among the many users. Dynamic Provisioning is the cloud can be provisioned and the services can be provided according to the dynamic user needs and requests. The service can be reduced or expanded based on the demands. Thus the cloud can be very scalable which paves way for efficiency and reliability. Access of Network is the cloud applications are accessed through the internet along a wide range of devices such as PCs, laptops, mobile and other handheld devices. The applications and services are based and solely depend on the internet access and delivery of service is not possible without networking. Managed Accounting is the cloud services use a separate accounting and management service to ease the billing and accounting information. The cloud usage of the user is accounted so that the user can pay for what they use only.

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. As such, the use of remote storage systems is gaining an expanding interest, namely the cloud storage based services, as it provides cost efficient architectures. These cloud architectures support the transmission, storage in a multi-tenant environment, and intensive computation of outsourced data in a pay per use business model. For saving resources consumption in both network band-

Cite this paper:

S. Muthurajkumar, "Secured Cloud Data Deduplication Using Tag Equality Testing", International Journal of Advances in Computer and Electronics Engineering, Vol. 7, No. 7, pp. 7-15, July 2019.

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 7, July 2019, pp. 7–15

width and storage capacities, client side de-duplication is used. A client could encrypt its file, under a user's key before storing it. But, common encryption modes are randomized, making de-duplication impossible since the SS (Storage Service) effectively always sees different cipher texts regardless of the data. Therefore a deterministic encryption technique is chosen to resolve the situation. The proposed approach checks for the duplicate file in cloud server before actually uploading the file to cloud server without affecting the confidentiality of the file.

Cloud storage is a cloud computing model in which data is stored on remote servers and can be accessed from the internet or cloud. The advent of cloud storage enables the organizations and enterprises to outsource their data to third party Cloud Service Provider (CSP). Cloud storage provides several benefits to customers that includes cost saving of resources, data availability, simplified convenience, mobility opportunities and scalable service, backup and disaster recovery. These great features attract more and more customers to utilize and store their personal data to the cloud storage. The data in cloud server is maintained, operated and managed by the cloud storage service provider. Moving the data onto the cloud offers significant advantages in resource saving. It also provides great convenience to users as they don't have to worry about the complexities of hardware, software and their maintenance. Stored files can be accessed from anywhere at any time via internet connection.

The main objective of the paper is to effectively utilise the cloud storage space allocated to the user by Cloud Service Provider (CSP). The system used to achieve client side de-duplication which avoids redundant data storage in the cloud even before uploading the file which saves the storage space and also effectively utilises the bandwidth. It also aims in providing security to the data by avoiding unauthorized users from accessing the data stored in the Cloud Server.

2. LITERATURE SURVEY

There has been several research works contributed in the literature on several cloud storage methods and their retrieval procedures. The different methods of Cloud Storage, methods of de-duplication and the types of encryption and decryption techniques that have been used are surveyed and studied in this section. Jingwei et.al [1] have proposed a method to achieve data integrity and de-duplication in cloud, a system called SecCloud is proposed, which helped clients generate data tags before uploading as well as audit the integrity of data stored in cloud. Another system called SecCloud+ was put forward, where clients encrypt the data before uploading the files.

Jia Yu et.al [2] proposed a model that employed the binary tree structure and the preorder traversal technique to update the secret keys for the client. The authors developed a novel authenticator construction to support the forward security and the property of block less verifiability. Baojiang Cui et.al [3] had addressed the practical problems like secure communication, storage and complexity which was largely neglected in the literature, by proposing the novel concept of key-aggregate searchable encryption and instantiating the concept through a concrete KASE scheme, in which a data owner only needs to distribute a single key to a user for sharing a large number of documents, and the user only needs to submit a single trapdoor to the cloud for querying the shared documents.

Kan Yang et.al [4] have proposed an expressive, efficient and revocable data access control scheme for multi-authority cloud storage systems, where there are multiple authorities co-exist and each authority is able to issue attributes independently. The authors proposed a revocable multi-authority CP-ABE scheme, and applied it as the underlying techniques to design the data access control scheme.

Chen et.al [5] proposed a method for Secure deduplication with efficient and reliable convergent key management, where each user holds an independent master key for encrypting the convergent keys and outsourcing them to the cloud. However, such a baseline key management scheme generated an enormous number of keys with the increasing number of users and requires users to dedicatedly protect the master keys. The authors proposed Dekey, a new construction in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers. Security analysis demonstrated that Dekey was secure in terms of the definitions specified in the proposed security model. As a proof of concept, the implementation of Dekey using the Ramp secret sharing scheme and demonstrated that Dekey incurs limited overhead in realistic environments.

Poornashree et.al [6] proposed a Provable data possession scheme where the customer outsources the data to the remote cloud service provider which is responsible for storing and preserving the data. Customers could rent the storage infrastructure from the cloud service providers to store their data by paying fees. Therefore the customers needed to verify whether the server possesses the original data and should have strong guarantee that the service provider is storing all the data copies issued as per the agreement. In such process the issues such as data security, data dynamics, integrity protection and multi cloud storage have remained the most important task. Various PDP techniques and its extensions were discussed in this paper for implementing the process.

Deepika et.al [7] have proposed a new way of uploading a file by encrypting sensitive data in order to overcome the attacks in the existing systems like pre-

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 7, July 2019, pp. 7–15

dicting files, creating secured channel and the content distribution attack. The result in avoiding deduplication and hence none of the attacks was possible. The files could be encrypted by the private key of the user. The key should not be disclosed and should be kept personal by each and every user. However, there is a possibility of dictionary attack. The private key of any user was vulnerable to offline dictionary attacks. Also, if two users by accident use same private key then deduplication would still occur. Another problem related with the option was regarding the generation of the personal key and the bookkeeping task for example that if a user forgot his personal key. In the "convergent encryption" method even when the users used different personal keys it is capable of generating identical encrypted files from original files. The method kept the essence of deduplication while providing a good level of security.

Kai He et.al [8] have proposed a public auditing scheme for cloud storage systems, in which deduplication of encrypted data and data integrity checking could be achieved within the same framework. The cloud server could correctly check the ownership for new owners and the auditor could correctly check the integrity of deduplicated data. Joseph et.al[9] had introduced a novel two-factor data security protection mechanism for cloud storage system, in which a data sender is allowed to encrypt the data with knowledge of the identity of a receiver only, while the receiver is required to use both his/her secret key and a security device to gain access to the data. The solution not only enhanced the confidentiality of the data, but also offered the revocability of the device so that once the device was revoked, the corresponding cipher text would be updated automatically by the cloud server without any notice of the data owner.

Wenjing et.al [10] proposed a secure cloud storage system supporting privacy-preserving public auditing. The author further extended the result enables the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis showed the proposed schemes were provably secure and highly efficient.

Hong Liu et.al [11] proposed a shared authority based privacy-preserving authentication protocol (SAPA) to address privacy issue discussed in the previous system for cloud storage. The SAPA had shared access authority which was achieved by anonymous access request matching mechanism with security and privacy considerations, attribute based access control which was adopted to realize that the user can only access its own data fields and the proxy re-encryption, applied to provide data sharing among the multiple users. Hence, the proposed protocol was attractive for multi-user collaborative cloud applications.

Jia Yu et.al [12] the author focussed on how to make the key updates as transparent as possible for the client and proposes a new paradigm called cloud storage auditing with verifiable outsourcing of key updates. In the paradigm, key updates could be safely outsourced to some authorized party, and thus the key-update burden on the client would be kept minimal. The security proof and the performance simulation showed that the detailed design instantiations were secure and efficient.

Bo Mao et.al [13] proposed Performance Oriented I/O Deduplication (POD) to improve the input/output performance of primary storage systems in the cloud. The existing data Deduplication scheme for primary storage systems were iDedupe and Offline-Dedupe, these two techniques were capacity oriented. POD was primarily designed for storage and reduced small write traffic, improving cache efficiency and read performance. Online RAID reconstruction was an important and integral part of parity-based RAID systems of Hard Disk Drive (HDD). Moreover, hash computing also consumed extra power. The experimental results of the online RAID reconstruction showed that data deduplication could reduce the reconstruction time, thus improving the system reliability.

Aparna et.al [14] had addressed secure data deduplication process for every uploaded data into public cloud by separating the process of sensitive data and non-sensitive data. While accessing data from public cloud only authorized users could access the data for the sake of data read/write. For the sake of data privacy from public cloud or attackers, only sensitive data was encrypted and privileges would be given by data owner.

JunbeomHur et.al [15] proposed server-side deduplication scheme for encrypted data. The system allowed the cloud server to control the outsourced data even when the ownership changes dynamically by exploiting randomized convergent encryption and secure ownership group key distribution. In the proposed scheme, file level deduplication was done in the system and it guaranteed data integrity against any attack. In this level, Single file was duplicated and eliminated the duplicate copies of the same file. Hash values were used for file checking function. The method enhanced data confidentiality and data privacy in cloud storage against which doesn't have a valid ownership of the data. Tag consistency was guaranteed and it allowed full advantage to be taken over encrypted data of deduplication. It was more efficient for communication cost. The method achieved more secure and fine-grained ownership management in cloud storage for secure and efficient data deduplication.

Zheng Yan et.al [16] the author focussed on a scheme to deduplicate encrypted data stored in cloud based on the ownership challenge and proxy re-encryption which integrated cloud data deduplication with access control. The method could flexibly sup-

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 7, July 2019, pp. 7–15

port data update and sharing with deduplication even when the data holders were offline. The result showed the superior efficiency and effectiveness of the scheme for potential practical deployment, especially the security model was very suitable for big data deduplication in cloud storage.

Arokiam et.al [17] proposed a new cryptographic technique which was applied to address the problems in the existing system. Encrypted data were stored on storage servers while secret key(s) are retained by data owner; access to the user was granted by issuing the corresponding data decryption keys. Along with encryption, obfuscation technique was used to increase the confidentiality of data. The proposed technique was secure to store the cloud users' data in the cloud storage. Encryption only provided maximum security to user's data in the cloud data.

Jia Yu et.al [18] had formalized the definition and the security model of auditing protocol with key-exposure resilience and employed the binary tree structure and the preorder traversal technique to update the secret key for the client and developed a novel authenticator construction to support the forward security and the property of blockless verifiability. The integrity of the data previously stored in cloud could still be verified even if the client's current secret key for the cloud storage auditing was exposed.

Jianghong Wei et.al [19] had proposed a notion called revocable-storage identity-based encryption(RS-IBE) to build a cost-effective and secure data sharing system in cloud computing, which could provide the forward/backward security of cipher text by introducing the functionalities of the user revocation and cipher text update simultaneously. The proposed RS-IBE scheme was proved adaptive-secure in the standard model, under the decisional l-DBHE assumption.

Bellare et.al [20] have proposed a Message Locked Encryption scheme, a symmetric encryption scheme in which the key used for encryption and decryption is itself derived from the message. The primitive had widespread deployment and application for the purpose of secure deduplication, but in the absence of a theoretical treatment, the authors had no precise indication of what the methods did or did not accomplish. Bandwidth refers to the amount of data that can be transmitted over the internet from one point to another. There is no guarantee for the security for the data stored on the cloud. Some attackers or even the service provider may steal the data stored in the cloud. Client side De-duplication is the most recent technology that is used to avoid redundant data storage in cloud before uploading the file to the cloud by finding the hash value of the file. More specifically, an attacker who knows the hash signature of a file can convince the storage service that it owns that file, hence the server lets the attacker download the entire file. To improve storage capacity of the cloud service providers. To effectively utilize the network bandwidth allocated to the user by cloud service provider. To improve security to the data stored by the user in the cloud storage. To avoid leakage of information stored in the cloud database. To reduce the time taken to upload the same file to cloud. To avoid redundant storage of data in the cloud. To prevent malicious user from proving that he is the owner of the data stored in the cloud in case if he gains knowledge of the file in the cloud. To provide security to the data by avoiding unauthorized users from accessing the data stored in the Cloud Server. There has been several research works contributed in the literature on several cloud storage methods and their retrieval procedures [21-30].

3. PROPOSED SYSTEM ARCHITECTURE

The aim of the work is to design a system that has benefits for both the client (cloud storage user) and the server (cloud storage provider). For the benefit of user, an improved security mechanism for the data stored on the cloud is provided by encrypting the data before even uploading the file to the storage server. On the other side, capacity of the cloud storage is optimally utilised by avoiding redundant storage of the same file on the server. Encrypting the same file by using two different encryption keys produces two different encrypted files which in turn affects de-duplication concept of the cloud service provider. So an efficient mechanism which generates the hash value from the content of the text file and then uses this hash value to encrypt the data is proposed. In order to avoid leakage of data attacks, hash value with unique file identifier id is found which stored in the database. So when the client uploads the file to the cloud server, the unique id of the file is compared with those stored in the cloud server. To achieve deduplication, the server creates a link to the file by avoiding redundant data storage. The proposed work also effectively utilises the bandwidth of the client because on redundant uploading of file, only the hash value is passed to the server and checked. The client can also provide access to the data stored by in cloud to the set of users with the corresponding access permission. When the users request to download the client's file, the cloud service provider checks the authentication of the user and gives access to the file based on the user's access permission.

Client: A client makes use of provider's resources to store, retrieve and share data with multiple users. A client can be either an individual or an enterprise.

Cloud Service Provider (CSP): CSP has significant resources to govern distributed cloud storage servers and to manage its database servers. These services can be used by the client to manage the data stored in the cloud servers.

Users: Users are able to access the content stored in the cloud, depending on their access rights whose

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 7, July 2019, pp. 7–15

authorizations are granted by the client, like the rights to read, write or re-store the modified data in the cloud.

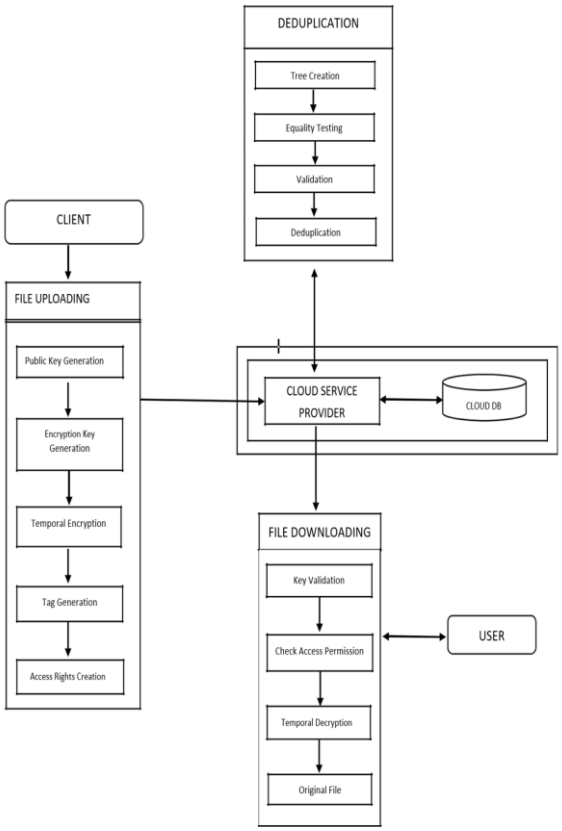

Figure 3.1 De-duplication Architecture Diagram

Access rights are specified to several groups of users. Each group is characterized by a set of users and their access rights. In practise, CSP provides a web interface for the client to store data into a set of cloud servers. In addition the web interface is used by the users to retrieve, modify and restore data from the cloud, depending on their access rights. Moreover, the CSP maintains database servers to map client identities to their stored data file identifiers and group identifiers.

Data Storage with Deduplication Check:

Step 1: The client's data file (f) to be uploaded is chosen.

Step 2: The hash value is generated from the contents of the data file using Message Locked Encryption technique, where the key for encryption is selected from the content itself.

Step 3: Secure Hashing Algorithm is used to generate the hash value (H).

INPUT: Plain Text File
ALGORITHM: Secured Hashing Algorithm
OUTPUT: Hash key (H)

Step 4: Then the client file is encrypted using the hash value (H) as the encrypting key since the data are stored enciphered in cloud servers.

INPUT: Plain Text File, Hash Key
ALGORITHM: Advanced Encryption Standard
OUTPUT: Cipher Text File.

Step 5: The encrypted file is hashed to create a unique identifier for the data file stored in the cloud server.

INPUT: Cipher Text File
ALGORITHM: B Tree Search
OUTPUT: Hash Value (Unique ID)

Step 6: The hash Value (H) generated for encrypting the file is encrypted before uploading to the cloud server.

Then client starts the storage process by sending ID of the file to verify the uniqueness of the generated ID. If the file does not exist in the cloud servers, the client sends the file that he intends to store in the cloud, and the data decrypting key enciphered with the public keys of authorized users. The client also specifies the group of authorised users with their access rights permission.

Data Sharing:

User Request Access: The Current User initiates sharing of a file to another user. When receiving this message, the CSP searches for the read/write permissions of the recipient, and then generates a Response Access message.

Response Access: CSP sends a random key to the authorised user mail Id. On successful validation of the key CSP sends enciphered file. Then user deciphers the associated symmetric key with its own private key. Then performs a symmetric decryption algorithm to retrieve the plaintext file.

4. CLOUD DATA DEDUPLICATION TAG EQUALITY TESTING ALGORITHM

Cloud Data Deduplication Tag Equality Testing Algorithm consists of 6 phases are Encryption Key Generation, Encryption, Decryption, Tree Creation, Equality Testing and Deduplication.

Encryption Key Generation:

Step 1: The input 'pp' is obtained from the PPGen function

Step 2: The contents of the file are taken as a string input 'm'

Step 3: The output derived 'km' is used as the encryption key

Step 4: KeyGen is the function to generate the output km ← KeyGen(pp, m)

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 7, July 2019, pp. 7–15

Step 5: The hashing algorithm SHA-256 is implemented for the KeyGen function due to its high collision resistant property

Encryption:

Step 1: The input 'km' is obtained from the KeyGen function which is the XOR of the two halves of SHA-256 output

Step 2: The contents of the file are taken as a string input 'm'

Step 3: The output derived 'c' is the encrypted cipher text

Step 4: Encpp is the function to generate the output c ← Encpp (km, m)

Step 5: The encryption algorithm AES is implemented for the Encpp function which takes 128 bit key from the hashing function

Decryption:

Step 1: The input 'km' is obtained from the KeyGen function

Step 2: The encrypted cipher text 'c' is taken as the input

Step 3: The output derived 'm' is the decrypted message

Step 4: Decpp is the function to generate the output m ← Decpp(km, c)

Step 5: Reverse AES is implemented for the decryption function

Step 6: Decryption is carried out at the time of downloading a file's contents

Tree Creation:

Step 1: B Tree is used for storing the hash value (Tag)

Step 2: The tree state 'ts' is obtained as output

Step 3: TreeInit is the function to generate the ouputts ← TreeInit()

Equality Testing:

Step 1: The tags τ1 and τ2 of two cipher texts are taken as input

Step 2: EQpp is the function to generate the output {0, 1} ← EQpp(τ1, τ2)

Step 3: It outputs 1 if the tags of the cipher texts are generated from identical messages and 0 otherwise

Deduplication:

Step 1: The tags τ1 and τ2 of two cipher texts are taken as input

Step 2: The current state of the B Tree 'st' is also passed to determine duplication

Step 3: Dedupp is the function to generate the output {0, 1} ← Dedupp(st, τ1, τ2)

Step 4: It returns 1 when there is a duplicate copy of the file found in the database else returns 0

Secured Hashing Algorithm (SHA) -256:

Step 1: The input message (M) should always be a multiple of 512 bits. Let M be message of length l, append 1 followed by k zero bits, where l+1+k=448mod512. Append 64 bit equal to the length of message (M).

Step 2: The padded message is parsed into N 512-bit blocks, M(1) ,…, M(N). Since the 512bits of the input block may be expressed as sixteen 32-bit words, the first 32 bits of message block i are denoted . M0(i), the next 32 bits are M1(i) and so on up to M15(i).

Step 3: The initial hash value, H(0), shall consist of the following eight 32-bit words, in hex:

H0(0) = 6a09e667
H1(0) = bb67ae85
H2(0) = 3c6ef372
H3(0) = a54ff53a
H4(0) = 510e527f
H5(0) = 9b05688c
H6(0) = 1f83d9ab
H7(0) = 5be0cd19.

The words were obtained by taking the first thirty-two bits of the fractional parts of the square roots of the first eight prime numbers.

For i=1 to N:

1. Prepare the message schedule, W(t)
2. Initialise the eight working variables a, b, c, d, e, f, g, h with (i-1)th hash valuelike a=H0(i-1) b=H1(i-1) till h=H7(i-1).
3. for t=0 to 63

```
{
    T1= h+ sum(e) + ch(e,f,g)+k(t)+w(t)
    T2= sum(a)+maj(a,b,c)
    h=g
    g=f
    f=e
    e=d+T1
    d=c
    c=b
    b=a
    a=T1+T2
}
```

4. Compute the i(th) intermediate hash value H(i)

H0(i) = a + H0(i-1)
H1(i) = b + H1(i-1)
H2(i) = c + H2(i-1)
H3(i) = d + H3(i-1)
H4(i) = e + H4(i-1)
H5(i) = f + H5(i-1)
H6(i) = g + H6(i-1)
H7(i) = h + H7(i-1)

After processing M blocks the resultant hash value H is concatenation of H1,H2 upto H7.

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 7, July 2019, pp. 7–15

H=H1||H2||H3||H4||H5||H6||H7

Advanced Encryption Standard:

The key size used for an Advanced Encryption Standard (AES) cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the ciphertext. The number of cycles of repetition are as follows:

10 cycles of repetition for 128-bit keys.
12 cycles of repetition for 192-bit keys.
14 cycles of repetition for 256-bit keys.

Step 1: KeyExpansion - round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.

Step 2: InitialRound AddRoundKey - each byte of the state is combined with a block of the round key using bitwise XOR.

Step 3: Rounds SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table. ShiftRows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps. MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column. AddRoundKey.

Step 4: Final Round (no MixColumns)

1. SubBytes
2. ShiftRows
3. AddRoundKey.

B Trees:

B-Tree is a self-balancing search tree. The main idea of using B-Trees is to reduce the access time. Height of B-Trees is kept low by putting maximum possible keys in a B-Tree node. The B-Trees are designed in such a way that more frequently access elements are kept closer to the root node than the less frequently accessed elements.

Properties of B-Tree:

All leaves are at same level. A B-Tree is defined by the term minimum degree‘t’. Every node except root must contain at least t-1 keys. Root may contain minimum 1 key. All nodes (including root) may contain at most 2t – 1 keys. Number of children of a node is equal to the number of keys in it plus 1. All keys of a node are sorted in increasing order. The child between two keys k1 andk2 contains all keys in range from k1 and k2. B-Tree grows and shrinks from root which is unlike Binary Search Tree. Binary Search Trees grow downward and also shrink from downward. Like other balanced Binary Search Trees, time complexity to search, insert and delete is O(logn).

5. PERFORMANCE ANALYSIS

CloudSim is a toolkit for the modelling and simulation of Cloud computing environments which provides system and behavioural modelling of the Cloud computing components. Initially the input is a set of data files to be stored in the cloud server by different users. Parameters such as bandwidth, file size, uploading time are considered. The output is a set of unique files stored in the cloud and shared between common users. The algorithm used initially is to avoid redundant data storage of identical file by a single user so that storage space and maintenance cost of cloud service provider is reduced. In previous existing methods, Cloud Service Provider (CSP) was responsible for providing security to the data stored on the cloud. So CSP had complete access to the uploaded data which affects the confidentiality and privacy of the user. Initially service providers offered vast storage capacity to their client which encouraged clients to upload enormous data to the cloud. The system increased the cost of maintenance for the CSP and also increases network bandwidth consumption associated to transmitting the same contents several times.

The proposed solution is effective in achieving both security and deduplication aspects of cloud storage. It effectively utilises the bandwidth and storage space allocated for the user. Besides, our solution is also shown to be resistant to unauthorized access to data and to any data disclosure during sharing process, providing two levels of access control verification. The algorithm proposed is effective in achieving cross-user deduplication, which allows more storage savings. Moreover, the system checks for the duplicate file in cloud server before actually uploading the file to cloud server which ignores the storage of redundant data in cloud servers and also reduces network bandwidth consumption associated to transmitting the same contents several times.

Time Complexity:

‘n’ denotes the number of files in the cloud B Tree based µR-MLE2 takes O(n * log n) for initialization Pair denotes the insert operation for the hash table Hash table has worst case complexity O(n) when collision resistance is low and whenever rehash is necessary

TABLE 5.1 TIME COMPLEXITY

Scheme	De-duplication Time Complexity
R-MLE2	O(n)
µR-MLE2 (Dynamic)	O(n)
B Tree based µR-MLE2	O(log n)

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 7, July 2019, pp. 7–15

The given graph in Fig.5.1 shows the performance analysis of redundant data in cloud servers and also reduces network bandwidth consumption associated to transmitting the same contents several times.

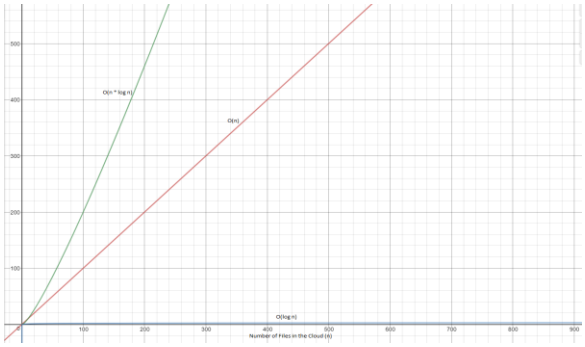

Figure 5.1 Performance Analysis

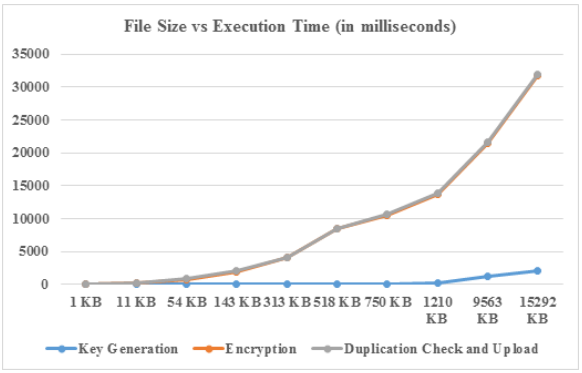

Figure 5.2 Execution Time Comparison

The given graph in Fig.5.2 shows the comparison of Execution Time for Key Generation, Encryption using AES and Duplication Check after uploading a file. It can be observed from the graph that the time for Encryption and Duplication check differs by very small values. The comparison is repeated for files of different sizes and tabulated to obtain the input for the graph.

6. CONCLUSION AND FUTURE WORK

Previously existing methods for finding deduplication in cloud storage have been successful in achieving either data integrity or storage efficiency but not both. Moreover, the existing methods do not provide support for cross user deduplication checking. The client side deduplication methods have failed to protect the privacy of the user.

The proposed method has been successful in effectively utilising the storage space allocated to the users by CSP. Our method also supports many users to share a common memory space which helps to save more space than the existing methods. It also prevents the confidentiality of user by checking their proof of ownership. It does not allow unauthorised users to access other files stored in cloud by random key generation method thus providing more security to the data files.

The work can be incorporated in real time environment for improving the efficiency and flexibility in auditing the files allowing the user to authenticate their file content stored in the cloud server without downloading the entire file. Another way of providing security is the obfuscation method which can be implemented in order to improve the security of the user file. Integration of obfuscation technique with the encryption technique will improve the confidentiality. The system can protect the data in the cloud storage from insiders as well as outsiders attack. Provide a brief conclusion here based on the paper.

REFERENCES

[1] Jingwei Li, Jin Li, Dongqing Xie and Zhang Cai (2016), 'Secure auditing and deduplicating Data in Cloud', *IEEE Transaction on Computers*, Vol. 65, No. 8, pp. 2386-2396.

[2] Jia Yu, KuiRen, Cong Wang and Vijay Varadharajan (2015), 'Enabling Cloud storage auditing with key-exposure resistance', *IEEE Transaction on Information Forensics and Security*, Vol. 10, No. 6, pp. 1167-1179.

[3] Baojiang Cui, Zheli Liu and Lingyu Wang (2015), 'Key-aggregate searchable encryption (KASE) for group data sharing via cloud storage', *IEEE Transaction on Computers*, Vol. 6, No. 1, pp. 1-13.

[4] Kan Yang, XiaohuaJia, Expressive (2014), 'Efficient and revocable data access control for multi-authority cloud storage', *IEEE Transactions in Parallel and Distributed Systems*, Vol. 25, No. 7, pp. 1735-1745.

[5] Li J., Chen X., Li M., Lee P. and Lou W. (2014), 'Secure deduplication with efficient and reliable convergent key management', *IEEE Transaction on Parallel Distribution System*, Vol. 25, No. 6, pp. 1615–1625.

[6] Poornashree B.R., Srividhya S. (2016), 'A survey on provable data possession in cloud computing systems', *International Journal Of Engineering Research & Technology*, Vol. 5, No. 7, pp. 271-292.

[7] Deepika Singh, Preetika Singh (2014), 'New challenges for security against deduplication in cloud computing', *International Journal of Advance Research in Computer Science and Management Studies*, Vol. 2, No. 5, pp. 653-667.

[8] Kai He, Chuanhe Huang, Hao Zhou, Jiaoli Shi, Xiaomao Wang, Feng Dan (2015), 'Public auditing for encrypted data with client-side deduplication in cloud storage', *Wuhan University Journal of Natural Science*, Vol. 20, No. 4, pp. 291–298.

[9] Joseph K. Liu, Kaitai Liang, Willy Susilo, Jianghua Liu, and Yang Xiang (2016), 'Two-factor data security protection mechanism for cloud storage system', *IEEE Transaction on Computers*, Vol. 65, No. 6, pp. 256-270.

[10] Wenjing Lou, KuiRen, Qian Wang, Sherman S.M. Chow , Cong Wang (2013), 'Privacy-preserving public auditing for secure cloud storage', *IEEE Transactions on Computer*, Vol. 62, No. 2, pp. 378-395.

[11] Hong Liu, Huanshen Ning, Qingxu Xiong, Luarence T.Yang (2015), 'Shared authority based privacy-preserving authentication protocol in cloud computing', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 26, No. 1, pp. 768-781.

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 7, July 2019, pp. 7–15

[12] Jia Yu, KuiRen, Cong Wang (2016), 'Enabling cloud storage auditing with verifiable outsourcing of key updates', *IEEE Transactions on Information Forensics and Security*, Vol. 11, No. 6, pp. 690-708.

[13] Bo Mao, Hong Jiang, Suzhen Wu and Lei Tian (2016), 'Leveraging data deduplication to improve the performance of primary storage systems in the cloud', *IEEE Transactions On Computers*, Vol. 65, No. 6, pp. 278-292.

[14] Aparna B., Kumar K. S. M. V. (2015), 'Privacy preserving and authorized data deduplication in public cloud framework', *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 5, No. 10, pp. 412-434.

[15] JunbeomHur, Dongyoung Koo, Youngjoo Shin and Kyungtae Kang (2016), 'Secure data deduplication with dynamic ownership management in cloud storage', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 99, pp. 571-586.

[16] Zheng Yan, Wenxiu Ding, Xixun Yu, Haiqi Zhu, Robert H. Deng (2016), 'De-duplication on encrypted big data in cloud', *IEEE Transactions on Big Data*, Vol. 2, No. 2, pp. 138-150.

[17] Arokiam L., Manikandan S. (2014), 'Efficient cloud storage confidentiality to ensure data security', *Computer Communication and Informatics (ICCCI), 2014 International Conference*, Vol. 4, No. 3, pp. 1126-1142.

[18] Jia Yu, Kui Ren, Cong Wang, Vijay Varadharajan (2015), 'Enabling Cloud storage auditing with key-exposure resistance', *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 6, pp. 1167-1179.

[19] Jianghong Wei, Wenfen Liu, Xuexian Hu (2015), 'Secure data sharing in cloud computing using revocable-storage identity-based encryption', *IEEE Transaction on Cloud Computing*, Vol. 14, No. 8, pp. 1-13.

[20] Bellare M., Keelveedhi S. and Ristenpart T. (2013), 'Message-locked encryption and secure deduplication', *Proceedings Advanced. Cryptography*, Vol. 5, No. 6, pp. 296–312.

[21] Liu, Joseph (2016), 'Two-factor data security protection mechanism for cloud storage system', *IEEE Transactions on Computers*, Vol. 65, No. 6, pp. 1992-2004.

[22] Cao, Ning (2014), 'Privacy-preserving multi-keyword ranked search over encrypted cloud data', *IEEE Transactions on parallel and distributed systems*, Vol. 25, No. 1, pp. 222-233.

[23] Hermine Hovhannisyan, Kejie Lu, Rongwei Yang, Wen Qi, Jianping Wang, Mi Wen. (2016), 'A novel deduplication-based covert channel in cloud storage service', *IEEE Global Communications Conference (GLOBECOM)*, Vol. 47, No. 1, pp. 1-6.

[24] Zheng Yan, Wenxiu Ding, Xixun Yu, Haiqi Zhu, Robert H. Deng (2016), 'Deduplication on encrypted big data in cloud', *IEEE Transaction on Big Data*, Vol. 2, No. 2, pp. 138-150.

[25] Prabhu Kavin B, Ganapathy S, "Data Mining Techniques for Providing Network Security through Intrusion Detection Systems: A Survey", *International Journal of Advances in Computer and Electronics Engineering*, Vol. 2, No. 10, pp.1-6, October 2017.

[26] Pratham Harshit Rajmahanty, S. Ganapathy, "Role of Decision Trees in Intrusion Detection Systems: A Survey", *International Journal of Advances in Computer and Electronics Engineering*, Vol. 2, No. 4, April 2017, pp. 09 - 13.

[27] Riyaz B, Ganapathy S, "Intelligent Soft Computing Techniques for Providing Network Security: A Survey", *International Journal of Advances in Computer and Electronics Engineering*, Vol. 2, No. 10, pp. 7-14, October 2017.

[28] Kannan Devibala, Saminathan Balamurali, Ayyanar Ayyasamy, Maruthavanan Archana, "Flow Based Mitigation Model for Sinkhole Attack in Wireless Sensor Networks using Time-Variant Snapshot", *International Journal of Advances in Computer and Electronics Engineering*, Vol. 2, No. 5, May 2017, pp. 14 - 21.

[29] Elangovan Gurumoorthi, Ayyanar Ayyasamy, Maruthavanan Archana, Jayabalan Vijaya Barathy, "Performance Enhancement for QoS in VoIP Applications over MANET", *International Journal of Advances in Computer and Electronics Engineering*, Vol. 2, No. 5, May 2017, pp. 47 - 54.

[30] Poomagal C. T, Sathish Kumar G. A, "Modular Multiplication algorithm in Cryptographic Processor: A Review and Future directions", *International Journal of Advances in Computer and Electronics Engineering*, Vol. 2, No. 2, February 2017, pp. 28 - 33.

Authors Biography

S. Muthurajkumar received M.E. degree in Computer Science and Engineering from Anna University, Chennai where he has completed Ph.D. and he is working as an Assistant Professor in the Department of Computer Technology, MIT Campus, Anna University, Chennai. He has published more than 8 articles in journals and conferences. His area of interest is Cloud Networks security, Cloud Computing and Data Mining.

Cite this paper:

S. Muthurajkumar, "Secured Cloud Data Deduplication Using Tag Equality Testing", International Journal of Advances in Computer and Electronics Engineering, Vol. 7, No. 7, pp. 7-15, July 2019.