Secure Data Deduplication System with Efficient and Reliable Multi-Key Management in Cloud Storage

R. Vignesh[1*], J. Preethi[2]

[1] Department of Computer Science, Muthayammal Engineering College, India
[2] Department of Computer Science, Anna University Regional Campus, India
vigneshapcse@gmail.com, preethi17j@yahoo.com

Abstract

The revolutionary growth in the processing and storage mechanisms over the Internet has given the enhancement to inexpensive and strong computing properties. Cloud computing is a rising technology, which offers the data storage facility also application accessing facility in online environment. This system stands countless opportunities also challenges. In that, security of data and the increasing similar data in cloud (duplication) are very important issues to be addressed. So, Deduplication method is developed to reduce the similar data that is present in the storage system. In this paper, a novel technique is proposed to remove the duplicate data from cloud also help to save the bandwidth access and storage space. The experimental results demonstrate that the proposed system provide the more security for data in cloud storage and also overcomes the main drawbacks of the existing systems. In one-server storage and distributed storage systems, we have created a solution which provides data security and space efficacy. The chunk data generates encryption keys consistently; the same chunk is therefore always encrypted with the same chip text. In addition, the keys cannot be derived from the chunk data encrypted. Because the information to be accessed and decrypted by each user is encrypted by using a key known to the user alone, even a complete system breach cannot expose which chunks are utilised by which users.

Keywords: Cloud storage, Data deduplication, Multi key management, Data security

1 Introduction

With the support of distributed and virtual machine technology, the cloud computing system provides the greatest services in efficient manner. At anywhere, the users can access the unlimited storage space on any time in cloud system. Cloud service supplier is responsive to enlarge the data storage area and combining data deduplication method into cloud system, while data deduplication eliminates redundant data that is present in cloud environment. Data privacy conserving also a vital topic to be deliberated and to recognize this data privacy. So as to support deduplication, new kind of deduplication method is utilized for encrypting the data before outsourcing in a cloud environment [1].

Because of the storage and networking environment, the majority of data is now kept on the cloud. Duplicate data on the disc is not recognized by the data disc. Duplicate data can eat up disc storage capacity. More duplicate data has an impact on disc performance, space, speed, and other performance parameters.

Cloud system offers its clients with great computing and storage services that benefits impressively. A survey done recently by the ‘New York Times’, they had study on leased ‘Amazon’s cloud computing’ and they establish that cloud computing services are utilized beyond 1.10 crore stories in one day to translate electronic document for customers’ to browse, and the whole rate was around $240 [1].

On the other hand, when there is a large amount of data, duplication of data may increase the user’s memory, cost, and time for data processing. By locating the exact data and storing it in a location where the leftover data is eliminated, the deduplication technique can alleviate all of the above-mentioned concerns while maintaining high security and confidentiality.

As cloud computing is becoming popular gradually as total number of consumers is growing, so security issues of the user data on cloud environment are emerging users may move its sensitive data during a cloud environment, so to supply a correct security control to guard integrity and privacy of user data remain main concern. Cloud computing technique is computing services through the internet delivering different services on demand. For users, there are different types of services that are provided by cloud computing for example SaaS, PaaS, IaaS and so on. Today’s cloud service providers offer client different type of services such large storage space and parallel computing of resources at very cheap price [1]. In the cloud system, data deduplication is a novel technique that works on quickly growing number of digital data in cloud data storage, these techniques used for identifying redundant data. The outcome of unique single copy is kept and it can be served to all or any of the authorized clients. The general system of data deduplication is illustrated in Figure 1.

The below Figure 1 illustrates the general process of deduplication system. For duplicate checking, user requests for duplicate check of selected file to administrator. A token which are sends towards administrator is in encrypted format, this encryption done with client’s password as encryption key. At administrator side, administrator decrypted this token by using client’s password as decryption key. Once decryption is completed, the token administrator checks this token in the database. At time of file uploading, file is get encrypted with token as encrypted keys and then uploaded this encrypted file

*Corresponding Author: R. Vignesh; E-mail: vigneshapcse@gmail.com

DOI: 10.53106/160792642022072304016

on cloud. So, the files are stored in encrypted format in public cloud.

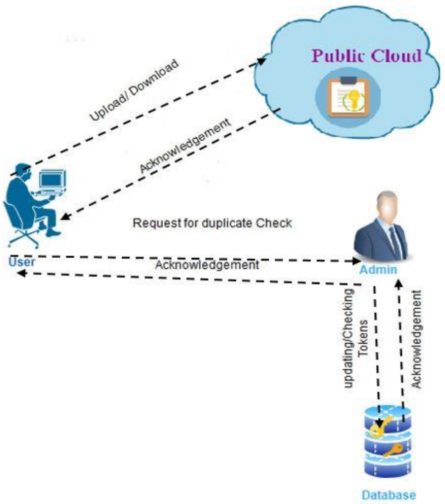

Figure 1. General data deduplication system architecture

Data deduplication have significant role to reduce the storage cost in cloud storage management system. Nowadays, many secure deduplication encryption systems have been implemented to secure the privacy of clients' information. Available systems cannot maintain a cost-effective cloud service to achieve the data system with a best duplication structure [2]. At this point, it will create a main challenge for the cloud users to be acquiring the deduplication scheme. This inattentive behavior has even lead to serious scandals. In order to attain the authorized deduplication, satisfy dynamic privilege updating and revoking and avoid data leakage, the proposed system implements a novel secure role encryption and decryption method with secured data deduplication in cloud environment. The experiment proves that the proposed system can efficiently moderate the response time and succeed the storage load of files more composed.

The rest of this paper is systematized as follows: In section 2, review of the previous related work done for data deduplication. In section 3, discussed Data Deduplication Design Criteria, Section 4 define proposed scheme of deduplication system, Section 5 described Implementation details, and section 6 illustrates result & discussion and various evaluation parameters.

1.1 Contributions of the Research Work

A reduction in data that must be moved, stored and managed in cloud storage systems becomes essential. Increasing the amount of files wastes hardware resources and complicates data centres, which further lowers cloud storage speed. Due to the huge quantity of duplicate files stored on cloud systems by various users, we recognised a research need. For this reason, this article offers a novel data index approach, which incorporates data de-duplication for storage optimization on cloud systems, in order to reduce the burden imposed by duplicate files.

- Data de-duplication with secure cloud storage system is proposed in this research paper. The key risks which we contribute on this work are data deduplication and cryptography based security policy.
- They can be static (for long-term use) or ephemeral (for short-term use) (designed to be used only for a single session or transaction). According to their intended usage, the crypto-period (i.e. lifetime) of static keys might range from days to weeks to months to years. It is vital to replace keys as necessary since the more a key is used, the more vulnerable it is to attack, and the more sensitive data is at danger (this process is called updating or cycling).
- In this proposed work, a novel key management mechanism is achieved to decrypt and access the de-duplicated data chunks.

There is a huge concern with data breaches, account hacking, malware injection, and the use of the insurance API to compromise user security. When multiple passwords are used with stronger encryption standards, the foregoing vulnerabilities can be mitigated to a greater extent.

The deduplication of data is crucial since this decreases your storage space demands substantially, saves you money and avoids the waste of bandwidth while moving the data from/to remote storage sites. In certain situations, data deduction can reduce storage needs by up to 95% but factors such as the type of data that you seek to deduplicate affect the deduction ratio you specify. Even if your storage requirements are decreased by less than 95%, the deduction of data can still lead to considerable savings and improvement in availability of your bandwidth.

2 Related Work

The term deduplication may be a recent developing survey area in that several researchers have supported from the previous couple of eras. Many researchers have suggested various methods and technologies to assist and widen data deduplication system. Literature survey section will elaborate the flow of research and techniques followed by other researchers in the existing system.

Halevi *et al.* [3] used an idea which is Hashing function SHA256 implemented for creating hash value for building Merkle tree for Proof of Ownership. The author proposed the Proof of ownership that deals a client proficiently influence a server which consume licensed one. During this system they kept just some kind of information about file instead of complete file. This is often done by using Merkle trees with certain encodings to examine their security level.

Hou *et al.* [4] find the approach to recognize cloud storage auditing associate with deduplication management for various security levels affording to data popularity. To obtain the optimal solution, this author elucidates the one key issue to ensure that the third party auditor who still can audit the integrity of cloud data after data popularity modifications. At that point, they suggested the main cloud storage auditing system with deduplication supporting various safety levels. This methodology implements convergent threshold cryptosystem to offer semantic safety for detested data also cipher text deduplication for common data. Clients are don't want to create new authenticators by himself as well as be online while the data popularity modifications.

Zhou *et al.* [5] developed EDedup system, which utilize a similarity aware encrypted deduplication method to accomplish the server based MLE at segment level for scale back computation ingestion. To prevent privacy leakage, the

EDedup system integrates source based related segment recognition and target based duplicate chunk checking. This system also creates arbitrary message derived file keys to manage the metadata also understands access control with revocation through accepting proxy based CP-ABE (Ciphertext Policy Attribute-Based Encryption) to encrypt file keys.

Zheng *et al.* [6] designed a deduplication system on cloud data to support the certificate less proxy re-encryption. It contains Certificate Less Proxy Re-Encryption (CL-PRE) and proof of ownership supported certificate less signature (PoW-CLS). They use certificate less cryptography to unravel the matter of key escrow and avoid things where a key generation center (KGC) impersonates a user to decrypt the cipher text. The certificate less cryptography method was used by this authors to undo the substance of key guarantee and avoid things where a KGC (Key Generation Center) imitates a client to decrypt the cipher text.

Xia *et al.*, [7] utilized ‘P-Dedupe’ with parallelized data deduplication method, which speed up deduplication method through dividing the deduplication method into 4 phases such as chunking, fingerprinting, indexing, and writing. Pipelining these 4 phases with chunks and files then parallelizing ‘CDC’ (Content Defined Chunking) also secure hash based fingerprinting phases to more ease the calculation block. This system will first split the information stream into numerous segments that is called ‘Map’, in this point every segment running ‘CDC’ in parallel with a liberated thread then it re-chunk as well as combine the boundaries of those segments that is called as ‘Reduce’ to validate the chunking efficiency of parallelized ‘CDC’ [8].

Yuan *et al.* [9] proposed an accessible with safe system for data deduplication through active client controlling, which modifies active cluster clients during a safe approach and confines the illegal clients of the profound facts influenced by using effective clients.

Shanshan Li [10] propose a safe also well-organized ‘CSED’ (Client Side Encrypted Data Deduplication) system sustained PoW and incorporate it into ‘CSED’ to repel illicit data circulation attacks. A fanatical key server is presented in producing ‘MLE’ (Message Locked Encryption) keys to repel ‘brute force attacks’ in CSED. Furthermore, a hierarchical storage planning is utilized to improve the ‘I/O’ effectiveness on the cloud server. They acquire the key server to aid the users in making the ‘MLE’ keys also adopt the rate limiting approach to break brute force attacks

3 Terminologies in Deduplication System

3.1 Data Deduplication Design Criteria

Data deduplication system developed to minimizing the cost of cloud users through reducing the storage, hardware and bandwidth cost. It diminishes the backup costs because buying/preserving least storage space will return us with the quicker outcomes.

Data deduplication is typically achieved by two main entities such as a client that indicates the data owner and Cloud Service Provider (CSP) that gives storage area for the outsourced data in cloud storage system. This may vary conferring to safety aims then design criteria [11]. The following Figure 2 categorize these systems as ‘Data Granularity’ (DG), ‘Deduplication Location’ (DL) and ‘Duplicate Check Boundaries’ (DCB).

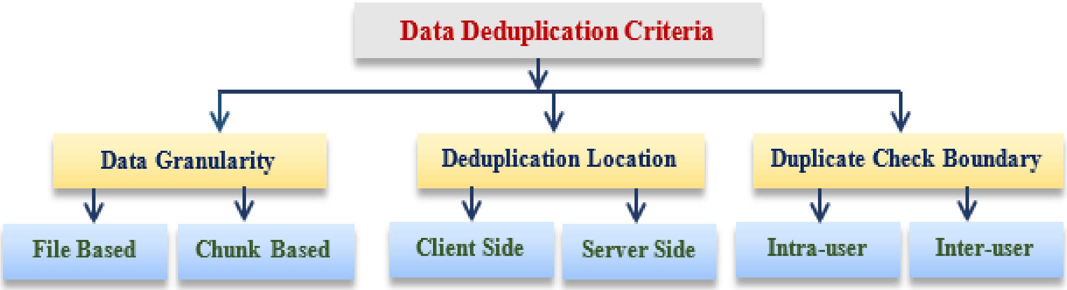

Figure 2. Classification of deduplication

3.1.1 Data Granularity

There are different approaches to segment information into essential units for the disposal of duplicates. In record level deduplication that is one of the clearest methodologies, a document is preserved as the essential unit of deduplication. Hash signatures are registered from the substance of the whole record and afterward used to discover duplicate documents in the cloud by looking at their hash signatures. Block level deduplication is alternative usual methodology, where a record is separated into different blocks and duplicates are checked for every segment. Then again, variable size utilizes Rabin fingerprinting method to produce hash signatures. It considers a record as a flow of bytes and searches for pre-determined delimiters in the byte stream [12]. All information chunk isolated by the delimiters is then rewarded as the fundamental unit of deduplication. Block level deduplication ordinarily has a higher deduplication proportion than document level deduplication yet acquires extensive overheads because of the enormous measure of metadata created for every block. Conversely, document level deduplication offers moderately low overhead in keeping up the metadata.

3.1.2 Deduplication Location

Data deduplication can be achieved on the server side, user side, and gateway side. In server-side deduplication, users who need to redistribute their information would consistently transfer it to a cloud storage server over the system. When the

CSP gets the information, it achieves deduplication by determining duplicates inside the saved information and dispensing with them.

In the customer side deduplication method, a customer initially calculates a hash value for the information to be transferred and sends it to the CSP before transferring the real content of the information. Compared with server-side deduplication, this methodology has a benefit in term of decreasing system transfer speed utilization. In some cloud computing situations, especially in hybrid models, a particular sort of machine server, alluded to as a storage gateway, and is normally conveyed on a client's reason private system. A capacity gateway gives a client access to an open cloud storage administration. Gateway side deduplication uses storage gateways to execute deduplication [13]. On getting redistributing information from a customer, the storage gateway performs deduplication alongside a CSP in the interest of the customer.

3.1.3 Duplicate Check Boundary

Deduplication strategies can be sorted into intra-client and inter-client methods in duplicate check boundary. When discovering duplicates in transferred information, intra-client deduplication exclusively considers information in the cloud storage that has been redistributed by similar information proprietor. Then again, between client's deduplication considers all stored information over numerous information proprietors in the cloud storage. Both intra-client and inter-client methods viably dispose of excess information in cloud storage. Since intra-client deduplication finds and evacuates duplicates of similar clients' information, this strategy is increasingly valuable for reinforcement frameworks, where copies are probably going to be found among rehashed reinforcements made by a solitary client. Between client's deduplication is compelling for capacity frameworks inside an association where an enormous number of copies exists among information possessed by various clients.

3.2 Encryption in Deduplication

Deduplication works by removing the redundant files, or data. In whole data file hashing, a complete file is first sent to hashing function. The preferred hashing functions are MD5 and SHA-1. Its result is a cryptographic hash which forms the basis for the identification of complete duplicate file. The complete file hashing has fast execution with low calculation and low metadata expense. But it prevents from matching two files that just differ by a byte of information divides two types, fixed-size chunking and variable-length chunking, can be used for dividing a file. In fixed size chunking, a file is divided into an amount of static or set sized pieces called "chunks," whereas in variable size chunking, data is broken down into chunks of differing length [14].

Deduplication works by expelling the repetitive records, information and blocks. In entire information document hashing, a total record is first sent to hashing function. The favored hashing functions are MD5 and SHA-1. Its outcome is a cryptographic hash which frames the reason for the determination of complete duplicate record. The total document hashing has quick execution with low figuring and low metadata cost. In any case, it keeps from coordinating two records that simply contrast by a byte of data isolates two sorts; fixed-size and variable-length chunking these can be utilized for partitioning a document. In fixed size chunking method, a record is partitioned into a measure of static or set estimated parts known as "chunks" while in variable size chunking, information is separated into pieces of contrasting length.

3.3 Chunking-based Data Deduplication

In this technique, each record is initially partitioned into an arrangement of opposing cushions, called pieces/ chunks. Each bit is an adjacent form of bytes from the document that is refined freely.

3.4 Hashing Mechanism for Deduplication

Hash impacts are probable issue with deduplication. The hash sum for every single chunk is made utilizing a method. Some much of the time utilized hashing methods are Rabin's calculation, Alder-32, Secure Hash Algorithm-1 (SHA-1), and Message digest (MD-5). Advanced Encryption Standard (AES) is not really utilized for this ability. The encoded information is out-sourced to the cloud condition. Rabin's fingerprinting structure produces fingerprints utilizing polynomials. It produces cryptographic hash of each stop in an information document [15].

4 Proposed Scheme

In the proposed deduplication system which are used are listed below.

4.1 Encryption

In the proposed system, the File is encrypted with its respecting Hash Key used for Convergent Encryption and been encrypted using AES algorithm as a whole. The encryption techniques used in the proposed system consist of below methods.

4.1.1 Hash Key

Hashing method is one among the chief normally utilized encryption techniques. A hash might be a unique function that performs single direction encryption. In view of document content hash key is produced. The tag, hash key produced is extraordinary for each record. Key generation method maps an information copy 'N' to a convergent/united key 'R' and it depends on the security boundary. The reason for creating hash key is to scramble the square of information with remarkable hash key.

4.1.2 Convergent Encryption

Encoding information doesn't approve in the de-duplication, if two indistinguishable records scrambled with various keys will yield diverse encoded information blocks which can never again be shared. Along these lines, to beat this, another method is presented which is known as 'Convergent Encryption' (CE).

CE gives information secrecy in deduplication. Besides, the client infers a tag for the data duplicate, such the tag will be used to distinguish duplicates. Here, we accept that the label rightness property holds, i.e., in the events that two information duplicates are a proportional, at that point their labels are a comparable. To distinguish copies, the client initially sends the tag to the server side to check whether the indistinguishable duplicate has been now stored. Note that both the convergent key and the tag are autonomously determined, and the tag can't be used to derive the convergent key and bargain information privacy. Both the scrambled information duplicate and its comparing label will be stored on the server side [16]. Officially, a CE strategy are regularly characterized with four primitive functions:

- KeyGence(M) is the key generation algorithm that maps a knowledge copy M to a convergent key K.
- Encryption is the symmetric encryption algorithm that takes both the convergent key K and the data copy M as inputs and then outputs a cipher text.
- Decryption is the decryption algorithm that takes both the ciphertext C and the convergent key K as inputs and then outputs the original data copy M.
- TagGence is the tag generation algorithm that maps the original data copy M and output as tag T. We allow TagGence to generate a tag from the corresponding ciphertext as in, by using

The CE work is to originate the encryption key from the hash of content of a plaintext. If two clients with two matching plaintexts will attain two equal cipher texts, then the encryption key is similar. Each file now features a distinct encryption key, certain technique is required for every data owner/client to record and retrieve the keys from connected information blocks. As the encryption key is produced on the plaintext so, there is no need for establish an agreement for key generation. Hence, convergent encryption is very good for deduplication in cloud environment.

4.1.3 Message Digest - 5 (MD5) Technique

In most of the system, the MD5 technique is a utilized to generate the hash value. MD is employed to make sure the reliability of a message communicated over an unconfident network. The message is trained a cryptographic hash function. It generates a compressed form of the message known as 'Digest'. Steps involved in MD5 is as follows,

i. Divide the input data file into the blocks
ii. At the end of last block, certain bits are injected
iii. Add the extra bits. if last block is < other block sizes,
iv. Routines above rounds to analysis the blocks
v. The MD5 digest is generated, after completing above steps.

4.1.4 SHA Family (Secure Hash Algorithm)

The SHA (Secure Hash Algorithm) Family assigns a group of six distinctive hash functions. It was intended to utilize for secure hashing in the 'US Digital Signature Standard'. The suggested framework comprises of SHA-1, SHA-256 and SHA-384 techniques. SHA-1 generates a MD of 160 bits, SHA-256 generates a MD of 256 bits, and SHA-384 generate a MD of 384 bits individually.

4.1.5 Advanced Encryption Standard (AES) Algorithm

AES is one of the comprehensively used 'symmetric encryption' method. AES utilizes the Rijndael block cipher. AES s encrypts a 128-bit fixed size block without a moment's delay. To encode 128-bit size block AES utilizes variable key sizes like 128 bit, 192 bit and 256 bit to accomplish diverse degree of security. AES takes a shot at various encryption rounds dependent upon key size like 10 round encryption for 128-bit key size, 12 rounds encryption for 192-bit key bits and 14 round encryption for 256-bit key size. Each round has 4 diverse handling ventures aside from the last round, which has 3 stages. These 4 stages comprise of byte replacement, move line activity, blend section activity and include round key activity. For encryption these 4 stages are utilized in each round and for decoding the backwards of these means are utilized these are converse replacement byte, opposite move line, reverse blend segment and reverse include round key. This method has following stages,

- The Byte Substitution Transformation
- The Shift Row Transformation
- The Mix Column Transformation
- Add Round Key

4.1.6 Rivest-Shamir-Adleman (RSA) Algorithm

RSA is outstanding amongst other known open key cryptosystems for key trade or encryption of blocks of information. It utilizes a variable size encryption block and a variable size key. It is an unbalanced (open key) cryptosystem dependent on number hypothesis that is a block cipher framework. It utilizes two prime numbers to create general public and private keys size is 1024 to 4096 bits. These two distinct keys are utilized for encryption and unscrambling reason. Sender encodes the message utilizing Receiver open key and when the message gets transmit to recipient, at that point beneficiary can decode it by utilizing his own private key [17]. RSA activities can be decayed in three wide advances; key generation, encryption and decryption.

Key generation

- Produce two large prime numbers a, b ($a \neq b$)
- Compute n = a x b
- Calculate $\phi(n)=(a-1) \times (b-1)$
- Choose random integer 'r', $1 < r < \Phi(n)$ such that gcd $(r, \Phi(n)) = 1$
- Compute exponent secret 'd',
- Calculate exponent secret 'd', $1 < d < \varphi$ (rd=1, mod $\Phi(n)$ =1)
- The private key is (d, n) and the public key (r, n) and the 'd', 'a', 'b' and 'φ' values are kept secret.

Encryption

- Determine the public key.
- The plaintext represented as a message positive as an integer positive.
- Cipher text is calculated.
- Cipher text is send to the receiver.

Decryption

- Use the private key (n,d) to compute plaintext: M = Cd mod (n).
- Extract the plaintext from the input message.

There are three main functional modules are described in this section such as Encryption, Decryption algorithms and key generation system. These are necessary background for this research to improve our proposed deduplication system in cloud storage management. The proposed system achieves the high deduplication output and good scalability by utilizing the above mentioned algorithms in effective way. The implementation details are defined in next section [18].

5 Implementation Details

5.1 Existing System

The Document uploaded to the server is been stored as a whole complete document and copy of the document is stored in the alternative server in case of loss of data in one server the other can be referred. The Document are being converted to hash key (Using Convergent Encryption) and been checked for duplication. In existing system, they are completely encrypted the file using some standard encryption methodology and stored the encrypted the file as whole without any partition. So if a loop hole may completely undergo the loosing of the information. The encryption system produces the same key and same cipher text for whole files [19-20].

Disadvantages

- Data confidentiality isn't achieved.
- In this case if one character or a space is being added to the document unknowingly still the generated hash key is totally different compared to the original hash key so the duplication cannot be identified.
- Once user lost the key, there was not possible to recover the content of the file. It's simple by attackers, then the user data are going to be leaked.
- The existing encryption system rule doesn't maintain the key management theme.

5.2 Proposed System

In existing work researchers have given good contribution in making deduplication system by utilizing standard encryption methodology and trusted execution environment to provide secure key management.

According to the recognized necessities in positions of security possessions plus system overhead in the earlier work, we examine advanced secure deduplication solutions that can be include the below lines:

1. The input File is encrypted with its respecting Hash Key used for Convergent Encryption and been encrypted using AES algorithm as a whole.
2. The File is also split into four and encrypted using their own hash function keys.
3. The files are hashed using different hash Function because if the hacker wants as found the hashing function of a particular split and try opens the remaining splits and joins to get the information of the file.
4. In this Proposed system, it is not possible as four splits in the four different server uses different hash function and keys and encryption by the keys also vary so the security is being very much advanced secure.
5. The Duplication is checked as chunks in all 5 servers with respect of its hashing techniques.

5.3 Proposed Deduplication System Model

5.3.1 Login

Figure 3 shows a user Login process; user first login communication is done with key based process. It is a type of authentication that may be utilized as a substitute to password authentication. As a replacement for requiring a user's password, it is probable to authorize the client's uniqueness by asymmetric cryptography method, with public and private keys. After successful completion of this process client can able to get verification code from code generating database then client verify that code authentication from code verification database. Then, it decrypts and store the data into file server. When it comes to this process, it is always exchange of data between client and server or passing of data through the network. And such passing of data can be dangerous if the data to be exchanged is not encrypted. So, here we used the robust method to secure the data. In this section, AES and RSA algorithms are utilized to generate the secured keys for login process.

When a user attempts to log in to the system for the first time, a common password is required. This password is logged as an input attempt along with the IP address as soon as the password is verified. If the user immediately enters the passphrase, a passphrase is required. The machine will check the IP address only if it matches the previous one, and only if you do this authentication, it will check the password entered by the user. Users grant access to the cloud path.

The validation was carried out when the user inputs the password, which is encrypted and the key was produced using both encryption standards. With the key saved in the database the generated key is checked. Before the key is sent to the API the original password is disguised fully by encryption standards.

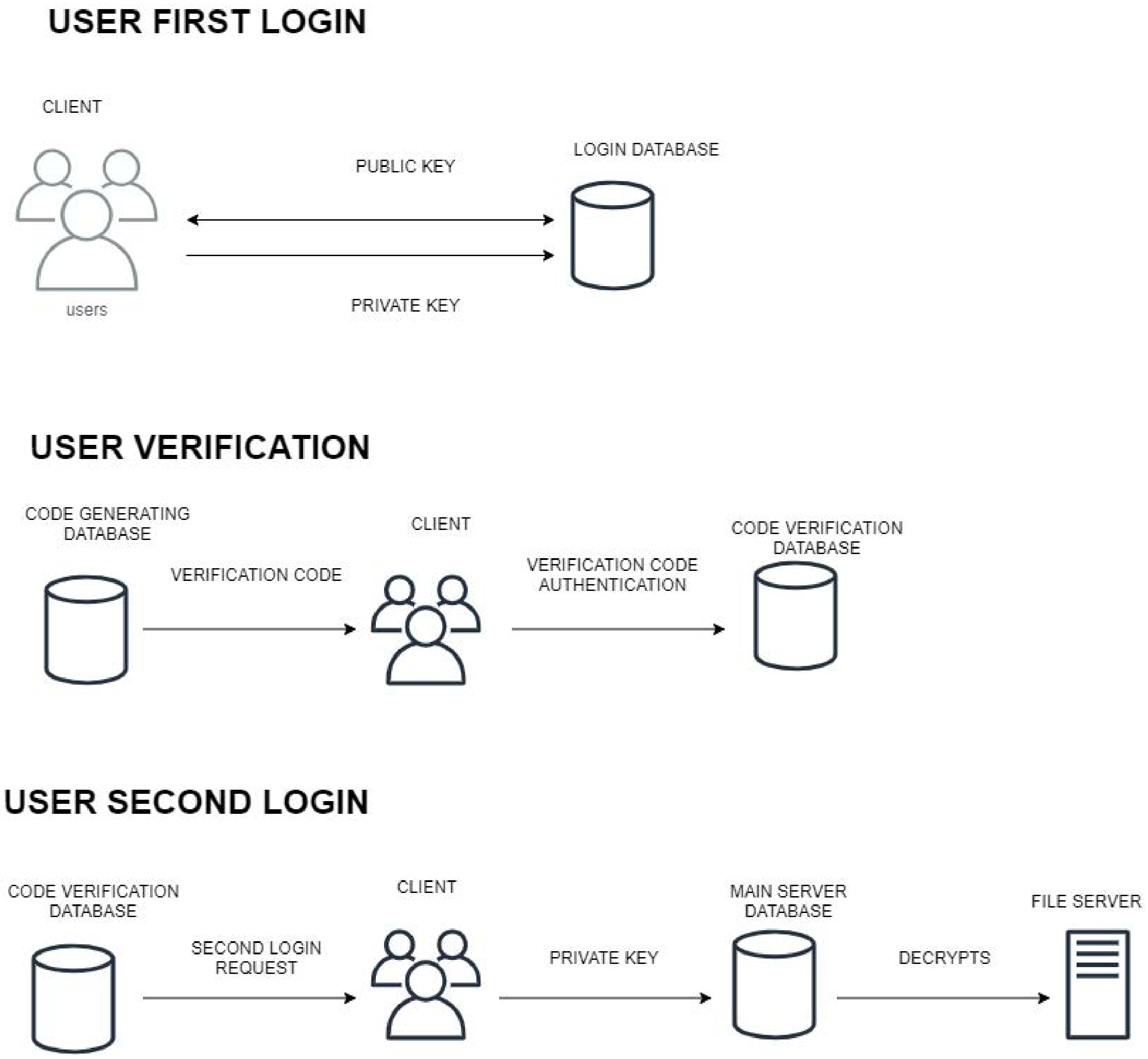

Figure 3. User login system

5.3.2 File Uploading

Figure 4 shows a process for file uploading. A registered user enters login id and password, after successful attempt user is ready to upload file. After completion of login process (As we mentioned in the above login section: Step 1 to Step 9) user get the login access of file server, the below steps involved in file uploading process,

- **Step 10:** User sends request to server to upload the file.
- **Step 11:** Check the user log for identifying whether both the login IP and access was success on bases of attempts
- **Step 12:** Request for the deduplication check
- **Step 13:** Response of the deduplication check is send to the user from the server.
- **Step 14:** if no duplicate found after comparison in server, then that file is ready to upload into the file log server
- **Step 15:** The uploaded file is split into four segments and the files are stored after encryption is completed

The document which is been uploaded is first stored as a whole in the one server and the copy of the document is been split into four parts and is been stored in the four different servers. Now the document is convert into hash not using the same hash function four different hashing key for four different servers and the duplication is checked according to the hash function. They are listed below,

- Server 1 –MD5
- Server 2 – SHA-1
- Server 3 – SHA-256
- Server 4 – SHA-384

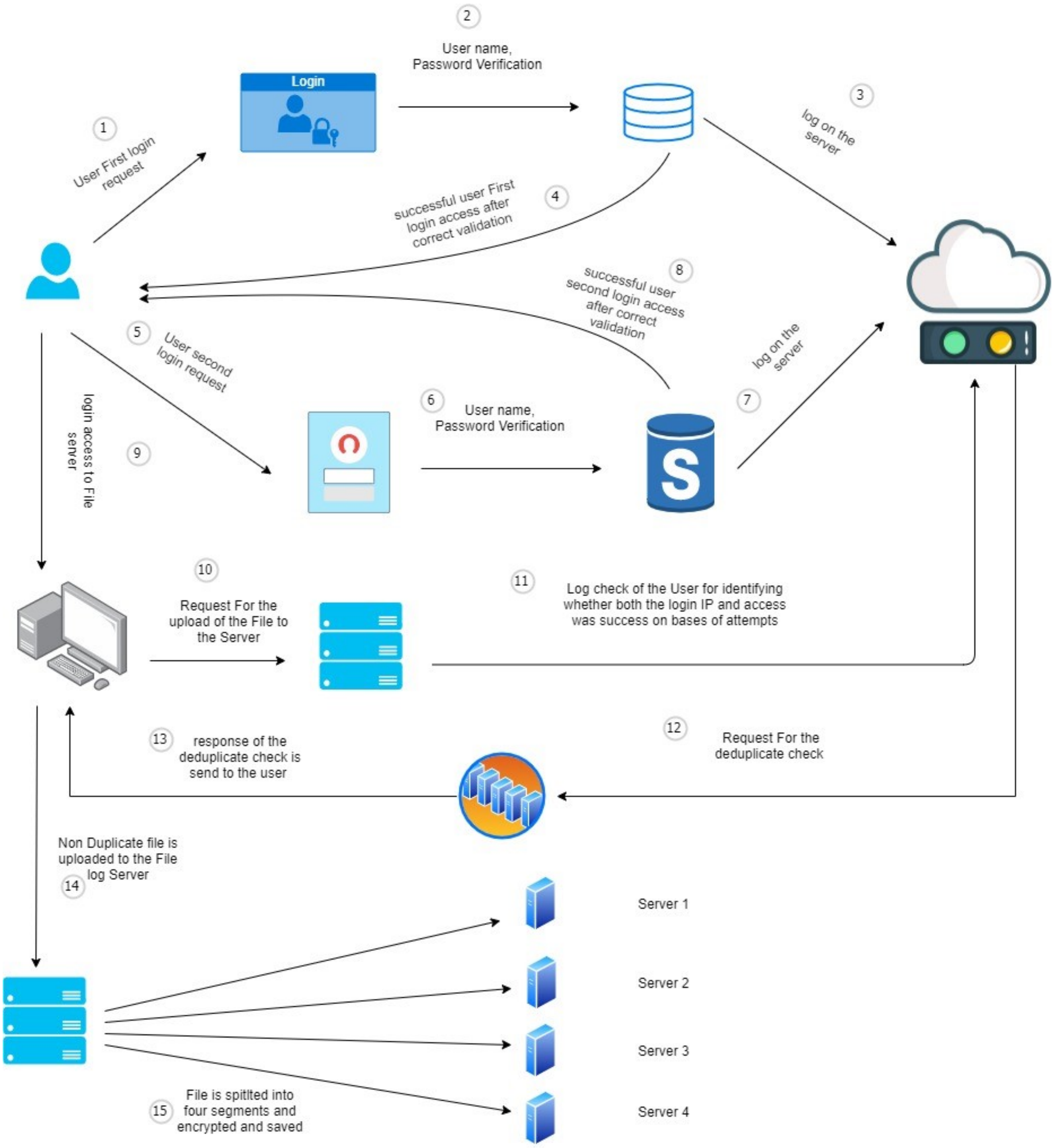
1
User First login request
Login
2
User name, Password Verification
3
log on the server
4
successful user First login access after correct validation
8
successful user second login access after correct validation
5
User second login request
9
login access to File server
6
User name, Password Verification
7
log on the server
10
Request For the upload of the File to the Server
11
Log check of the User for identifying whether both the login IP and access was success on bases of attempts
12
Request For the deduplicate check
13
response of the deduplicate check is send to the user
Non Duplicate file is uploaded to the File log Server
14
Server 1
Server 2
Server 3
Server 4
15
File is spitlted into four segments and encrypted and saved

Figure 4. File uploading process in cloud storage system

5.3.3 Request for Download:

In this module, client downloads the files by applying below steps which are illustrated in Figure 5. A registered user enters login id and password, after successful attempt user is ready to download the file. After completion of login process (As we mentioned in the above login section: Step 1 to Step 9) user get the login access of file server, the below steps involved in file downloading process,

- **Step 10:** User sends request to server to download the file
- **Step 11:** Admin can check the identity of the client
- **Step 12:** Check the user log for identifying whether both the login IP and access was success on bases attempts
- **Step 13:** Response is received from the log server whether the attempt is success or failure
- **Step 14:** if that attempt is success, then the client gets grant access to file with a time constraint even after download

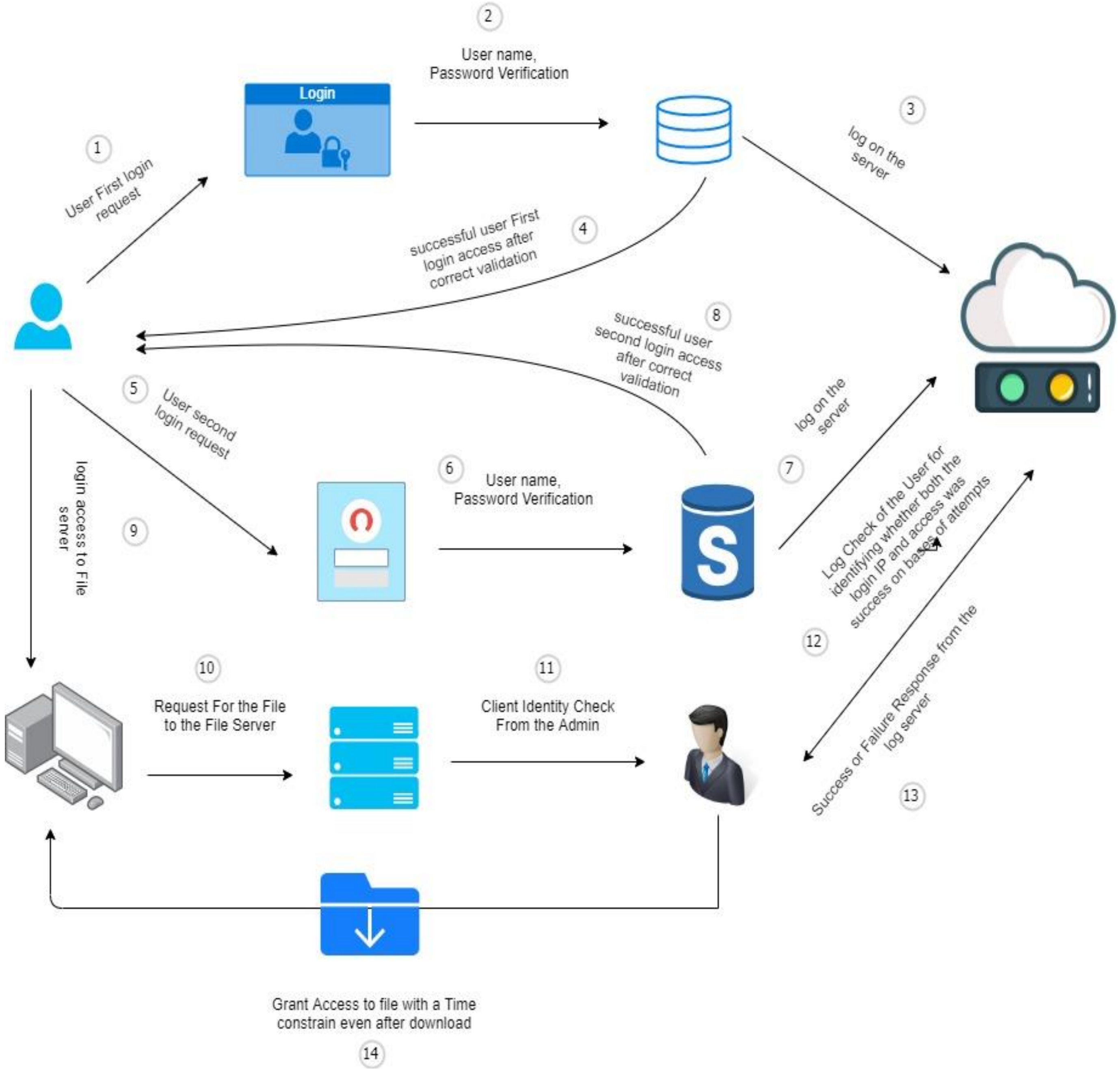
1
User First login request
Login
2
User name, Password Verification
3
log on the server
4
successful user First login access after correct validation
8
successful user second login access after correct validation
5
User second login request
6
User name, Password Verification
7
log on the server
9
login access to File server
Log Check of the User for identifying whether both the login IP and access was success on bases of attempts
12
10
Request For the File to the File Server
11
Client Identity Check From the Admin
Success or Failure Response from the log server
13
Grant Access to file with a Time constrain even after download
14

Figure 5. File downloading process in cloud storage system

5.3.4 Request for Own File Download

To download their own file from cloud server, user need to follow the below steps which are illustrated in Figure 6. Since they also need to complete the above mentioned login process (Step 1 to Step 9).

- **Step 10:** User sends request to server to download the file
- **Step 11:** Admin can check the identity of the client
- **Step 12:** Check the user log for identifying whether both the login IP and access was success on bases attempts
- **Step 13:** Response is received from the log server whether the attempt is success or failure
- **Step 14:** If that attempt is success, then the client gets grant access to file with a time constraint to the stored file log server
- **Step 15:** Finds the split files according to the log files
- **Step 16:** File is merged and access is granted to file with a time constraint even after download
- **Step 17:** File path is linked to the requested file server
- **Step 18:** Finally, the required file is downloaded

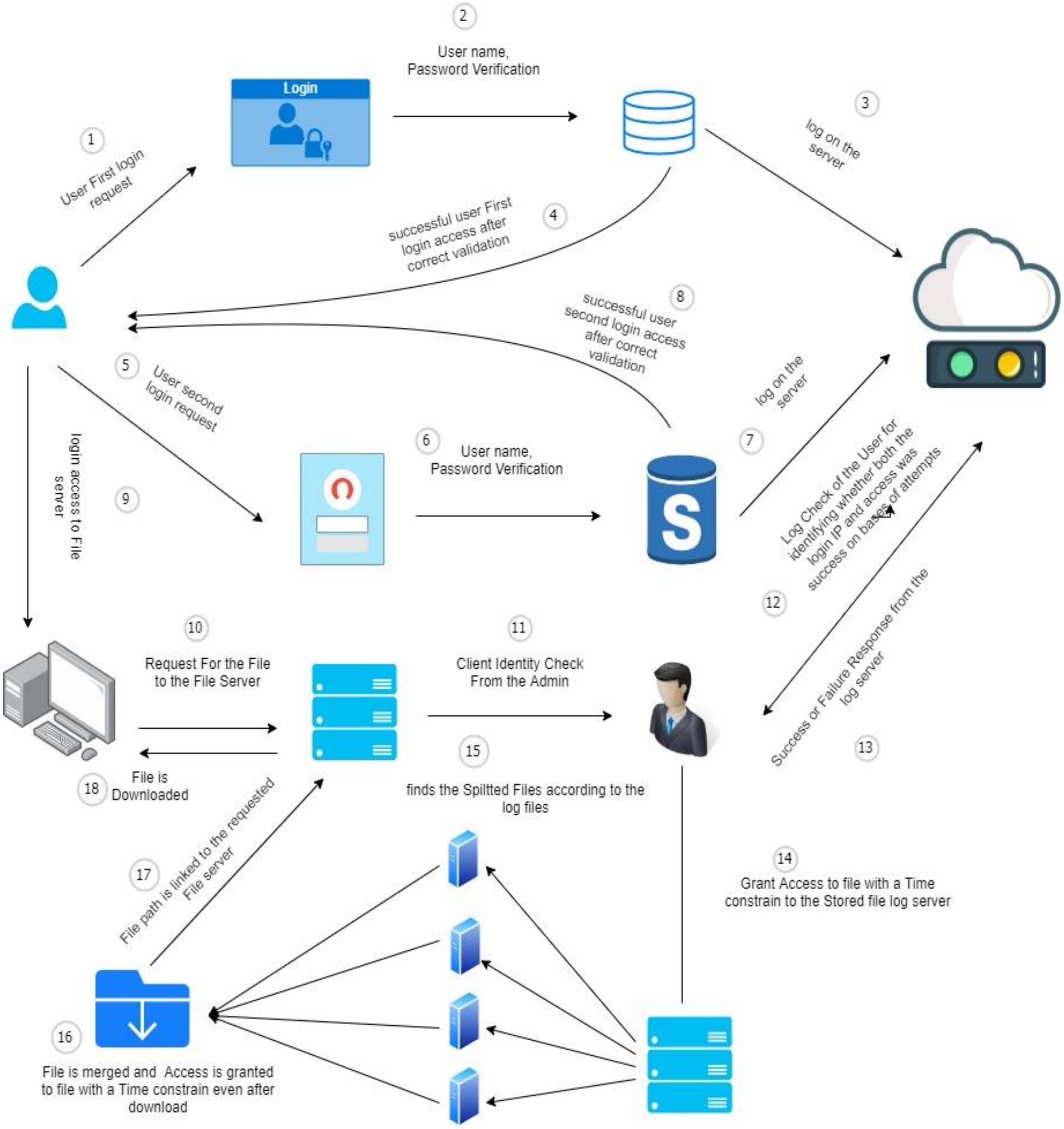
1
User First login request
Login
2
User name, Password Verification
3
log on the server
4
successful user First login access after correct validation
5
User second login request
6
User name, Password Verification
7
log on the server
8
successful user second login access after correct validation
9
login access to File server
10
Request For the File to the File Server
11
Client Identity Check From the Admin
12
Log Check of the User for identifying whether both the login IP and access was success on bases of attempts
13
Success or Failure Response from the log server
14
Grant Access to file with a Time constrain to the Stored file log server
15
finds the Spiltted Files according to the log files
16
File is merged and Access is granted to file with a Time constrain even after download
17
File path is linked to the requested File server
18
File is Downloaded

Figure 6. File request for own file downloading process from cloud storage system

6 Experimental Result

In this section, we evaluate the proposed deduplication system with encoding and decoding process using various key generation algorithms. All our experiments were performed on an Intel Xeon E5530 (2.40 GHz) server with windows-10 OS.

The data is collected from an example login page, where many people are requested to login using their own password. Those passwords were implemented to produce the encrypted key in the algorithm thereafter with time indicated to finish the full procedure.

6.1 Initial Password- Dataset

Table 1 illustrates the sample data (Initial password) with the AES and RSA parameters. We also calculate the Dec_Time (NANO SEC) and Total Execution time (SEC) for key generation process.

Table 1. Initial password generation factors

S.NO	UP	UP_KL	UP_KS	AES Key _Len	AES Key SIZE	RSA Key length	RSA Key Size	RSA KEY Generating Time (NANO SEC)	Dec_Time (NANO SEC)	Total Execution time (SEC)
1	Aenkdfn@	8	56	40	120	15	96	0.198	0.062	1.63
2	a5dfE5dg@	9	56	40	120	15	96	0.023	0.024	0.182
3	gg545hGH56	10	64	37	112	16	112	0.024	0.022	0.843
4	DDFFffgg@@3	11	64	40	120	15	88	0.021	0.017	0.235
5	2D2g3D5h#$1	11	64	40	120	16	104	0.024	0.018	0.173
6	ghRRh@dg$$$d	12	64	40	120	14	96	0.17	0.016	0.241
7	EESs334Fyu9df	13	64	37	112	16	104	0.016	0.016	0.384
8	g$fkgjWlh#5g4GH	15	72	37	112	14	96	0.017	0.016	0.25
9	!@#jjewreEERRR%ghg	18	80	37	112	29	176	0.031	0.022	0.206
10	!@#bsdfkj&&&%klklhj	19	80	40	120	30	152	0.022	0.031	0.272

6.2 Phrase Password

Table 2 illustrates the sample data (Phrase password) with the AES and RSA parameters. We also calculate the Dec_Time(NANO SEC) and Total Execution time(SEC) for key generation process.

6.3 Login

Table 3 shows the initial, Phrase and total login time in seconds.

Table 2. Phrase password generation factors

UP	UP_KL	UP_KS	Base64 Key_Len	Base64 Key SIZE	RSA Key length	RSA Key Size	RSA KEY Generating Time (NANO SEC)	Dec_Time (NANO SEC)	Total Execution time (SEC)
Hi buddy i am gone hack you	27	96	40	120	44	120	0.114	0.062	0.00208
your are always My hero	23	88	40	120	44	128	0.029	0.029	0.00069
Open if you can	15	72	40	120	24	120	0.022	0.022	0.00096
HItting into people always	26	96	40	120	44	120	0.025	0.023	0.0008
this is my game and u cannot play	33	104	40	120	64	120	0.032	0.029	0.00066
login in the pubg rack	22	88	40	120	44	120	0.025	0.032	0.00063
killing the robber to hack	26	96	40	120	44	120	0.023	0.024	0.00299
my first money hiest	20	80	40	120	44	120	0.025	0.023	0.00123
Going into break the mint	25	88	40	120	44	120	0.025	0.025	0.00818
the eneminity of aminities	26	96	40	120	44	128	0.037	0.027	0.00086

Table 3. Login time in seconds

S.NO	Ini_Login (SEC)	Phrase _login (SEC)	Login Time (SEC)
1	1.63	0.00208	1.63208
2	0.182	0.00069	0.18269
3	0.843	0.00096	0.84396
4	0.235	0.0008	0.2358
5	0.173	0.00086	0.17386
6	0.241	0.00066	0.24166
7	0.384	0.00063	0.38463
8	0.25	0.00299	0.25299
9	0.206	0.00123	0.20723
10	0.272	0.00818	0.28018

6.4 Upload

The file that has been uploaded is first stored as a whole in the one server and the copy of the document is been split into four parts and is been stored in the four different servers. Due to security concerns, each server utilizing various algorithms (MD5, SHA-1, SHA-256 and SHA-384) to generate the key and encrypt the files (Copy 1 to Copy 4).

Table 4. demonstrates the size of the files also key generation and encryption process timings. Finally, we calculate the file upload timing process. Figure 7 shows the uploading time for various file size.

Table 4. File size, key generation and encryption process timing

File Type	Total size of file	Size of file 1	Copy 1 key Gen_Time (SEC)	Encryption Time1 (Nano Sec)	Size of file 2	Copy 2 Gen_time (SEC)	Encryption Time2 (NANO Sec)	Size of file 3	Copy 3 Gen_Time (SEC)	Encryption Time3 (NANO SEC)	Size of file 4	Copy 4 Gen_Time (SEC)	Encryption Time4 (Nan0 SEC)	Upload time (SEC)
Text file 1	117kbs	20kbs	0.03063	0.00476	20kbs	0.00025	0.00754	20kbs	0.00042	0.00334	20kbs	0.01342	0.00382	0.34543
Text file 2	156Kbs	26kbs	0.00286	0.00171	26kbs	0.0003	0.00199	26kbs	0.00037	0.00163	26kbs	0.00247	0.00183	0.21803
Music 1	1.04mbs	20,142kbs	0.17362	0.24712	23,173kbs	0.16391	0.28368	26,347kbs	0.27465	0.30358	26,325kbs	0.36943	0.23765	0.97108
Music 2	1.06mbs	20,142kbs	0.13096	0.19218	23,173kbs	0.17493	0.24465	26,447kbs	0.24054	0.22853	26,399kbs	0.24387	0.24899	0.82402
Image 1	61,054Kbs	12,333kbs	0.09101	0.11954	14,288kbs	0.1011	0.1273	15,336kbs	0.1456	0.16013	15,271 kbs	0.14729	0.1387	0.48412
Image2	61,054Kbs	12,333kbs	0.08105	0.12616	14,288kbs	0.10602	0.13109	15,336kbs	0.13987	0.13227	15,271 kbs	0.11737	0.15665	0.49533
Video 1	2.45mbs	48,393kbs	0.41873	0.47438	56,205kbs	0.40067	0.54221	61,363Kbs	0.57875	0.63231	60,826kbs	0.45279	0.61916	1.44375
Video2	2.45mbs	48,393kbs	0.37719	0.59702	56,205kbs	0.76581	0.62839	61,363Kbs	0.57458	0.56389	60,826kbs	0.62967	0.55914	1.54989

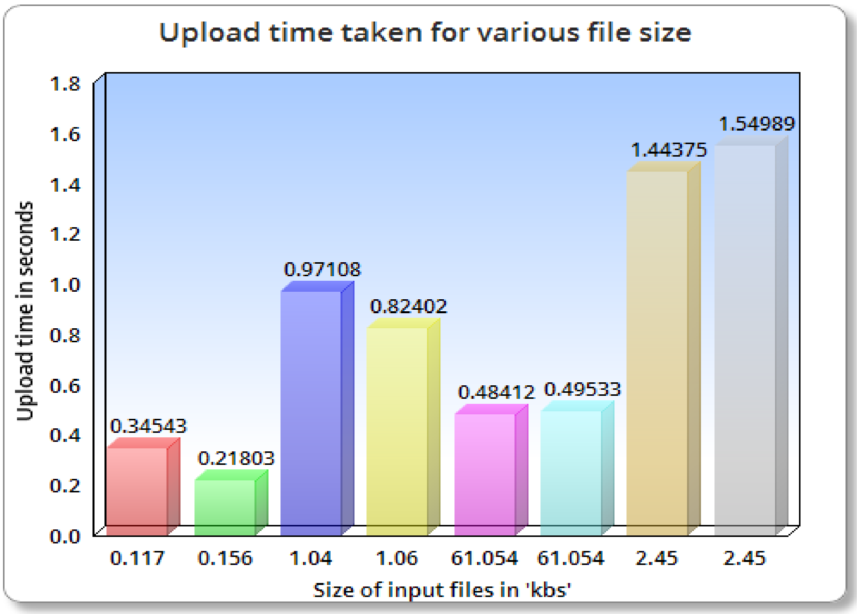

Figure 7. Upload time taken for various file size

6.5 Download

As we discussed in previous section, user sends a download call to cloud service supplier to acquire the file from log server. Utilize the file id in log server to acquire every encrypted chunk from the cloud storage supplier and decrypt every chunk by using the appropriate key. Finally, reconstruct the original file.

Table 5 illustrates the decryption time in nano-seconds for each chunk, total file merge time in seconds and size of the downloaded file. Figure 8 shows the downloading time for various file size.

Table 5. Decryption process timing

S.NO	File Type	Decryption Time 1 (NANO SEC)	Decryption Time 2 (NANO SEC)	Decryption Time 3 (NANO SEC)	Decryption Time 4 (NANO SEC)	Total Merge Time (SEC)	File Size (kbs)
1	Text file 1	0.01013	0.0031	0.00581	0.00347	0.02251	117kbs
2	Text file 2	0.00078	0.00256	0.00062	0.00045	0.00441	156Kbs
3	Music 1	0.24135	0.24324	0.32559	0.29622	1.1064	1.04mbs
4	Music 2	0.32896	0.25487	0.27488	0.30439	1.1631	1.06mbs
5	Image 1	0.16417	0.1614	0.16541	0.20014	0.69112	61,054Kbs
6	Image2	0.14245	0.15678	0.16267	0.17245	0.63435	61,054Kbs
7	Video 1	0.53731	0.62084	0.79836	0.64468	2.60119	2.45mbs
8	Video2	0.54866	0.62814	0.76312	0.645	2.58492	2.45mbs

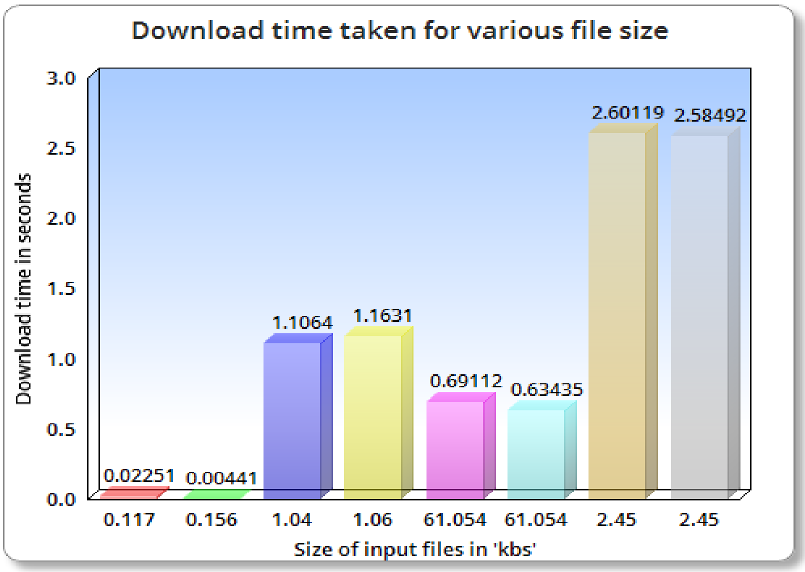

Figure 8. Download time taken for various file size

Examining the experimental results in graphs and tables, it shows that the encoding/decoding processing time. Exactly, at the file upload, the encoding time of 117kbs is 0.34543 and 2.45mbs is 1.54989, and is less than that of encrypting a data chunk in existing system result. At the file download, the decoding time is smaller than the time of encrypting a data chunk. The system can pipeline both the encoding as well as decryption modules, creating the decryption portion taking less time. This proposed system has also verified the cases with other file sizes (like 156Kbs, 1.04mbs, 1.06mbs, 61,054Kbs, 61,054Kbs and 2.45mbs) and made similar observations. With the size of file raising, the computational price of creating keys, symmetric encryption and decryption has expressively raised.

6.6 Evaluation Parameters

In proposed deduplication system, we assess the overhead by changing various elements such data/file size, amount of stored file and deduplication rate etc. factors which are mentions are given below.

- **File Size:** This is factor effects on time required on processing of file on authorized deduplication system. The time required on encryption, upload increase with respect to increase in file size, but the other operation such as token generation and duplicates check time remain constant throughout.
- **Deduplication Ratio:** The deduplication ratio is described as percent of storage space that has save by using this proposed deduplication system.

7 Conclusion

Cloud storage services deals on request virtualized storage resources also clients only pay for the area they essentially spent. As the growing demand and data store in the cloud, data deduplication is one of the systems utilized to advance storage proficiency. Data deduplication is a focused data compression method for removing duplicate copies of data in storage. This paper proposes secure deduplication with the aid of multiple key generation mechanism. This data deduplication method contributes a lot of advantages, along with security as well as privacy concerns are also become solve. As the client's profound data are safe from both insiders also outsider of the attacks. The stored file content in cloud storage should not be exposed to anybody excluding the client who owns the data. In proposed deduplication systems, only a valid client/user who owns the original data (communication is done with key based process) should be confirmed as an authentic owner cloud storage.

The quantity of ciphertexts generally increases fast in cloud storage. So we must set aside enough types of ciphertext for future extension. If not, we require the public key to extend. We cannot compromise both on security and duplication of data across storage locations in this information-dense environment. A plan has to be developed to improve storage optimization and to provide deduplication technology to data storage servers where the data provided is encrypted without negotiating encryptions.

References

[1] V. Waghmare, S. Kapse, Authorized Deduplication: an Approach for Secure Cloud Environment, *Procedia Computer Science*, Vol. 78, pp. 815-823, 2016.

[2] B. D. Aldar, V. Devmane, A survey on secure deduplication of data in cloud storage, *International Journal of Innovations in Engineering and Technology*, Vol. 6, No. 1, pp. 13-20, October, 2015.

[3] S. Halevi, D. Harnik, B. Pinkas, A. Shulman-Peleg, Proofs of ownership in remote storage systems, *18th ACM Conference on Computer and Communications Security*, Chicago, Illinois, USA, 2011, pp. 491-500.

[4] H. Hou, J. Yu, R. Hao, Cloud storage auditing with deduplication supporting different security levels according to data popularity, *Journal of Network and Computer Applications*, Vol. 134, pp. 26-39, May, 2019.

[5] Y. Zhou, D. Feng, Y. Hua, W. Xia, M. Fu, F. Huang, Y. Zhang, A similarity-aware encrypted deduplication scheme with flexible access control in the cloud, *Future Generation Computer Systems*, Vol. 84, pp. 177-189, July, 2018.

[6] X. Zheng, Y. Zhou, Y. Ye, F. Li, A cloud data deduplication scheme based on certificateless proxy re-encryption, *Journal of Systems Architecture*, Vol. 102, Article No. 101666, January, 2020.

[7] W. Xia, D. Feng, H. Jiang, Y. Zhang, V. Chang, X. Zou, Accelerating content-defined-chunking based data deduplication by exploiting parallelism, *Future Generation Computer Systems*, Vol. 98, pp. 406-418, September, 2019.

[8] P. Singh, N. Agarwal, B. Raman, Secure data deduplication using secret sharing schemes over cloud, *Future Generation Computer Systems*, Vol. 88, pp. 156-167, November, 2018.

[9] H. Yuan, X. Chen, T. Jiang, X. Zhang, Z. Yan, Y. Xiang, DedupDUM: Secure and scalable data deduplication with dynamic user management, *Information Sciences*, Vol. 456, pp. 159-173, August, 2018.

[10] S. Li, C. Xu, Y. Zhang, CSED: Client-Side encrypted deduplication scheme based on proofs of ownership for cloud storage, *Journal of Information Security and Applications*, Vol. 46, pp. 250-258, June, 2019.

[11] Y. Shin, D. Koo, J. Hur, A Survey of Secure Data Deduplication Schemes for Cloud Storage Systems, *ACM Computing Surveys*, Vol. 49, No. 4, pp. 1-38, December, 2017.

[12] J. Paulo, J. Pereira, A survey and classification of storage deduplication systems, *ACM Computing Surveys*, Vol. 47, No. 1, pp. 1-30, July, 2014.

[13] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, M. Theimer, Reclaiming space from duplicate files in a serverless distributed file system, *Proceedings of the 2002 22nd International Conference on Distributed Computing Systems*, Vienna, Austria, 2002, pp. 617-624.

[14] M. V. Kakde, N. B. Kadu, Survey paper on deduplicating data and secure auditing in Cloud, *International Journal of Computer Science and Information Technologies*, Vol. 7, No. 1, pp. 94-95, January-February, 2016.

[15] R. N. Widodo, H. Lim, M. Atiquzzaman, A new content-defined chunking algorithm for data deduplication in cloud storage, *Future Generation Computer System*, Vol. 71, pp. 145-156, June, 2017.

[16] H. Cui, R. H. Deng, Y. Li, G. Wu, Attribute-based storage supporting secure deduplication of encrypted data in Cloud, *IEEE Transactions on Big Data*, Vol. 5, No. 3, pp. 330-342, September, 2019.

[17] J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, W. Lou, Secure Deduplication with Efficient and Reliable Convergent Key Management, *IEEE Transactions On Parallel and Distributed Systems*, Vol. 25, No. 6, pp. 1615-1625, June, 2014.

[18] R. S. Abdeldaym, H. M. Abd Elkader, R. Hussein, Modified RSA Algorithm Using Two Public Key and Chinese Remainder Theorem, *Journal of Electronics and Information Engineering*, Vol. 10, No. 1, pp. 51-64, March, 2019.

[19] D. N. Tran, T. N. Nguyen, P. C. P. Khanh, D. T. Trana, An iot-based design using accelerometers in animal behavior recognition systems, *IEEE Sensors Journal*, pp. 1-14, January, 2021.

[20] T. G. Nguyen, T. V. Phan, D. T. Hoang, T. N. Nguyen, C. So-In, Efficient SDN-Based Traffic Monitoring in IoT Networks with Double Deep Q-Network, *International Conference on Computational Data and Social Networks*, Dallas, TX, USA, 2020, pp. 26-38.

Biographies

R. Vignesh is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Muthayammal Engineering College (Autonomous), Namakkal. He completed his Bachelor of Engineering degree in Computer Science and Engineering in the year 2014 from PGP College of Engineering and Technology, Anna University, Chennai. He did his Master of Engineering in Computer Science and Engineering from PGP College of Engineering and Technology, Anna University, Chennai in the year 2016. He was awarded with Visalatchi Award for Outstanding Student of the year. He is now Pursuing Ph.D in Computer Science and Engineering from Anna University, from 2017 to present. His research interests include Cloud Computing Techniques in both Storage and Security.

J. Preethi is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Anna University, Regional Campus, Coimbatore. She completed her Bachelor of Engineering degree in Computer Science and Engineering in the year 2003 from Sri Ramakrishna Engineering College, Bharathiyar University, Coimbatore. During her studies, she has been awarded Sri Ramaswami Naidu memorial award for securing the Top rank in B.E (CSE) during the academic year 2000-2001. She did her Master of Engineering in Computer Science and Engineering from Government College of Technology, Coimbatore, Anna University, Chennai in the year 2007. She obtained her Ph.D in Computer Science and Engineering from Anna University, Chennai in the year 2013. Her research interests include Soft Computing Techniques, Medical Image Processing, Data mining and Heterogeneous Wireless Networks. She has received OUTSTANDING FACULTY IN ENGINEERING for exceptional academic records, initiatives and developments in the field of Computer Science and Engineering for the year 2018 from Centre for Advanced Research and Design, Venus International Foundation.