Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

A Novel Data Storage Solution for Cloud

Bhagya S.L.
Dept. of Computer Science
Mar Baselios College of Engineering & Technology,
Thiruvananthapuram, Kerala, India

Prof. Raju K. Gopal
Dept. of Computer Science
Mar Baselios College of Engineering & Technology,
Thiruvananthapuram, Kerala, India

***Abstract*— Data deduplication is the latest data compression method which will succours to reduce the data storage space by eliminating redundant data stored and keeps exactly one copy of the data. The amount of digital information is growing drastically due to increase in the use of internet and IoT [Internet of Things] devices. Since multiple users are dealing with same data, miscellaneous copies of same data may be generated and stored as multiple copies, which results huge amount of storage space requirements. By performing deduplication, exactly one physical copy of the data is stored and all duplicate copies of the data are referenced. So this technique eliminates storing multiple copies of the same data and save lot of storage space. It produces much better result than other text data compression methods.**

Keywords- Data compression; Deduplication; AES encryption; SHA 256

I. INTRODUCTION

Starting year 2000, the digital data in the world was growing explosively and poses tremendous challenges to the Information Technology (IT) infrastructure managers/administrators, in facilitating and managing the storage space. Statistics shows that in 1980's less than 1% of information were stored in digital format and it became 94% in late 2000s and by 2014 almost 99% of data in the world became digital. As a result of the rapid increase in the number of Internet and IOT (Internet of things) users resulted in the proliferation of hand held gadgets like tablet computers, smart phones and sensors which leads to the explosive growth of digital data in the world. Humongous amount of information has been generated in the past two years compared to the previous years. Almost 4.4 zettabyte of digital data were generated in 2013. It is predicted that in year 2020 about 1.7 megabytes of new data will be developed every second per user/device. Therefore the estimated amount of digital information that will be generated during 2020 will be 44 zettabyte. That is digital universe is huge and it is growing exponentially, which means we requires much more storage space to satisfy the needs of upcoming generations.

According to the studies conducted by Microsoft they found that about 50% to 80% of the data in their storage system are redundant. Same data are handled by different users and they are getting stored multiple times in different location. The existence of additional information along with original data causes redundancy and it will leads to unnecessary increase in size of storage space. The excessive data causes data instability, data corruption and it will decrease the efficiency of storage media. Data deduplication also called as intelligent compression or single-instance storage is a redundant data reduction technique which minimizes the required storage space by eliminating the multiple copies of same data and by keeping exactly one copy. Each such copy can be interpreted based on various granularities: which may refer to file level deduplication (refer to a whole file) or block-level deduplication (which refers to more fine-grained fixed or variable size data blocks). After performing deduplication redundant data will be replaced with a pointer to the unique data. With this approach, only one instance of the file is actually stored and each subsequent copy is just referenced back to the saved one.

Data deduplication can be of two types, chunk level and file level deduplication [2]. In chunk level deduplication the incoming file will be divided into chunks/blocks and unique chunks will be stored by comparing every incoming chunk for identifying its duplication. This method achieves better deduplication efficiency because it does exact deduplication. But its throughput is low, since it checks every incoming chunk for duplication. File level deduplication method causes the chunks of similar files to be compared against duplicates. File level deduplication is used to identify the duplicates in attachments, emails, folders. However, the deduplication efficiency is comparatively low as some duplicate chunks may be found on different groups.ie. Chunk level deduplication is better than file level deduplication.

Ensuring security to valuable and sensitive data in an Internet environment is highly challenging. From user's point of view, security and privacy are the most important aspects while outsourcing their data. To intensify security and privacy, user need to encrypt their data before uploading on a remote storage system. Encryption is the most coherent way to attain security. The concept of data deduplication is compactable with convergent encryption method which will ensure data confidentiality [3]. Since this encryption process is deterministic, similar data copies will produce same key and same cipher text which will allow deduplication on the encrypted data. For example, user derive the key K from the message M such that K=H(M) where H is cryptographic hash function and thus the cipher text C=E(K,M)=E(H(M),M), which means user get the same cipher text for the same file. That is convergent encryption is a good option for both encryption and deduplication.

This paper is organized as follows. In Section 2 various related works are explained. Our proposed architecture is explained on Section 3. The proposed system is explained on section 4. The performance analysis has been given in Section 5. Finally, we draw conclusion in Section 6.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

II. RELATED WORKS

Data compression is a technique that plays a vital role in data storage and transmission. Compression means reduction in the amount of bits required to represent data. It can save huge storage space, reduce costs of storage hardware and can minimize network bandwidth. Compression consists of two stages, an encoding algorithm that uses a message as input and generates a compressed result and a decoding algorithm that regenerates the original information or some approximation of the original from the compressed representation. Lossless and lossy compressions are the two categories of data compression [5]. Lossless compression eliminates statistical redundancy and have compression ratio of around 2:1. It is reversible. While lossy compression reduces data by identifying unnecessary information and its compression ratio is around 50:1 and it is irreversible. For network related applications lossy compression is mainly used. The lossless compression is mainly applicable for the preservation of text and images for legal reasons, storage and transmission of medical information or images and so on [1].

Various methods for data compression are Huffman coding, Arithmetic encoding, Adaptive Huffman coding, Run-Length encoding, Lempel Zev Welch algorithm and Shannon Fanon algorithm etc [5]. Claude E Shannon formulates the theory of data compression. He introduced the theory of information entropy.

E.g. Let string = abaaacabba

Let a = 6

b = 3

c = 1

Therefore, Entropy of $a = H(a) = -\log_2 0.6 = 0.737bits$

Entropy of $b = H(b) = -\log_2 0.3 = 1.737bits$

Entropy of $c = H(c) = -\log_2 0.1 = 3.322bits$

Entropy of string $H(\{a,b,c\}) = H(a) \times 0.6 + H(b) \times 0.3 + H(c) \times 0.1 = 1.295bits$

Huffman Coding: David Huffman developed Huffman coding algorithm which is a statistical model based lossless data compression method also called entropy coding was introduced during 1950's. It identifies redundancy at the byte level. Here it uses a frequency sorted binary tree to generate the optimal prefix codes for entropy coding. This method will separates the input into the component symbols and replace with a shorter code and it is used as the back end of GZIP, JPEG and many other applications. During 1960's Elias proposed arithmetic coding which achieves the basic limit of data compression, for better compression ratio the whole data was encoded into several decimals. During 1970's Dictionary model based coding was proposed by Lempel and Ziv [4]. It consists of basically two algorithms LZ77 and LZ78 [6]. It will perform data compression at the string level based on dictionary of previously seen things. It uses a sliding window to identify and replace repeated strings by length and position of matched data. After that, variant of LZ compression approach was introduced during 1980s. Some examples are DEFLATE, LZMA [which will improve compression ratio], LZO, LZW [which will speed up the compression process]. During 1990's Delta Compression has been introduced. It performs the compression of homogeneous files [the similar files can be different versions of files or similar chunks]. It stores and transmits information in the form of delta (differences) between sequential data rather than entire document. This is also known as data differencing. Software revision control systems, delta compression at the file system level, software distribution, exploring file differences, improving http performance, efficient web page storage are the various applications of delta compression. At the end of the taxonomy of data compression methods i.e. during 2000's data deduplication was proposed. It helps the process of compression in large scale storage systems. Today's mercantile cloud storage services like Drop-box, Mozy and Memopal have been using deduplication to storage space.

Along with efficient data storage, data security is also indispensable. In traditional encryption scheme, different users encrypt their data with their own key, which leads to the generation of different cipher texts for identical copies of different users and thus deduplication becomes impossible. Baseline approach is inefficient and unreliable, as it generate huge number of keys with the increasing number of users and each users are requested to protect their own master key [3]. This inspires to investigate, how efficiently and reliably secure data deduplication can be achieved.

III. PROPOSED ARCHITECTURE

Based on the literature survey, our architecture consists of three entities: user, processing unit and storage system.

Here, user is an entity, who can either upload the file to storage system or can download the file from storage system. This system is developed based on user's point of view. Second entity is the processing unit which will perform all the system activities. Activities like file optimization, dictionary matching, data encryption/ decryption, de-tagging, de-optimization are performed by the processing unit. Third entity is the storage system which can be a cloud that can efficiently store the data. There is an inbuilt dictionary which contains list of words corresponding to each alphabet. The efficiency of the system depends on the strength of dictionary. As the strength of the dictionary increases we get more efficient result.

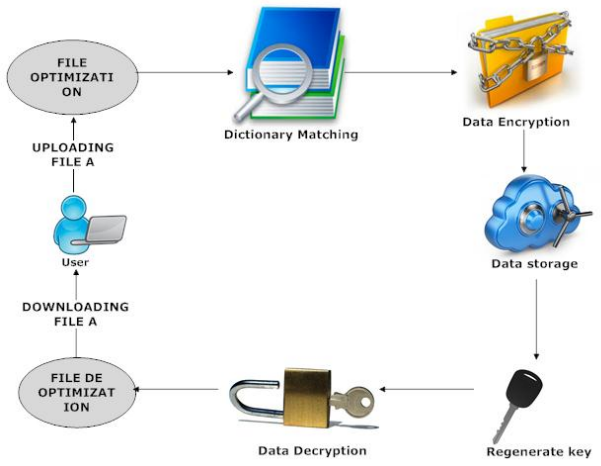

Fig.1 Proposed architecture

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

IV. PROPOSED SYSTEM

In this section, we present our system, which aims to provide required security properties of secure deduplication and which alleviate the key management overhead and offers the fault tolerance assurance for key management.

For example, consider that a user wants to upload a file A. First, perform chunk level or block level deduplication after which we have to identify the distinctive block that have to be uploaded. Each block is related to a tag for the duplicate check. On the input file A, the processing unit has to compute the optimized file block by separating special characters from File A and sent the optimized data to next stage. The first set of compression has been completed during this stage. On the optimized file, the following computations will be performed: divide the file A (optimized file) into a set of blocks (here each block consists of a word), compute block tag and sent the block tag to dictionary matching section. Block tag is calculated based on a predefined dictionary where each word (or block) will be replaced with the file index. The strength of dictionary actually determines the space that can be saved after performing deduplication. As the strength of dictionary increases, more words can be converted to tags and the redundancy of huge number of words can be eliminated. If the block is not in the dictionary, instead of replacing the block, the block will be kept as such. The result of this dictionary matching section is actually our plain text for encryption.

Convergent encryption contributes various options to implement data confidentiality while perceiving deduplication. Plain text is encrypted using convergent key which is obtained by computing the cryptographic hash value of data. Homogeneous data copies will create same convergent key and identical cipher text. That is the data copy is encrypted using the key derived from the data copy itself. This helps to perform deduplication on the cipher text. Hash function, SHA-256 [7] is used to generate a hash key of size 32 bytes for each data blocks. Secure Hash Algorithm is one of the cryptographic hash functions, where hashes are similar to a signature for a text or a data files. Generally, this type of hash algorithm is used to perform message integrity checks and digital signatures in miscellaneous information security applications. The encryption algorithm used is AES (Advanced Encryption Standard) [8] [9] which is a symmetric-key encryption algorithm. It is a block cipher encryption method, which is hard to break. Here the input is a block of 128 bits of plain text and key of size 256 bits. After encryption, the cipher text is uploaded to the storage system. In order to ensure key security, visual cryptography is performed on the key i.e. converting the key into bitmap format and splitting each pixel of the image into two shares, white and black, which is then kept aside using password. Only after combining both the shares, the original key can be retrieved.

Suppose that the user wants to download the file A. For downloading a file user has to go through various authentication processes. First of all, file authentication has to be performed by entering proper file id and file password, then the key has to be retrieved from both the shares which is obtained after visual cryptography through next level authentication. The decryption has to be performed on the information using the same key which is used for encryption. That is, the user has to pass almost five levels of authentication in order to retrieve the original file. Through multistage authentication process, the system can ensure complete security to the data. After decryption, using dictionary the data has to be detagged. The detagged information is actually the optimized file block, so that data have to be de-optimized inorder to generate the original file.

V. ANALYSIS

The proposed method has been evaluated by considering five unique text files of various sizes. This technique is efficient in providing higher reduction in size of the data by eliminating redundant data and it ensures higher level of security through multilevel authentication. An average of 38.316% of space can be saved through this method.

Table I shows the space required for five unique text files before and after data deduplication, along with the percentage of space saved through this method. If n users are uploading the same file, then for normal case, it requires n times the space for storing single copy, but by applying the proposed method, it does not require that much huge amount of storage space. It requires only the space for saving a single copy.

From this table, it is clear that using this method, we can reduce the storage space in a much better way and it is proved that even though the digital universe is growing, it is not necessary to have a proportionate growth is storage space.

TABLE I Space required before and after deduplication

File Name	File Size (in Bytes)	File Size After Tag Generation (in bytes)	% of space saved
FILE A.txt	188	77	59.04
FILE B.txt	366	248	32.24
FILE C.txt	546	342	37.36
FILE D.txt	631	450	28.68
FILE E.txt	715	470	34.26
Total Space	489.2	317.4	

Total space needed before deduplication = 489.2 bytes
Total space needed after deduplication = 317.4 bytes
Total space saved = 171.8 bytes

Figure 2 compares the size of distinct text files of different file sizes before and after performing data deduplication. The space required for data storage can be minimized based on the strength of the dictionary. The percentage of space that we can save is proportional to the strength of the dictionary and the number of users using same file.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 05, May - 2017

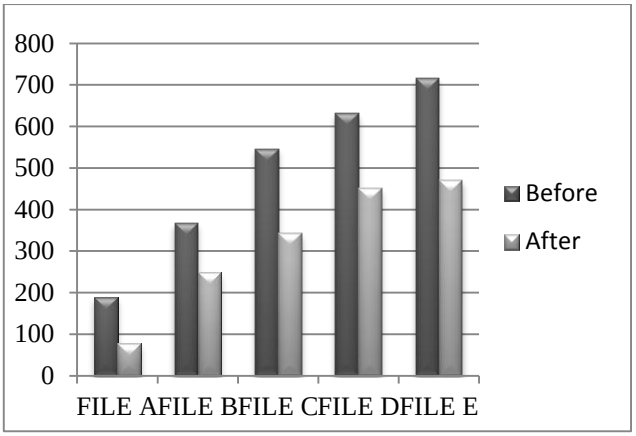

Fig.2.Comparison of file size before and after deduplication

VI. CONCLUSION

As a result of the worldwide use of digital gadgets like digital cameras, digital surveillance cameras, digital televisions there is a huge increase in the digital universe. Other fast-growing corners of the digital universe are those associated to Internet access, sensor-based IOT (Internet of things) applications, and activities supporting "cloud computing" and social networks, which consist of digital data created by many millions of online users. IT organizations must place a high priority on boosting the efficiency, reliability and security of their data storage system.

With the growth of digital data, the challenges associated are also increasing. Efficient data storage is one of the challenges faced with this huge growth. Data security and data compression are the basic requirements for an efficient storage system. An efficient data storage method is outlined here, which eliminates redundancy through data deduplication and provides higher level security through encryption. Data deduplication is a redundant data reduction technique which is basically used for large scale storage systems. Through this technique we are able to minimize the storage space by eliminating redundant data and demonstrated that even though there is an immense growth in digital data it is not necessary to have such a huge data storage space. To provide security, one of the ramp secret sharing methods, SHA 256, a convergent encryption technique, Advanced Encryption Standard (AES) algorithm and Visual Cryptography technique has been implemented. The proposed system achieves high level security since the encryption process goes through five levels of authentication.

The method is capable to optimize storage space and it ensure efficient data store by ensuring data security and key security.

REFERENCES

[1] Kodabagi M.M, M.V. Jerabandi and Nagaraj Gadagin,"Multilevel security and compression of text data using bit stuffing and huffman coding," Applied and Theoretical Computing and Communication Technology , International Conference on IEEE, 2015

[2] Deepu S.R, Bhaskar R, Shylaja B.S.,"Performance Comparison Of Deduplication Techniques For Storage In Cloud Computing Environment", Asian Journal of Computer Science And Information Technology, 2014, pp. 42-46

[3] Li, Jin, "Secure deduplication with efficient and reliable convergent key management," IEEE transactions on parallel and distributed systems 25.6, 2014, pp.1615-1625.

[4] Xia, W., Jiang H., Feng D., Douglis, F., Shilane, P., Hua, Y., Fu, M., Zhang, Y. and Zhou, Y., "A comprehensive study of the past, present, and future of data deduplication," Proceedings of the IEEE, 104(9), 2016 , pp.1681-1710.

[5] Kodituwakku S.R, U.S. Amarsinghe, "Comparison of lossless data compression algorithms for text data," Indian journal of computer science and engineering 1.4,2010, pp. 416-425

[6] Williams, Ross N., "An extremely fast Ziv-Lempel data compression algorithm," In Data Compression Conference, 1991. IEEE, 1991, pp. 362-371.

[7] Sklavos, Nicolas, Odysseas Koufopavlou, "On the hardware implementations of the SHA-2 (256, 384, 512) hash function," Circuits and Systems, 2003. ISCAS'03, Proceedings of the 2003 International Symposium, IEEE 2003, Vol. 5.

[8] Lu, Chih-Chung, and Shau-Yin Tseng, "Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter," Application-Specific Systems, Architectures and Processors, 2002. Proceedings, The IEEE International Conference, 2002, pp. 277-285.

[9] Zhang, Xinmiao, and Keshab K. Parhi, "Implementation approaches for the advanced encryption standard algorithm," IEEE Circuits and systems Magazine, 2002, pp. 24-46.