International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)

International Conference on Information Systems and Computing (ICISC-2013), INDIA.

# REMOTE DATA INTEGRITY IN CLOUD SECURITY SERVICES

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

[1] *Author, Department of CSE, KIT, Krishnankoil, Tamilnadu, India*
[2] *Co -Author, Department of CSE, KIT, Krishnankoil, Tamilnadu, India*

Email: dhanamjai55@rediffmail.com, ramesh_8607@yahoo.co.in

***Abstract***

**Cloud Structure delivers infrastructure, platform and software as a service which are made available in pay-as-you go model. Cloud storage is a model of networked online storage where data is stored in virtualized pools of storage which are generally hosted by third parties. Users interact with the cloud servers through Cloud Storage Providers (CSP) to access or retrieve the data. Users have no time to monitor the data online. Hence, it entrust the auditing task to an optional Third Party Auditor (TPA). Its main aim is to achieve the integration of storage correctness across multiple servers and data error localization. Homomorphic pre computation is used in this paper to identify the error corruption. User precomputes the token for the data file; server computes signature over specified blocks. If the signature is not matched with the precomputed token which denotes the data corruption. Byzantine Fault tolerant algorithm or data error localization algorithm is used to identify in which server the data gets corrupted or which server is not behaving properly. After detecting the error, reed Solomon algorithm is used to recover the corrupted data. In this paper, precompute the token for large number of data will improve the performance of error detection in storage services.**

***Keywords*-- Cloud computing, Storage verification, Error localization, Data storage, Tokens**

## I. INTRODUCTION

With the cloud it is easier than ever to consume resources from multiple data centers and providers, making low cost, redundant infrastructure possible for organizations of all sizes. Fully achieving the potential of highly scalable and resilient cloud-based applications requires a way to manage and automate your infrastructure and applications across data centers and providers. Cloud storage involves storing data on multiple virtual servers that are generally hosted by third parties. Organizations have to deal with to isolate the data. Individual users store their data redundantly across multiple physical servers. Make sure the storage correctness across multiple servers using distributed protocols. Focus on static data. Utilize homomorphic token, whenever data corruption detected and identify misbehaving servers. To provide a good balance, use algebraic property of token computation and coded data. Users safely delegate the integrity checking tasks to third party auditors. Cloud storage allows multiple users to access files simultaneously, which is great for sharing documents and passing materials back and forth - but a serious problem from a security standpoint.

With cloud storage, IT pros are concerned about potential data breaches or unauthorized releases of sensitive data. Detect data corruption that may be due to random Byzantine failures.

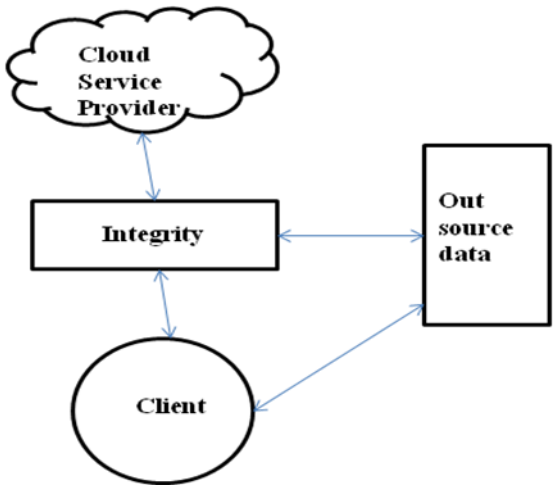


**Figure 1: Cloud Storage Services**

## II. WORKING ARCHITECTURE

*Cloud storage architecture has three entities:*

*User:* has data to be stored

*Cloud server (CS):* managed by CSP to provide data storage service

*Third Party Auditor (TPA):* (optional) trusted to access and expose risk of cloud storage services

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

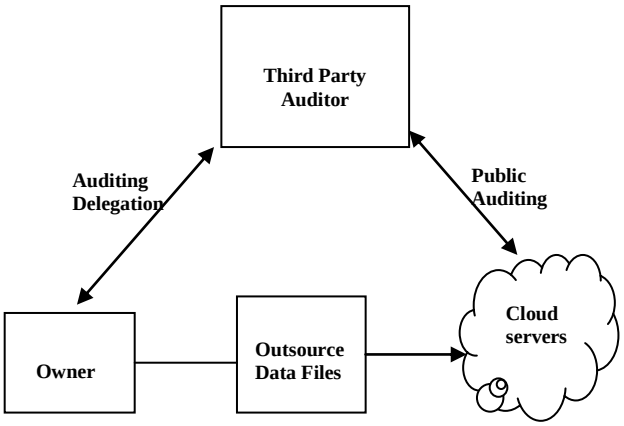


**Figure 2: System Model**

Users interact with the cloud servers via CSP to access or retrieve his data. User may need to perform block level operations on his data. Ensure user that their data are correctly stored and maintained.

Users have no time to monitor the data online. Delegate the data auditing tasks to an optional trusted TPA. Any possible leakage of user's outsourced data toward TPA through the auditing protocol should be prohibited.

- F – File to be stored. It is denoted as matrix of m equal- sized data vectors each consisting of l blocks.
- R – Dispersal matrix used for Reed-Solomon coding
- C – Encoded file matrix
- PRF – Pseudo random Function
- PRP – Pseudo Random Permutation
- Ver – Version number. It denotes number of times the block has been modified. At initial state, ver is 0.

## III. Cloud Storage Services

With cloud computing, users need not know about network and server maintenance. They can access different servers around the world without necessarily knowing where the servers are located.

### *3.1 File Distribution*

In cloud data storage, disperse the data file F across set of n=m+k distributed servers. R (m+k, k) is the Reed-Solomon erasure-correcting code used to create k redundancy parity vectors from m data vectors. By placing each m+k vectors on different servers, the data file can survive the failure of any k without any loss and with a space overhead k/m.

Each file F is represented as column vector.

All blocks are elements of Galois Field GF (2p).
Information dispersal matrix A can be written as,
A=(I|P)

Where I is the m*m identity matrix
P is the secret parity generation matrix.

**Table 1**
**Notations**

| | |
|---|---|
| l | Data Vector size in blocks |
| GF | Galois Field |
| n,m | Parity matrix with row and column vectors |
| P | Parity Generation Matrix |
| w | Parity Vector |

### *3.2 Homomorphic Token Computation*

To verify the correctness of user's data & to locate the errors, we entirely rely on the pre-computed verification tokens. These tokens are calculated before file distribution & they are very short. Compute the tokens by pseudorandom function PRF& pseudorandom permutation function PRP. Pre-computes short verification tokens on individual vector, each token covering a random subset of data blocks. Assume block size as 256 bits & r as 8 number of verification per indices. We have three data devices and three checksum devices. Then n=3 and m=3. Choose w=4, since 2w>n+m.

Construct P to be a 3*3 matrix, defined over GF(24)

*Algorithm: TOKEN PRE-COMPUTATION*

1. Begin
2. Choose file F to upload & encrypt the file using AES
3. Generate n*m Vector Matrix D on file F.
4. Create Reed Solomon Matrix P over Galois Field GF($2^w$)where w=4.
5. Generate Matrix C=D*P. It is Checksum Matrix created for fault tolerance.
6. Compute Token over Matrix C i.e., Compute Token(c,l,t,r) where l-block size, t- no. of tokens, r- indices per verification. Compute the tokens by pseudorandom function & pseudorandom permutation function.
7. Store these precomputed tokens on the main cloud server.
8. Disperse the file over the Cloud. i.e. Disperse File Matrix) D.
9. End.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

### 3.3 *Verifying Data distribution*

To eliminate the errors in storage systems key prerequisite is to locate the errors. The newly computed tokens from servers for each challenge are compared with precomputed tokens to determine the correctness of the distributed storage. This also gives information to locate potential data errors.

*Algorithm: CORRECTNESS VERIFICATION*

1. Begin Challenge, for, where n- total number of cloud servers.
2. Get TokenArr () Getting precomputed tokens from main cloud server.
3. HandleChallenge ()// Reading file blocks from all cloud servers for calculating new tokens.
4. Generate Vector Matrix D on all file blocks that are read in step 3.
5. Create Reed Solomon Matrix P
6. Generate Matrix C=D*P. On this matrix, new tokens will be computed.
7. Compute token on Matrix. ComputeToken(c,l,t,r)
8. If (PrecomputedToken=Newly Computed Token)then,

Data is intact .Else Data is Corrupt. For that, initiate the recovery.

9. End

### 3.4 *Error Recovery & File Retrieval*

Once the data corruption is detected, next important step is to recover the corrupted data and bring data storage back to consistent state. The comparison of precomputed tokens and received response values can guarantee the identification of misbehaving server. Therefore user can recover the corrupted data. Our system recovers data from backup server & distributes all data vectors to corresponding servers. This will results in successful recovery of corrupted data. But due to file splitting we made at the time of file distribution, user's need to recover file from all the servers. Error localization is limited to misbehaving servers only, i.e. servers giving false assurance of posing user's data.

*Algorithm: Error Recovery*

1. Assume the block corruptions have been detected among the specified rows;
2. Assume $s \leq k$ servers have been identified misbehaving
3. Download r rows of blocks from servers;
4. Treat s servers as erasures and recover the blocks.
5. Resend the recovered blocks to corresponding servers.

## IV. DYNAMIC OPERATIONS

In cloud data storage, there are many potential scenarios where data stored in the cloud is dynamic, like electronic documents, photos, or log files etc. Therefore, it is crucial to consider the dynamic case, where a user may wish to perform various operations of update, delete and append to modify the data file while maintaining the storage correctness assurance. The straightforward and trivial way to support these operations is for user to download all the data from the cloud servers and recompute the whole parity blocks as well as verification tokens. This would clearly be highly inefficient.

### *4.1. Update Operation*

In cloud data storage, sometimes the user may need to modify some data stored in the cloud, from its current value to a new one. Refer this operation as data update.

To perform update operation on particular data block client need to recalculate the verification token on updated data. Also client need to update this value of newly calculated token to all the replicas of file in storage cloud.

When user want to perform update operation, the splitted file from all storage servers is merged and given to the user to perform data updates.

Once user has finished with the updating the data, new tokens are calculated on whole file and they are stored on main cloud server. After this, updated file is splitted back and dispersed onto corresponding cloud storage servers. Update operations include modifying file, inserting data, as well as deleting data from file.

### *4.2. Delete Operation*

Sometimes, after being stored in the cloud, certain data may need to be deleted. The delete operation we are considering is a general one.

When user wants to delete some file, he can simply delete it. 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.

### *4.3. Append Operation*

In some cases, the user may want to increase the size of his stored data in file by adding data at the end of the data file, which we refer as data append operation. So in case of append operation whenever user append data to his file, new verification tokens are calculated & stored on main cloud server & file is splitted as before and dispersed among the cloud storage servers.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

In case of insert operation, we are treating as a part of update operation and we are relaying on update operation for insert operation.

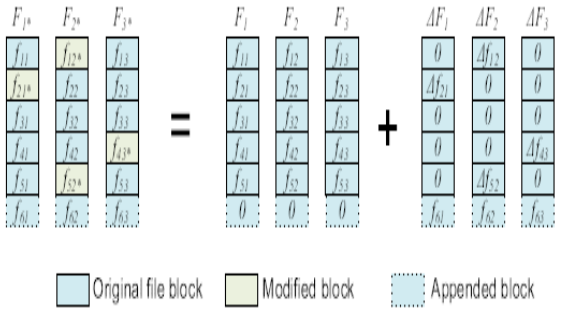


**Figure 3: Logical Representation of Dynamic Operations**

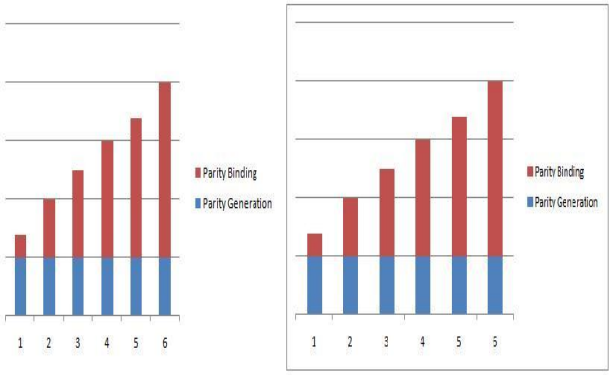


**Figure 4: Performance Comparison between two Different Parameters. a. m is fixed, k is decreasing b. m,k is fixed.**

Redundancy parity vectors are calculated via multiplying the file matrix F by secret parity generation matrix P.

Pick blocks from among the data and parity vectors to establish a set of m.k linear equations. Once having the knowledge of P, server can modify any part of the data blocks and calculate corresponding parity blocks.

K determines how many parity vectors are required before outsourcing. Growth of k means large number of parity blocks required to be blinded.

Take the Tokens dynamically to find the corruptions in the cloud servers. If we choose random data for token precomputation it will not check the entire data hence it cannot detect the corruptions systematically.

In this paper, we are evaluating the parity blocks with low cost overhead and increasing the performance speed. This performance can be achieved by large tokens on pre computation.

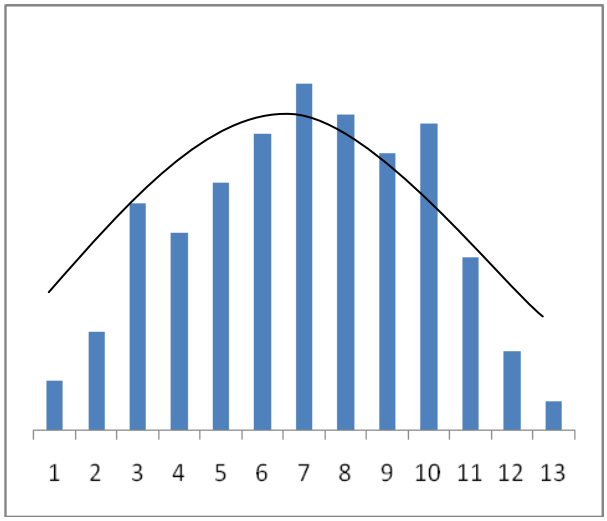


**Figure 5: Detection of data distribution when tokens computed dynamically**

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 is identified by the byzantine fault tolerance algorithm.

## V. CONCLUSION

Individuals or companies considering using cloud storage services are advised to check whether a cloud provider meets these security requirements.

As a major result, the paper shows that most of the analyzed cloud storage providers are aware of the extreme importance of data security and privacy; hence they have taken protection measures. The data is often of great value and its irrecoverable loss or damage could be a total disaster for its owner. This requires secure methods of preserving important data in order to prevent unrecoverable data loss, whilst constantly keeping up with increasing demands for storage space.

With the cloud it is easier than ever to consume resources from multiple data centers and providers, making low cost, redundant infrastructure possible for organizations of all sizes. Fully achieving the potential of highly scalable and resilient cloud-based applications requires a way to manage and automate your infrastructure and applications across data centers and providers.

REFERENCES

*1. Journal Papers*

[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

*2. Text Book*

[5] C.Cahin, R.Guerraoui, L.Radrigies, Introduction to reliable and secure distributed programming, 2nd edition, Published by Springer,2011.

*3. Conference Proceedings*

[6] 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.

[7] K. D. Bowers, A. Juels, and A. Oprea, "Hail: A high-availability and integrity layer for cloud storage," in Proc. of CCS'09, 2009, pp. 187–198.

[8] 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.

*4. Generic Website*

[9] http://www.distributedprogramming.net

[10] http://msdn.microsoft.com/en-us/library/windowsazure/ee706734.aspx

[11] http://www.sit.fraunhofer.de/content/dam/sit/en/studies/Cloud-Storage-Security_a4.pdf

[12] https://**cloudsecurity**alliance.org/

[13] http://www.opensciencegrid.org/bin/view/storage/HadoopInstallation

[14] http://www.thegeekstuff.com/2012/02/hadoop