Secure Auditing and Maintaining Block Level Integrity with Reliability of Data in Cloud

Ms. Aishwarya R. Kumthekar,
Dept. of computer,
RMDSSOE , Pune
Aish275@gmail.com

Prof. Jyoti Raghatwan
Dept. of computer,
RMDSSOE, Pune
jyotiraghatwan2@gmail.com

Abstract—Cloud storage systems are becoming increasingly popular and popular and the cloud computing is getting enhance day by day it needs to provide more security with secure auditing. For storing large and large amount of data in cloud, requires more space and data can be replicated which will increase the space and cost too unnecessarily. To avoid this deduplication needs to be done. So , in this paper, pondering the main issue of honesty and secure deduplication on cloud information. Specifically, going for achieving both information uprightness as well as deduplication in cloud. And in this paper, proposing the algorithm which will audit securely and provide block level deduplication as well as it will maintain reliability of data in cloud

Index Terms—*Secure auditing, Deduplication, Reliability, Cloud computing, Third Party Auditor .*

*****

I. Introduction

Despite the fact that cloud stockpiling framework has been generally embraced, it neglects to oblige some critical emerging needs, for example, the capacities of auditing the integrity of cloud by cloud customers and then detecting copied by cloud servers. Shows the issues underneath. The more issue is known as integrity auditing. Cloud server has the capacity alleviate customers from the substantial weight of capacity administration and maintenance. The distinction of the cloud stockpiling from the customers in-house stockpiling is that the data is exchanged by means of the Internet and put away into some uncertain domain, and not under the control of the customers by any kind of stretch of the imagination, which inevitably raises the customers worries on the integrity of their data. These worries originate from the way that cloud stockpiling is defenseless to the security dangers from both the sides i.e. from outside and from inside of the cloud, and also the uncontrolled cloud servers might inactively conceal the some amount of data misfortune the incidents from the customers to maintain their notoriety. In addition to this is that for saving money and the space, the cloud servers may effectively and purposely dispose of once in a while got to data less belonging to an ordinary customer.

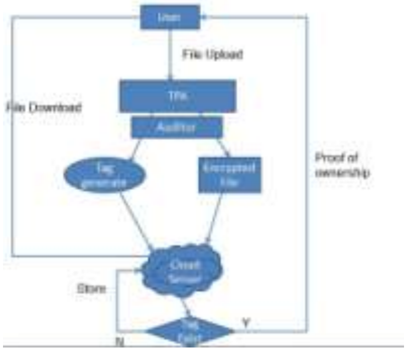

Fig.1.Flow model.

In the figure, user will upload file in the cloud, then third party auditor(TPA) will generate tag and encrypt the file. Blocks will be verified , deduplication of block will be taken place. If the tag exists then it wont generate the tag again. For each file it will generate secret key while when user will register into system, that time also secret key will be generated with some random number. Clients of the cloud have vast information files to put away and it is depends on the cloud for information support and calculation. They can be the singular buyers or business associations.

The rest of the paper is organized as follows:
- Section 2 discusses Literature Survey. Section 3 provides an overview of Proposed Work and implementation details Section 4 is consists of proposed algorithm section 5 consists of results and discussion and section 6 discusses about conclusion.

II. Literature Survey

1) Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing:
Cloud Computing is considered as the next generation architecture of IT Enterprise. It moves the database and the application software to the centralized large data centers, where the management of all the data and the services may not be fully trustworthy[5]. This factor is unique and which brings about many security challenges, which have not been well understood yet. In this paper, it studies the problem of ensuring the data integrity of all the storage in the Cloud Computing. In particular, it can be consider the task of allowing a TPA, on the behalf of the cloud client and which verifies the integrity of all the data stored in the total cloud storage but only the dynamic data. Here introduction of TPA eliminates the total involvement of client through auditing, whether its data is stored in the cloud is indeed intact[5]. The support for data dynamics of data operation, such as block modification,

updation,deletion , insertion is also a significant step towards practicality.[5].

2) Proofs of Ownership in Remote Storage Systems:
Cloud storage systems are becoming very popular and popular. It is the promising technology which keeps their cost down and is removing duplication of file , which stores only single copy of the data repeating[4]. In the Client side deduplication , it tries to attempts to identify the deduplication opportunities which are already present at the client and it saves the bandwidth of uploading the copies of existing files to the server which is harmful. In this paper trying to identify the attacks that exploit client-side deduplication, which is allowing an attacker to gain the access to arbitrary size files of other users based on a very small hash signature of these files[4]. An attacker , if he knows the hash signature of a file then he can convince the storage service that it owns that file, hence that server lets the attacker download the entire file.

3) DupLESS: Server-Aided Encryption for Deduplicated Storage :
The cloud storage service providers such as Dropbox and others performs the deduplication to save the space by only storing one copy of each file uploaded on it. Should clients conventionally encrypt their files, however their savings are lost. Message locked encryption (the prominent manifestation of which is convergent encryption) resolves this all the tension[2]. However it is inherently subject to the brute force attack that can recover files falling into some known set. It enables the client for storing encrypted data with an existing service that have the service perform deduplication on their behalf, and yet achieves strong confidentiality guarantees[2]. Showing that encryption for the de-duplicated storage can achieve the performance and the space savings close to that of using all the storage service with plaintext data.

4) Provable Data Possession at Untrusted Stores :
It Introduce a model for provable data possession that is PDP , which allows a client that has stored data at an untrusted server for the verification of the server possesses the original data without retrieving[3]. This model generates the probabilistic proofs of the possession by sampling some random sets of blocks from the server, which drastically reduces the I/O costs[7].

III. Implementation Details

3.1 SOFTWARE SPECIFICATIONS

For implementation we have used :
1. Coding Platform: Java
2. IDE : Eclipse
3. Database : MySQL

3.2 MATHEMATICAL MODEL

Fig. 2. Mathematical Model

Set
Theory:
Let S be the system object It consist of following
Where S={U,F,TPA,CSP}
S denoted the System which consists of the following ,
That is U denotes Users , where F for Files , TPA denotes Third Party Auditor and CSP denotes Cloud Service Provider. Where input

1. I={U,F}
U={u1,u2,u3,..un} that is users can be infinite
F={f1,f2,f3,..fn} and files can be infinite. P that is process consists of
2. P={TG,C,PF,V,POW, DD,BD,PF,F}
CSP ={DD,BD,PF,F}
DD= Deduplication
BD=Block level Deduplication
PF=proof if duplicate tag exist.
F= store files if tag not exist
TPA={TG,C,PF,V,POW}
TG= tag Generation

C=challeng
e
PF =Proof by CSP
V= Verification by TPA
POW= Proof of ownership
O is for the output
3. O={Result}
Save Tag and encrypted file if not exist.

3.3 PROPOSED SYSTEM ARCHITECTURE

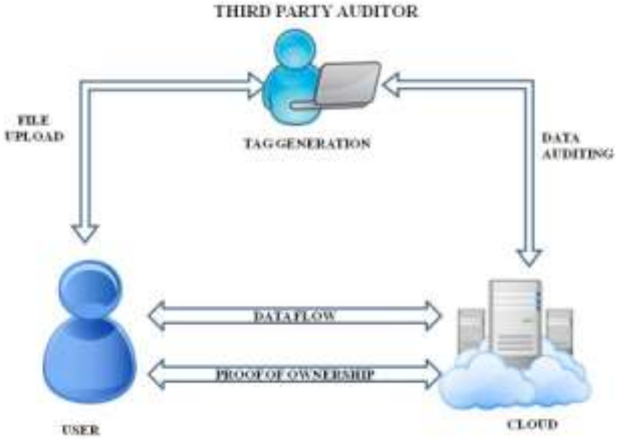

Fig. 3 Architecture

Cloud Clients:
Cloud Clients have large data to be stored and it depends on the cloud for the maintenance and computation of the data. They can be either the

individual consumers or may be the commercial organizations.

Cloud Servers:
Cloud Servers are virtualize the resources ac- cording to the requirements of the clients and then it expose them as the storage pools. Cloud clients may buy or lease storage capacity from cloud servers, and then it stores their individual data in these bought or the rented spaces for future utilization.
Auditor : Auditor helps the clients to upload and au- dit their outsourced data which maintains a MapReduce cloud and acts like a certificate authority. This assumption presumes that the auditor is associated with a pair of public and private keys. Public key is made available to other entities in the system.

IV. Proposed Algorithm

4.1 PHASES :

1. Setup Phase :
The challenger which first runs the setup algorithm with all the required security parameter and the other public parameter as input. Then, it generates the public and secret key pair as (pk; sk). Then the public key pk is forwarded to the adversary A.

2. Query Phase :
The adversary A , is allowed to query the file upload as file F. Then, the file with the correct tags are generated and uploaded to the cloud storage server. These all the tags can be publicly verified with respect to the public key pk[1].

3. Challenge Phase.:
Here A can adaptively send file F to the file tag then tag comes, C runs the integrity verification protocol known as IntegrityVerifyA C(pk; tag) with A.

4.Forgery:
Here A outputs a file tag and then the description of a prover Pt.And can be said that a prover Pt on tag is -admissible, if the following two conditions are hold:
(1) tag is a file tag output by a previous upload query.
(2) Pr[IntegrityVerifyPt C(pk; tag) = 1] .

4.2 Protocols Using

1.File Uploading Protocol: This protocol used to allow the clients to upload files via the auditor.

Algorithm 1 Convergent encryption

```
1: KeyGen: Input = file content F Output = the
   convergent key.
2: Encrypt: Input = convergent key, and file content
   Output = Ciphertext.
3: Decrypt: Input = convergent key, and ci- phertext
   Output = Plain text
4: TagGen: Input= Takes a file content F Out- put =
   Generate the tag. i=1
```

Algorithm 2 AES: AES use for encryption and decryption (for block level it uses 16 byte block)

```
1: At the start derive the set of round keys from the cipher key.
  2: Initialize the state array with the block data
     (plaintext data).
  3: Add the initial rounds key to the starting state of
     the array.
  4: Perform nine rounds of state manipulation.
  5: Perform the tenth and final round of state
     manipulation.
  6: Copy the final state array out as the en-
crypted data (ciphertext).
```

plays the role of prover, while the auditor or client works as the verifier of the data.

3.Proof of Ownership Protocol: This protocol is typically comes with the file uploading protocol to prevent the leakage of side channel information. On the contrast to integrity auditing protocol, in PoW the cloud server works as verifier, while the client plays the role of the prover.

Algorithm 3 SHA1- Block level deduplication (it is not encryption it is used for hash compu- tation)

```
1: Append Padding Bits
2: Append the Length to it
3: Prepare the processing functions
4: Prepare processing constants
5: Initialize the buffers
6: Processing message in 512-bit blocks
```

Parameter	AES	DES
Key Size	128, 192 and 256 bits	64 (8:parity, effective key length:56 bits)
Block Size	128 bits	64 bits
Rounds	10,12,14	16
Flexible	Flexible	Not flexible
Features	Replacement for DES.	Not enough structure.

Table : Difference between AES and DES.

2.Integrity Auditing Protocol: The cloud server

Algorithm 4 Block level deduplication

1. User will register to the system with his information
2. Random secret key generation using Random( ) function.
3. When user will login to the system, secret key must be verified.
4. User will successfully login to the system , user will upload any file to cloud .
5. File will have unique secret key.
6. Setup Phase : The auditor initializes the public key and private key

$$P_k = (g, \{u_i\}^t)\ S_k \leftarrow \alpha$$

7. KeyGen(F) : This key generation algorithm takes a file content F as input and outputs the convergent key ckF of F; K eyGen(F) → ckF.

8. Encrypt(ckF; F) : Here the encryption algorithm takes the convergent key ckF and file content F as input and outputs the cipher- text ctF;
9. Decrypt(ckF; ctF) : Here the decryption algorithm takes the convergent key ckF and ciphertext ctF as input and outputs the plain file F;
10. TagGen(F) : Then the tag generation algorithm takes a file content F as input and outputs the tag tagF of F.

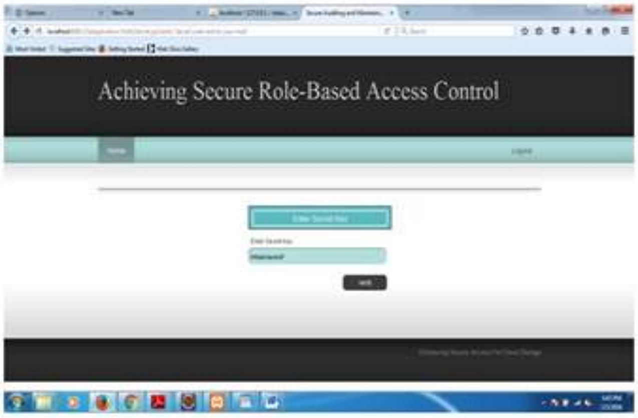

Fig. 4. This screen is used for checking the role of user whether he is admin or user or TPA

$\alpha_{ij} = [Hash(ID_F ||B_i)\Pi^{t}_{u}{}^{B_{ik}}]^{j}$ k=1 k

11: Dividing the file into no of blocks First divide
F ile into no of blocks and generate separate block
id for each block by using

12: File store on server in the form of (ID, F, $\alpha^{S}_{i=1}$)

4.3 Why to use these algorithms?

1. Why **Convergent** encryption ?

- If using file and users key then it will not be able to detect the duplicate files since cipher texts of two files will be different.

2. Why AES

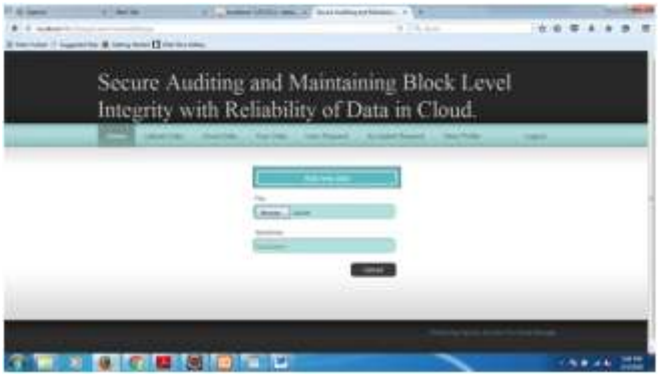
Secure Auditing and Maintaining Block Level
Integrity with Reliability of Data in Cloud.

Fig. 5. Here secret key needs to enter using this screen.

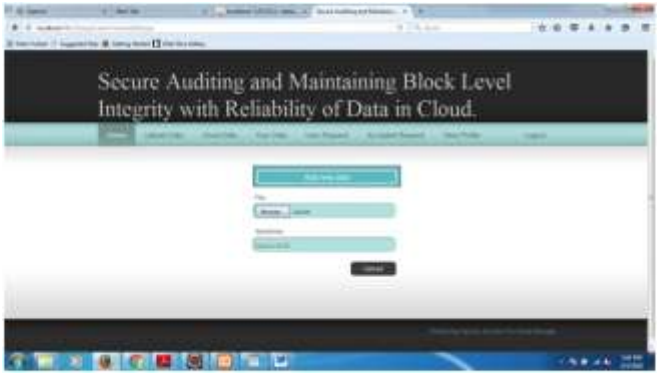
Secure Auditing and Maintaining Block Level
Integrity with Reliability of Data in Cloud.

Fig. 6. User needs to upload file using this screen.

V. Conclusion

Aiming to achieve both that is data integrity as well as block level deduplication in cloud, proposed in the above algorithm.

This helps the clients to generate the block tags before uploading as well as it helps to audit the integrity of data having been stored in cloud.

Additionly, this will enable secure block level deduplication through introducing a Proof of Ownership protocol and preventing all the leakage of side channel information in data deduplication. Thus secure auditing,maintaining block level integrity and will provide reliability of data.

References

[1] Secure Auditing and Deduplicating data in cloud, Jingwei Li , Jin Li, Dongqing Xie and Zhang cai ,IEEE Transactions on computer vol . PP no 99 YEAR 2015”

[2] S. Keelveedhi, M. Bellare, and T. Ristenpart, Dupless: Server- aided encryption for deduplicated storage, in Pro- ceedings of the 22Nd USENIX Conference on Security, ser. SEC13. Washington, D.C.: USENIX Association, 2013, pp. 179194.

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, Remote data checking using provable data possession, ACM Trans. Inf. Syst. Secur., vol. 14, no. 1, pp. 12:112:34, 2011.

[4] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, Proofs of ownership in remote storage systems, in Proceedings of the 18th ACM Conference on Computer and Communications Security. ACM, 2011, pp. 491500.

[5] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, Enabling public verifiability and data dynamics for storage security in cloud computing, in Computer Security ESORICS 2009, M. Backes and P. Ning, Eds., vol. 5789. Springer Berlin Heidelberg, 2009, pp. 355370.

[6] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, Provable data possession at un- trusted stores, in Proceed- ings of the 14th ACM Conference on Computer and Communications Security, ser. CCS 07. New York, NY, USA: ACM, 2007, pp. 598 609.

[7] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, Scal- able and efficient provable data possession, in Proceedings of the 4th International Conference on Security and Privacy in

Communication Netowrks, ser. SecureComm 08. New York, NY, USA: ACM, 2008, pp. 9:19:10.

Ms. Aishwarya Kumthekar is currently pursuing M.E (Computer) from Department of Computer Engineering, RMDSSOE, Savitribai Phule Pune University,Pune. She has received her B.E (Computer) Degree from Savitribai Phule Pune University, Pune. Her area of interest is cloud computing and security.

Prof. J.S. Raghatwan has master degree in information tech- nology from Pune Uni- versity. She has 7 years of experience in teaching field. Currently working as Assistant Professor at RMDSSOE ,Warje Pune.Her area of interest is information security