AN EFFICIENT DATA COMPRESSION AND STORAGE TECHNIQUE WITH KEY MANAGEMENT AUTHENTICATION IN CLOUD COMPUTING

K. MuthuLakshmi [1] ,Associate Professor Member, *IEEE*, K.Lalitha [2] .Assistant Professor ,S.Uma [3] Associate Professor ,Panimalar Engineering College
akmuthube@gmail.com [1], lali.kalai@gmail.com [2] , umaokj@gmail.com [3]

Abstract

Large scale data processing is increasingly common in cloud computing systems like MapReduce, Hadoop, and Dryad in recent years. In these systems, files are split into many small blocks and all blocks are replicated over several servers. The high demand for data processing and leads to high computational requirement which is usually not available at the user's end Compression algorithms reduce the redundancy in data representation thus increasing effective data density. Data compression is a very useful technique that helps in reducing the size of text data and storing the same amount of data in relatively fewer bits resulting in reducing the data storage space, resource usage or transmission capacity.In this paper a hybrid data compression algorithm is propose which involves the combination of LZW and run length encoding. This hybrid algorithm increase the compression ratio and minimize the compression ,decompression time while comparing with existing algorithms.

Keyword: Data compression, LZW algorithm, run length encoding, Third party auditing, Key management authentication.

1. Introduction

Cloud storage become an important aspect in IT industry. Cloud storage is an application of cloud computing. It's the most developed part in cloud application. It depends on the cluster application, grid technology and distributed file system, and provides storage service to user through internet. In most conditions, Cloud storage can provide high reliability and security storage service at competitive price. The most developed cloud storage application is online backup or file-syncing. Disaster may happen at any time: fire, flood, tornado, hard driver failure. These disasters can destroy all local-stored data. User can use remote backup to protect their data from disasters. The other choice is to store copies of files in the cloud storage[1]. Online backup is an Internet based system that is set to automatically back up all selected files. These files are stored online, and can be accessed anywhere. Which is especially useful in case of local

computer or server gets lost or damaged. The benefit of using online storage services does not limited in protecting data. Cloud storage services make it easy to share files from different machines and mobiles.

The benefits of the cloud storage are flexible with reduced cost and they also manage the data loss risk and so on. Recently many work focus towards third party auditing and the remote integrity checking, providing the data dynamics. Remote archive service is responsible for properly preserving the data. The remote data integrity checking protocol detects the data corruption and misbehaving server in the cloud storage. In the proposed work Data partitioning technique, remote data integrity checking is analyzed in internal and external ways. Partitioning happens in alphabetical order by using of index method whereby the data being used is controlled.[2]

The security mechanism is also emphasized in order to prevent unrecoverable data loss. Storage and retrieval process are simplified by reducing the storage space when there is need to store and retrieved by merging technique. Cloud invisibly backs up the files and folders and elevates the potentially endless and costly search for extra storage space from music files to pictures to sensitive documents. Using cloud storage services means that you and others can access and share files across a range of devices and locations. Files such as photos and videos can sometimes be difficult to email if they are too large or you have a lot of them. We can quickly circulate a URL and can share the files with anyone we choose, which would mean uploading to the cloud storage.[3]

Management of data becomes paramount in maintaining service levels and securing the critical business information because all data in a cloud lives in the same shared system. Small business under cloud computing is also dependent on the reliability of our internet connection. But the internet outages make the most reliable cloud computing service providers to suffer now and again. It is also noteworthy that the premier cloud computing service providers suffer much outage. As the smaller organizations plan to move forward with cloud computing, outages which happen everywhere could have a profound impact on them [4]. Cloud computing enables to be excessively dependent on the internet. The availability of the robust and reliable internet for all the time is the premise on which the cloud computing exists. For the purpose of upholding the efficiency and effectiveness of cloud computing in data management even in the case of reduced bandwidth and irrespective of the device used to retrieve data, a feasible solution could be to compress the data on cloud. Data compression helps to reduce the consumption of expensive resources in this regard. For example, the disk space or transmission bandwidth.[5]

2.Literature survey:

Many researchers have proposed the use of compression in cloud computing which leads to effective use of storage disks and bandwidth in the cloud.Wang et al. [6] was the first to propose the scheme which can support public verification and fully dynamic data at the same time because previous studies only supported to modify and delete on a data file. They define public auditability which implies public verification is delegated by a trusted third party auditor (TPA) to verify. They propose a scheme to improve complex the file index information because this needs to consume a lot of computed resource.

Stanek et al. [7] The innovative encryption scheme which provides many different security of known and unknown data. For known information that are not especially delicate or sensitive, the traditional or classic ordinary encryption is performed. An alternate two-layered encryption plan with higher security while giving support to deduplication is proposed for unknown information. Along these lines, they accomplished better tradeoff between the proficiency and security of the outsourced information.

Xu et al. [8] additionally tended to the problem and demonstrated a protected convergent encryption for effective encryption, without considering problems of the block level deduplication and key-management. There are likewise different implementations of convergent encryption for secure deduplication. It is realized that some business cloud storage suppliers, for example, Bitcasa, likewise send convergent encryption.

Wang et al. [9] proposed a privacy protection scheme which is considered user's data privacy in the public auditability. Data privacy implies personally identifiable information or sensitive information whether they can be shared with third parties. As far as users are concerned what they depend on TPA just for the outsourced storage security of their data.

However, most studies do not consider the protection of clients' private information in the auditing phase. This is a serious problem because an auditor may leak information without the client's authorization. Besides, there are legal regulations, such as the Health Insurance Portability and Accountability Act (HIPPA), it guarantees patient confidentiality for all healthcare-related data and demands the outsourced data not to be leaked to external parties.

In [10] proposed an architecture that ensures the privacy of data stored in cloud storage. The proposed architecture can directly applicable to existing clouds without any modifications or any changes in cloud database. It can be process that connects directly to an encrypted cloud database without an intermediate

devices or systems with geographically distributed clients and it also allowed executing independent and operations including those changing the database structure. Moreover the proposed architecture removes intermediate proxies that limit the scalability, elasticity and availability properties that are intrinsic in cloud-based solutions.

Khobragade P. B. et al [11] reviewed a number of compression techniques for compressing image such as Lempel-Ziv-Welch, Huffman coding, Run Length encoding,, compression based on discrete cosine transform, discrete wavelet transform (DWT), integer wavelet transform (IWT) based compression and concluded that integer wavelet transform based compression techniques along with lifting scheme gives better compression ratio and retains quality of data.

5. Mukherjee, Tilak et al [12] proposed lossless compression approach using wavelets in which integer wavelet transforms are used and wavelet coefficients are converted to integer values and lifting scheme is proposed for obtaining better peak signal to noise ratio and less time execution for compression, hence resulted in good compression results.

3. Research methodology:

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 datas are compressed and stored in the network using hybrid data compression.

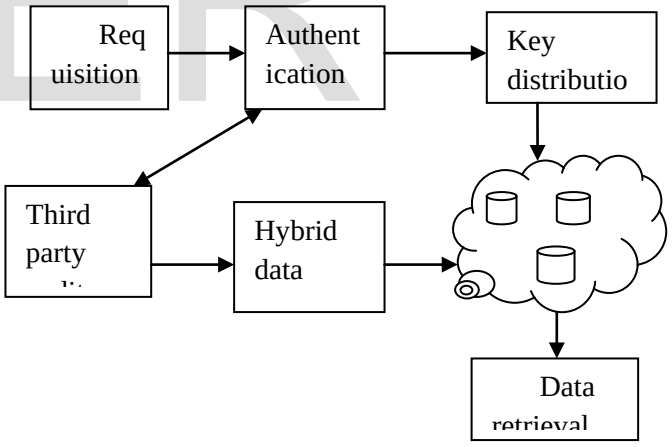

Figure 1. System architecture of proposed methodology

3.1 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's

attributes 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).

A.Setup the key: This phase outputs the public key and the master key.

B.Encryption: Encrypt the message M with the set of attributes X, but the attributes are Xhide hidden.

C.Key generation: Key generation can be done by access structure as input and produces the output.

D.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.

3.2 Key distribution centre:

A Distributed Key Distribution Center (DKDC, for short) is a set of n servers of a network that jointly realizes the same function as a KDC. In this setting, users have secure point-to-point channels with all servers. A user who needs to communicate with other users securely, sends a key-request message to a subset at his choice of at least k out of the n servers[13]. With this approach, the concentration of secrets and the slow down factor which arise in a network with a single KDC are removed. A single server by itself does not know the secret keys, since they are shared between the n servers. Moreover, each user can send a key-request in parallel to different servers. Hence, there is no loss of time in computing a key, compared with a centralized setting. Finally, the users can obtain the keys they need even if they are unable to contact some of the servers.[14]

3.3 Hybrid data compression algorithm:

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.

In this case, the encoded data consists entirely of 12 bit codes, each referring to one of the entries in the code table. In the encoding process, the cumulative probabilities are calculated and the range is created in the beginning. While reading the source character by character, the corresponding range of the character within the cumulative probability range is selected. Then the selected range is divided into sub parts according to the probabilities of the alphabet. Then the next data is read and the corresponding sub range is selected. In this way, datas are read repeatedly until the end of the file is encountered. Finally a number should be taken from the final sub range as the output of the encoding process. This will be a fraction in that sub range. Therefore, the entire source message can be represented using a fraction. To decode the encoded message, the number of characters of the source message and the probability/frequency distribution are needed.

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.

4. Performance analysis

(A) Compression ratio

Compression Ratio is the ratio between the size of the compressed file and the size of the source file.[15]

$$\text{Compression ratio} = \frac{size\ after\ compression}{size\ before\ compression}$$

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

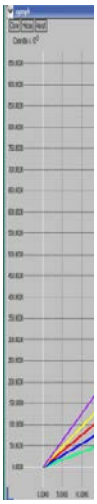

Figure 2 -Graphical representation of compression ratio

(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

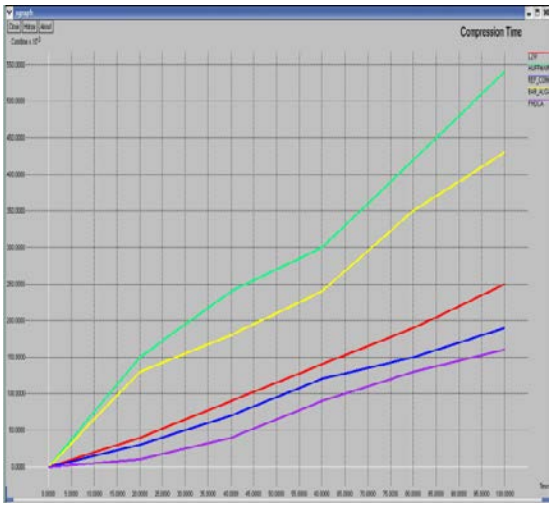

Figure 3 -Graphical representation of compression time

(b) De-compression time:

Table-3 Decompression time between existing and proposed algorithm

Algorithm type	Average de-Compression time for 100 kb
LZW	0.380 sec
Huffman encoding	0.54 sec
Reference based compression	0.57 sec
BAR algorithm	0.64 sec
Proposed Hybrid data compression algorithm	0.21 sec

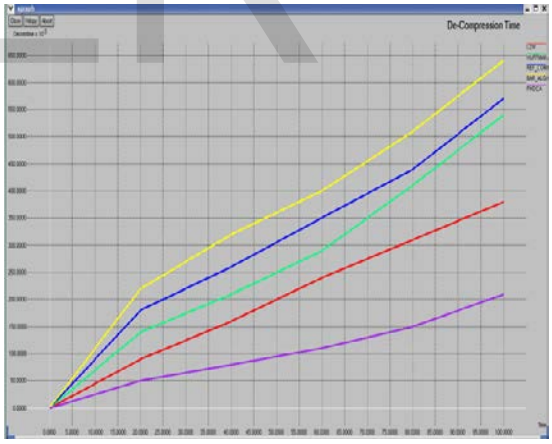

Figure 4 -Graphical representation of decompression time

(e) Packet loss rate:

Table-5 PLR between existing and proposed algorithm

Algorithm type	Packet loss ratio
LZW	0.97

Huffman encoding	0.86
Reference based compression	0.58
BAR algorithm	0.23
Proposed Hybrid data compression algorithm	0.15

Figure 5 -Graphical representation of Packet Loss rate (PLR)

5. Conclusion:

This paper work studies the security issues of ensuring the integrity of data storage in Cloud Computing. We outline the challenges associated with the retrieval of data from cloud in an appropriate manner. As the data gets compressed, it leads to a more optimized way of retrieving data from cloud. The use of compression in cloud computing leads to effective use of storage disks and bandwidth. This work enables the user to fine-tune the trade-off between storage costs, computation time and bandwidth costs. Different computations of characters can be represented by fewer numbers of bits in compression, which is an efficient way of retrieving data in the cloud environment. As a result the proposed algorithm achieves the compression ratio about 0.932 with the compression time 0.16 sec and decompression time 0.21 sec for 100 kb file.

References:

[1] 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. IEEE, 2011.

[2] Qian Wang, Cong Wang, Jin Li, Kui Ren, 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

[3] Dr. Anil G.N, Mrs. Swetha M.S & Mr. Muneshwara M.S "A Smarter Way of Securing and Managing Data for Cloud Storage Applications Using High Throughput Compression in the Cloud Environment" IJARCSMS Volume 2, Issue 9, September 2014.

[4] "PPM performance with BWT Complexity: A fast and effective data compression algorithm", M. Effros,Proceedings of the IEEE, 88(11), 1703-1712, (2000).

[5] Yuan, Jiawei, and Shucheng Yu. "Secure and constant cost public cloud storage auditing with deduplication." Communications and Network Security (CNS), 2013 IEEE Conference on. IEEE, 2013.

[6] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 847–859, 2011.

[7] Stanek, Jan, et al. A secure data deduplication scheme for cloud storage. Technical Report, 2012.

[8] Li, Jin, et al. "Secure deduplication with efficient and reliable convergent key management." (2013) 1-1

[9] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy-preserving public auditing for secure cloud storage,” IEEE Transactions on Computers, vol. 62, no. 2, pp. 362–375, 2014.

[10]. L. Ferretti, M. Colajanni, M. Marchetti. Distributed, concurrent, and independent access to encrypted cloud databases. IEEE transactions on parallel and distributed systems, 2014; 25(2), 437-446.

[11] Khobragade, P. B., and S. S. Thakare. "Image Compression Techniques-A Review." International Journal of Computer Science and Information Technologies (IJCSIT) 5.1 (20145272-275

[12] Mukherjee, Tilak, and M. Koteswara Rao. "Efficient Performance of Lifting Scheme Along With Integer Wavelet Transform In Image Compression."International Journal of Engineering Research and Applications (IJERA) 3.4 (2016): 1950-1953.

[13] S.S.M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, “Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions,” Proc. ACM Conf. Computer and Comm. Security,pp. 152-161, 2010.

[14] Cheng-Kang Chu ,Sherman S.M. Chow ,Wen-Guey Tzeng , Jianying Zhou, and Robert H. Deng “Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage ” , IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 2, February 2014 .

[15]Gurjeevan Singh, Ashwani Kumar Singla, K.S.Sandha, “Through Put Analysis of Various Encryption Algorithms”, IJCST Vol.2, Issue3, September 2011.