Secured Storage through Remote Data Auditing

Keerthana S[1] , Balasubramanian V[2]
[1]PG Student, Department of Computer Science and Engineering, SSN College of Engineering, Chennai, India
Email: keerthana.sekar1@gmail.com
[2]Assistant Professor, Department of Computer Science and Engineering, SSN College of Engineering, Chennai, India
Email: balasubramanianv@ssn.edu.in.

Abstract—Cloud computing is visualized as the next generation architecture of IT Enterprise. Cloud Storage, a service provided by Cloud Service Provider (CSP) for the data owners to remotely store the data is advantageous with respect to low cost, ease of access and dynamic retrieval. However, the integrity of the outsourced data cannot be certified and is a constant threat to the data owners. Merkle Hash Tree is a deterministic approach to authenticate the data file blocks in cloud storage. A highly efficient method that helps in the reduction of server time as there is no overhead of generation the metadata for every block. The MHT that is generated by the client is verified by comparing the root hash in order to check the data integrity. It provides an improvement over the existing Remote Data Auditing (RDA) techniques like Provable Data Possession (PDP).

Keywords*-Cloudcomputing, Cloudstorage, Data Integrity, Merkle Hash Tree(MHT).*

*****

I. INTRODUCTION

Cloud Computing provides computing, softwareand storage as services to users on-demand over the internet. The other characteristics of cloud computing include device and location independence which means that the users can access from any location and from any device (e.g.,laptop, PC or mobile). With the advent of network and computing technology, the cloud computing service has attracted noticeable attention. RDA mechanism is used to check the integrity of the data stored in the cloud. It refers to sampling of the outsourced data in the cloud and evaluating the data against various criteria such as validity,accuracy and integrity. Remote Data Auditing consists of two components such as Data owner(DO), Cloud Service Provider(CSP). Sometime the CSP may delete the rarely accessed data files and pretends that the outsourced data ofthe data owners is still correctly stored in the Cloud Storage Server. Also theCSP cover the data loss caused by the hardware or software failures in the cloud for their reputation. RDA scheme allows the data owner to audit the integrity of the outsourced data on the storage server.

The following are the features that need to be minimized in designing data integrity techniques for outsourced data

- Computation overhead for the Data Owner
- Computation overhead of the Cloud Service Provider
- Storage overhead of metadata

An authenticated data structure (ADS) is a data structure in which the operations will bedone by an untrusted party and the results of those operations will be audited by the data owner. It support the maintenance and processing of outsourced data to an untrusted party without loss of integrity.

II. RELATED WORK

Provable Data Possession [1],allows a client to verify the outsourced data at the server without retrieving it. PDP scheme generates probabilistic proofs of possession by sampling random sets of blocks instead of checking the whole data using spot checking technique. In PDP scheme, the client maintains the metadata of each block to verify the integrity. PDP supports large data sets in widely distributed storage. PDP model uses Homomorphic Verifiable Tags to verify the probabilistic proof. The goal of the PDP scheme that achieves the probabilistic proof of data possession is to detect server misbehaviour when the server has deleted the fraction of a file. PDP scheme gives a probabilistic proof of data possession. It does not give deterministic proof of data possession. In this PDP scheme, the file will be divided in to separate blocks and the tag is generated for each block. After that the file with corresponding tag will be stored on the server. PDP scheme is efficiently adopted to the spot checking technique. The spot checking technique allows the client to randomly sample the number ofblocks for checking rather than checking the whole data. But it is vulnerable to small corruption attack.

In paper [2], there are three entities namely cloud, Data owners and auditor. The cloud is managed by the cloud service provider and provides data storage service to the authorized users. The Data owners create and store their data to the cloud. The auditor is capable of checking the integrity of the outsourced data. The data which is outsourced will be further divided into number of blocks. There are two kinds of threats namely Internal threats and External threats. To check the integrity of the outsourced data the Data owner compute a signature foreach data block. Then the auditor uses the signature to check the integrity of the data block. There are five stages in the construction of the proposed scheme such as Setup, Signblock, Challenge, ProofGen and ProofVerify. In the Setup stage, the Data owner generates master key and some global random numbers to initialize the system. In the Signblock, the Data owner computes the signature for each data file block and upload the data blocks to the cloud with the corresponding signatures. In the challenge stage, the Data owner sends the challenge message to the cloud and the auditor to check the integrity of the outsourced data. Inthe ProofGen stage, the cloud compute the proof by doing the sum of the selected blocks and send it to the auditor. In the ProofVerify stage, the auditorchecks the proof sent by the cloud and return the result to the user. If the proofis correct, then it sends truemessage to the Dataowner otherwise it sends falsemessage to the Dataowner.This scheme has been expanded to support data dynamics which means that the user can perform dataupdate, data insertion .

In the public auditing scheme [3], everyone who has the verification parameters can be able to check the integrity of the outsourced data. Privacy associated with public auditing scheme motivates the Designated verifier Auditing (DVA) scheme. In this scheme the cloud service provider specifies the particular verifier who can do the integrity check of the outsourced data by providing the private key to protect the privacy. Data owner is the signer, cloud server is the depositor and third party is the verifier. Both signer and depositor have access to the data but the verifier doesn't have the access to data. Only cloud server is allowed to generate designated signature in this scheme to avoid the problem in which the verifier pretend to act as a data owner and generate the signature.Only the authorized Designated Verifier can verify the integrity of the data. This scheme also provides batch verification, which helps verifier to do the auditing task simultaneously for multiple data owners. Though DVA scheme is fully secured it doesn't support dynamic operation which is the future work.

In paper [4], Algebraic Signature based Remote Data Possession Checking protocol(RDPC) was proposed. The goal of RDPC is to check whether the data is correct. In this scheme the user needs to store only two secret keys and random numbers. The scheme focuses on how to verify that the cloud server store the client's data without retrieving the original data. Algebraic Signature is a kind of hash function in which the signature of the original file will be as same as the sum of the selected blocks. The algebraic signature is similar to MD5 and SHA-1. The algebraic signature is not secure because it can generate same signature for two blocks but it is ideally suited for use in cloud storage. The advantage of using Algebraic Signature is that there is no use of public key, no retrieval of original data from cloud to check the integrity and the use of Signature. This scheme does not support dynamic operations which is the future work.

In paper [5], the index table management method was proposed for secure and efficient mechanism to audit the dynamic shared data. The index table contains the sequence of data blocks and identifiers. Since, the index table is managed by the Cloud Service Provider(CSP)there is a chance of two attacks such as forge at- tack and replace attack. In replace attack the duplicate data will be verified instead of the damaged data. In forge attack the duplicate verifying term will be used instead of original verifying term. In order to prevent these attacks the index table should be managed by the Trusted Third Party Auditor (TPA) and the user. In privacy preservation, the random masking and bilinear map techniques are used where the TPA cannot be able to learn the information about the data. Soin the proposed scheme, the index table is managed by the TPA and the identifier is renewed by the CSP. Therefore, by using this scheme any user in the group can update the shared block efficiently.

III. MERKLE HASH TREE

A Merkle Hash Tree (MHT) is a good authenticated data structure which is used to prove that the data are unaltered and undamaged. In MHT, each leaf nodes contains the hash of the data block such H(1) holds the hash of the first block. Internal nodes contain the hash of the concatenated hashes of their children nodes such as $H(1, 2) = H(H(1)|H(2))$ where '|' is concetanation. To audit the integrity of any block, the whole tree of hashes does not be transmitted to the CSP. The dataowner transmits only the hashes of nodes which are in the path of the challenged block. Figure 1 shows the example of Merkle Hash Tree.

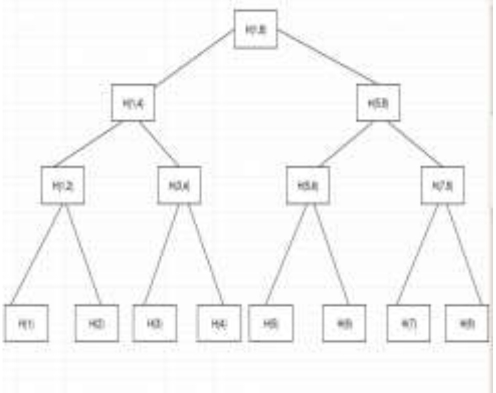

Fig.1:OverviewofMerkle Hash Tree

For example, if the Data Owner wants to check the integrity of the data block 2 then only H(1), H(3,4) and H(5,8) need to be transferred to the CSP. Then the CSP can calculate the H(2) from the data block 2. H(1,2) will be calculated from the received H(1) and calculated H(2). In the same way, H(1,4) can be calculated and then H(1,8). The CSP sends the calculated root hash to the Data owner. The data owner then can compare the received H(1,8) with the already stored H(1,8). When both the hashes match then the integrity of data block 2 is confirmed.Some important facts of Merkle's Signature Scheme are as follows:

- Security of this signature scheme depends on the security of the hash function.
- Only one hash needs to be maintained/shared securely.

IV. OVERVIEW OF PROPOSED SYSTEM

In this proposed system, the data owner split the file into several blocks. The blocks are encrypted . For each block, the tag is created and Merkle Hash Tree is constructed. The data owner keeps the secret key, public key and sends the public key and data blocks to the CSP. Then the data owner creates a challenge message and send it to the CSP. In CSP,the root hash will be generated and send to the data owner. The data owner compares the already stored hash and received hash. The system model is shown in figure 2.

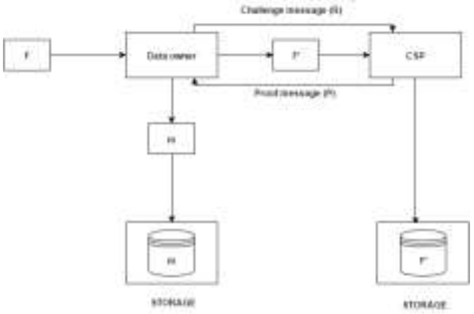

Fig.2:OverviewofProposed System

A. Key Generation

The data owner generates the public key and secret key using the RSA cryptographic algorithm. After generating the keys, the file F is splitted into manageable blocks. Then the public key is used to encrypt the splitted blocks using RSA algorithm. After encrypting, the data owner stores the data blocks in cloud.

B. Hash Generation

The data owner creates the hash foreach encrypted blocks. For generating hash for each encrypted block, SHA-256 algorithm is used. The SHA-256 algorithm takes the encrypted blocks and public key as inputs and returns the hash as output for each encrypted block. The hashes are concatenated until the root hash is generated.

C. Proof Generation

The data owner generates a challenge message by randomly selecting the data block. Then sends the challenge message and the hashes of the data blocks in the path of the challenged block to the Cloud Service Provider. The CSP takes the public key,challenged data block and generates the hash of the challenged block and then generates the root hash using the generated and received hashes.

D. Proof verification

The CSP sends the generated root hash to the data owner. The data owner compares the received hash and already stored hash to check the integrity of the challenged data block. If both the root hashes holds then there is no modification in the challenged data block.

V. IMPLEMENTAION AND RESULTS

Probabilistic Checking of CSP misbehaviour

Data owner chooses some random blocks from total number of blocks to check the integrity of data stored in cloud. By using Spot checking technique only the probabilistic proof can be achieved. In spot checking technique the number of challenged blocks increases when there is a small number of corrupted blocks to detect the CSP misbehaviour with high probability. If the CSP deletes 1% of 10,000 blocks then the data owner can challenge 460 and 300 blocks to achieve probability of 99% and 95%. Figure 3 shows the example of CSP misbehaviour.The following is the formula to calculate the probability of CSP misbehaviour.

$$P_X = P\{X \geq 1\} = 1 - P\{X = 0\} = 1 - \{\frac{n-r}{n} \cdot \frac{n-1-r}{n-1} \cdot \frac{n-2-r}{n-2} \cdots \frac{n-c+1-r}{n-c+1}\}$$

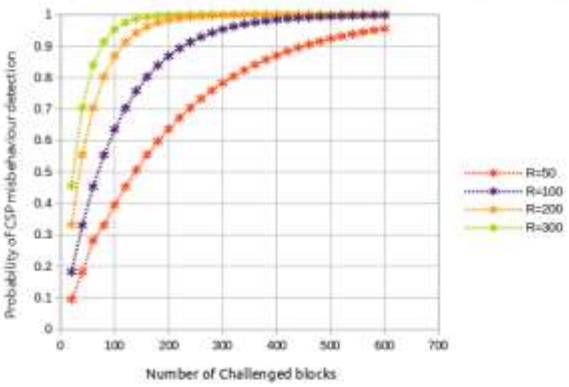

Fig 3: Probability of CSP misbehaviour

Key Generation and Splitting of File

The algorithm outputs the public key and secret key. The file is split into several data blocks.The public key is used to encrypt the data blocks which are then stored in the cloud by the data owner. Figure 4 shows the example of key generation*and splitting of file*

```
-----BEGIN RSA PRIVATE KEY-----
MIISKwIBAAKCBAEKhAtWGRyLmEH2p0+1gucv1uObs4aQnJ+q40TWQq4dJwl42Z2l
7OFMFosft95Xf6+PQMRfAm3BvYNAQSOQvfLQ/uKnu3ydqZzdf3lc0Ki3pafRoeTU
KCELK/5iKWjdAwYThpST0VVWarfDVYfruhob0ydj47/y6zMZOdHAvjmvd8IbZjAz
2GgSBMBSCwy42Rg8PjAg/iGUV6RDH3G9vPZEC+GKhfh5JvbXZ3VosI2si8b1XgSL
Hxhv+4pYSMpO+Palgbit7hEUmBAAuhY2zlCybzgjkHB8k91Cv8BfpsczkBqIA0d+
BhwAa6uSUaRDyVyluJAdcKVOaQTW80Ejh//0dS9yBxRXvWONFJpoy9W2Fx82r/AN
BL6KiS01XBomBTUGNXCL6ZMyu6xTehAyzHJAIcIC80RAkzyh+05qw06y001qAKu1
oFA7y7rvpVGx034pNOxS/pZKv/3vBKhkkmtKIPSeDbtFqglHz6gD7SS6mBDXhEpe
WJQLyPnFrwqshzC1bfVE29gn506hSlg3IDYCUsAthbD6sydZ0NMIgvBZRoXgNsIM
Dflc35dCEgWS0ecQGoJrDTaAy3QYPEy3ivVVn5e9Vl68bSqa12Pk07uRyOebEvctW
P70ZcUbDa2PQkiB3iV5yKUKRsyh1O2zpPk+JW8vv+t/RycTBmtOLtTO5KwrHY6w3
wjRI6oLZe0t609gnbP8GUg4lj9goTplkoY6VKwt/yJdUD5X/XqTYNQGtTlBATUoL
bNcAlT1cg1Mfeg9GeRBakMtOGMz5QpJlFHRapWobVQXS5HZqUBafWjtE5CXDXzAT
I3ujh1izQeScMGRyfKMyyeSBsh32uTaH6PuqR9ow29HkpldXQwbuDuv4SF4l4Zjr
KqC5tQi32lHRbkoSQTwrRrHn1kxjiG0QI7eWmVB5/c9YEHUE7GDDsMNd4KvUPFsP
djZAtbPdlDg0iwpTTCySwd/NbWgvVHUA4sGNfoygz1egBtRYUSnAASze0fgTVlYB
W0owmk29nFBMZA3LJSIbO1naHGXTdX0flTCqHj5iRDFejmeRst/E86EMCx4nndmB
43qP2sPHaq5N9fUSh4B7ShSXASg1PoTgknyVxXDZ8/Tfx4YO/yY/zk8nVRo4adVV
IZvA2txq46vVlAc1G/d6vD9Kw7Ma8JXoxQlJVGHRLPsoM7MGw6YpUz9bKOU37b14
Gpn1fSqY1ABM/YKYgaUckaQ/jS4rZtREZqFzV/MvaK/AiqgJGFOg46FTd1buJaer
N3nUFRLpFQGqIMeJ5EXWEmL3GFA++u5uBt0gtdoCIG9oSMh/UNr3dvzAxYLefbAp
VQssII+Xgx+rFtiwkmJOHCZJwTye7rlQMObwfQIDAQABAoIEAQTgxeV1lqFGyugh
KD+Z7qZpd4HLoHl6vjJmZKtDmUMatVgSUqgCn2xNVN1s1+4c5BJ+x9kO5l6mDTmE
PfaJsNaBNe1e8cetPIAfPuwIOwudwe8iCiVETDfDeeltbwJ/AZWTK176jyycKoap
I7jExllhCTO0FXmZE0YT2TKUy2uTES7mk1upenVJUPF+ALPKuPDyqR7USbzM/vzI
HPoYyuA7Sq70gIupwHyhmWws0juF85OQw5jMDuJ7dWzRDOvfX4LdDw9XzQKBnm3LK
s7BlrZpoL1GIEhrJfaBW0v2uvtN1Ipt/fCzApgKn76vyrFFpmhg8p1SQyY369zXf
zKN/1gLPTWgG0q/0wbZTJWGRlJx4U1lp+eAAIGdhROBozhTcdW1g4sl0W36AQ+sh
R1xZLHKFdlSp/7I42rqFOhYabWpuzKdsKjqZMD5BvkzHkaPR6JwyZaG5SaNSNlFn
0rpIlYS4hSDAW+vv+YgfUuAnZI90NRnUWy8yxdJFs1+SWCk6OSUgUdNUbAvup7H4
K72+gm8fd8hoGPX1n1FapkNfglOXrwlYPiGbZdSoNeFHPOCjkB8PPwprwanzNzJ6
3leSFu6JpIU+7NKrKX8XpxqfGsb28w+lvy1OOekQe/Pw4AeDkZ4B0hcA/zlZo13m
eBo2dmzQcWTsKmFFyIjKynEXaFvwEXh9aw7bXLlrQ6cKfsy8r594cGuPV2xmZTat
SHGnum+eBRav017gkSDgiJm1yBZ8dscgLAjjtrpv2tWkfqKnyvKCRXNN6MOWAMO7
Jnnc+iPRWTLeXM2euUk9OqZhYC2rtt2pkq0pMw+kS8PS1hYKEUYsvph22O5VoJuk
```

Fig 4a): Generation of private key

```
-----BEGIN PUBLIC KEY-----
MIIEIjANBgkqhkiG9w0BAQEFAAOCBA8AMIIECgKCBAEKhAtWGRyLmEH2p0+1gucv
1uObs4aQnJ+q40TWQq4dJwl42Z2l7OFMFosft95Xf6+PQMRfAm3BvYNAQSOQvfLQ
/uKnu3ydqZzdf3lc0Ki3pafRoeTUKCELK/5iKWjdAwYThpST0VVWarfDVYfruhob
0ydj47/y6zMZOdHAvjmvd8IbZjAz2GgSBMBSCwy42Rg8PjAg/iGUV6RDH3G9vPZE
C+GKhfh5JvbXZ3VosI2si8b1XgSLHxhv+4pYSMpO+Palgbit7hEUmBAAuhY2zlCy
bzgjkHB8k91Cv8BfpsczkBqIA0d+BhwAa6uSUaRDyVyluJAdcKVOaQTW80Ejh//0
dS9yBxRXvWONFJpoy9W2Fx82r/ANBL6KiS01XBomBTUGNXCL6ZMyu6xTehAyzHJA
IcIC80RAkzyh+05qw06y001qAKu1oFA7y7rvpVGx034pNOxS/pZKv/3vBKhkkmtK
IPSeDbtFqglHz6gD7SS6mBDXhEpeWJQLyPnFrwqshzC1bfVE29gn506hSlg3IDYC
UsAthbD6sydZ0NMIgvBZRoXgNsIMDflc35dCEgWS0ecQGoJrDTaAy3QYPEy3ivVn
5e9Vl68bSqa12Pk07uRyOebEvctWP70ZcUbDa2PQkiB3iV5yKUKRsyh1O2zpPk+J
W8vv+t/RycTBmtOLtTO5KwrHY6w3wjRI6oLZe0t609gnbP8GUg4lj9goTplkoY6V
Kwt/yJdUD5X/XqTYNQGtTlBATUoLbNcAlT1cg1Mfeg9GeRBakMtOGMz5QpJlFHRa
pWobVQXS5HZqUBafWjtE5CXDXzATI3ujh1izQeScMGRyfKMyyeSBsh32uTaH6Puq
R9ow29HkpldXQwbuDuv4SF4l4ZjrKqC5tQi32lHRbkoSQTwrRrHn1kxjiG0QI7eW
mVB5/c9YEHUE7GDDsMNd4KvUPFsPdjZAtbPdlDg0iwpTTCySwd/NbWgvVHUA4sGN
foygz1egBtRYUSnAASze0fgTVlYBW0owmk29nFBMZA3LJSIbO1naHGXTdX0flTCq
Hj5iRDFejmeRst/E86EMCx4nndmB43qP2sPHaq5N9fUSh4B7ShSXASg1PoTgknyV
xXDZ8/Tfx4YO/yY/zk8nVRo4adVVIZvA2txq46vViAc1G/d6vD9Kw7Ma8JXoxQlJ
VGHRLPsoM7MGw6YpUz9bKOU37b14Gpn1fSqY1ABM/YKYgaUckaQ/jS4rZtREZqFz
V/MvaK/AiqgJGFOg46FTd1buJaerN3nUFRLpFQGqIMeJ5EXWEmL3GFA++u5uBt0g
tdoCIG9oSMh/UNr3dvzAxYLefbApVQssII+Xgx+rFtiwkmJOHCZJwTye7rlQMObw
fQIDAQAB
-----END PUBLIC KEY-----
```

Fig 4b): Generation of public key

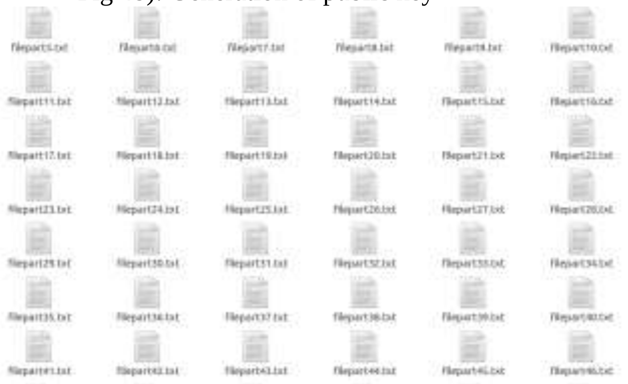

Fig 4c): Splitting of file

Encryption of data blocks

The Splitted data blocks are encrypted using the public key and these encrypted data blocks are stored in cloud.Figure 5 shows the example of encryption of data block.

Fig 5:Encryption of data block

Data Storage in Cloud

Cloud storage has become one of the most common and easiest ways to store, share or back up data to a secure location. The data in these remote cloud can be easily accessed at any time, from anywhere by the users and individuals around the world.Amazon S3 bucket names are globally unique, regardless of the AWS region in which the bucket is created. The name will be specified at the time the bucket is created. Each encrypted blocks are stored. Figure 6 shows the example of data storage in Amazon Web Services(AWS).

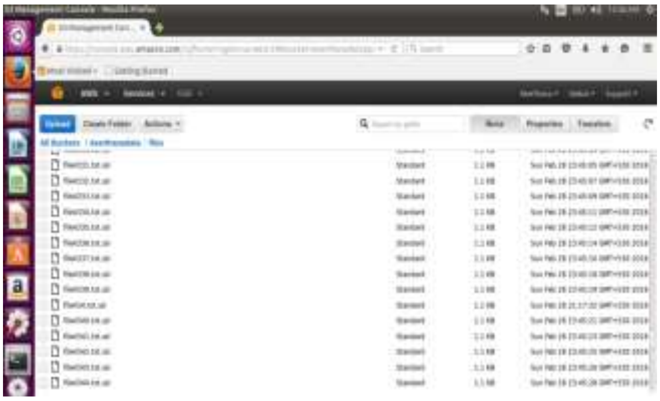

Fig 6: Data storage in AWS

V. CONCLUSION

Several works have been done in the literature to check the integrity of the outsourced data. In this paper we propose a Merkle Hash Tree based Technique to check the integrity of data stored in cloud. Merkle Hash Tree technique is used to check the integrity of data stored in cloud without downloading the data from the cloud. Further work has to be done on generating challenge and checking integrity of data stored in cloud.

REFERENCES

[1] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring *Provable data posession at untrusted stores*,International Journal of Computer Science and Information Technologies,2012.

[2] LUO Yuchuan, FU Shaojing, XU Ming, and WANG Dongsheng *Enable Data Dynamics for algebraic signatures based remote data pos- session checking in the cloud storage*, IEEE Transactions on Informa- tion Forensics and Security,2014.

[3] Solomon Guadie WORKU, Chunxiang XU, and Jining ZHAO ,*Cloud data auditing with designated verifier*, Higher Education Press and Springer-Verlag Berlin Heidelberg, 2013.

[4] Lanxiang Chen *Using algebraic signatures to check data possession in cloud storage*, Jan, 2012.

[5] Ohmin kwon, Dongyoung koo, Yongjoo Shin, Hyunsoo Yoon *A secure and efficient audit mechanism for dynamic shared data in cloud storage*, May, 2014.

[6] Mehdi Sookhak, Abdullah Gani, Adman Akhunzada, Muhammed Khurram Khan and Nor Madrul Anuar *Towards dyanmic remote data auditing in computational clouds*, July 2014.

[7] Hemalata A.Gosavi, Prof.Manish R.Umale *Auditing and data dynamics 27 28 for storage security in cloud computing*, International Journal of engi- neering trends and technology , May, 2014.

[8] Muhammad Saqib Niaz, Gunter Saake *Merkle Has Tree based Techniques for Data Integrity of Outsourced data*, International Journal of engineering trends and technology , May, 2015

[9] Pro. Bharat Tidke, Poonam M. Paradeshi *Improving Data Integrity for Data Storage Security in Cloud Computing* , International Journal of Computer Science and Information Tech- nologies , May, 2014