A brief Survey on Data Integrity and Compression in Cloud Computing

Abhijit Choudhury
PG Student
Department of CSE, SMIT,
East Sikkim, India

Santanu Kumar Misra
Associate Professor
Department of CSE, SMIT,
East Sikkim, India

Bijoyeta Roy
Assistant Professor
Department of CSE,SMIT
East Sikkim, India

ABSTRACT
Cloud computing is an emerging technology that can allow user's large amount of data to store in cloud and it can be access from anywhere. It provides various kinds of services to users such as Software as a service, Infrastructure as a service, and Platform as a service etc. It can manage to provide cost effective, easy to manage, elastic, and powerful resources over the Internet. It uses optimal and shared utilization to enhance the ability of the hardware resources .These features help user and organizations to switch their applications and services to the cloud. There are two important aspects of cloud storage- (1) Data integrity is an important aspect while attain validation, violation and correctness of data on cloud storage. Data integrity checking mechanism plays a major role in Cloud Computing. This deals with checking the integrity of data at remote cloud storage server. It ensures that data at the sender and receiver side are same. User can detect data integrity violations with the help of this mechanism while retrieving from remote cloud storage server. It refers to the completeness, accuracy and consistency of data over its entire life cycle. This can be determined by the absence of alteration between two instances of data.(2)- Data compression is a process of eliminating redundancies in order to reduce storage space and cost on cloud storage .It implies sending & storing smaller number of bits. It involves manipulating and modifying bits structure of data in such a way that it reduce size. This paper propose a new methodology to improve the performance of data integrity checking and to optimize the compression ratio.

General Terms
Compression ratio, Cloud Computing, Dictionary, Index

Keywords
Data integrity checking, Auditing Mechanism

1. INTRODUCTION
With the rapid growth of processing and storage technologies and the revolution of the Internet, computing resources have become cheaper and more powerful than ever before. This technological trend has enabled the realization of a new computing model cloud computing model [3, 5]. It refers to the delivery of computing resources over the Internet. Instead of storing data on hard drive or updating applications for needs,

Use a service over the Internet, at another location, to store information or use its applications. It is one type of Internet-

Based computing, where different services — such as servers, storage and applications are delivered to computer and devices

Through the Internet. The cloud computing has five essential [3] characteristics are presented below:

1.1 Characteristics
1.1.1 On-demand self-service
Internet based services such as email, applications and networks service can be provided automatically to customer for the purpose of managing and requesting the usage of services to Cloud service provider(CSP) without requiring human interaction. The entire process works through web services and management interfaces.

1.1.2 Broad network access
Cloud capabilities supports services, data and customer's applications. Capabilities of cloud are available on entire the network and standard protocol has been introduced for accessing cloud services which are promoted by independent platforms (e.g., mobile phones, tablets, laptops, and workstations).

1.1.3 Resource pooling
Resource pooling is a process of sharing cloud resources among multiple customers to serve in a multi-tenant model. Transparency defined on location of resources that are accessed by customers.

1.1.4 Rapid elasticity
Sometimes cloud service can provide resources for scaling out and releasing .Capabilities which are available provided by cloud are unlimited in nature and it can be purchase based on customer demand at any time in anywhere.

1.1.5 Measured service
The resources of cloud computing usage are measured and controlled for the betterment of transparency between provider and customer. It has metering capability which allows to optimize the usage of resources as per customer demands. Services provided to customer are charged as per usage metrics and generated a bill.

Maintaining data integrity [4, 9] is one of the major challenges in cloud computing because the user has no control over the security mechanisms that are used to protect the data. It is one of the important fundamental component for securing data .It can protect data in case of loss and damage due to hardware and software failure. Inconsistency and inaccuracy of data can occur by malicious attacks. The term data integrity refers to state, process and function. It applies on the field of data quality. It provides an assurance to the user that the data is not modified or corrupted by the cloud service provider or other users. The data will be stored in the cloud by a user and the integrity of the data will be checked by Auditing mechanism

Data integrity performance can be measured based on the error detection rate. Apart from error detection rate, some data integrity mechanism also use for correcting data. In the field

of data security, data integrity is an essential component. Nowadays security is a major concern in cloud computing. So for that reason data integrity mechanism needs to be improved. Data integrity requires because of growing demand of cloud in recent technological trends. The efficiency of data integrity is measured using the parameters like time for processing the data, storage cost, and memory for storage. For assuring data integrity which includes Data encryption, which encrypt and decrypt data by cipher, Data backup, which stores a copy of data in remote location, Access controls, permission to access of data, Data validation, to certify uncorrupted transmission and Using Error detection and correction of data when transmitting data. The scope of the Data Integrity assurance [2] mechanism can be classified into two levels: To prevent data corruption, to detect and correct data violation.

With the rapid growth of digital information, the cost of storage infrastructure, management cost and storage space also increases. Therefore it becomes a serious concern to reduce the large amount of data in order to be transferred, stored, and managed in cloud storage systems. People tends to store a lot of files inside theirs storage system which leads to waste of hardware resources and increases complexity of data center that will near future degrades the performance of cloud storage. When the storage reaches it limit, they then try to reduce those files size to minimum by using data compression methods.

Data compression [11] is a method to reduce storage space by eliminating redundancies that occurs in most files. It is not only use for optimizing the limited storage space but also helpful in save time and optimal usage of resources which are insufficient in recent days. There are two types of compression, lossy and lossless.Lossy compression reduced file size by eliminating unwanted data after decoding, this is often used by video and audio compression. On the other hand, Lossless compression, manipulates each bit of data to minimize the size without losing any data after decoding. Data compression can be used for network processing technique to save energy. This paper surveys various data integrity methods to check storage correctness and analyze various well known data compression methods to achieve better compression ratio. After surveying and analyzing, propose a methodology which is a combination of efficient data integrity checking method and data compression method in cloud computing .This methodology provides efficient method for data integrity assurance as well as produces better compression ratio.

The rest of the paper is organized as follows. Section2 discuss related works, Section3 describes the proposed methodology and Section4 describes by conclusion.

2. RELATED WORKS

2.1 For Data Integrity

G. Sivathanu [10] et.al talks about Mirroring Technique which can detect integrity violations caused by data corruption, but cannot help in recovering from the data. Recovery is possible using rules of mirroring is 3-way or more. A malicious user who can easily copies of the data, unless the location of the copies is maintained in a confidential manner. It also cannot detect integrity violations caused by user errors, because in most cases user modifications are carried out in all mirrors. It is also known as data replication method. It maintains two or more copies of the original data. Integrity check is made by comparing these copies and if any differences exists, it will indicate a possible corrupted data. There are many weaknesses with this technique. If the original and mirrored data have the same modifications, it cannot detect the data corruption. If a malicious users inserts the same value into the Original and mirrored data, the it cannot detect the integrity violation, and the inserted data is considered as part of the original data.

J. A. Ghaeb [9] et.al. Discusses on checksum method, it is an error detection mechanism which is created by "summing up" all the bytes or words in a data word to create a checksum value, which is added to the transmitted data word. The checksum of the retrieved data word is computed and compared with the received checksum value. If the computed and received checksum are not matched, then the data is corrupted during transmission .It can be possible that some altered bits in the transmitted word caused by an erroneous data word matching the transmitted checksum value. In addition, if there is a corruption in retrieved data, the It does not provide information about which components in the data are corrupted. There are many types of checksum techniques. The simplest checksums involves a "sum" function apply on all bytes or words in a data word. Although, there are at least three types of error which cannot be detected by this method. The first one while the data is reordering. Second is when zero values are inserted in the data or deleted from it. The third can occur if multiples errors amount to zero. Three other commonly used simple "sum" functions are XOR two's complement addition, and one's complement addition. These type of checksums provide fairly weak error detection coverage, but holds very low computational cost.

P.Premkumar [4] et.al. Discusses Hamming code method that can be employed as an error detection mechanism. It depends on parity bit. A code word is created by adding the original data bits and the check bits to form Hamming Code word. These check bits are assisting in detecting data integrity violation. It is acquired by XORing the weights of bits. The new check bits of the newly code word are recomputed and XORed with the received check bits. If the result is zero, the data has no corruption during the transmission .It has many flaws. If the original data bits are reordered or if the check bits are modified in a way to give zero XORing with the new check bits at receiver, this method cannot detect the data corruption. It is determined by binary word. For 15-bit representation, 4-bits of them are reserved for the check bits. This will increase the frame length of data and increase the data size and transmission cost.

P.Premkumar [4] et.al. Describes a hash function is any function that can be used to map data of arbitrary size to data of fixed size. It takes an input or message and returns a fixed size string, which is called the hash value also called as a message digest, a digital fingerprint, a digest or a checksum. The hash functions like as encryption file systems which provide some degree of integrity assurance but they do not detect the data violation. The malicious users can easily modify the hash codes of data that leads to data violation in the original data. For a large number of data, it is impossible to get a unique hash code. Hashing is one form of cryptographic security which differs from encryption, Whereas encryption is a two-step process that can help to first encrypt and then decrypt a message, hashing shorten a message into an fixed-length value or hash. Two of the most common hashing algorithms are seen in networking are MD (Message Digest) 5 and SHA (Secure Hash Algorithm). Hashing is used for verify the data; the original message cannot get from a hash.

2.2 For data compression

Nelson [12] et.al describes Run-length encoding (RLE) is one

of basic technique for lossless data compression. The idea behind this method is: If data item d occur n successive times in the input stream, replace the n occurrences with the pair nd. RLE is mainly used for compress runs of the same byte. This method is useful when repetition often occurs inside data.

Nelson [12] et.al.describes Burrows-wheeler transform (BWT) works in block mode while mostly others work in streaming mode. This algorithm classified into transformation algorithm because the main idea behind is to rearrange (by adding and sorting) and concentrate symbols. These concentrated symbols can be used as input for another algorithm to achieve good compression ratios.

Salomon [11] et.al discusses Move to front transform (MTF) is a transformation algorithm which is one of the main basic technique for data compression. It does not compress data but can help to reduce redundancy. The basic idea is to move to front the symbols that mostly occur, so these symbols will have smaller output number. This technique is to be used as optimization for other algorithm likes Burrows-wheeler transform.

Salomon [11] et.al.discusses Arithmetic coding (ARI) is using statistical method to compress data. It starts with a certain interval, it reads the input file symbol one by one and uses the probability of each symbol narrowing the interval. To specify a narrower interval requires more bits, then the number constructed by the algorithm growing continuously. To achieve compression, this algorithm is designed in such a way that a high-probability symbol narrows the interval less than a low-probability symbol, with the result that high-probability symbols contribute fewer bits to the output.

Salomon[11] et.al talks about Huffman coding method starts by constructing a list of all the alphabets in descending order of their probabilities after that constructs a tree, with a symbol at every leaf, from the bottom up approach. At each step the two symbols with smallest probabilities are selected, added to the top of the partial tree, deleted from the list, and exchanged with an auxiliary symbol representing both of them. When the list is minimized to just one auxiliary symbol, the tree is complete. The tree is then travel across to determine the codes of the symbols.

Salomon [11] et.al. Describes LZ77 algorithms achieve compression by replacing repeated occurrences of data with references to a single copy of data which is existing earlier in the input data stream. A match is encoded by a pair of numbers called a length distance pair. To spot matches, the encoder must keep track of some amount of the most recently data. The structure in which data is maintain is called a sliding window that is why LZ77 is sometimes called sliding window compression.

Salomon [11] et.al. Describes LZ78 algorithm, in this algorithm to achieve compression by replacing repeated occurrences of data with references to a dictionary that is constructed based upon the input data stream. Each dictionary entry is of the form dictionary [...] = {index, character}, where index is the index to a previous dictionary entry, and character is appended to the string represented by dictionary [index].

3. PROPOSED METHODOLOGY

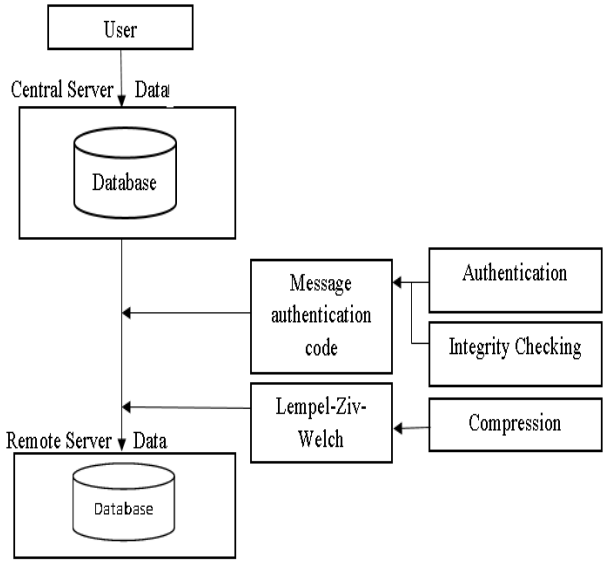

Fig 1: Block diagram of proposed methodology

3.1 Message Authentication Code (MAC)

In this method, it can can be assumed that the file consists of a set of blocks m1,m2.One of the simplest way to ensure the data integrity is to recomputed MACs [1, 2] for the entire file. It is one of the important cryptographic checksum technique which uses MAC algorithm with symmetric key to provide authentication for securing message. For establishing MAC process, sender can send message with secret key to MAC algorithm which produces MAC value and on the receiver side, after receiving message with MAC value, receiver compute Mac value with the help of algorithm and shared key. If Mac value is matched on both sides then it proves that message is authenticated as well as integrity is checked. There are several types of algorithms used in MAC.Most commonly used algorithm is HMAC (Hash Message Authentication Code).MAC uses same key on both sender and receiver side and secret keys as done in case of encryption also.

Let us now try to understand the entire process in detail –

1. MAC function compresses an arbitrary long input into a fixed length output.
2. The sender uses some publicly known MAC algorithm, inputs the message and the secret key K and produces a MAC value.
3. The sender forwards the message along with the MAC.
4. On receipt of the message and the MAC, the receiver feeds the received message and the shared secret key K into the MAC algorithm and re-computes the MAC value.
5. If the computed MAC does not match the MAC sent by the sender, the receiver cannot determine whether it is the message that has been altered or it is the origin that has been falsified.
6. The receiver now checks equality of freshly computed MAC with the MAC received from the sender. If they match, then the receiver accepts the message and assures himself that the message has been sent by the intended sender.

3.2 Lempel-Ziv-Welch (LZW)

It is a universal lossless data compression algorithm created by Abraham Lempel, Jacob Ziv, and Terry Welch.LZW[6,7] compression maintains table which contains two attributes-Dictionary and Index. Table can be used for encoding. Dictionary has list of possible characters and index has list of serial numbers which is assigned according to input characters entered in table. Dictionary can be updated if dictionary does not match with result of addition operation. It generally performs best on files with repeated substrings, such as text files.

3.2.1 Principles of the LZW algorithm [8]:

The LZW algorithm's coding steps are as follows:

Step1: Initialize dictionary. Dictionary contains all single characters in the data stream.

Step2: Set the Prefix P Null.

Step3: Read the next character in the data stream as the current character C.

Step4: Judge whether the string P + C is in the current dictionary.

1) Yes, set P = P + C, that is extending P with C.
2) No.

①output P's corresponding code to the encoded data stream.

②Judge whether the dictionary achieve to the maximum capacity:

If it doesn't, add the string P + C to the dictionary, otherwise don't do that.

③Define P = C (P only contains C right now)

Step5: Judge whether there are characters in the data stream.

1. Yes, return step3 to continue the encoding process.
2. No, output P's corresponding code to the encoded data stream.

Step 6: End.

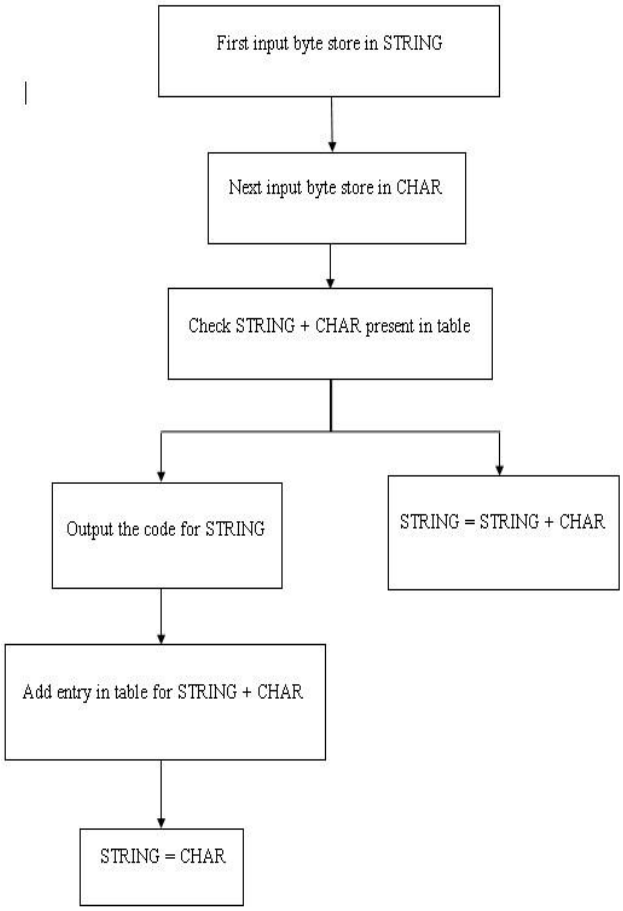

Fig 3: LZW diagram

4. CONCLUSION

The study has found that the data integrity issues exist while transferring data to and from main server to remote server and data stored in remote server from the main server needs to be compressed in order to utilize storage space in the remote server. To assure the data integrity, it is necessary to detect any error inserted to the data irrespective of its type and for assuring storage space, data compression needs to be implemented for efficient utilization of remote server storage using compression algorithms. This paper presents a combine approach which will be useful for improving the detection of data integrity violation and for better compression ratio. In this paper, discussed theoretical performance analysis of selected data integrity checking methods and data compression methods. The proposed methodology is simple and straight forward and is well-suited for two important aspects of cloud computing. The limitation of this methodology for data integrity is that Establishment of Shared Secret and Inability to Provide Non-Repudiation and for data compression, the proposed compression method cannot be used in variant color images or gray scale image or natural images that contain shadows or gradient. So there is scope for further research in the area where MAC method preferred to use of shared secret key among predecided legitimate users and on the other hand, MAC cannot provide Non-repudiation service. This service provides the assurance that a message originator cannot deny any previously sent messages and commitments or actions. For Further research in compression technique, there is a lot of scope in optimizing LZW method because it waste some time to find characters.

5. ACKNOWLEDGMENTS
The authors would like to thank Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology for providing useful guidance to accomplish this survey.

6. REFERENCES
[1] M.T.Patil, A Survey on Different Techniques Used in Decentralized Cloud Computing, International Journal of Science and Engineering Applications, 2016.

[2] V.Kalpana and V. Meena, Study on Data Storage Correctness Methods in Mobile Cloud Computing, Indian Journal of Science and Technology,2015.

[3] M.Ali, S. U. Khan and A. V. Vasilakos, Security in cloud computing: Opportunities and challenges, Information Sciences 305 (2015) 357–383.

[4] P.Premkumar and Dr.D.Shanthi, An Efficient Dynamic Data Violation Checking Technique for Data Integrity Assurance In Cloud Computing, International Journal of Innovative Research in Science, Engineering and Technology, 2014.

[5] L. Wei , H. Zhu , Z. Cao , X. Dong , W. Jia , Y. Chen and A. V. Vasilakos, Security and privacy for storage and computation in cloud computing, Information Sciences,6 May 2013.

[6] I. M. A. D.Suarjaya "A new algorithm for data compression optimization." arXiv preprint arXiv: 1209.1045 (2012).

[7] K.Govinda and Kumar, Storage Optimization in Cloud Environment using Compression Algorithm ,International Journal of Emerging Trends & Technology in Computer Science (IJETTCS),2012.

[8] F.Zhang, Li, Z. Wen, M. C., Jia, X., & Chen, C. (2011). Implementation and optimization of LZW compression algorithm based on bridge vibration data. Procedia Engineering, 15, 1570-1574.

[9] J.A.Ghaeb, M.A. Smadi, and J.Chebil, 2011. A high performance data integrity assurance based on the determinant technique. Future Generation Computer Systems, 27(5), pp.614-619.

[10] G.Sivathanu, Wright, C.P. and Zadok, E., 2005, November. Ensuring data integrity in storage: Techniques and applications. In Proceedings of the 2005 ACM workshop on Storage security and survivability (pp. 26-36). ACM.

[11] D.Salomon, 2004. Data compression: the complete reference. Springer Science & Business Media.

[12] M.Nelson, "Data compression with the Burrows-Wheeler transform." Dr. Dobb's Journal 9 (1996): 46-50.