Global Journal of Engineering Science and Research Management

# DYNAMIC PROVABLE DATA POSSESSION IN MULTI CLOUD ENVIRONMENT

**N.R. Rejin Paul*, Vedashree. E, Pooja. G.K, Suganya. P**
*Asst.professor ,Velammal Institute of Technology.
student,Velammal institute of Technology.
student,Velammal institute of Technology.
student,Velammal institute of Technology.

**KEYWORDS:** Multi cloud, Dynamic block generation, Verifiable integrity checking, Dynamic reallocation.

## ABSTRACT

In multi cloud environment, the remote data integrity checking has major part in the field of security. It ensures Data Integrity on User Uploaded data in Multi cloud maintaining security and privacy on cloud data and also provide access confidentiality through dynamic re-allocation of data at every user access. In our proposed systemfile is split into blocks using Dynamic Block generation Algorithm and stored in a Multi cloud environment. File Allocation Table (FAT) File System has proper Indexing and Metadata's for the different Chunks of the Cloud Storage. If Attacker corrupts data in Multi cloud, the continuous auditing process helps the Verifier to perform Block level and File level checking for remote data Integrity Checking using Verifiable Data Integrity Checking Algorithm. Cloud provides random blocks to Verifier for Integrity Checking which is to protect user privacy from Verifier (Third Party). File recovery is done by the Verifier automatically if the data gets corrupted during checking. User can complaint Cloud for File Recovery. At every access of the file by the user, blocks of the data will be dynamically reallocated between the Cloud Servers.

## INTRODUCTION

Cloud computing has become an important theme in the computer field. Essentially, it takes the information processing as a service, such as storage, computing. It relieves of the burden for storage management, universal data access with independent geographical locations. The foundations of cloud computing lie in the outsourcing of computing tasks to the third party. It entails the security risks in terms of confidentiality, integrity and availability of data and service. The issue to convince the cloud clients that their data are kept intact is especially vital since the clients do not store these data locally. Remote data integrity checking is a primitive to address this issue. For the general case, when the client stores his data on multi cloud servers, the distributed storage and integrity checking are indispensable. On the other hand, the integrity checking protocol must be efficient in order to make it suitable for capacity-limited end devices. Thus, based on distributed computation, we will study distributed remote data integrity checking model and present the corresponding concrete protocol in multi cloud storage. The current works are based on the development of the information society has caused a vast majority of such infrastructures to critically depend on the correct operation of the underlying information systems—that is why such information systems are often referred to as critical information infrastructures. Indeed, any service interruption, malfunction or, even worse, partial or total destruction of those information systems as a consequence of an accident or a terrorist attack can result in huge material or even human casualties. This reality supports the claim that one of the biggest challenges in infrastructure protection is the underlying information security problem information systems should be dependable. Storage outsourcing is a rising trend which prompts a number of interesting security issues, many of which have been extensively investigated in the past. However, Provable Data Possession (PDP) is a topic that has only recently appeared in the research literature. The main issue is how to frequently, efficiently and securely verify that a storage server is faithfully storing its client's (potentially very large) outsourced data. The storage server is assumed to be untrusted in terms of both security and reliability. (In other words, it might maliciously or accidentally erase hosted data; it might also relegate it to slow or off-line storage.) The problem is exacerbated by the client being a small computing device with limited resources. Prior work has addressed this problem using either public key cryptography or requiring the client to outsource its data in encrypted form. [1]. In our paper verifier sets an audit time so that the data can be checked in that particular timing and the file is not visualize to the attacker. Here a FAT (File Allocation Table) is used to store the index of the data which is used to identify where the data is stored.

## RELATED WORK

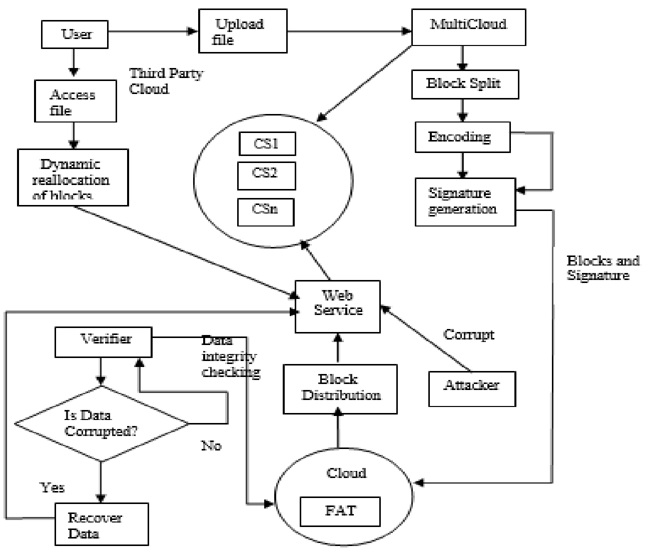
Remote data integrity checking is a crucial technology in cloud computing, it enables a server to prove to an auditor the integrity of a stored file. The clients massive data is outside his control. The malicious cloud server may corrupt the clients' data in order to gain more benefits. Existing protocols can permit to check that a remote server can access an uncorrupted file with the help of a third party verifier The major challenge to RIC is the data confidentiality and privacy of data against potential adversary who might be interested in the information of the stored data item.

The existing sytem talks about PKI (public key infrastructure), is a system for the creation, storage, and distribution of digital certificates which are used to verify that a particular public key belongs to a certain entity. Provable data possession protocol needs public key certificate distribution and management.PDP scheme, which allows checking data possession without retrieving the data from the server and without having the server access the entire data.The model of PDP can be used for RIC,a client that has stored data at an untrusted server can verify that the server possesses the original data without retrieving it.Here the client preprocesses the data and then sends it to an untrusted server for storage while keeping a small amount of meta-data. It usually supports large sets in distributed storage systems [1,3,]. It will incur considerable overheads since the verifier will check the certificate when it checks the remote data integrity.

PPDP is a matter of crucial importance when client cannot perform the remote data possession checking ,this protocol is designed using bilinear pairing[1,6].In addition to the heavy certificate verification, the system also suffers from the other complicated certificates management such as certificates generation, delivery, revocation, renewals, etc. In cloud computing, most verifiers only have low computation capacity. Identity-based public key cryptography can eliminate the complicated certificate management.Recently,there has an increased intensity in research on IBC.It enjoys most of the functionality of public key cryptography without need for certificates and the problems that these bring.Many proposal for identity based schemes related to signatures,signcryption,key agreement and so on have been proposed[13,14].In the existing paper,FAT file system is exposed to the verifier.Here user privacy will be affected because of providing the user's file to verifier.The user data always stores in static location of the cloud server which may help the attacker to trace the user's block by analyzing traffic patterns in each file access by user. Remote data integrity checking on MultiCloud Storage is not done yet.

## PROPOSED WORK

In MultiCloud environment, remote data integrity checking is required to secure user's data. User will upload file to Cloud. This file is split into blocks using Dynamic Block generation Algorithm and stored in a MultiCloud environment. File Allocation Table (FAT) File System has proper Indexing and Metadata's for the different Chunks of the Cloud Storage. If Attacker corrupts data in MultiCloud, the continuous auditing process helps the Verifier to perform Block level and File level checking for remote data Integrity Checking using Verifiable Data Integrity Checking Algorithm.Cloud provides random blocks to Verifier for Integrity Checking which is to protect user privacy from Verifier (Third Party). File recovery is done by the Verifier automatically if the data gets corrupted during checking. User can complaint Cloud for File Recovery.At every access of the file by the user, blocks of the data will be dynamically reallocated between the Cloud Servers.



## PROPPOSED METHODOLOGY

**Admin Configuration:**
Admin configure MultiCloud server setup.Server IP Address and Port number is given by the admin for each Cloud. If the admin has to reconfigure the old MultiCloud server setup, it can be done.For old server setup, FAT file can be modified or remain same.

**Data Upload and Block Split:**
User has an initial level Registration Process at the web end. After Registration, user can upload files to the server. Uploaded files will be stored in a Server.When the user upload the data to different cloud by the time it is Splitted into different blocks using Dynamic block generation Algorithm and each block will be appended with Signatures before Storing the data in FATFS.Signature generated using MD5 Algorithm. Also the data gets encoded using for Base64 Algorithm.

**Verifier Integrity Checking:**
Cloud allocates random combination of all the blocks to the Verifier, instead of the whole file. This is to protect user privacy from a third party (Verifier). Verifiable Data Integrity Checking Algorithm is done in two steps: Block Checking and File Checking. In Block Checking step: Three signatures are generated for Block level Checking.

1) A signature of a block retrieved from a FATFS
2) A new signature is generated for block to be
3) Checked
4) A Signature is retrieved from the block appended with the signature which is stored in the Cloud

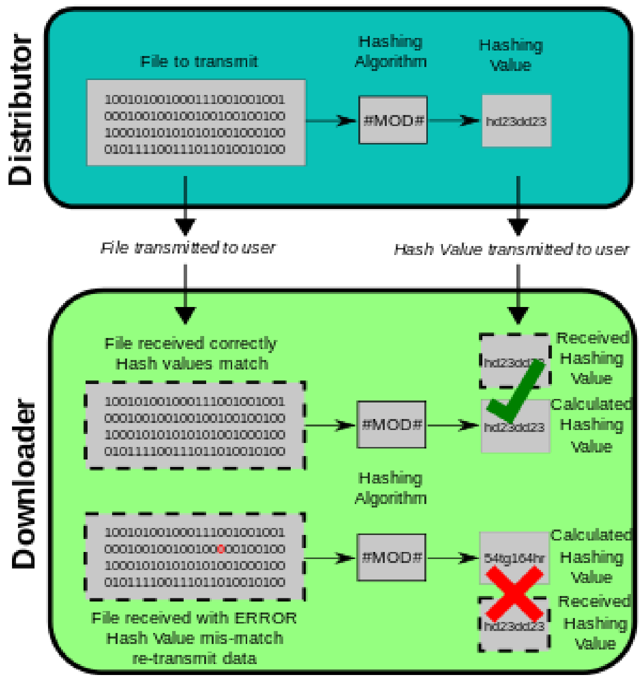
The above three signatures are cross checked for Block level Integrity Checking. And the block contents are appended to verify with File level Integrity Checking.

**Automatic and On Demand File Recovery:**
Attacker can corrupt data in any one of the cloud servers. User can complaint to the Cloud if the user file get corrupted. Verifier hold the others verification process and check the corrupted data.Recovery Process will be done by the verifier automatically when data gets corrupted.Whenever user access file, Blocks will be reallocated dynamically and updated in FAT

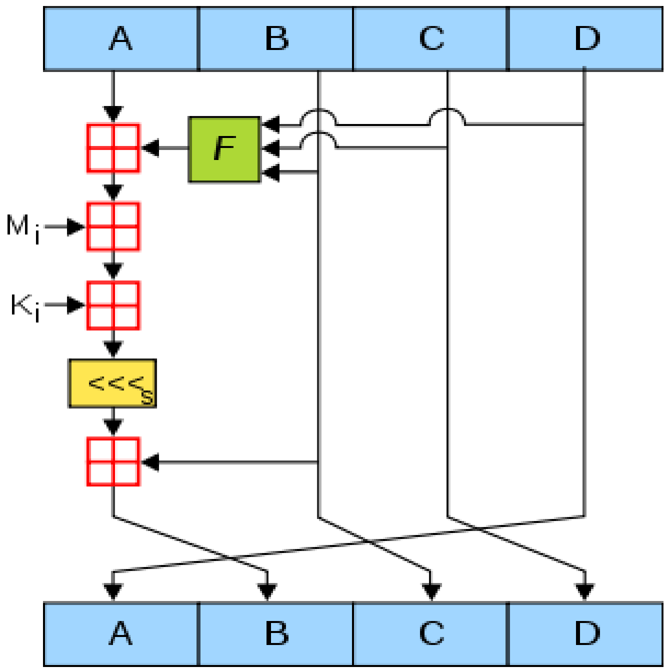
**ALGORITHM:**

***BASE-64 :***
Base64 algorithm is designed to encode any binary data, an stream of bytes, into a stream of 64-printable characters. Base64 encoding algorithm was first presented in "RFC 1421 - Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures" in 1993 by John Linn. It was later modified slightly in "RFC 1521 - MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies" in September 1993 by N. Borenstein. The Base64 encoding process is to: Divide the input bytes stream into blocks of 3 bytes. Divide 24 bits of each 3-byte block into 4 groups of 6 bits. Map each group of 6 bits to 1 printable character, based on the 6-bit value using the Base64 character set map. If the last 3-byte block has only 1 byte of input data, pad 2 bytes of zero (\x0000). After encoding it as a normal block, override the last 2 characters with 2 equal signs (==), so the decoding process knows 2 bytes of zero were padded. If the last 3-byte block has only 2 bytes of input data, pad 1 byte of zero (\x00). After encoding it as a normal block, override the last 1 character with 1 equal signs (=), so the decoding process knows 1 byte of zero was padded. Carriage return (\r) and new line (\n) are inserted into the output character stream. They will be ignored by the decoding process.

***MD5:***
MD5 digests have been widely used in the software world to provide some assurance that a transferred file has arrived intact. For example, file servers often provide a pre-computed MD5 (known as md5sum) checksum for the files, so that a user can compare the checksum of the downloaded file to it. Most unix-based operating systems include MD5 sum utilities in their distribution packages; Windows users may install a Microsoft utility,[44][45] or use third-party applications. Android ROMs also utilize this type of checksum.



However, now that it is easy to generate MD5 collisions, it is possible for the person who created the file to create a second file with the same checksum, so this technique cannot protect against some forms of malicious tampering. Also, in some cases, the checksum cannot be trusted (for example, if it was obtained over the same channel as the downloaded file), in which case MD5 can only provide error-checking functionality: it will recognize a corrupt or incomplete download, which becomes more likely when downloading larger files.MD5 can be used to store a one-way hash of a password, often with key stretching.[46][47] Along with other hash functions, it is also used in the field of electronic discovery, in order to provide a unique identifier for each document that is exchanged during the legal discovery process. This method can be used to replace the Bates stamp numbering system that has been used for decades during the exchange of paper documents.



ISSN 2349-4506
Impact Factor: 2.265

Figure 1. One MD5 operation. MD5 consists of 64 of these operations, grouped in four rounds of 16 operations. *F* is a nonlinear function; one function is used in each round. $M_i$ denotes a 32-bit block of the message input, and $K_i$ denotes a 32-bit constant, different for each operation. $_s$ denotes a left bit rotation by *s* places; *s* varies for each operation. denotes addition modulo $2^{32}$.MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit words); the message is padded so that its length is divisible by 512. The padding works as follows: first a single bit, 1, is appended to the end of the message. This is followed by as many zeroes as are required to bring the length of the message up to 64 bits fewer than a multiple of 512. The remaining bits are filled up with 64 bits representing the length of the original message, modulo $2^{64}$.The main MD5 algorithm operates on a 128-bit state, divided into four 32-bit words, denoted *A*, *B*, *C*, and *D*. These are initialized to certain fixed constants. The main algorithm then uses each 512-bit message block in turn to modify the state. The processing of a message block consists of four similar stages, termed *rounds*; each round is composed of 16 similar operations based on a non-linear function *F*, modular addition, and left rotation. Figure 1 illustrates one operation within a round.

There are four possible functions *F*; a different one is used in each round:

$$
\begin{aligned}
F(B,C,D) &= (B \wedge C) \vee (\neg B \wedge D) \\
G(B,C,D) &= (B \wedge D) \vee (C \wedge \neg D) \\
H(B,C,D) &= B \oplus C \oplus D \\
I(B,C,D) &= C \oplus (B \vee \neg D)
\end{aligned}
$$

denotes the XOR, AND, OR andNOT operations respectively.

## EXPERIMENTAL RESULT

Remote data integrity checking is done here to secure user's data. This file is split into block and stored in a MultiCloud environment. Random blocks is verified to protect user privacy from Verifier. File recovery is done for the corrupted data.At each access of the file the data will be dynamically reallocated.

## CONCLUSIONS

Thus the remote data integrity checking is done by the verifier effectively for maintaining security and privacy in MultiCloud. Data Recovery is done on Integrity checking process when data gets corrupted.

## REFERENCE

1. Huaqun Wang,'' Identity-Based Distributed Provable DataPossession in Multicloud Storage''IEEETrans,2014,pp.328-340.
2. F. Sebe´, J. Domingo-Ferrer, A. Martı´nez-Balleste´, Y. Deswarte,and J. Quisquater, ''Efficient Remote Data Integrity Checkingin Critical Information Infrastructures,'' IEEE Trans. Knowl.Data Eng., vol. 20, no. 8, pp. 1034-1038, Aug. 2008.
3. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner,Z. Peterson, and D. Song, ''Provable Data Possession atUntrusted Stores,'' in Proc. CCS, 2007, pp. 598-609.
4. G. Ateniese, R. DiPietro, L.V. Mancini, and G. Tsudik, ''Scalableand Efficient Provable Data Possession,'' in Proc. SecureComm,2008, pp. 1-10.Fig. 8. Probability curve PX of detecting the corruption.WANG: IDENTITY-BASED DISTRIBUTED PROVABLE DATA POSSESSION IN MULTICLOUD STORAGE 339
5. C.C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia,''Dynamic Provable Data Possession,'' in Proc. CCS, 2009,pp. 213-222.
6. H.Q.Wang. (2013, Oct./Dec.). Proxy Provable Data Possession inPublic Clouds. IEEE Trans. Serv. Comput. [Online]. 6(4), pp. 551-559.Available: http://doi.ieeecomputersociety.org/10.1109/TSC.2012.35
7. Y. Zhu, H. Hu, G.J. Ahn, andM. Yu, ''Cooperative Provable DataPossession for Integrity Verification in Multicloud Storage,''IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231-2244,Dec. 2012.
8. Y. Zhu, H.Wang, Z. Hu, G.J. Ahn, H. Hu, and S.S. Yau, ''EfficientProvable Data Possession for Hybrid Clouds,'' in Proc. CCS, 2010,pp. 756-758.

9. R. Curtmola, O. Khan, R. Burns, and G. Ateniese, ‘‘MR-PDP:Multiple-Replica Provable Data Possession,’’ in Proc. ICDCS,2008, pp. 411-420.
10. A.F. Barsoum and M.A. Hasan, ‘‘Provable possession and replication of data over cloud servers,’’ Centre Appl. Cryptogr.Res., Univ. Waterloo, Waterloo, ON, Canada, Rep. 2010/32. [Online]. Available: http://www.cacr.math.uwaterloo.ca/ techreports/2010/cacr2010-32.pdf
11. Z. Hao and N. Yu, ‘‘A Multiple-Replica Remote Data PossessionChecking Protocol with Public Verifiability,’’ in Proc. 2nd Int. Symp. Data, Privacy, E-Comm., 2010, pp. 84-89.
12. A.F. Barsoum and M.A. Hasan, ‘‘On verifying dynamic multipledata copies over cloud servers,’’ Int. Assoc. Cryptol. Res., New York, NY, USA, IACR eprint Rep. 447, 2011. [Online]. Available: http://eprint.iacr.org/2011/447.pdf
13. A. Juels and B.S. Kaliski, Jr., ‘‘PORs: Proofs of Retrievability forLarge Files,’’ in Proc. CCS, 2007, pp. 584-597.
14. H.W. Lim, ‘‘On the Application of Identity-Based Cryptographyin Grid Security,’’ Ph.D. dissertation, Univ. London, London,U.K., 2006.
15. D. Boneh, B. Lynn, and H. Shacham, ‘‘Short Signatures from theWeil Pairing,’’ in Proc. ASIACRYPT, vol. 2248, LNCS, 2001,pp. 514-532
16. Q. Zheng and S. Xu, ‘‘Fair and Dynamic Proofs of Retrievability,’’in Proc. CODASPY, 2011, pp. 237-248.
17. Y. Dodis, S. Vadhan, and D. Wichs, ‘‘Proofs of Retrievability viaHardness Amplification,’’ in Proc. TCC, vol. 5444, LNCS, 2009, pp. 109-127.
18. Y. Zhu, H. Wang, Z. Hu, G.J. Ahn, and H. Hu, ‘‘Zero-KnowledgeProofs of Retrievability,’’ Sci. Chin. Inf. Sci., vol. 54, no. 8, pp. 1608-1617, Aug. 2011.
19. C. Wang, Q. Wang, K. Ren, and W. Lou, ‘‘Privacy-PreservingPublic Auditing for Data Storage Security in Cloud Computing,’’ in Proc. IEEE INFOCOM, Mar. 2010.
20. Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, ‘‘Enabling PublicAuditability and Data Dynamics for Storage Security in Cloud Computing,’’ IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 5, pp. 847-859, May 2011.
21. C. Wang, Q.Wang, K. Ren, N. Cao, and W. Lou, ‘‘Toward Secure and Dependable Storage Services in Cloud Computing,’’ IEEETrans. Serv. Comput., vol. 5, no. 2, pp. 220-232, Apr./June 2012.
22. Y. Zhu, G.J. Ahn, H. Hu, S.S. Yau, H.G. An, and S. Chen. (2013, (Apr./June). Dynamic Audit Services for Outsourced Storages in Clouds. IEEE Trans. Serv. Comput. [Online]. 6(2), pp. 227-238. Available: http://doi.ieeecomputersociety.org/10.1109/TSC. 2011.51.
23. O. Goldreich, Foundations of Cryptography: Basic Tools. Beijing,China: Publishing House of Electronics Industry, 2003, pp. 194-195.
24. D. Boneh and M. Franklin, ‘‘Identity-Based Encryption from theWeil Pairing,’’ in Proc. CRYPTO, vol. 2139, LNCS, 2001,pp. 213-229.
25. A. Miyaji, M. Nakabayashi, and S. Takano, ‘‘New ExplicitConditions of Elliptic Curve Traces for FR-Reduction,’’ IEICE Trans. Fundam., vol. E84A, no. 5, pp. 1234-1243, May 2001.
26. K.D. Bowers, A. Juels, and A. Oprea, ‘‘Proofs of Retrievability:Theory and Implementation,’’ in Proc. CCSW, 2009, pp. 43-54..
27. H. Shacham and B. Waters, ‘‘Compact Proofs of Retrievability,’’in Proc. ASIACRYPT, vol. 5350, LNCS, 2008, pp. 90-107.
28. S. Yu, K. Ren, and W. Lou, ‘‘FDAC: Toward Fine-GrainedDistributed Data Access Control in Wireless Sensor Networks,’’ IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 4, pp. 673-686, Apr. 2011.
29. S. Yu, K. Ren, and W. Lou, ‘‘Attribute-Based on-DemandMulticast Group Setup with Membership Anonymity,’’ Calc. Netw., vol. 54, no. 3, pp. 377-386, Feb. 2010.
30. P.S.L.M. Barreto, B. Lynn, and M. Scott, ‘‘Efficient Implementationof Pairing-Based Cryptosystems,’’ J. Cryptol., vol. 17, no. 4, pp. 321-334, Sept. 2004.
31. A.F. Barsoum and M.A. Hasan, ‘‘Integrity Verification ofMultiple Data Copies’’