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

[1]S.Naveen kumar, [2]Dr.K.Nirmala
[1]Research scholar, university of madras, Chennai, India,
[2]Research supervisor, Quaid-E-Millath Government College of women, Chennai, India,

Abstract

Extensive scale data handling is progressively normal in distributed computing frameworks like MapReduce, Hadoop, and Dryad lately. In these frameworks, documents are part into numerous little squares and all squares are repeated more than a few servers. The appeal for data preparing and prompts high computational necessity which is typically not accessible at the client's end Compression calculations diminish the repetition in data portrayal subsequently expanding compelling data thickness. Data compression is an extremely valuable system that aides in diminishing the extent of content data and putting away a similar measure of data in moderately less bits bringing about decreasing the data storage room, asset utilization or transmission limit. In this paper a half breed data compression calculation is proposes which includes the blend of LZW and run length encoding. This cross breed calculation increment the compression proportion and limit the compression, decompression time while contrasting and existing calculations.

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

1. Introduction

Cloud storage turn into a critical viewpoint in IT industry. Cloud storage is a use of cloud figuring. It's the most grown part in cloud application. It relies upon the bunch application, network innovation and disseminated document framework, and gives storage administration to client through web. In many conditions, Cloud storage can give high unwavering quality and security storage benefit at focused cost. The most created cloud storage application is online reinforcement or record matching up. Catastrophe may occur whenever: fire, surge, tornado, hard driver disappointment. These fiascos can obliterate all nearby put away data. Client can utilize remote reinforcement to shield their data from catastrophes. The other decision is to store duplicates of records in the cloud storage[1].

Online reinforcement is an Internet based framework that is set to naturally back up every single chosen document. These documents are put away online, and can be gotten to anyplace. Or, in other words in the event of nearby PC or server gets lost or harmed. The advantage of utilizing online storage administrations does not restricted in securing data. Cloud storage administrations make it simple to share documents from various machines and mobiles. The advantages of the cloud storage are adaptable with lessened expense and they likewise deal with the data misfortune chance et cetera. As of late many work center towards outsider reviewing and the remote honesty checking, giving the data elements. Remote document benefit is in charge of appropriately protecting the data. The remote data uprightness checking convention identifies the data defilement and getting out of hand server in the cloud storage. In the proposed work Data parceling procedure, remote data honesty checking is dissected in inward and outer ways. Parceling occurs in sequential order arrange by utilizing of file technique whereby the data being utilized is controlled [2].

The security system is additionally accentuated with the end goal to avert unrecoverable data misfortune. Storage and recovery process are streamlined by decreasing the storage space when there is have to store and recovered by blending method. Cloud undetectably backs up the records and organizers and hoists the conceivably perpetual and expensive scan for additional storage space from music records to pictures to touchy reports. Utilizing cloud storage administrations implies that you and others can access and offer records over a scope of gadgets and areas. Documents, for example, photographs and recordings can some of the time be hard to email on the off chance that they are too expansive or you have a considerable measure of them. We can rapidly circle a URL and can impart the records to anybody we pick, which would mean transferring to the cloud storage.[3]

Administration of data winds up fundamental in keeping up administration levels and anchoring the basic business data since all data in a cloud lives in the equivalent shared framework. Private venture under cloud figuring is likewise subject to the unwavering quality of our internet association. However, the internet blackouts make the most solid cloud processing specialist organizations to endure occasionally. It is likewise imperative that the head cloud figuring specialist organizations endure much blackout. As the littler associations intend to push ahead with cloud registering, blackouts which happen wherever could profoundly affect them [4].

Cloud computing empowers to be too much subject to the internet. The accessibility of the vigorous and solid internet for all the time is the start on which the cloud figuring exists. To uphold the proficiency and adequacy of cloud registering in data administration even on account of diminished transmission capacity and independent of the gadget used to recover data, a practical arrangement could be to pack the data on cloud. Data compression decreases the utilization of costly assets in such manner. For instance, the plate space or transmission bandwidth.[5]

2.Literature survey:

Numerous analysts have proposed the utilization of compression in cloud registering which prompts powerful utilization of storage plates and transmission capacity in the cloud.Wang et al. [6] was the first to propose the plan which can bolster open check and completely unique data in the meantime in light of the fact that past examinations just upheld to adjust and erase on a data record. They characterize open auditability which infers open confirmation is designated by a confided in outsider reviewer (TPA) to check. They propose a plan to enhance complex the record file data since this needs to expend a considerable measure of processed asset.

Stanek et al. [7] The imaginative encryption conspire which gives a wide range of security of known and obscure data. For known data that are not particularly fragile or touchy, the conventional or great standard encryption is performed. A substitute two-layered encryption plan with higher security while offering backing to deduplication is proposed for obscure data. Thusly, they achieved better tradeoff between the capability and security of the re-appropriated data.

Xu et al. [8] moreover kept an eye on the issue and showed a secured joined encryption for compelling encryption, without considering issues of the square level deduplication and key-administration. There are in like manner distinctive usage of focalized encryption for secure deduplication. It is understood that some business cloud storage providers, for instance, Bitcasa, in like manner send joined encryption.

Wang et al. [9] proposed a security insurance plot which is viewed as client's data protection in general society auditability. Data security suggests by and by identifiable data or delicate data whether they can be imparted to outsiders. To the extent clients are concerned what they rely upon TPA only for the re-appropriated storage security of their data. Be that as it may, most examinations don't consider the security of customers' private data in the inspecting stage. This is a significant issue in light of the fact that an evaluator may spill data without the customer's approval. Additionally, there are legitimate controls, for example, the Health Insurance Portability and Accountability Act (HIPPA), it ensures quiet classification for all human services related data and requests the redistributed data not to be spilled to outer gatherings.

In [10] proposed an engineering that guarantees the protection of data put away in cloud storage. The proposed engineering can specifically appropriate to existing clouds with no alterations or any adjustments in cloud database. It very well may be process that associates straightforwardly to a scrambled cloud database without a middle gadgets or frameworks with geologically dispersed customers and it additionally permitted executing autonomous and activities including those changing the database structure. Besides the proposed engineering evacuates middle of the road intermediaries that farthest point the versatility, flexibility and accessibility properties that are inherent in cloud-based arrangements.

Khobragade P. B. et al [11] evaluated various compression systems for packing picture, for example, Lempel-Ziv-Welch, Huffman coding, Run Length encoding,, compression dependent on discrete cosine change, discrete wavelet change (DWT), whole number wavelet change (IWT) based compression and reasoned that whole number wavelet change based compression strategies alongside lifting plan gives better compression proportion and holds nature of data.

Mukherjee, Tilak et al [12] proposed lossless compression approach utilizing wavelets in which whole number wavelet changes are utilized and wavelet coefficients are changed over to number qualities and lifting plan is proposed for getting better pinnacle flag to clamor proportion and less time execution for compression, thus brought about great 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.

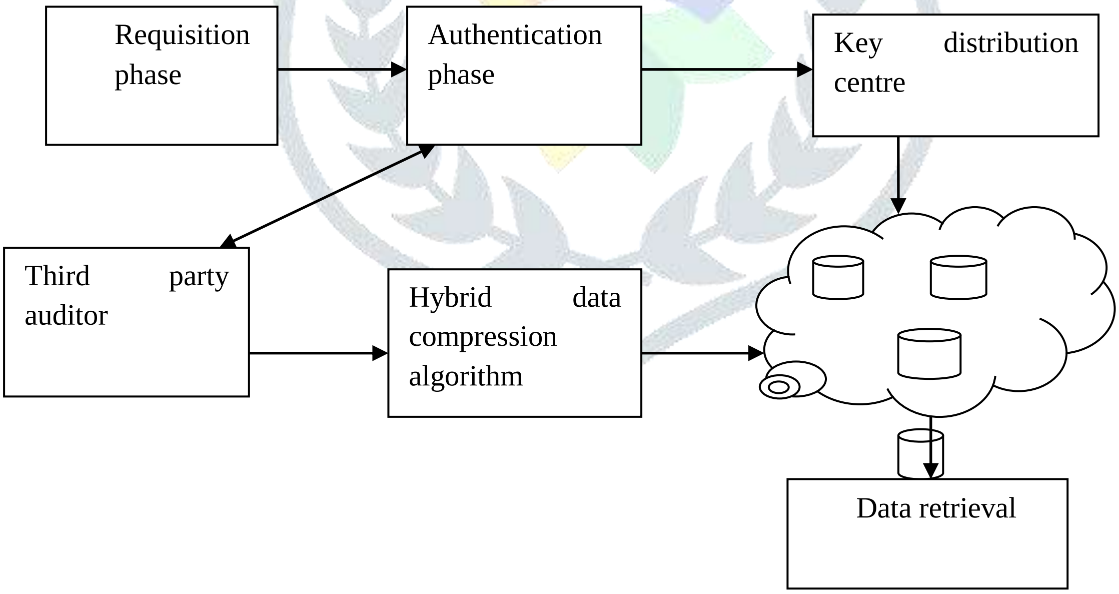

Figure 1. System architecture of proposed methodology

3.1 Efficient key management authentication algorithm:

The approved clients from the demand stage produce a different key. For following each quality clients has the one of a kind personality. These personality and client's traits are escaping the clients. Through this can't take in anything from the figure messages about the traits coordinating or crisscrossing. The characteristics are named the shrouded typical qualities (HN) and the concealed personality properties.

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 gathering Auditor) is a substance, which has mastery and capacities for Encryption and decoding Service. At the point when customer need to store data at the cloud storage around then TPA (encryption/decoding administration) Encrypt the data and return back to client for storage reason.For delicate traits the strategy picks a hash work which confirms the personality of client with the assistance of TPA (Third gathering Auditor) and once the character check gets cleared then the entrance leeway is figured. At the point when the client ask for clears both the administration is satisfied and the touchy qualities are scrambled utilizing the particular key which could be decoded by the client. For non-delicate characteristics the technique utilizes an open key based encryption which can be decoded by the client.

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

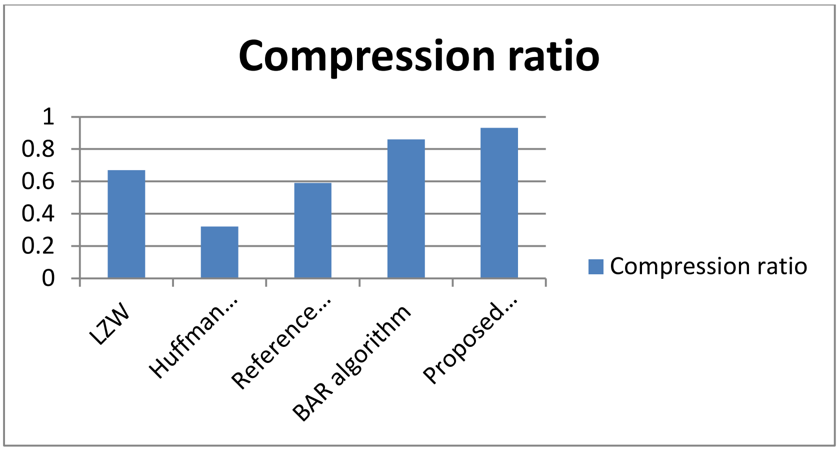

Figure 2 -Graphical representation of compression ratio

(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	0.16 sec

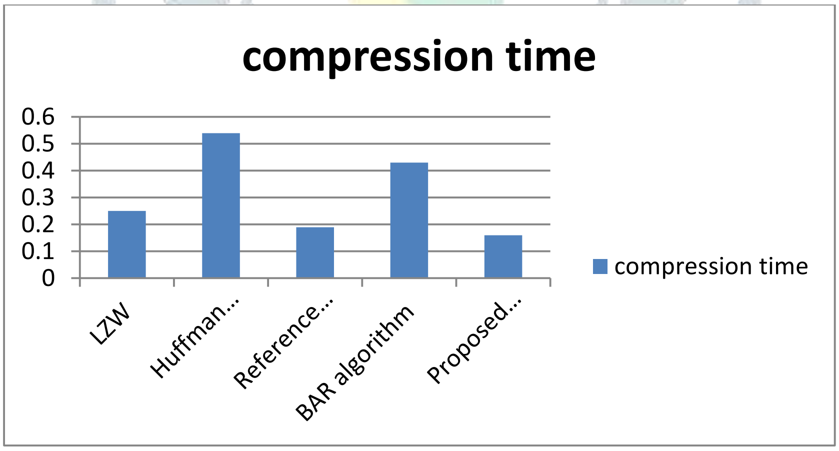

Figure 3 -Graphical representation of compression time

(b) De-compression time:

Table-3 Decompression time between existing and proposed algorithm

Algorithm type	Average Compression time for 100 kb
LZW	0.38 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

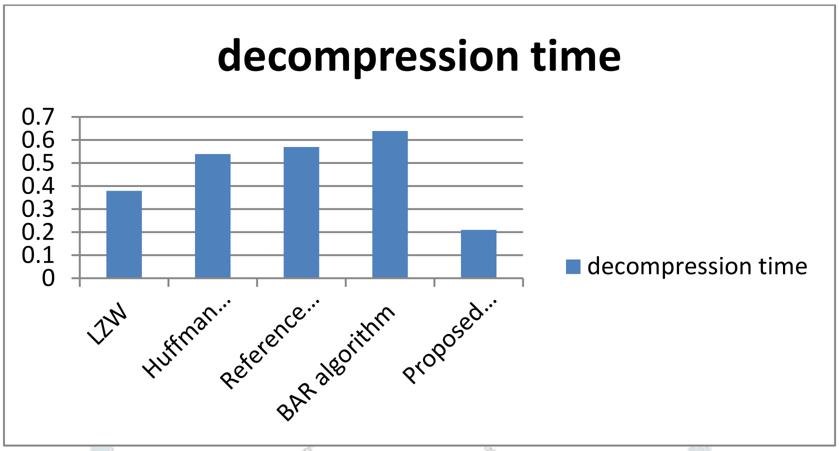

Figure 4 -Graphical representation of decompression time

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 disk 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.