Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 2, February 2014, pg.1 – 7

RESEARCH ARTICLE

Distributing, Ensuring and Recovery of Data Stored in Cloud

Dr. B. F. Momin[1], Mr. Girish Bamane[2]

[1]Computer Science and Engineering, Shivaji University, Walchand College of Engineering, India

[2]Computer Science and Engineering, Shivaji University, Walchand College of Engineering, India

[1] bfmomin@yahoo.com; [2] girishb6390@gmail.com

Abstract*— *Cloud Computing can be seen as buzzword these days. As sight to traditional systems where IT services are under the control of Limitations like computing, hardware etc and everything is in personal. The Cloud Computing moves all things like applications, databases, and huge data to the large data centres, where management of the data and services may not be trustful and reliable. So, new attributes are arriving to security, which are still not well understood. Here in this paper we are going to know about data storage security. To ensure the correctness of data stored in a cloud, we are going to propose a new scheme of distributed storage. In this scheme, we are verifying the correctness of data stored in the cloud and also identifying the storage server which is misbehaving or being compromised (data error localization). Though there are lots of methods are present to prevent the data modification also for the recovery of data, like data replication method, and parity generation. The scheme proposed in this paper is more efficient than others because it prevents data loss from single point to multipoint failure storage servers.

Keywords*— *Data and services; Correctness of data; Parity generation; Multipoint failure

I. INTRODUCTION

The Cloud Computing provides internet based on demand services .It allows end users to use or access software's and hardware's from a remote location and managed by third parties. The cloud allows access to the information and computer resources from any place where a network connection is available. The cloud provides data storage, networks, computing power, shared resources, different applications to end user. In short Cloud is "Set of IT resources located on a remote server, where content and applications stored and used". The impression of Cloud Computing is cultivated across IT industries because it has a large number of solutions for huge challenges faced by IT industries today like poor performing servers and their aging infrastructure, on-demand provision of technology, complexities comes while deploying new applications, frequently unable to achieve optimal utilization or resources. So Cloud Computing is a paradigm shift that provides computing over internet. Cloud consist of highly optimized virtual data centres. They provide various software's hardware's and information resources for use when we need it. Organizations can simply connect to cloud and use the available resources to pay as you go basis. This helps uses or organizations to avoid capital expenditure on additional expenses on premises infrastructure resources and instantly scale up and scale down as per business requirements. Fig. 1 shows basic architecture of cloud.

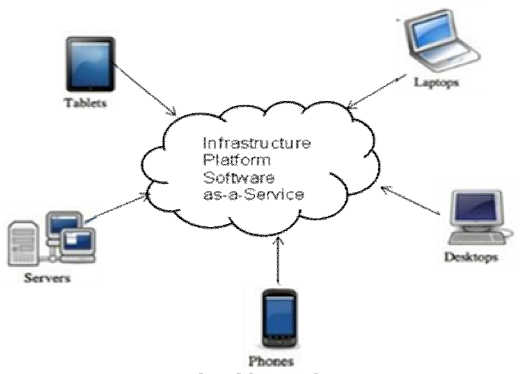

Fig 1 Cloud basic diagram

At the cheaper but powerful processor with SaaS (Software-as-Service) model are transforming large data to huge data centres. As the reliability and increasing bandwidth of networks made it possible such that the end user can easily subscribe for high quality service from data centres remotely. While moving the data to cloud end user doesn't have to care about the hardware complexities because Cloud vendors like Amazon Simple Storage Service (S3), Amazon elastic compute cloud (EC2) [1]. Web Services provides huge storages to end user along with scalable computing resources, which causes removal of the responsibilities maintaining local machines for data maintenance. One side of the Cloud is powerful and reliable infrastructure as compared to personal computers but still internal and external threads on securing the data are exist like data loss, availability [2]. But on the other hand user don't have data in his hand, he always conscious about the unfaithful behaviour of a Cloud Service Provider (CSP), or even sometimes CSP tries to hide the accident of data loss to maintain the reputation. If high value data stored in the cloud then, the necessity of CSP to provide strong assurance of the data integrity with availability. To ensure the integrity and availability of data along with quality of cloud storage services, the correctness verification of data has been designed on behalf of the user. The facts say that the user doesn't have physical control over data and also the cloud doesn't adapt directly the traditional cryptosystems for the protection of data integrity. So verification of cloud storage must be done without knowing the data from files.

Some of the techniques are there to check the integrity of data stored in cloud but they are focused on the single server scenario. This is useful for testing of QoS (Quality of Service) but not guarantees the availability of data in case of server failure [8]. So in this paper we are going to use the distributed approach to ensure the storage correctness across a number of servers. This technique can drastically reduce the storage overhead problem that caused by traditional system like the data replication method, it also reduces the problem of communication overhead. The data dependability can be provided by using erasure –correcting code i.e. parity mechanism. This scheme focuses on archive or static data only.

When we go for checking the data tampering or corruption simultaneously we find the misbehaving servers. This can be done by utilizing the homomorphic token.

Our work done is by considering the distributed data storages. So as only binary results are provided by old systems, our method moves further and identifying misbehaving servers. Considering the performance and security issues our scheme is much strong against Byzantine failures, data tampering

II. PROBLEM STATEMENT

The figure 2 shows the basic architecture of cloud data storage [5]. Cloud entities user who stores data in the cloud , CSP (Cloud Service Provider) who has storage servers, TPA [6] (Third Party Auditor) which is optional but has extra-ordinary capabilities that don't have to user. Though the data not stored locally the owner of the data must have some security means so that he makes assurance of his data stored in the cloud. If the user doesn't want the overhead of checking the data then he may take help of TPA. Note that we don't take the issue of data privacy in our mind.

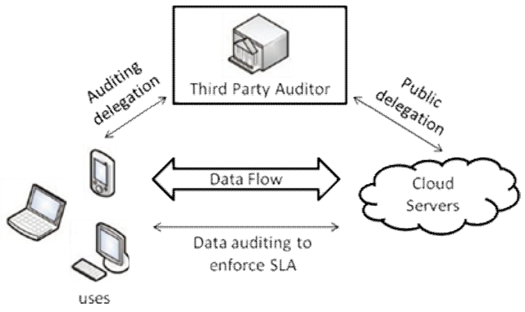

Fig 2 architecture of cloud data storage

We have threats from two sides; one is from CSP itself when it tries to hide the occurred data loss accidents and second is people who are interested in data like hackers and crackers. Once they compromise the server/s they can pull all the data from servers and introduce their own fake data to the original file.

A. Design Goals:

To ensure the correctness of user's data stored in cloud servers and target is to design data verification.

The goals are:

I. Storage Correctness of the user's data to ensure that data stored is appropriate.

II. Fast localization of data errors to locate malfunctioning servers.

III. Dependability against modification of data and Byzantine failure.

IV. Lightweight such that user can check with minimum overhead.

B. Preliminaries and Notations:

- F- data file
- Dispersal/Distribution matrix
- G- Encoded file matrix
- fkey (·) -pseudorandom function
- □ key (·) -pseudorandom permutation
- ver- version number
- S_{ij}^{var} - seed for PRF

III. ENSURING CLOUD DATA STORAGE

As the data stored in cloud by, data no longer in control of the user. So it must be needed to validate the correctness and availability of data, for unauthorized data modification, corruption, and server compromise like key issues. And also, does the identification of the data server/s where data errors are present.

To solve these problems we have solutions, first part contains the file distribution across cloud storage servers and the second part consists of homomorphic token. Functions used for the computation of the token are chosen from the family of universal hash functions [4] to preserve homomorphic properties, which are combined with the verification of erasure coded data [5].

A. File Distribution Preparation

To allow multiple distributed storage failures in file distribution system the erasure correcting codes can be used. So we trust in the system and distribute the file F across n file distribution servers redundantly. The n servers we are using, out of my servers are for storing the data vectors and remaining k servers are for storing the parity (n=m+k). The Reed-Solomon erasure correcting codes are used to create k parity vectors. From m data vectors, k parity vectors are generated, such that we can easily recover the original data from the loss using those k parity vectors. As we are placing those m+k vectors across n servers, we can survive from the failure of any k servers out of m+k servers. But on the other side it has a space overhead of k/m.

Let data file F can be divided into (F_1,F_2,F_3……F_m), and further each Fi=(f_{1i},f_{2i},f_{3i}….f_{li}) ^{T} where $l \leq 2^p$-1 and these blocks are elements of Galois Field GF(2^p)[7].

The file distribution matrix A can be computed form Vandermonde matrix of m* m+k size. So matrix A is derived from Vandermonde matrix it consumes the invertible property such that m columns from m+k form invertible property.

$$A = (I|P) = \begin{pmatrix} 1 & 0 & \ldots 1 & p_{11} & p_{12} & \cdots & p_{1k} \\ 0 & 1 & \ldots 0 & p_{21} & p_{22} & \cdots & p_{2k} \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & & . & . & . & & . \\ 0 & \ldots & 1 & P_{m1} & p_{m2} & \cdots & p_{mk} \end{pmatrix}$$

I is Identity matrix of size m*m and P is secrete parity generation matrix.

Encoded file can be obtained by multiplying F by A

$$G = F\cdot A = (G^{(1)}, G^{(2)},\ldots,G^{(m)}, G^{(m+1)},\ldots G^{(n)})$$

$$= (F_1, F_2,\ldots, F_m, G^{(m+1)},\ldots, G^{(n)})$$

B. *Challenge Token Pre-computation*

Pre-computed token verification scheme used to check the correctness of users data and error localization. Before the distribution of file across storage servers the token pre-computation is done, such that these tokens are used for the verification of the individual data vector $G^{(j)}$ where j€ (1,…,n) and each token has randomly selected data blocks. When user wants to ensure the correctness of data stored in cloud he challenges the cloud server by randomly selecting the blocks indices. When the server receives the challenge he generates the signature for the specified blocks and then returns the value to the user. The signature generated by the server should match with the tokens, pre-computed by the user. When a token is generated than the user can store it anywhere in a safe, like in encrypted form or locally.

The user can generate t number of tokens to challenge cloud servers t times for checking correctness of data. So for each $G^{(j)}$ pre-computation of tokens is done using PRF, PRP, Challenge key and master permutation key. This is done to compute ith token for server j.

The token is indicated as $v_i^{(j)}$, when user challenges server for specific data blocks that time he have faith that severs will send same token.

C. *Correctness Verification and Error*

Localization Legacy systems provides binary results only, but our scheme combines correctness verification along with error localization so the response values received from server not only checks the correctness, but also goes for error localization that is provides methods to locate data error/s.

The protocol for checking the correctness of data i. e challenge-response protocol for ith token on jth server can be done s follow:

- User knows the α_i and permutation key $k_{prp}^{(i)}$ for all servers.
- Index values specified in permutation key, used by servers to aggregate the rows specified in the permutation key to linear combination and generates $R_i^{(j)}$.
- When a user receives Ri$^{(j)}$ he takes away blind values and generates actual $R_i^{(j)}$.
- The secrete matrix is used to validate codeword which verifies the received values.

$$(R_i^{(1)}, \ldots ,R_i^{(m)}) \cdot P = (R_i^{(m+1)}, \ldots ,R_i^{(n)})$$

The equation shown above is if holds then the data is correct else there exists an error in the blocks in specified rows.

Once it have been detected that error lies in specified blocks then we can go for using the pre-computed tokens to identify where the errors lies in data. The $R_i^{(1)}$ is computed in same way as $v_i^{(1)}$,so it is very much easy to user to identify the server from which data is tampered. As follow equation:

$R_i^{(1)} = v_i^{(1)}$ where j Є (1,…,n)

D. File Retrieval and Error Recovery

As we are using the matrix file representation user can easily able to construe the file by downloading the data vectors from the first m servers which are only used for storing the data by users while remaining are used for storing the parity. The storage correctness assurance is probabilistic because of the random spot checking used by our scheme. As we are using unique parameters like (*i, j, l, r*), we can assure that the file is retrieved successfully.

When data corruption is detected response values received form servers and pre-computed token are compared, so that misbehaving server/s are identified. The erasure correcting code is used to correct the lost data and that recovered data blocks are redistributed. The error recovery is done by following:

- Corrupted block are detected amongst specified r rows.
- Suppose $s \le r$ servers are misbehaving.
- Download those blocks.
- S servers are treated as erasure and are used as recovery.
- Re-distribute blocks across servers.

IV. PROVIDING DYNAMIC DATA OPERATION SUPPORT

Till here we are only considered the static data, but in some cases the data stored in cloud is dynamic in nature like electronic documents, log files. Hence it is very important to consider the dynamic data case where user wants to perform the operations on data blocks like Update, Append, Delete and Insert with maintaining the correctness assurance of data. The simple way to do all such update operation is to download all files from all servers concatenate it and do operations. This scheme is totally inefficient, so it is necessary to handle dynamic data.

A. Update Operation:

As when user updates some data block/s then value of that block changes from f_{ij} to $f_{ij} + \Delta f_{ij}$. The Reed-Solomon code has linear property, so by using only Δf_{ij} update operation of data and parity can be done. The update matrix is as shown:

$$\Delta F = \begin{pmatrix} \Delta f_{11} & \Delta f_{12} & \dots & \Delta f_{1m} \\ \Delta f_{21} & \Delta f_{22} & \dots & \Delta f_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta f_{l1} & \Delta f_{l2} & \dots & \Delta f_{lm} \end{pmatrix} = (\Delta F_1, \ \Delta F_2, \ \dots \ \Delta F_m)$$

Note, unchanged blocks are denoted by 0. Parity vectors are maintained by multiplying ΔF by A ,so the update information for data vectors and parity vectors can be generated as:

$\Delta F \cdot A = (\Delta F_1, \Delta F_2, \dots, \Delta F_m, \Delta G^{(m+1)}, \dots, \Delta G^{(n)})$, $\Delta G^{(m+1)}$ denotes the update information of parity vectors.

As the pre-computed tokens having old value of updated data, it needed to replace it with new value. This can be done using homomorphic design of tokens .The a version number *var* used with the index for each blocks, it keeps the information about number of times the block is modified.

B. Delete Operation

When the data is stored on cloud sometimes user interested in deleting the data blocks from storage or sometimes certain data blocks are needed to be deleted. It is simple to delete data blocks but rather than directly deleting the data block, if we look at it it is just an update operation of data block. We simply keep the value of Δf_{ij} in ΔF as - Δf_{ij} along this modification are performed in both data and parity vectors.

C. Append Operation

For files stored in storage server, user wants to append data like to log files auditing records. Appending the data blocks is nothing but adding some extra rows at the bottom of matrix format for file F. As matrix shows, it has l rows initially so we have to add m rows for m data blocks at the end of file. User append blocks at end of file and it is denoted as ($f_{l+1,1}$, $f_{l+1,2}$, … $f_{l+1,m}$). Also append blocks for each parity server are calculated.

D. Insert Operation

An append operation is referred to perform the insert operation, to insert data block in at particular position without breaking the structure of whole file. By shifting all blocks by j+1, the $F_{[j]}$ block is inserted. The re-computation of many tokens is done because insert operation may affect the all subsequent blocks

CONCLUSIONS

In this paper we have discussed the problem of data security in cloud as well we proposed the effective and flexible scheme to ensure our data stored across distributed storage serves. Our scheme also provides dynamic data support. To ensure the erasure coded data our scheme uses distributed verification by utilizing homomorphic token. So the integrity of data and data error localization is done. The storage server where data corruption is detected during verification, we can guarantee the identification of misbehaving server. Our scheme is highly efficient against Byzantine failure and even with server colluding attacks.

.

ACKNOWLEDGMENT

I would like to express my gratitude towards my Seminar Guide Dr. B. F. Momin for their constant guidance during my seminar, and to the college, especially to the Computer Science and Engineering Department for making its facilities available to me.

I take this opportunity to express my sincere thanks to all the staff members of Information Technology Department for their help whenever required. Finally I express my sincere thanks to all those who helped me directly or indirectly in this seminar.

REFERENCES

[1].Joseph Baron, Sanjay Kotecha “Amazon Web Services Storage Options in the AWS Cloud” October 2013

[2]. Cloud Security Alliance “Top Threats to Cloud Computing V1.0” March 2010

[3]. Cong Wang, , Qian Wang,Kui Ren, Ning Cao, and Wenjing Lou,” Toward Secure and Dependable Storage Services in Cloud Computing” IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 5, NO. 2, APRIL-JUNE 2012

[4]. T. S. J. Schwarz and E. L. Miller, “Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage,” Proc. of ICDCS’ 06, pp. 12–12, 2006.

[5] J. Hendricks, G. Ganger, and M. Reiter, “Verifying Distributed Erasure coded Data,” Proc. 26th ACM Symposium on Principles of Distributed
Computing, pp. 139–146, 2007.

[6] Shuai Han, Jianchuan Xing “Ensuring Data Storage Security Through A Novel Third Party Auditor Scheme In Cloud Computing”, Proceedings of IEEE CCIS2011 978-1-61284-204-2/11/$26.00 ©2011 IEEE

[7] James Westall,James Martin" An Introduction to Galois Fields and Reed-Solomon Coding", School of Computing Clemson University Clemson, SC 29634-1906, October 4, 2010

[8] Xiao Zhang, Hong-tao Du ,Jian-quan Chen, Yi Lin, Lei-jie Zeng "Ensure Data Security in Cloud Storage" 978-0-7695-4355-0/11 $26.00 © 2011 IEEE DOI 10.1109/NCIS.2011.64