INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING & TECHNOLOGY (IJCET)

ISSN 0976 – 6367(Print)
ISSN 0976 – 6375(Online)
Volume 4, Issue 2, March – April (2013), pp. 229-236
© IAEME: www.iaeme.com/ijcet.asp
Journal Impact Factor (2013): 6.1302 (Calculated by GISI)
www.jifactor.com

IJCET
© I A E M E

PERCEIVING AND RECOVERING DEGRADED DATA ON SECURE CLOUD

V.Ramesh[1], P.Dhanalakshmi[2]

[1,2] Department of Computer Science, Kalasalingam Institute of Technology, Krishnankoil-626126

ABSTRACT

Cloud computing is Internet-based computing, whereby shared resources, software and information, are provided to computers and devices on-demand. . Cloud Storage deals with file blocks, simplifying storage management and eliminating metadata concern. Data are continuously distributed through multiple servers in cloud. The token is computed dynamically. If data lost, then it must find out that which server gets corrupted. It can be done with byzantine fault tolerance system. The usual way of detecting corrupted data is by computing a signature for the token when it enters the cloud, and whenever it is transmitted across a cloud that is unreliable and hence capable of corrupting the data. The data is deemed to be corrupt if the newly generated signature doesn't match the original signature precomputed by the user. Third Party Auditor (TPA) is responsible for verifying the token they receive before displaying the data and its signature. The TPA verifies all the tokens distributed through multiple server. Distributed cloud server stores replicas of file blocks; it can heal corrupted blocks by retransmitting the corrupt replica block. The RS algorithm is used to guard against corruption due to data loss/node loss by supporting the retransmission. The main aim of this paper is making the file system tolerate node failure without suffering data loss.

Keywords: Cloud Storage, Data Lost, Distributed System, Fault Node, File Blocks, Recovery, Retransmission, Secure Outsourcing, Tokens

I. INTRODUCTION

Cloud computing refers to the method by which files are transferred from a computer or Smartphone, or tablet to physical servers. Example Web-based e-mail is a form of cloud

computing. Email messages are saved on Cloud servers which is, e-mail can be checked from anywhere. Similarly, the file can be retrieved on nearly any Web-enabled device, when the files are saved or backed-up with a cloud service. The benefits of cloud computing include lowered costs, higher performance and availability. Businesses with a cloud-based IT infrastructure do not have to spend on any additional on-site hardware or personnel. The cloud symbol traditionally represents the Internet. Cloud computing refers to both the applications delivered as services over the Internet and the hardware and system software in the data centers that provide those services.

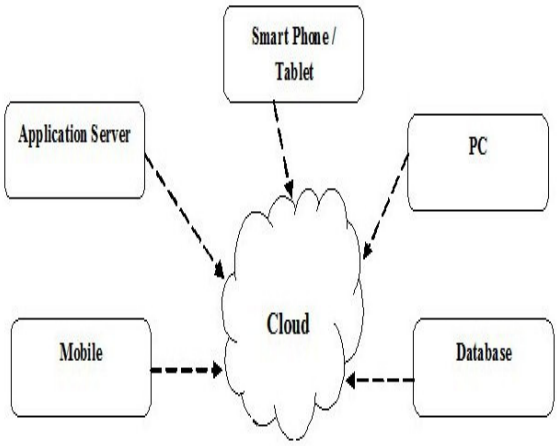

Figure 1: Cloud

Cloud storage system can be considered to be a network of distributed data centers which typically uses cloud computing technologies like virtualization, and offers some kind of interface for storing data. Cloud storage providers usually provide client software which assists users in setting up their synchronization or backup schemes on the local devices. The actual transmission of all data with the remote storage servers is also handled by the client software.

Distributed File System is a file system designed for storing very large files with streaming data access patterns, running on clusters of service hardware. Cloud Storage System manages the storage across a network of machines. Cloud Storage deals with file blocks, simplifying storage management and eliminating metadata concern.

II. TRUSTED EVALUATION

In this paper, we focus on applications where the latency of the computation should be minimized, i.e., the time from submitting the query until receiving the outcome of the computation should be as small as possible. To achieve this, Fig.2, show how to combine a trusted hardware token with Secure Function Evaluation to compute arbitrary functions on secret data where the computation leaks no information and is verifiable. The file blocks are cracked into tokens. The delegation token is generated by the server with the signature. The tokens are distributed through multiple nodes and they are integrated at the user. It is finally verified by the TPA. If any token mismatch, it must be replicated.

The time to transfer a large file is made of multiple blocks. Because by making a block large enough, the time to transfer the data from the disk can be made to be significantly larger than the time to seek to the start of the block.

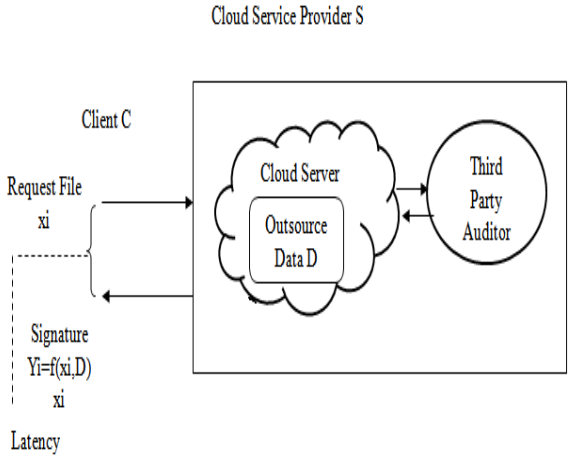

Figure 2: Token Evaluation

III. TOLERANCE VALIDATION

The client opens the file it wishes to read by requesting it from the ser ver. During the transfer from server to client, if the user encounters an error while communicating with other nodes then it will try the next closest one for that block to retransmit. It will also remember all other nodes that have failed so that it doesn't needlessly retry them for later block transfer. TPA also verifies the checksum for the data transferred to it from the other nodes.

1. Send each file block from s0 to sk
2. When a process finishes transfer, it collects all blocks
3. TPA analyze the signature of the encoded source blocks
4. K+1 block provide k fault tolerance
5. If any encoded signature is missing, it shows faulty data attainment
6. When a Byzantine failure has occurred, the system may respond in unpredictable way
7. Locate the server failure, with the checksum stored with each block
8. Sends the integrated blocks to user

IV. DATA RESTORATION

Erasure coding concept is used to improve the throughput and to find the faulty system. The file is segmented into multiple blocks of fixed length. Multiple smaller blocks of encoded data are likely to easily experience the data loss. Each block is protected by the recovery reed Solomon algorithm. The encoded symbols are verified at the user. If any data loss, it can be recovered. Data Block Units can be recovered by retransmission of fault data block. Fermat Number Transform based algorithm operates as fast as XOR based implementations for small file blocks.

1. Let r be an file of order n-1 in the finite field
2. Use Discrete Fourier Transform(DFT) to take a vector a=(a0,…,an-1) of size n as input in Finite Field
3. Use Fermat Number Transform in finite fields to process DFT
4. Generate n encoded data block units from a set of k source data block units
5. Let s=(s0,s1,…,sk) be a source vector, where k is the number of data blocks
6. Let e=(e0,e1…,ek) be the encoded vector of size n
7. Interpolating the data block with encoded signatures on the k data block positions
8. Apply decoding on the received data block units
9. k symbols have been received for a polynomial of degree k-1.
10. If some source symbols are missing i.e., some of the information in e could have been lost
11. Apply same decoding on the previously received k data block units
12. Apply same DFT on the server encoding side to recover the full encoded data block vector e

Fig.3, Shows the general working functionality of the cloud to check the integrity violation

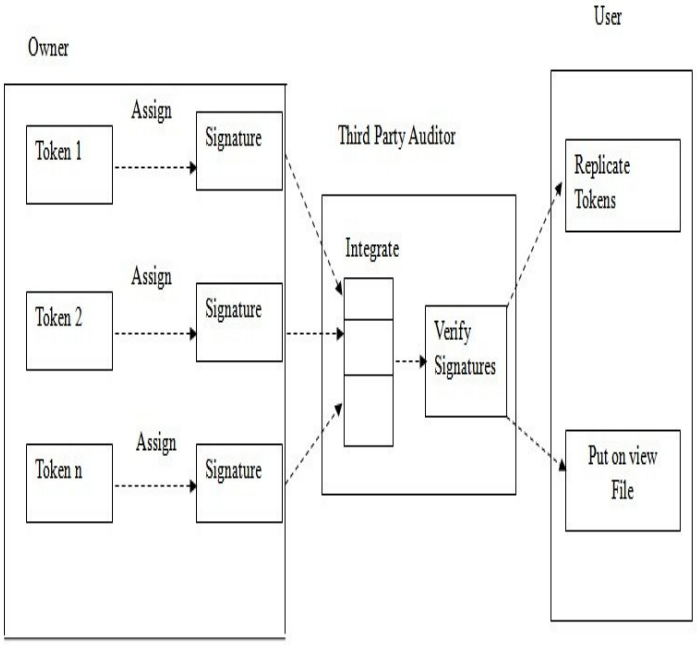

Figure 3: Working Functionality

V. FUNCTIONAL EXECUTION

UPDATION

Clients submit one update operation to their local server, wait for proof that the update has been ordered, and then submit their next update.

APPEND

The append operation allows authorized user to modify an already written file by opening it and writing data from the final offset in the file.

DELETION

In delete operation, file blocks that are distributed among cloud storage servers are all deleted. Once file is deleted, we cannot perform any recovery of deleted files as there won't be any backup available in main cloud server.

VI. PERFORMANCE EVALUATION

Having many file blocks means the time taken to process each token is small compared to the time to process the whole input. If size of tokens is too small, then the overhead of managing the tokens begins to dominate the total job execution time. Hence the file must be cracked as a maximum of large size tokens to reduce the large overhead.

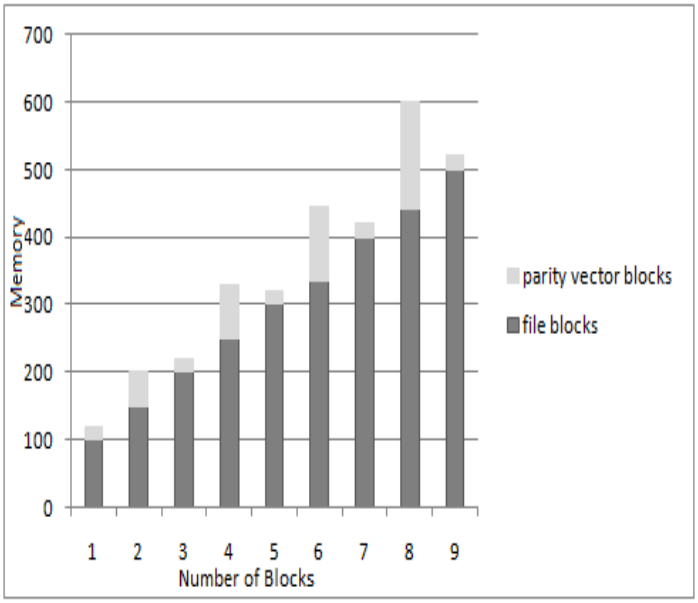

Figure 4: Parity Generation Cost

e determines how many parity vectors are required before outsourcing. Growth of e means large number of parity blocks required to be blinded. The files with variable file size blocks will increase the parity vector size according to the file token size. Hence, the parity generation cost will also be increased with increase in file size.

In Fig. 4, the fixed file blocks have fixed parity vector size. Hence, the cost is limited according to the token precomputed by the owner. Data are continuously distributed through multiple servers in cloud. The token is computed dynamically.

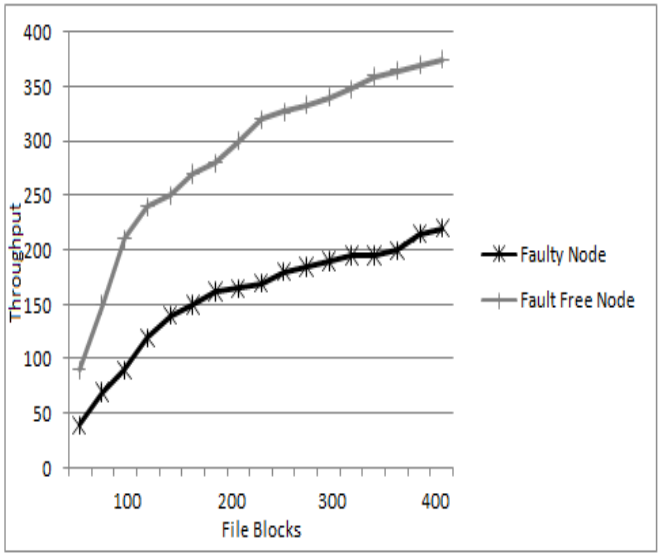

Figure 5: Comparison of Throughput for faulty and non-faulty nodes

The throughput results are different when some of the nodes are attacked. In Fig.5, The slope of the curve corresponding to Node under attack is less steep than when it is not under attack due to the delay added by the malicious attack.

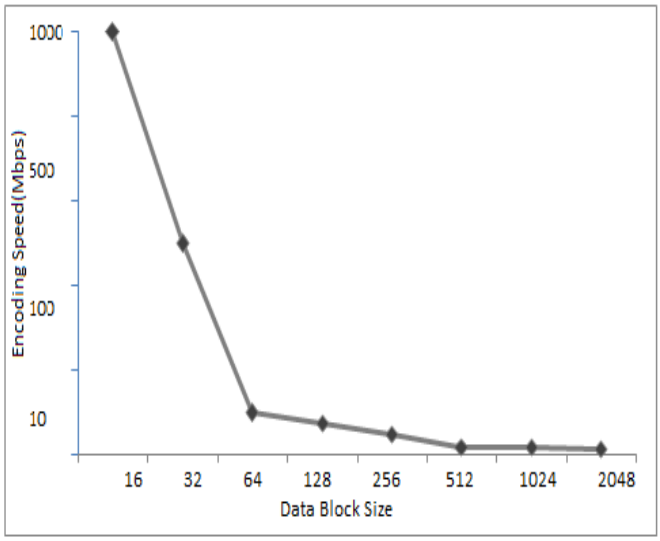

Figure 6.a: Encoding Speed for Various File Block Size

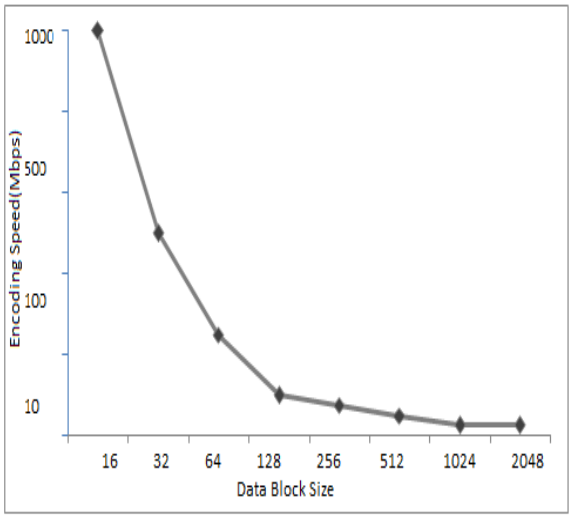

Figure 6.b: Decoding Speed for Various File Block Size

Fig.6, provides the encoding and decoding speed for the FNT based reed Solomon algorithm. However, the encoding/decoding complexities of the file depend on the size of the data blocks. For cloud storage, the encoding/decoding methods are adapted more for smaller file block sizes. If the file is transferred as whole, then the encoding speed will be more. But if the file is cracked into multiple blocks, then the encoding speed will be low. This will improve the performance of the throughput.

VII. CONCLUSION

Cloud Computing requires security methods of preserving important data in order to prevent unrecoverable data loss. Secure outsourcing of computation to cloud service provider is becoming more and more important. This paper open the way for practical implementation of Byzantine Fault Tolerance and Reed Solomon for smaller multiple number of file blocks. The file generates a distinct signature on all file blocks and appends the signature to display the file. When the data blocks are aggregated on the user, the auditing mechanism can speed up the verification of signature at TPA. Indeed, FNT is a special case of reed Solomon algorithm could benefit from recovery of faults. This paper describes the efficient dynamic data verification method and efficient recovery of faulty data blocks.

REFERENCES

[1] A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 584–597.
[2] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy preserving public auditing for storage security in cloud computing," in Proc. of IEEE INFOCOM'10, San Diego CA, USA, March 2010.
[3] C. Wang, K. Ren, W. Lou, and J. Li, "Towards publicly auditable secure cloud data storage services," IEEE Network Magazine, vol. 24, no. 4, pp. 19–24, 2010.

[4] C. Wang, Q. Wang, K. Ren, and W. Lou, "Towards Secure and Dependable Storage Services in Cloud Computing", in IEEE Transactions on Services Computing, vol.5, no.2, pp. 220-232, 2012
[5]Alexandre Soro, Jerome Lacan, "FNT based Reed Solomon Erasure Codes", ISAE/DMIA,Toulouse, France
[6] C.Cahin, R.Guerraoui, L.Radrigies, Introduction to reliable and secure distributed programming, 2nd edition, Published by Springer, 2011.
[7] T. White. Hadoop: The Definitive Guide. O'Reilly, 2009.
[8] C.Wang, Q.Wang, K.Ren and W.Lou, "Ensuring data storage security in cloud computing", proc.17th Intl Workshop Quality of Service(IWQoS'09),pp.1-9,July 2009.
[9] K. D. Bowers, A. Juels, and A. Oprea, "Hail: A highavailability and integrity layer for cloud storage," in Proc. of CCS'09, 2009, pp. 187–198.
[10] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS'09, volume 5789 of LNCS. Springer-Verlag, Sep. 2009, pp. 355–370.
[11] Veronese.G.S, Correia.M, Bessani.A.N, and Lung.Spin one's wheels? Byzantine fault tolerance with a spinning primary. In Proceedings of the 28th IEEE Symposium on Reliable Distributed Systems, Sept. 2009.
[12] Mr. Hemantkumar Wani and Dr. N. Mahesh, "Security Issues in Cloud Computing for MSMES" International Journal of Advanced Research in Management (IJARM), Volume 3, Issue 2, 2012, pp. 21 - 28". ISSN Print: 0976 – 6324, ISSN Online: 0976 – 6332.
[13] Abhishek Pandey, R.M.Tugnayat and A.K.Tiwari, "Data Security Framework for Cloud Computing Networks", International journal of Computer Engineering & Technology (IJCET), Volume 4, Issue 1, 2013, pp. 178 - 181, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.
[14] http://www.sit.fraunhofer.de/content/dam/sit/en/studies/Cloud-Storage-Security
[15] http://pagespro.isae.fr/jerome-lacan
[16] http://www.opensciencegrid.org/bin/view/storage/Hadoop Installation
[17] http://www.gmplib.org/

Mr. V.Ramesh, The author is an Assistant Professor in Computer Science and Engineering Department at Kalasalingam Institute of Technology. He received his BE from Syed Ammal Engineering College; affiliated to Anna University, and M.Tech., Degree from Kalasalingam University. His research interests are in the areas of network security, Data mining and cloud computing security.

Ms. P.Dhanalakshmi, The author is currently a ME student in Computer Science and Engineering Department at Kalasalingam Institute of Technology. She had completed BE from PSR Engineering College; affiliated to Anna University.