International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 04 | April - 2021 ISSN: 2582-3930

A NEWMANITY FOR SENSING AND SECURE DATA DEPOT IN CLOUD USING LOSSLESS COMPRESSION ALGORITHM

Dhivya S[1], Kaviya E[2], Logapriya A[3], Rajkumar V[4]

[123]Student, [4]Assistant Professor,

Department of Computer Science & Engineering, Krishnasamy College of Engineering and Technology,Cuddalore.

Abstract

Compression technique is adopted to solve various big data problems such as storage and transmission. The growth of cloud computing and smart phone industries has led to generation of the digital data. Digital data can be in various forms as audio,video,imagesand documents. Which data can be compressed using a cloud_assissted compressive sensing_based data gathering system by compression sensing with encryption in communication(CSEic).Along with increase the data size, the computation overhead for decoding becomes unaffordable on the user/decoder side and occur issue in security of ensuring the integrity of data storage in cloud. In this paper, we propose a lossless compression method which involves the combination of Run Length encoding is effective for finding to repeating characters, Huffman encoding algorithm is used to minimize the bit variable length code and Lempel-Ziv-Welch algorithm is used for compressing the file. By minimizing the number of bits for encoding will aid in improving the encoding efficiency and high compression with cost-effective. Meanwhile this model increase the compression ratio and reduce the average computational time for both compression and decompression is reduced without breaching security. Experimental results show that the proposed algorithm achieves the compression time 0.16 sec and decompression time 0.21 sec for 100kb file.

Keyword:Compressive sensing,Compression ratio, Data Compression,Data Decompression, Cloud assistance, Security, dupicity, Encryption and Decryption.

I.INTRODUCTION

Data compression is one of the enabling technologies for multimedia applications. It would not be practical to put images, audio and video on websites if do not use data compression algorithms. Mobile phones would not be able to provide communication clearly without data compression. With data compression techniques, we can reduce the consumption of resources, such as hard disk space or transmission bandwidth. Data Compression is the process of encoding data so that it takes less storage space or less transmission time. Compression is possible because most of the real world data is very redundant. In this survey, first we introduce the concept of lossy and lossless data compression techniques.

Classification of compression methods:-We have two types of compression methods:

Lossless compression: - It is used to reduce the amount of source information to be transmitted in such a way that when compressed information is decompressed, there is not any loss of information.

Lossy compression: - The aim of lossy compression is normally not to reproduce a exact copy of the information after decompression. In this case some information is lost after decompression

II. LITERATURE REVIEW

SUNG-HSIEN HSIEH[1,3], TSUNG-HSUAN HUNG[2,3], CHUN-SHIEN LU[3], YU-CHI CHEN[4], SOO-CHANG PEI[1], "A Secure Compressive Sensing-based Data Gathering System via Cloud Assistance", 10.1109/ACCESS.2018.2844184, IEEE Access [1]

In this system, involving three parties of sensor, cloud, and user, possesses several advantages. First, in terms of security, for any two data that are sparse in certain transformed domain, their corresponding ciphertexts are indistinguishable on the cloud side. Second, to avoid the communication bottleneck between the user and cloud, the sensor can encrypt data individually such that, once the cloud receives encrypted data from sensor, it can immediately carry out its task without requesting any information from the user. Third, we show that, even though the cloud knows the permuted support information of data, the security never is sacrificed. Meanwhile, the compression rate can be reduced further. Theoretical and empirical results demonstrate that our system is cost-effective and privacy guaranteed and that it possesses acceptable reconstruction quality.

Kasmeera K S[a], Shine P James[a], Sreekumar K, "Efficient Compression of Secured Images using Subservient Data and Huffman Coding" , Peer-review under responsibility of the organizing committee of RAEREST 2016[2]

This paper proposes a scheme of compressing encrypted data with the help of a subservient data and Huffman coding. For encrypting the original image, it is manipulated with a pseudorandom number sequence generated using a secret key. The subservient data is also created by the content owner. The encrypted data is then compressed using a quantization mechanism and Huffman codingAt the reconstruction side the principal content of the data is reconstructed. Experimental results show that the compression ratio distortion performance of this method is superior to the existing Techniques. The compression ratio of encrypted image is improved to the range 10 to 20

Saidhbi Sheik, ThirupathiRao Komati, "A Way to Secure the Data in Cloud Data Storage by Using Cloud Data Compression Mechanism", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7 Issue-4S2, December 2018[3]

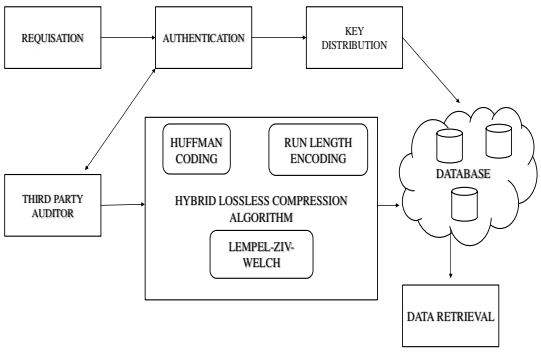
Cloud computing significantly plays a role in the aspect of effective resource utilization and service consumption. Irrespective of the type of clouds (ex. Private, public, hybrid or inter-cloud), every service providers concentrates on the data residing in cloud servers. Each and every moment, the researchers and scholars are proposing multiplicity of security algorithms to secure cloud data during the transactions. Most of the cloud data secure algorithms are focusing on the way to secure to cloud data in a single direction by using cryptographic algorithms. In this research paper focuses on a new direction to combine the features of data compression with the cloud data in order to secure the cloud data storage.

III.OUR CONTRIBUTION

In this paper, we propose a lossless compression method which involves the combination of Run Length encoding is effective for finding to repeating characters, Huffman encoding algorithm is used to minimize the bit variable length code and Lempel-Ziv-Welch algorithm is used for compressing the file. By minimizing the no.of.bits for encoding will aid in improving the encoding efficiency and high compression with cost-effective. Meanwhile this model increase the compression ratio and reduce the average computational time for both compression and decompression is reduced without

breaching security.

IV. SYSTEM ARCHITECTURE



V. PROPOSED SYSTEM

The main objective of the proposed work is to improve the storage capacity in cloud and secure the data in the cloud from the unauthorized users in order to achieve better throughput and end to end delay. Figure 1 demonstrates of the general architecture for the proposed work. The proposed framework is sub divided into three stages: Requisition phase, authentication phase and storage phase. The primary stage involves the requisition phase which uses the basis login and password requisition model. The secondary stage executes authentication and authorization, whereas the authorized user have separate key to access the data in order to do that an efficient method named efficient key management authentication algorithm. In the final stage the data are compressed and stored in the network using hybrid data compression.

A. Efficient key management authentication algorithm:

The authorized users from the requisition phase generate a separate key. For tracing each attribute users has the unique identity. These identity and user'sattributes are hiding from the users. Through this cannot learn anything from the cipher texts about the attributes matching or mismatching. The attributes are classified as the hidden normal attributes (HN) and the hidden identity attributes (HID).

1. **Setup the key:** This phase outputs the public key and the master key.
2. **Encryption:** Encrypt the message M with the set of attributes X, but the attributes are Xhide hidden.
3. **Key generation:** Key generation can be done by access structure as input and produces the output.
4. **Decryption:** Decryption can be done with decryption keys for each attributes of users.

TPA (Third party Auditor) is an entity, which has expertise and capabilities for Encryption and decryption Service. When client want to store data at the cloud storage at that time TPA (encryption/decryption service) Encrypt the data and return back to user for storage purpose.

For sensitive attributes the method chooses a hash function which verifies the identity of user with the help of TPA (Third party Auditor) and once the identity verification gets cleared then the access clearance is computed. When the user request clears both the service is fulfilled and the sensitive values are encrypted using the specific key which could be decrypted by the user. For non-sensitive attributes the method uses a public key based encryption which can be decrypted by the user.

B. Lossless Compression Methods:-

Run Length Encoding:

The first step in this technique is read file then it scans the file and find the repeating string of characters [6].when repeating characters found it will store those characters with the help of escape character followed by that character and count the binary number of items it is repeated. This method is useful for image having solid black pixels. This algorithm is also effective for repeating of characters. But it is not effective if data file has less repeating of characters. We can compress the run-length symbols using Huffman coding, arithmetic coding, or dictionary based methods.

Huffman Coding:

The Huffman coding algorithm is named after its inventor, David Huffman, who developed the method as a student in a class on information theory at MIT in 1950[1]. Huffman Coding Algorithm— it is a bottom-up approach 1. Initialization: Put the old nodes in a list sorted according to their frequency counts. 2. Repeat the following steps until the sorted list has only one node left: (1) From the list pick two nodes with the lowest frequency counts.

Form a Huffman sub tree that has these two nodes as child nodes and create a parent node. (2) Assign the sum of the children's frequency to the parent node and insert it into the list such that the order is maintained. (3) Delete the children from the sorted list. 3. Assign a 0 and 1codeword to the two branches of the tree on the path from the root. After the Huffman tree, the method creates a prefix code for each node from the alphabet by traversing the tree from the root to the node. It creates 0 for left node and 1 for a right node.

LZW (Lempel-Ziv Welch) compression method:-

LZW is the most popular method. This technique has been applied for data compression. The main steps for this technique are given below:- Firstly it will read the file and given a code to each character. If the same characters are found in a file then it will not assign the new code and then use the existing code from a dictionary. The process is continuous until the characters in a file are null.

Compression technique is mainly used to reduce the space of storage and increases the capacity of the resources. The data or information which occupies more space is compressed using a compressing technique (i.e) Lossless compression technique. Then the compressed data can again be decompressed to obtain the original information for future usage. This is mainly used to reduce the resources storage space and hence increase its productivity. In this section, we are going use the Lossless data compression technique where the data or information which is compressed to minimize its storage size does not undergo any loss of data or information. The lossless compression technique is highly secured. Our work comprises of the combination of LZW algorithm and run length encoding. The LZW algorithm is very fast and simple to implement but it has the limitation of compressing the file that contain repetitive data whereas this can be overcome by run length encoding.

Compression is achieved by taking each code from the file, and translating it through the code table to find what character or characters it represents. Codes 0-255 in the code table are always assigned to represent single bytes from the input file. For example, if only these first 256 codes were used, each byte in the original file would be converted into 12 bits in the LZW encoded file, resulting in a 50% larger file size. During compression, each 12 bit code would be translated via the code table back into the single bytes.

VI. PERFORMANCE ANALYSIS

A. COMPRESSION RATIO

Compression Ratio is the ratio between the size of the compressed file and the size of the source file.

$$\text{Compression ratio} = \frac{sizeaftercompression}{sizebeforecompression}$$

Table-1 Compression ratio between existing and proposed algorithm

Algorithm type	Compression ratio
LZW	0.67
Huffman encoding	0.32
Reference based compression	0.59
BAR algorithm	0.86
Proposed Hybrid data compression algorithm	0.932

B. COMPRESSION TIME:

Table-2 Compression time between existing and proposed algorithm

Algorithm type	**Average compression time for 100 kb**
LZW	0.25 sec
Huffman encoding	0.54 sec
Reference based compression	0.19 sec
BAR algorithm	0.43 sec
Proposed Hybrid data compression algorithm	0.152 sec

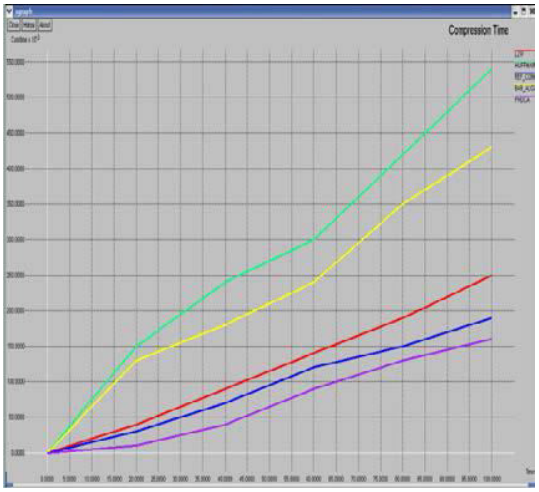


Figure 3 -Graphical representation of compression time

VII. CONCLUSION

Our project work as the data get compressed in a more optimized way of retrieving data from cloud by effective use of storage disks and reduce bandwidth, which studies the security issues of ensuring the integrity of data storage, minimize the network cost and reduce the duplicity problem. We model this design to increase the compression ratio of output and reduce the average computational time for both compression and decompression by without breaching security. The Experimental results demonstrate that our proposed method achieves the compression time as 0.16 sec and decompression time as 0.21 sec for 100kb file.

REFERENCE

[1] SUNG-HSIEN HSIEH[1,3], TSUNG-HSUAN HUNG[2,3], CHUN-SHIEN LU[3], YU-CHI CHEN[4], SOO-CHANG PEI[1], "A Secure Compressive Sensing-based Data Gathering System via Cloud Assistance", 10.1109/ACCESS.2018.2844184, IEEE Access.

[2] Saidhbi Sheik, ThirupathiRao Komati, "A Way to Secure the Data in Cloud Data Storage by Using Cloud Data Compression Mechanism", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7 Issue-4S2, December 2018

[3] K. MuthuLakshmi ,K.Lalitha and S.Uma , "An Efficient Data Compression And Storage Technique With Key Management Authentication In Cloud Computing", International Journal of Scientific & Engineering Research Volume 5, Issue 1, January-2018.

[4] C.C. Tan, Q. Liu, and J. Wu. "Secure locking for untrusted clouds in Cloud Computing (CLOUD)" , 2011 IEEE International Conference on, pages 131–138.

[5] Qian Wang, Cong Wang, Jin Li, KuiRen, and Wenjing Lou," Enabling Public Audit ability and Data Dynamics for Storage Security in Cloud Computing", IEEE computer society , Vol 22, No 5, May 2011.