International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 3 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT1952181

Data De-Duplication Using SHA (Secure Hash Algorithm)

G. Kalyani[1], D.S.L. Neethika[2], Ch. Jayasri[3], A. Divya[4], Sk. Wasim Akram[5]

[1,2,3,4] Department of Computer Science and Engineering, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India

[5]Assistant Professor Department of Computer Science and Engineering, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India

ABSTRACT

Now-a-days the number of users using cloud storage has increased so that the data stored has been increased in exponential rates. The data should be secured and the storage should be used efficiently. But a lot of duplicate data is present as two or more users may upload the same data. To make use of the cloud storage efficiently we have to reduce the redundant data hence improving the resources like storage space, disk I/O operations of the cloud vendors. Data De-Duplication is the process to remove redundant data and store only one instance of duplicate data. The objective of the proposed system is to increase the efficient comparison of hash values of a different data blocks and security of data. This paper includes a method for data deduplication using SHA (Secure Hash Algorithm) and AES. SHA is used as it is more secure than other hashing algorithms. The data is encrypted using AES at owner machine itself and using SHA the redundant data will be eliminated.

Keywords : Redundant, Encrypt, Hash value, Secure Hash Algorithm, AES, DeDuplication.

I. INTRODUCTION

Cloud processing gives versatile, minimal effort and area free administrations over the web. The administrations gave ranges from basic reinforcement administrations to distributed storage foundations. The quick development of information volumes has enormously expanded the interest for systems for space and transmission capacity [1]. Distributed storage administrations like Drop box, Google Drive pick a deduplication procedure where the cloud server stores just a solitary duplicate of repetitive information and makes connects to the duplicate as opposed to putting away genuine duplicates. The security of clients' information turns into another test. Consequently the clients encode the information before redistributing to the cloud. During that process we will be facing a problem of duplication. To solve that we have to perform data deduplication.

There is a huge increment in the measure of information produced every day and in 2020 it is normal 44 zettabytes of information will be delivered. But storing and managing these large amounts of data is really a difficult task. Cloud computing offers a new way of service provisions by rearranging the resources over the internet. Cloud storage is the most popular among all the storage providers as cloud storage is the most efficient one. Data duplication occurs when the same data is being shared to the cloud storage by multiple users [2].

Data de-duplication keeps only one physical copy and eliminates multiple data copies. Through this consumption of resources will be reduced and saves the disk space and network bandwidth. Cloud users upload their information. Security and privacy are the major issues though data deduplication promises lots of benefits. Data needs to be encrypted and store in the cloud which ensures security and user privacy [3].

Let us consider three users user1, user2, user3 they are uploading some amount of data through this we will get an idea how deduplication results in.

User1 uploads -----→ a, b, c

User2 uploads -----→ d, a, b

User3 uploads -----→ d, c, a

The similar kind of data has been uploaded which reduces storage and efficiency of resources [4]. By de-duplication only the files "a, b, c, d" will only be stored into the cloud. This eliminates the repeated data and only stores first unique instance of any data. Whenever the user tries to store the data which is already present in the stored in the cloud it only creates a pointer to the existing one rather than creating redundant data. In block level for each file or chunk a unique hash value will be generated using hashing algorithms like SHA or MD5. Whenever user needs to upload a file a hash value will be generated foe that and it will be compared with the existing hash values. If the hash value is not present then the hash value and the file will be stored else it will not store. But in this we will be facing a problem whenever the hash algorithm produces same hash values for different chunks of data then collision occurs. This hash collision leads to data loss [5].

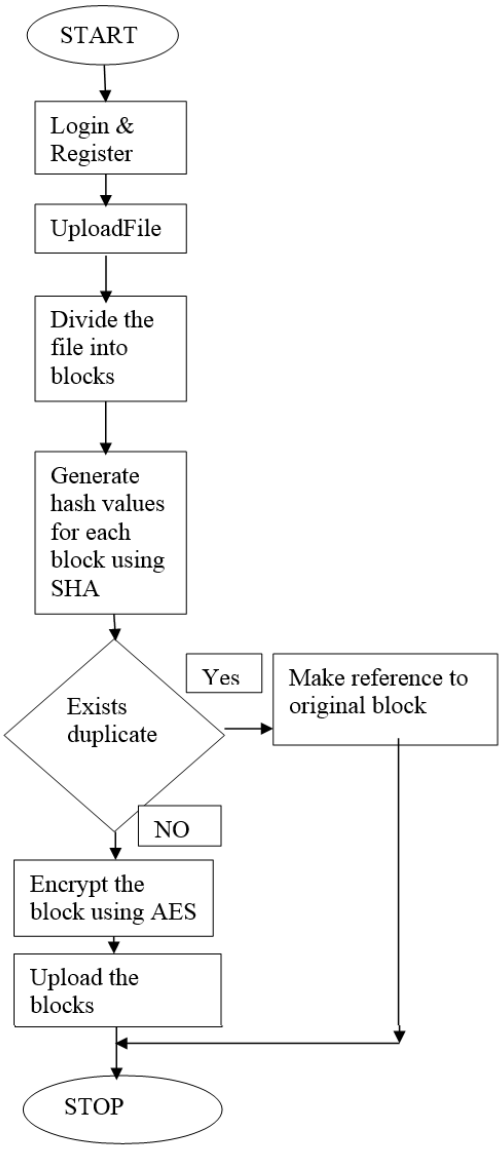
II. METHODS AND MATERIAL

The main objective of the proposed system is to eliminate redundant data and to provide security using SHA and AES. This section explains how SHA and AES work. SHA is the most secure hashing algorithm as the no of bytes is more and the chance of collision is very less. In order to store the data in to the cloud storage the following steps are followed:

First user needs to login (or) register to the cloud services.

Secondly, we need to upload the file within the user computer encryption is done which provides security for the data. This is done through AES algorithm [7].

After encryption the file is divided into number of chunks and generates the hash values using SHA algorithm for every chunk to the file which the user is uploaded. The hash values are compared with already existing hash values. If they are different the hash values are stored and the data is stored otherwise it creates a pointer or a reference to the original chunk that is already present [6].



SHA Algorithm:

Step 1: Slice data into chunks (fixed or variable).

Step 2: Generate hash per chunk and save.

A_h B_h C_h D_h E_h

Step 3: Slice next data chunks and look for hash map.

Step 4: Generate hash per chunk.

A_h B_h C_h D_h E_h

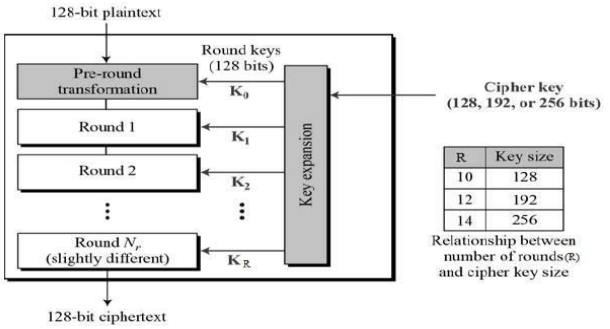
Step 5: Refer the previous hash values and store and secure [9].

AES Algorithm:

The more outstanding and extensively grasped symmetric encryption computation at risk to be encountered nowadays is the Advanced Encryption Standard (AES). It is discovered no under six times speedier than triple DES. A swap for DES was required as its key size was pretty much nothing. With growing figuring power, it was seen as unprotected against far reaching key interest attack. Triple DES was proposed to vanquish this drawback anyway it was found moderate.

Operations of AES:

AES is an iterative model. It depends on 'substitution– stage arrange'. It contains a progression of connected tasks, some of which include supplanting contributions by explicit yields (substitutions) and others include rearranging bits around (stages). Strangely, AES plays out the entirety of its calculations on bytes as opposed to bits. Thus, AES treats the 128 bits of a plaintext hinder as 16 bytes. These 16 bytes are organized in four sections and four columns for handling as a grid. In contrast to DES, the quantity of rounds in AES is variable and relies upon the length of the key. AES utilizes 10 rounds for 128-piece keys, 12 rounds for 192-piece keys and 14 rounds for 256-piece keys. Every one of these rounds utilizes an alternate 128-piece round key, which is determined from the first AES key. The schematic of AES structure is as follows:

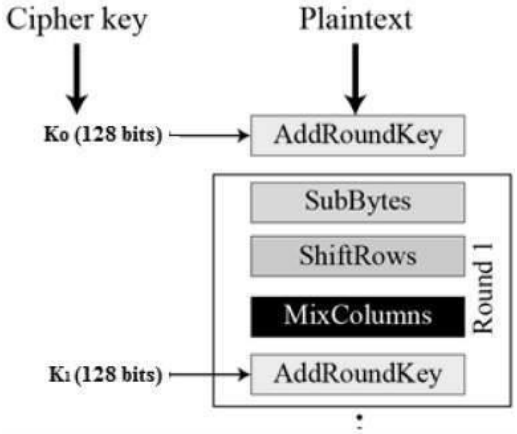


R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Encryption Process:

Each round contains four sub-forms. The first-round procedure is given below:



1. Byte Substitution: The 16 input bytes are substituted by looking up a fixed table (S-box) as given in the design. The result is in a matrix of four rows and four columns:

2. Shift Rows: Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows

A) First row is not shifted.

B) Second row is shifted one (byte) position to the left.

C) Third row is shifted two positions to the left.

D) Fourth row is shifted three positions to the left.

E) The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

3. Mix Columns: Every section of four bytes is presently changed utilizing an extraordinary scientific capacity. This capacity takes as info the four bytes of one segment and yields four totally new bytes, which supplant the first segment. The outcome is another new grid consists of 16 new bytes. It ought

to be noticed that this progression isn't performed in the last round.

4. Addroundkey: The 16 bytes of the framework are currently considered as 128 bits and are XOR the 128 bits of the round key. In the last round, at that point the yield is the cipher text. Something else, the subsequent 128 bits are deciphered as 16 bytes and we start another comparable round.

Decryption Process:

It is similar to encryption but in reverse order

- AddroundKey
- Mix Columns
- Shift rows
- Byte Substitution

This proposed plan gives a protected way to deal with distinguish copy information in cloud by utilizing SHA (secure hashing calculation) and AES (Advanced encryption standard) and get proficient outcomes. It gives the consequences of capacity estimate, deduplication proportion and decreases the capacity size and time. When a user upload a new file which is not present on the cloud then the file is divided into the fixed size blocks or chunks and perform SHA on each chunk. The generated hash value is compared with the hash values that are already present in the index table. If hash value is not found in index table then upload that block on cloud otherwise make a reference to the previous block on cloud. SHA is more secure and takes less time than MD5 and provides more collision resistance[8].

III. RESULTS AND DISCUSSION

This proposed plan gives a protected way to deal with distinguish copy information in cloud by utilizing SHA (secure hashing calculation) and AES (Advanced encryption standard) and get efficient outcomes. It gives the storage size, deduplication ratio and reduces the capacity size and time. When a user upload a new file which is not present on the cloud then the file is divided into the fixed size blocks or chunks and perform SHA on each chunk. The generated hash value is compared with the hash values that are already present in the index table. If hash value is not found in index table then encrypt the block and upload it on the cloud otherwise make a reference to the previous block on the cloud. SHA is more secure and takes less time than MD5 and provides more collision resistance.

IV. CONCLUSION

Data de-duplication is the emerging trend and the secured deduplication is the important concerns of the cloud users. This paper focuses on the basics of de- duplication and how deduplication is done, what are the various papers based in this with different algorithms, and how this is different from others. The proposed plan will be applied for block level deduplication and the outcomes represent the reduction in storage size and take less effort to store efficient information. At present, an improved strategy for capacity has been tried just for text files. In future work, it may extend for different sorts of files, for example, video and audio files.

V. REFERENCES

[1]. Priyadharshini.P, Dhamodran.P, Kavitha.M.S "A Survey on De-Duplication in Cloud Computing" in IJCSMC vol.3, Issue 11 (November 2014), ||pp.149-155https://www.ijcsmc.com/docs/papers/November2014/V3I11201435.pdf

[2]. https://data-flair.training/blogs/features-of-cloud-computing/

[3]. https://docs.microsoft.com/en-us/windows-server/storage/data-deduplication/understand

[4]. Rohini Sharma, "Data De-Duplication in Cloud Computing: A Review" in IJEAST vol. 2, Issue 2455-2143(February-March2017),||pp.26-29

[5]. ttp://www.ijeast.com/papers/26-29,Tesma204,IJEAST.pdf

[6]. Sachit-Ghimire,D.Venkata Subramanian "Chunking Algorithm for Data deduplication" in IJSRD Vol. 2, Issue(05,2014),||ISSN2321-0613 http://www.ijsrd.com/articles/IJSRDV2I5286.pdf

[7]. Ider Lkhagvasuren1, Jung Min So1, Jeong Gun Lee1, Chuck Yoo2, Young Woong Ko1, "Byte-index Chunking Algorithm for Data Deduplication System", in ijsia Vol. 7, No. 5(2013), ||pp.415-424 https://pdfs.semanticscholar.org/0880/325749067aefa23dce9bf63c4e92c6478773.pdf

[8]. Tannu1, Karambir2, "Detection of De-Duplication Using SHA-512 and AES-256 in Cloud Storage" in IASIRhttp://iasir.net/AIJRSTEMpapers/AIJRSTEM17-323.pdf

[9]. https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm

[10]. https://pibytes.wordpress.com/2013/02/09/deduplication-internals-hash-based-part-2

Cite this article as :

G. Kalyani, D.S.L. Neethika, Ch. Jayasri, "Data De-Duplication Using SHA (Secure Hash Algorithm) ", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 3, pp. 01-05, May-June 2019. Available at doi : https://doi.org/10.32628/CSEIT1952181
Journal URL : http://ijsrcseit.com/CSEIT1952181