Data Integrity and Compression in Cloud Computing

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

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

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

ABSTRACT
Cloud computing is an emerging internet based computing technology that uses internet and maintain the servers for data storage and other applications. Nowadays, cloud storage provides data owners to upload their files and delete the local copy of the data, which helps to reduce the maintenance process of data and here data are remotely stored which leads to less aware of security threads . Many auditing schemes and protocols are used for verifying the checking integrity of outsourced data without downloading option and data modification are done by several unauthorized users or malicious users. To solve the issues, propose an integrity checking approach for remote storing data and also propose a novel mechanism for compression data with integrity.

General Terms
Auditing scheme, Cloud computing, Compression ratio.

Keywords
Data integrity violations, Main cloud, Remote cloud.

1. INTRODUCTION
With the help of Cloud computing technology, users can store large amount of data in cloud and it can be access from anywhere Cloud storage is a model of data storage in which the digital data is stored in logical pools, the physical storage spans multiple servers (and often locations), and the physical environment is typically owned and managed by a hosting company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running. People and organizations buy or lease storage capacity from the providers to store user, organization, or application data. Cloud storage services may be accessed through a co-located cloud computer service, a web service Application Program Interface (API) or by applications that utilize the API, such as cloud desktop storage, a Cloud storage gateway or web-based content management system. Besides the application of cloud storage, there are intrinsic security risks. A real time example is that when data owners outsource their data to the cloud, they generally lose physical possession of their data and may have no idea where their data are actually stored or who has the permission to getting access to their data. That is to say, it is the cloud servers who control the fate of the data after the data owners uploading their files to the cloud. While most cloud service providers are honest (e.g. due to their vested interest in ensuring a good reputation and avoiding civil litigations), data loss incidents are inevitable. Here the user's data may loss may because of several deletion based on cloud servers for making the available storage space to other file in order to get more profit. All-in-all data owners require more strong integrity guarantees for their outsourced data and they want to make sure that cloud servers always store their data correctly. Here the cloud data integrity [2, 3] is particularly important for secure and reliable cloud storage services. On the other hand, data compression [4] method requires to utilize the storage space of cloud. Cloud storage provider counts every byte of space in cloud. The more data a cloud storage providers can fit onto their servers, the more money they can make. That is why they often compress image files, which are usually quite large. The problem is that users rarely know it's happening until they experience issues with the quality of photos for posting or printing downloaded from the cloud. In this paper an attempt has been made to consider the problem of various data integrity violations and storage issues in remote cloud. Also, authentication issues will take into consideration.

2. LITERATURE SURVEY
C.Dinesh [5] established new proposed system for this using our data reading protocol algorithm to check the integrity of data before and after the data insertion in cloud. Here the security of data before and after is checked by client with the help of CSP using our effective automatic data reading protocol from user as well as cloud level into the cloud‖ with truthfulness. Also proposed the multi-server data comparison algorithm with the calculation of overall data in each update before its outsourced level for server restore access point for future data recovery from cloud data server. Proposed scheme efficiently checks integrity in efficient manner so that data integrity as well as security can be maintained in all cases by considering drawbacks of existing methods. All cloud server storage resources are managed by high performance and high-availability storage area network. Many cloud solutions operate on local disks from the host system, which means any computing or storage failure can result in down time and potential data loss. As cloud servers are autonomous, if there happens any server crack in stored data, these can be protected against internal and external attacks.

A.Upadhyay et.al. [6] Published a paper focuses on the infrastructure services dealing with storage and network usage. Propose at a new architecture for deduplication on the cloud using functionalities like segmentation, compression and binning. This paper talks about local block-level deduplication which is verified using the Eucalyptus environment. A global deduplication across various users can have security issues. We have designed multiple metadata structures which enable faster lookups and enhance user experience. A cloud based on the proposed architecture has better storage efficiency and lesser bandwidth consumption. This architecture benefits both cloud service providers and users. Considerable amounts of space can be saved in the cloud by means of deduplication and compression. Segmentation of a file reduces it to smaller chunks which are easier to transfer over the Internet. Sending unique compressed segments minimizes bandwidth consumption significantly.

B. Priyadharshini and P. Parvathi [12] surveys protocols that verify remote data possession. These protocols have been

proposed as a primitive for ensuring the long-term integrity and availability of data stored at remote untrusted hosts. In this survey, analyzing several of these protocols, compare them with respect to expected security guarantees and discuss their limitations. One of the biggest concerns with cloud data storage is that of data integrity verification at untrusted servers. For example, the storage service provider, which experiences Byzantine failure occasionally, may decide to hide the data errors from the clients for the benefit of their own. How to efficiently verify the correctness of outsourced cloud data without the local copy of data files becomes a big challenge for data storage security in Cloud Computing.

Bogdan Nicolae [7] evaluates the trade-off resulting from transparently applying data compression to conserve storage space and bandwidth at the cost of slight computational overhead. He aims at reducing the storage space and bandwidth needs with minimal impact on I/O throughput when under heavy access concurrency. Our solution builds on BlobSeer, a highly parallel distributed data management service specifically designed to enable reading, writing and appending huge data sequences that are fragmented and distributed at a large scale. Compression/decompression can be performed either at application level by the user explicitly or it can be handled transparently by the storage service. Explicit compression management may have the advantage of enabling the user to tailor compression to the specific needs of the application, but this is not always feasible. Many applications are build using high-level paradigms specifically designed for data-intensive applications (such as Map Reduce). This paradigms abstract data access, forcing the application to be written according to a particular schema which makes explicit compression management difficult. For this reason it is important to integrate compression in the storage service and handle it transparently.

3. PROBLEM DEFINITION

There are several issues can occur while storing from main cloud to remote cloud as mentioned below:

1) Data integrity issues exist while transferring data to and from main cloud to remote cloud.

2) Authentication between remote cloud and main cloud can be compromised by malicious attacks.

3) Data stored in remote cloud from the main cloud needs to be compressed in order to utilize storage space in the remote cloud.

4. PROPOSED SOLUTION STRATEGY

The proposed solution strategy is intended to address the problems mentioned in the problem definition as:

1) Data integrity checking mechanism is required to certify uncorrupted transmission by Message authentication code (MAC).

2) Authentication between remote cloud and main cloud will be provided by employing techniques such as Message Authentication Code (MAC).

3) Data compression needs to be implemented for efficient utilization of remote cloud storage using compression algorithm like Lempel-Ziv-Welch (LZW)[9,10].

5. METHODOLOGY

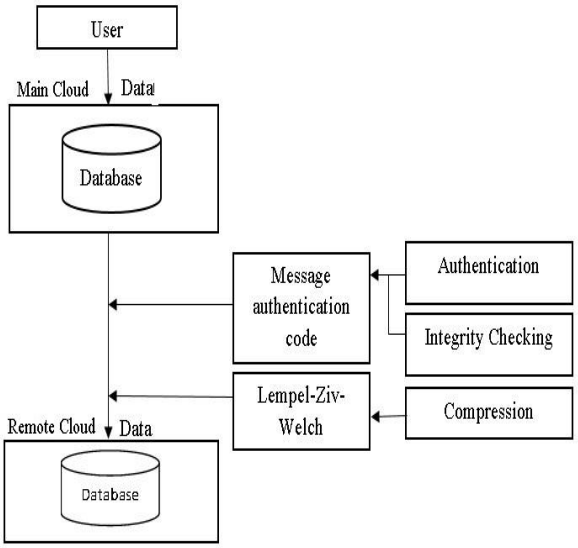

Fig 1: Block diagram of methodology

Create two clouds, send text files like (pdf, doc, and docx) and image file like (jpg, png, gif) from Cloud 1 to Cloud 2. Before both files storing into Cloud 2, apply Data Integrity algorithm (HMAC (Keyed-hash message authentication code)-SHA 512) for checking data integrity of both files then we need to apply Data Compression algorithm (LZW [11, 13] (Lempel-Ziv-Welch)) for compressing data on both files like text and image. After integrity checking and compressing both files successfully, then store the compressed files into Cloud2.

Remote cloud such as Dropbox, Google Drive enable you to host, edit, share and sync files, but not much else. Main cloud which is called as central server will run an operating system designed to support many users, multi-user applications, databases, and much more.

5.1 Message Authentication Code (MAC)

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

1) The sender takes input as a message and the secret key K and using MAC algorithm to produces a MAC[15] value.

2) The sender forwards the message along with the MAC value.

3) 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.

4) If the same MAC is found then the message is authentic and integrity checked.Otherwise, the message has been modified.

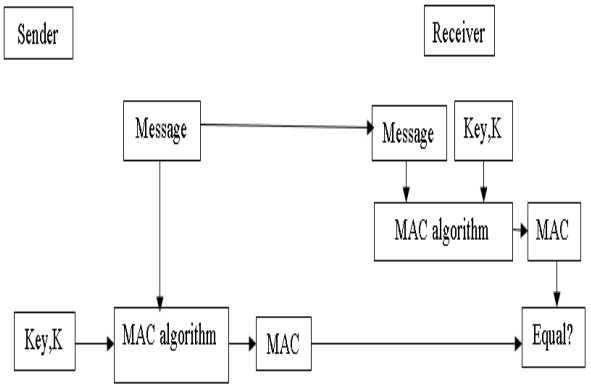

Fig 2: MAC diagram

To compute a MAC over the data 'text' using the HMAC[8](keyed-hash message authentication code) function, the following operation is performed:

$$\mathrm{MAC}(text)_t = \mathrm{HMAC}(K, text)_t = \mathrm{H}((K_0 \oplus \mathrm{Opad}) \| H((K_0 \oplus \mathrm{ipad}) \| text))_t$$

H – An hash function.

text-The data on which the HMAC is calculated.

t- The number of bytes of MAC.

B-Block size (in bytes) of the input to the Approved hash function.

K- Secret key shared between the originator and the receiver.

Ko- The key K after any necessary pre-processing to form a B byte key.

ipad- Inner pad; the byte x'36' repeated B times.

Opad- Outer pad; the byte x'5c' repeated B times.

⊕- Exclusive-Or operation.

‖- Concatenation

5.2 HMAC algorithm

Step 1: If the length of K=B: set K_0=k, Go to step 4.

Step 2: If the length of K>B: hash K to obtain an L byte string, then append (B-L) zeros to create a B-byte string (i.e., K_0 (i.e., = H (K) || 00...00). Go to step 4.

Step 3: If the length of K>B: append zeros to the end of K to create a B-byte string K_0 (e.g., if K is 20 bytes in length and B = 64, then K will be appended with 44 zero bytes 0x00).

Step 4: Exclusive-Or K0 with ipad to produce a B-byte string: K_0 + ipad.

Step 5: Append the stream of data 'text' to the string resulting from step 4:(K_0+ ipad) ‖text.

Step 6: Apply H to the stream generated in step 5: H ((K_0+ipad))‖text).

Step 7: Exclusive-Or $K_0$with opad: K_0+opad

Step 8: Append the result from step 6 to step 7:(K_0⊕ opad)‖H((K_0⊕ ipad)‖text).

Step 9: Apply H to the result from step 8:H((K_0⊕ opad)‖H((K_0⊕ ipad)‖text)).

Step 10: Select the leftmost t bytes of the result of step 9 as the MAC.

5.3 Secure Hash Algorithm (SHA)

1) The aim of this step is to make the length of the original message equal to a value, which is 64 bits less than an exact multiple of 512.

2) After padding bits are added, length of the original message is calculated and expressed as 64 bit value and these 64 bits are appended to the end of the resultant message.

3) Divide the input into 512-bit blocks.

4) Initialize chaining variables.

5) SHA[14] has a total of 80 iterations (4 rounds X 20 iterations). Each iteration consists of following operations:-

abcde = (e +Process P + S^5**(a) + W[t] + K[t]),a,**S^{30}**(b) , c .**

Where,

abcde = The register made up of 5 variables a, b, c, d, e.

Process P = The logic operation.

S^t = Circular-left shift of 32-bit sub-block by t bits.

W[t] = A 32-bit derived from the current 32-bit sub- block.

K[t] = One of the five additive constant.

Table 1: Process P in each SHA round

Round	Process P
1	(b AND c) OR ((NOT b) AND (d))
2	b XOR c XOR d
3	(b AND c) OR (b AND d) OR (c AND d)
4	b XOR c XOR d

The values of W[t] are calculated as follows:

1) For the first 16 words of W (i.e. t=0 to 15),the contents of the input message sub-block M[t] become the contents of W[t].

2) For the remaining 64 values of W are derived using the equation

W[t] = S^1(W[t-16] XOR W[t-14] XOR W[t-8] XOR W[t-3])

5.4 Lempel-Ziv-Welch (LZW)

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

2) Set the Prefix P Null.

3) Read the next character in the data stream as the current character C.

4) Judge whether the string P + C is in the current dictionary.

a)Yes, set P = P + C, that is extending P with C.

b) 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)

5) Judge whether there are characters in the data stream.

a) Yes, return step 3 to continue the encoding process.

b) No, output P's corresponding code to the encoded data stream.

6) End.

6. 6. RESULTS AND DISCUSSION

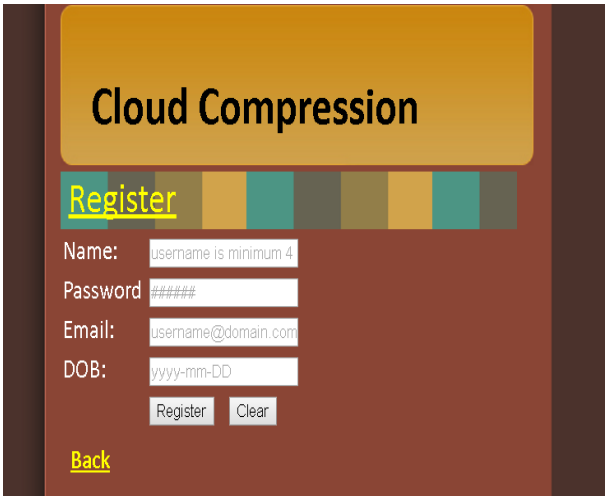

Fig 3: Registration for new user, if not an member

The user who has to avail the services of cloud storage, has to provide detailed information about him to the servers and gets registered. User information is monitored by admin. It is assumed that the admin is a trusted entity. User has to fill up the above details like name, password, email and date of birth.

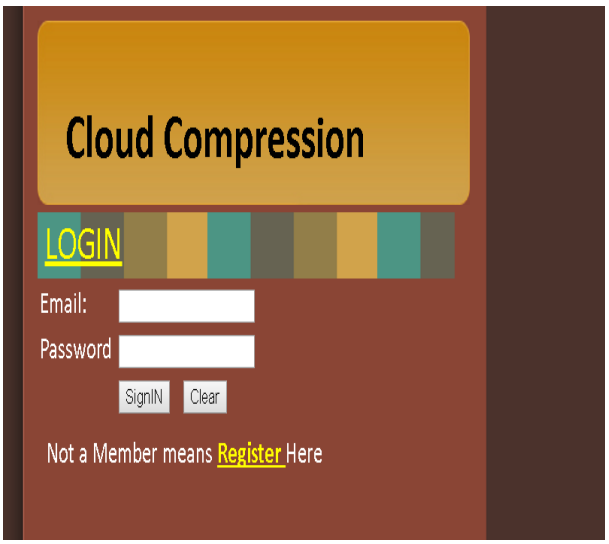

Fig 4: Login process for new user

Create login page for the users or data owner to register the clouds. Login the users to verify the authentication if valid user precedes the cloud process else return the invalid user page. Authorized data user upload, download and delete the filesfromclouds. Once the registration is done, the user can now login on the cloud server. After checking the login credentials, the user gets authorized to store data store on the cloud. The registered information is cross checked with the both the servers. While storing the file on the cloud server, the user does the login using the username and the password which he has developed during the registration process.

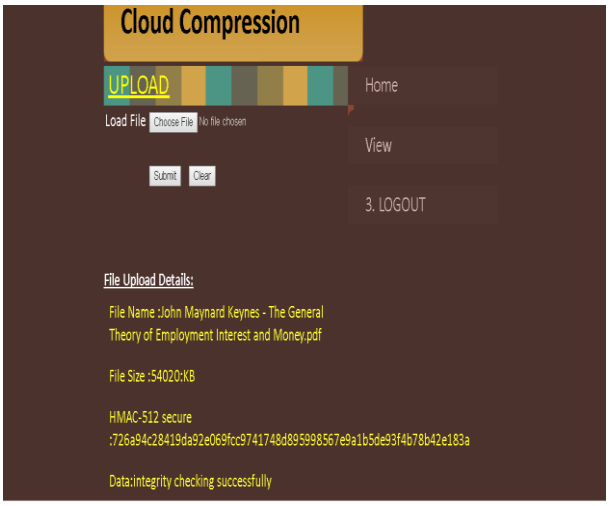

Fig 5: Displaying the uploaded file (PDF) details in main cloud

Data user use HMAC-SHA512 and during sending files (52 MB) from Cloud 1 to Cloud 2 communication use HMAC-SHA512 and checking files integrity. If file integrity failed then do not perform compression operation. User should upload or delete data in cloud 1(consider as Main Cloud) send to Cloud 2(Remote cloud) then perform .User should upload or delete data in cloud 1 send to Cloud 2 then perform HMAC-SHA512. After login page, it will automatically redirected to the upload page where it offered to choose a file like PDF or image file(jpg, PNG etc)which can be uploaded from any drive of the computer and then automatically stored the uploaded file onto cloud server 1 folder. Click submit button to submit a file, then it shows the file name ,file type and file size in kb.HMAC-SHA512 create a MAC value which will sending to the Cloud server, again recomputed MAC value , if same MAC is found then the message is authenticated and display output which shows data integrity checking successfully.

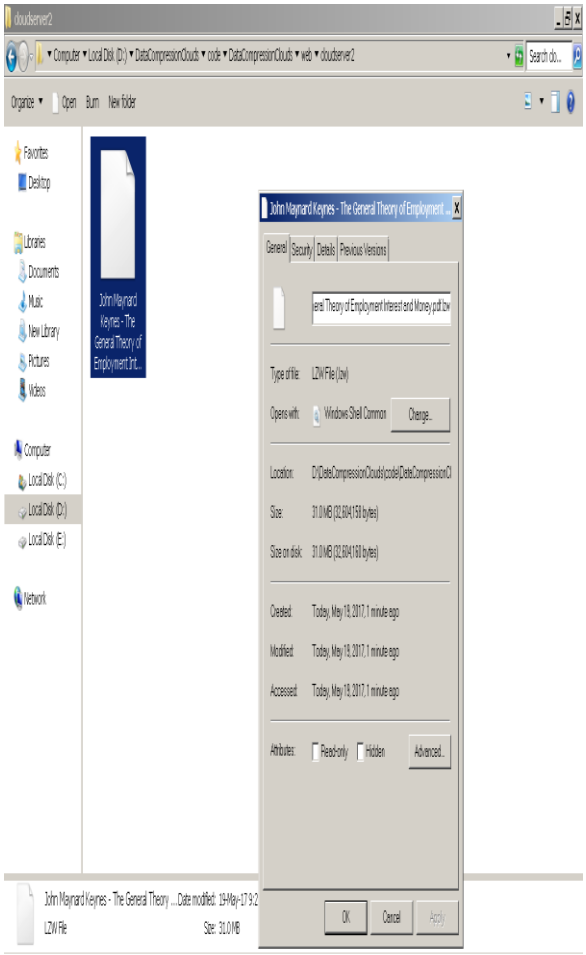

Fig 6: After Compressing File Stored in Remote Cloud

After successfully checking data integrity, compressed file automatically stored onto cloud 2(remote cloud) folder in the computer. Right click on file properties to show the file size. But, it cannot view as an image. Only storing purpose cloud2 will be used and compressed file (31 MB) stored in remote cloud. After compressing, user downloads or deletes data from cloud 2(Remote Cloud).

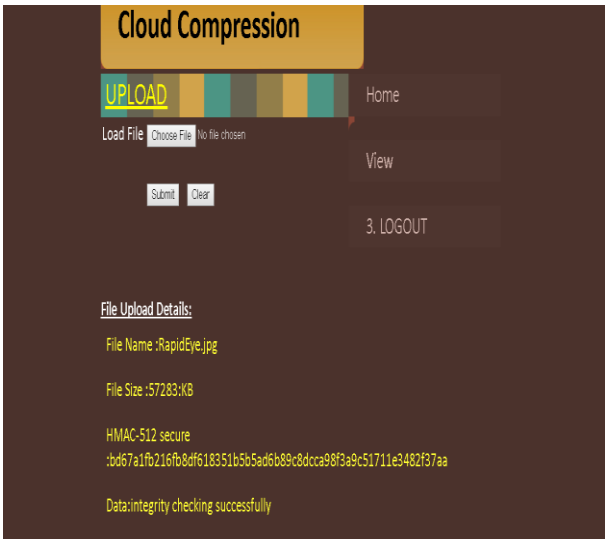

Figure 7: Displaying the Uploaded File of 55 MB (jpg) Details in Main Cloud

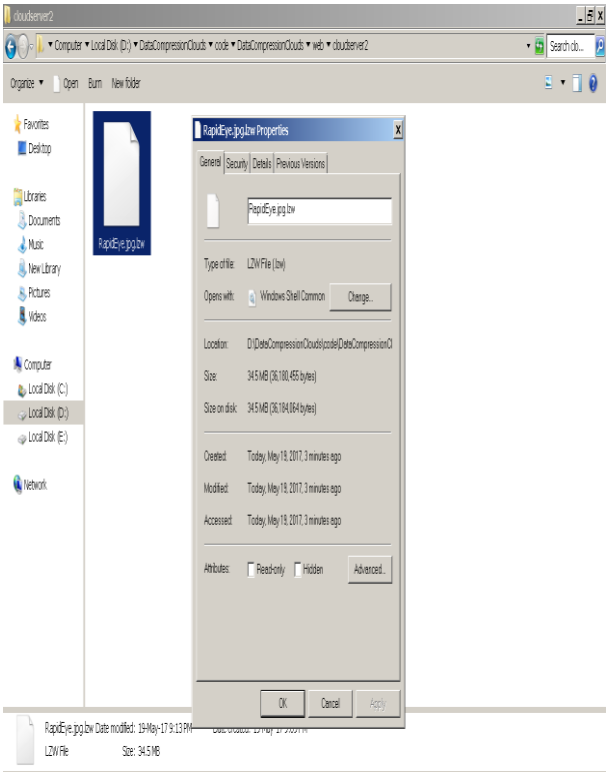

Figure 8: Compressed file of 34.5 MB (Jpg) stored in Cloud 2(Remote Cloud)

LZW algorithm is implemented using java and achieved around 50% of compression as shown in above figure. In the proposed approach, the findings say that the data stored on the cloud is in compressed format, needing less space as that of the original file. The security parameters (integrity and authentication) are evaluated in the proposed system. As for the proposed system all the parameters are satisfied. In proposed method size of uploaded file is same as original results less communication overhead. Time Complexity for various file Size is measured. As file size increases time increases. This time includes encoding, uploading and downloading time. Time taken also depends upon the network bandwidth and delay.

7. 7. CONCLUSION

The study has found that 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. On the other hand, data compression deals with 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. For Further research in compression technique, there is a lot of scope in optimizing LZW method. For optimization, a new method called forward-moving on frequently-used entries can be implemented to avoid waiting some time to find the codes which can make compression time long.

8. 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 project.

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

[2] Ghaeb,J.A.,Smadi,M.A and Chebil,J., 2010. A high performance data integrity assurance based on the determinant technique. Future Generation Computer Systems, 27(5), pp.614-619.

[3] Premkumar,P and Dr. Shanthi,D.,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.

[4] Quinlan,T.and Dorward, S., 2002, January. Venti: A New Approach to Archival Storage. In FAST (Vol. 2, pp. 89-101)

[5] Dinesh,C., Data Integrity and Dynamic Storage Way in Cloud Computing, IEEE 2010, Vol.2,pp 201-205.

[6] Upadhyay,A.,Balihalli,P.,Ivaturi.,S.andRao,S.,Deduplication and Compression Techniques in Cloud Design,IEEE 2012,Vol.4,pp 11-16.

[7] Nicolae,B., High Throughput Data Compression for Cloud Storage, Springer 2012.Vol.11,pp 48-54.

[8] Braden R, Borman D, Partridge C. Computing the Internet Checksum. Intenet Request For Comments RFC 1071. ISI, September. 1988.

[9] Yan,H.,Lu.H and Gao,Q., 2012.A BP-LZ78Compression Algorithm Based on BP Network. In Advances in Electronic Engineering, Communication and Management Vol. 2 (pp. 211-216). Springer Berlin Heidelberg.

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

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

[12] Priyadharshini .B and Parvathi,P., Data integrity in cloud storage. In Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on (pp. 261-265). IEEE.

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

[14] https://en.wikipedia.org/wiki/Secure_Hash_Algorithms

[15] https://en.wikipedia.org/wiki/Message_authentication_code