RESEARCHERID
THOMSON REUTERS

[Pote* *et al.*, 6(6): June, 2017]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

OPOR: PROOF OF RETRIEVABILITY IN CLOUD COMPUTING WITH RECOVERY AND REMOTE INTEGRITY CHECK

Shubham Pote[1], Vipul Jain[2] , Rutuja Shinde[3], Prasanna Paigude[4], Prof.Deepali Ahir [5]
(Computer Engineering ,Modern Education Society's College Of Engineering,
Pune , Savitribai Phule Pune University)

DOI: 10.5281/zenodo.802818

ABSTRACT

Cloud computing is popular ,and adopted because there is many security and privacy. A issue found in cloud storage is, when client out-source data to the cloud storage, the clients don't know that their data is damaged or not. Also the computational burden is too high. To tackle the issue, OPoR, another distributed storage method including a distributed storage server (DSS) and a TPA is proposed here. TPA is thought to be semi-legitimate. Specifically, we consider the assignment of permitting the TPA for the cloud clients, to pre-process the information before transferring to the DSS and later confirming the information quality. OPoR outsources the overwhelming calculation of the label era to the cloud review server and takes out the contribution of client in the examining and in the pre-processing stages. Besides, we secure the Proof of Retrievability(PoR) model to support information integrity, and in addition assurance security against reset assault dispatched by the DSS in the transfer stage.

KEYWORDS: Distributed Storage Server (DSS), Outsource Proof Of Retrievability, Cloud Audit Server, Cloud Service Provider(CSP) , TPA.

INTRODUCTION

The Outsourced Proof of Retrievability (OPoR) is an archive,that provides a brief proof that the user can recover the targeted file. OPoR is an important tool set for semi trusted on-line archive. The users can view their file in the archive but they can not adjust their data in the file. The goal of a OPoR is to ensure these checking without user's having to download the file themselves. And also in OPoR the cloud storage must prove to a verifier for the client that is storing all the client's data. Although OPoR provide many advantages some of disadvantages are also found with PoR. The users or the clients can not change their data in the file. Some security problems are also found and computational cost is found to be very high with PoR. Also some integrity issues also found. To overcome all the challenges faced by PoR a new method OPoR (Outsourced Proof of Retrievability) is used . It include two independent servers the cloud audit server and the cloud storage server. The cloud audit server has some additional capabilities that the client does not have and it is also responsible for pre-processing the data instead of the client. By using OPoR dynamic data operations can be performed .And all the security concerns are avoided .

Provable data possession (PDP): PDP technique are used by clients to check the data that is stored on cloud servers. It ensures client that the data is untouched. Client maintain some constant amount of meta-data to verify proof. It supports large data set in widely distributed networks.
Proof of Retrievability (PoR): In PoR system data storage centre must have to give a proof to a data owner (client) that client's data is intact on storage. Also it allow client to recover his outsourced data. In PoR prover and verifiers both doesn't needed to have knowledge of file F .

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

REALTED WORK

In [1] new scheme is proposed to check the integrity of outsourced data. TPA is offered to scale down the computational load of client. TPA does the task of auditing the data by challenging the CSS. Scheme provide public verifiability along with dynamic data operation . In PoR model provide safety against the reset attack launched by cloud storage server within the upload phase. TPA stores the tag of file to be uploaded and uses these tags to check integrity.\\

In [2] author defined a PDP model. It gives probabilistic proof that third party stored a files. User can access small block of file for producing the proof. Challenge and responses method is used in this technique. Some amount of metadata of client's data is stored at client side. Locally stored meta-data is used to verify proof which is given by servers .Client gives challenge to server for proving possession and wait for response. Server then compute and sent proof to client . Metadata are used to check correctness of response. RSA based Homomorphic variable tags are used to achieve goal. PDP accesse random sets of block and sample servers storage. Limitations of PDP's it gives only probabilistic proofs not a deterministic proofs . It can not supports dynamic data possession.\\

In [3] this a new scheme known as proof of retrievability (POR) is proposed . Using this schemes ,verifiers (users) can determine that whether Prover (servers) hacked his files or not .Schemes uses sentinels(called disguised blocks). Sentinels are hidden among usual file block for detecting data amendment by way of the server. Verifier challenge prover by specifying locations where sentinels are collected and asking to return associated value. Values are compared then check integrity of data. In this approach single cryptographic key is calculated and stored by verifier. Key is calculated using key hash algorithm. Error resiliency of their system is improved due to error correction code . This schemes increases larger storage requirement and computational overhead on prover.\\

In [4] this proposed new technique to obtain PoR . Two schemes are proposed in this. Pubic verifiability is implemented in 1st scheme. Here shortest query response of any POR obtained which is secure in the random oracle model. Second scheme provides shortest response with private retrivability. It is secure in standard model. Two homomorphic authenticators are used. 1st is based on PRF's and 2nd based on BLS signature. Only one authentication values is allowed in both schemes. Here, erasure encoded file is broken into n blocks by users. Each file block is accompanied by way of authenticator of equal size. Use of BLS signature give smaller sized proof as compared with RSA. It also accept higher error rate. But this scheme still work on static data only, dynamic data updates are not supported.\\

In [5] PDP model is expanded. Verifiable updates on stored data are provided. It makes use of new variation of authenticated dictionaries. These dictionaries are centered on rank knowledge. Rank knowledge is used for organizing dictionary entries. To check the integrity of file blocks, authentication skip list is used. Untrusted server stores File F and its skip list. Root meta-data is stored at client side. File f is divided into blocks. Client issues question atRank(i) to the server when he desires to verify integrity of block I. Server then computes tag T(i) as its proof and send to client. Clients compare proof given by server with stored meta-data and check for integrity. Also to update the data client issue atRank(i) (for insertion) and atRank(i-1) (for deletion).It does not allow for public variability of the stored data.
Scheme proposed in Paper [6] provides provable security and desirable efficiency simultaneously. Two servers are used. Particularly one for auditing and other for storing data. Third party Auditor (TPA) is used for auditing purpose. TPA screens information stored in cloud storage as well as transactions between data owner and cloud storage server (CSS). Public verifiability is provided. All the Computation is done by server instead of client. This leads to reduction of computational overhead at client side. Security of this scheme is analysed under variant of [2] which supports public verifiability. This is the game between challenger C (client) and storage server (adversary A) played to get proof of retrievability from Adversary A. If proof is valid for fraction of challenges, client can extract the file F.

EXISTING SYSTEM

The existing scheme can simultaneously provide provable security in the more enhanced security model and enjoy excepted efficiency, there is no scheme can resist reset attacks while supporting efficient public verifiability and dynamic data operation simultaneously PoR model is first to supports dynamic update operation and security against reset attacks in a verification schemes. The robustness against reset attack assure that a malicious storage servers can not gain any advantage of passing the verification of an incorrectly stored files by resetting the client (or the audit server) in the upload phase. We'll see that most of existing PoR schemes can not ensure this security for cloud storage.

SYSTEM OVERVIEW

An arch. for cloud data storage is shown in Fig. 1. The architecture consist of three different network entities known as Clients. An entity that has large data files to be stored in the cloud. Cloud storage server . An entity ,which is managed by cloud service provider.Cloud audit server. A TPA, which has capabilities and expertise that client do not have.In the cloud paradigm, the clients outsource their data to the TPA also known as the cloud audit server to be relieved of the stress of storage and computation. As clients no longer access their data , they should ensure and check that their data are being correctly stored and maintained .

The third party auditor then upload their data to the cloud storage which is managed by the (CSP)cloud service provider.

Figure:

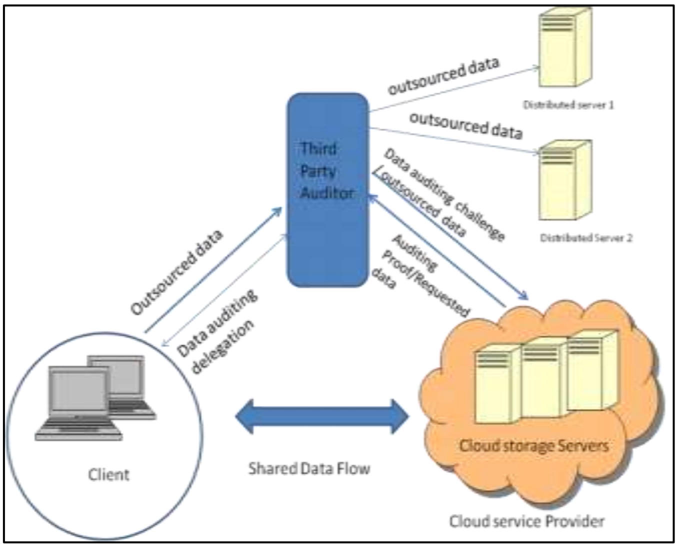

Figure 2. System Arch.

TABLES

Table 1.Overview of the proof of storage schemes

POS scheme	Confidentiality	Integrity	Availability	Public Verifiability	Type
PDP	Yes	Yes	Yes	Yes	Static
POR for large files	Yes	Yes	Yes	Yes	Static
Compact POR	Yes	Yes	Yes	Yes	Static
DPDP	Yes	Yes	Yes	No	Dynamic
POR with public	Yes	Yes	Yes	Yes	Dynamic

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

auditing					

PROBLEM STATEMENT AND SOLVING APPROACH

The main objective of this work is to find out whether the user's outsourced data is original or whether it is affected by some malicious intruder. For this auditing is performed with the help of hash values.To reduce the computational burden of making hash values and integrity verification at client side , TPA(Third Party Auditor) is introduced. Also public verifiability and dyanamic data operation are provided. PoR model is the first to support dynamic update operations and security against reset attack in a verification scheme. The robustness against reset attack ensures that a malicious storage server can never gain any advantage of passing the verification of an incorrectly stored file by resetting the client (or the audit server) in the upload phase.

Also recovery of deleted file is done by TPA. All the process is transparent to user.AES algorithm is used for encryption of file. And SHA256 is used for hashing purpose. Use of these algorithms improves security of file.

PROPOSED SYSTEM IMPLEMENTATION

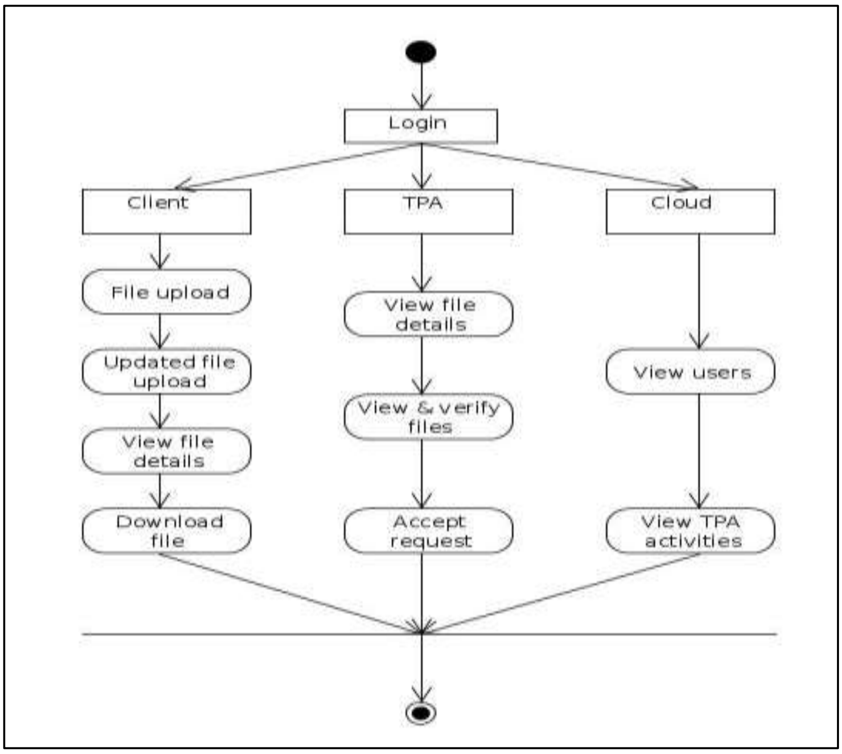

Figure 2.System Flow

The System flow for this gives a complete detailed views of the functionalities provided which will not only facilitate the user to store documents and search, but also we provide advance security to file and we also ensure the integrity of file ,we show how the general client can access view and search the files and share file, that have public scope . Additionally we have also added a admin functionality where in the admin can manage entire system, all through web interface.

- System contain main 3 entities which are client,cloud and TPA.

Client - Client is a public which can access this service. Client can select and upload file at his side. At the time of uploading one random key is generated at back end as secret key for a AES encryption algorithm.File is divided in 5 equal parts/chunks the file parts are encrypted at back end while uploading and at same the time for same

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

parts Hash value is generated using SHA256. Client can share file between multiple clients for that trapdoor is created and 1 key to open trapdoor is Mailed to registered mail id of shared client to open trapdoor and search the shared file.

Client can download file if it is unchanged or corrupted.If it is corrupted he can request to TPA for verification its integrity.

TPA - TPA works as a admin to this sysytem. He can handle or manage all work of verification of file,integrity checking using SHA256. He can view all clients an their request status which is Pending or Verified. Also TPA store file database at his side.If file are corrupted then he recover file from cloud database.

Cloud - Cloud is used to Deploy web services and to store all database and if file is corrupted at TPA side the TPA can retrieve previous file from cloud server.

MATHEMATICAL MODEL

- **Procedure (P):**

$P=\{S, Up, H^T{}_{n}, H^C{}_{n}, Ud, R\}$

Where,

1. Setup (Key generation)

S= {K1, K2, .Kn} Where, K is the set of keys generated.

Setup (security Parameter) $\longrightarrow$ Key K

2. Upload

Up= {Up1, Up2, Upn} Where Up is the set of files uploaded to cloud storage.

Ek (F, K) $\longrightarrow$ UPi

F is encrypted and stored to cloud server.

3.Integrity verification using hash values

SHA256(F*) $\longrightarrow$ H

- $H^T = \{ H^T{}_1, H^T{}_2, \ldots\ldots\ldots\ldots, H^T{}_n \}$

where $H^T{}_n$ is set of hash values stored at TPA.

- $H^c = \{ H^c{}_1, H^c{}_1, \ldots\ldots\ldots\ldots, H^C{}_1\}$

where $H^C{}_n$ is set of hash values stored at cloud storage server.

$$\text{Equal}(H^T{}_I, H^c) = \begin{cases} 1 & \text{if F* passes verification} \\ 0 & \text{if F* fails verification} \end{cases}$$

4. Update

Ud= {Ud1,Ud2…Udn} where Ud is the set of files to be updated

5. Recovery

$$\text{Request (t)} \xrightarrow{Recover(F*,t)} \text{Recovered file Fr*}$$

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

APPLICATION RESULTS

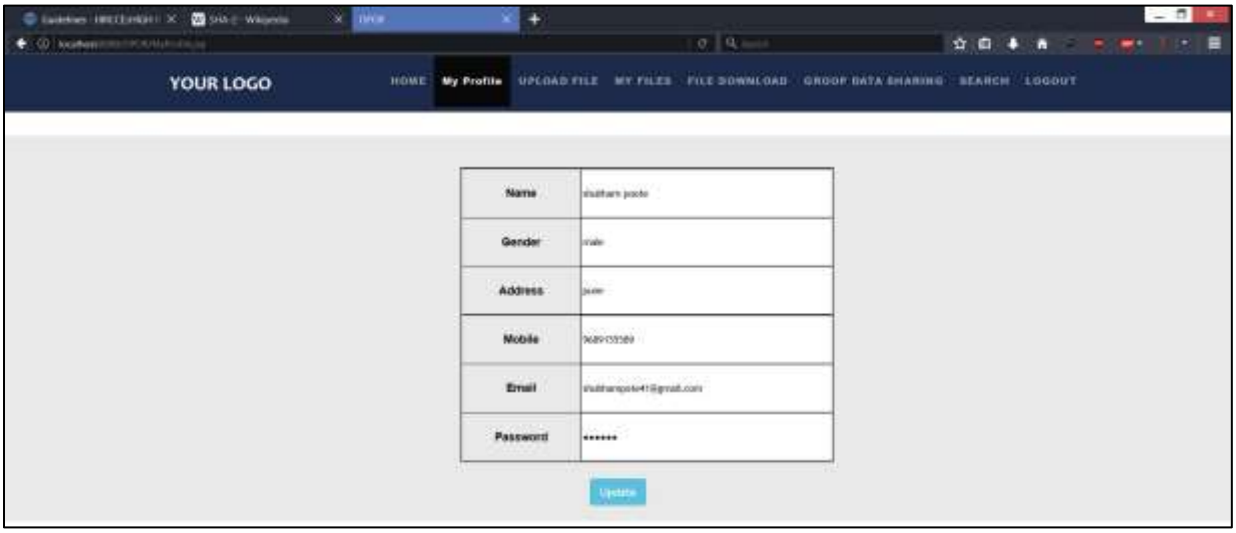

Figure 3.User Profile

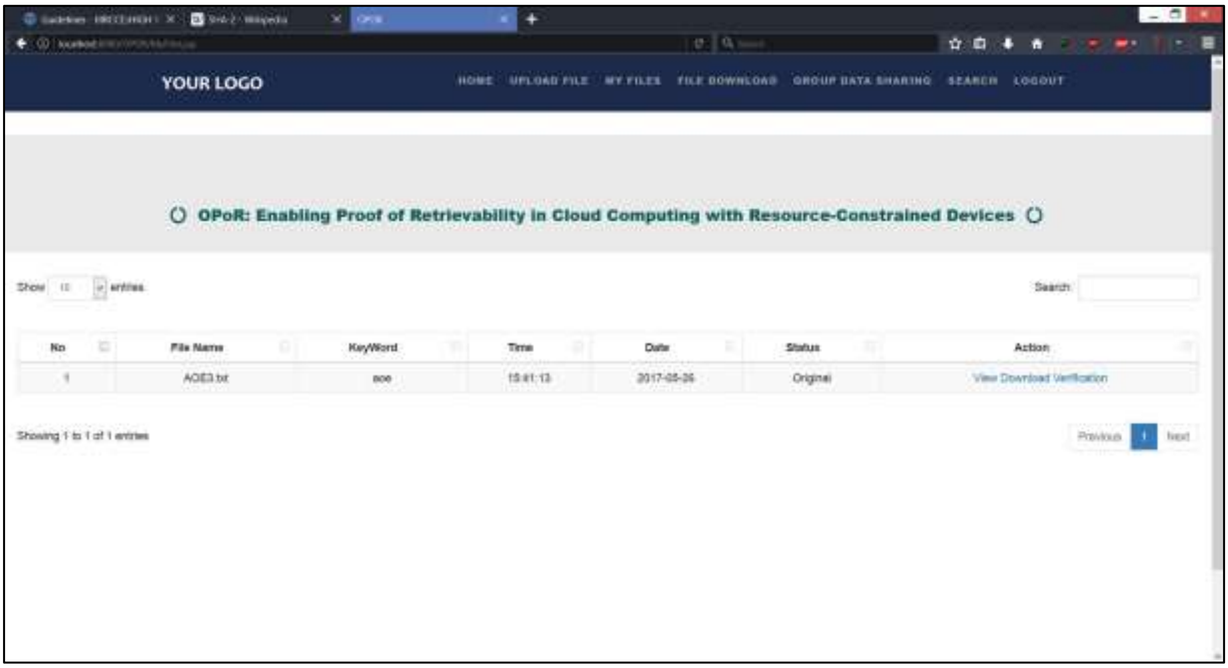

Figure 4.List of uploaded files at client side

Figure 5.File sharing

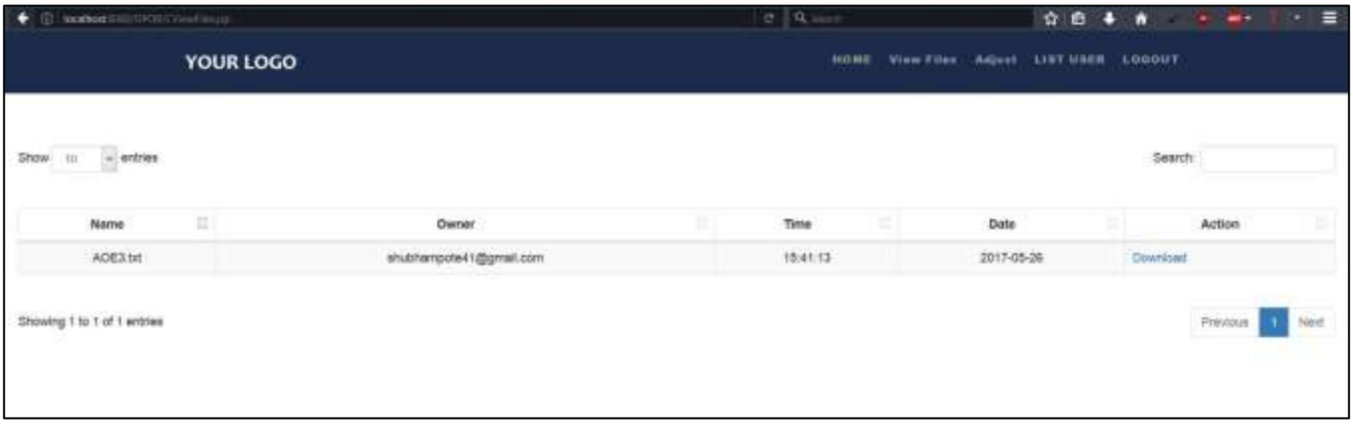

Figure 4.List of uploaded files at Cloud/TPA side

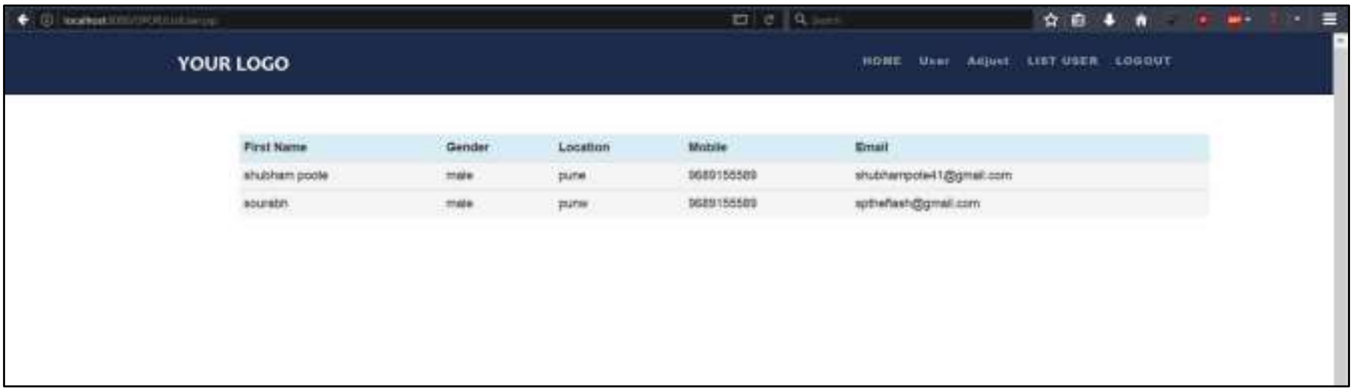

Figure 4.List of Users at cloud side

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

CONCLUSION

An integrity verification scheme is proposed here which gives an idea of asking proof of retrievability for cloud storage. Also feature for recovery of corrupted data is introduced. Here a third party auditor is presented for the purpose of preprocessing, uploading the data on cloud storage server and recover the corrupted data behalf of client. The third party auditor also performs the data integrity verification or updating the outsourced data upon the clients' request. Use of TPA scales down the computational burden for tag generation on client.

ACKNOWLEDGEMENT

We would like to take this opportunity to express our hearty gratitude and sincere thanks towards our guide **Prof. Deepali Ahir** for her invaluable assistance that we have received, throughout the development of our project. We express our sincere thanks to the H.O.D. of Computer **Dept., Prof. Dr. (Mrs.) N. F. Shaikh** and our respected Principal **Dr. A. A. Keste** for permitting us to present this paper and also to all staff members for their encouragement and suggestions during the completion of the project. We also acknowledge the literature which helped us directly or indirectly in the preparation of this project .

REFERENCES

1. J.Li,X.Tan.XChen and D.S.Wong,OPoR : Enabling proof of retrievability in cloud computing with Resource constrained Devices, IEEE transactions on cloud computing on volume:XX No:20
2. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, Provable data possession at untrusted stores, in CCS 07: Proceedings of the 14th ACM conference on Computer and communications security. New York, NY, USA: ACM, 2007, pp. 598609.
3. A. Juels and B. S. K. Jr., Pors: proofs of retrievability for large files, in CCS 07: Proceedings of the 14th ACM conference on Computer and communications security. New York, NY, USA:ACM, 2007, pp. 584597.
4. H. Shacham and B. Waters, Compact proofs of retrievability,in ASIACRYPT 08: Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 90107.
5. C.Erway,A.Kupcu,C.Papamanthou, and R.Tamassia,Dynamic provable data possession,cryptography e print archive,Report 2008,432,2008/432,2008, http: // eprint. iacr.org/. SYNOPSIS
6. J.Li,X.Tan.XChen and D.S.Wong,An efficient proof of retrievability with public auditing in cloud computing ,in / NCoS ,2013, pp, 93-98
7. C.Wang,Q.Wang,K .Ren, and W.Lou, Privacy preserving public auditing for data storage security in cloud computing,in INFOCOM, 2010 Proceedings IEEE .I EEE ,2010 pp.1-9
8. E.-C.Chang and J.Xu,Remote integrity check with dishonest servers,in preceedings of ESORICS 2008 ,volume 5283 of LNCS.springer-verlag ,2008,,pp.223-237
9. Q.Zheng and S.Xu,Fair and dynamic proofs of retrievability, in CODASPY,2011, pp.237-248
10. C.Wang,Q.wang and K.Ren,Ensuring data storage security in cloud computing,in preceedings of IWQoS 2009, Charleston, USA 2009

CITE AN ARTICLE

Pote, S., Jain, V., Shinde, R., Paigude, P., & Ahir, D., Dr. (2017). OPOR: PROOF OF RETRIEVABILITY IN CLOUD COMPUTING WITH RECOVERY AND REMOTE INTEGRITY CHECK. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY, 6*(6), 58-65. doi:10.5281/zenodo.802818