DEDUPLICABLE DYNAMIC PROOF OF STORAGE AND FILE SHARING FOR MULTI USER ENVIRONMENT

Mangesh Todkari(*Author*)
BE IT SKNSITS
Lonavala, India

Saif Khan (*Author*)
BE IT SKNSITS
Lonavala, India

Mayur Kadam(*Author*)
BE IT SKNSITS
Lonavala, India

Kaustubh Borate (*Author*)
BE IT SKNSITS
Lonavala, India

Snehal Khartad (*Guide*)
SKNSITS
Lonavala, India

ABSTRACT

Dynamic Proof of Storage (PoS) is a useful cryptographic primitive that enables a user to check the integrity of outsourced _les and to efficiently update the files in a cloud server. Although researchers have proposed many dynamic PoS schemes in single user environments, the problem in multi-user environments has not been investigated sufficiently. A practical multi-user cloud storage system needs the secure client-side cross-user deduplication technique, which allows a user to skip the uploading process and obtain the ownership of the files immediately, when other owners of the same files have uploaded them to the cloud server. To the best of our knowledge, none of the existing dynamic PoSs can support this technique. In this paper, we introduce the concept of deduplicatable dynamic proof of storage and propose an efficient construction called DeyPoS, to achieve dynamic PoS and secure cross-user deduplication, simultaneously. Considering the challenges of structure diversity and private tag generation, we exploit a novel tool called Homomorphic Authenticated Tree (HAT). We prove the security of our construction, and the theoretical analysis and experimental results show that our construction is efficient in practice.

Keyword - *DeyPoS , cloud, deduplication ,PoS etc*.

1. INTRODUCTION

STORAGE outsourcing is becoming more and more attractive to both industry and academic due to the advantages of low cost, high accessibility, and easy sharing. As one of the storage outsourcing forms,cloud storage gains wide attention in recent years. Many companies, such as Amazon, Google, and Microsoft, provide their own cloud storage

services, where users can upload their files to the servers, access them from various devices, and share them with the others. Although cloud storage services are widely adopted in current days, there still remain many security issues and potential threats .Data integrity is one of the most important properties when a user outsources its files to cloud

storage. Users should be convinced that the files stored in the server are not tampered. Traditional techniques for protecting data integrity, such as message authentication codes (MACs)and digital signatures, require users to download all of the files from the cloud server for verification,which incurs a heavy communication cost. These techniques are not suitable for cloud storage services where users may check the integrity frequently, such as every hour. Thus, researchers introduced Proof of Storage (PoS) for checking the integrity without downloading files from

the cloud server. Furthermore, users may also require several dynamic operations, such as modification, insertion, and deletion, to update their files, while maintaining the capability of PoS. Dynamic PoS is proposed for such dynamic operations. In contrast with PoS, dynamic PoS employs authenticated structures, such as the Merkle tree. Thus, when dynamic operations are executed, users regenerate tags (which are used for integrity checking, such as MACs and signatures) for the updated blocks only, instead of regenerating for all blocks. To better understand the following contents, we present more details about PoS and dynamic PoS. In these schemes, each block of a file is attached a(cryptographic) tag which is used for verifying the integrity of that block. When a verifier wants to check the integrity of a file, it randomly selects some block indexes of the file, and sends them to the cloud server. According to these challenged indexes, the cloud server returns the corresponding blocks along with their tags. The verifier checks the block integrity and index correctness. The former can be directly guaranteed by cryptographic tags. How to deal with the latter is the major difference between PoS and dynamic PoS In most of the PoS schemes, the block index is “encoded” into its tag, which means the verifier can check the block integrity and index correctness simultaneously. However, dynamic PoS cannot encode the block indexes into tags, since the dynamic operations may change many indexes of non-updated blocks, which incurs unnecessary computation and communication cost.

2. PROPOSED SYSTEM

-In this System system model considers two types of entities: the cloud server and users, For each file, original user is the user who uploaded the file to the cloud server, while subsequent user is the user who proved the ownership of

the file but did not actually upload the file to the cloud server.

-There are five phases in a de duplicatable dynamic PoS system: pre-process, upload, deduplication, update, and proof of storage. In the pre-process phase, users intend to upload their local files.

- The cloud server decides whether these files should be uploaded. If the upload process is granted, go into the upload phase; otherwise, go into the deduplication phase.

• In the upload phase, the files to be uploaded do not exist in the cloud server.The original users encodes the local files and upload them to the cloud server.

• In the duplication phase, the files to be uploaded already exist in the cloud server. The subsequent users possess the files locally and the cloud server stores the authenticated structures of the files. Subsequent users need to convince the cloud server that they own the files without uploading them to the cloud server.

• Note that, these three phases (pre-process, upload, and deduplication) are executed only once in the life cycle of a file from the perspective of users.That is, these three phases appear only when users intend to upload files. If these phases terminate normally, i.e., users finish uploading in the upload phase, or they pass the verification in the deduplication phase, we say that the users have the ownerships of the files.The duplicate files are mapped with a single copy of the file by mapping with the existing file in the cloud.The comprehensive requirements in multi-user cloud storage systems and introduced the model of deduplicatable dynamic PoS.

2.1SYSTEM ARCHITECURE

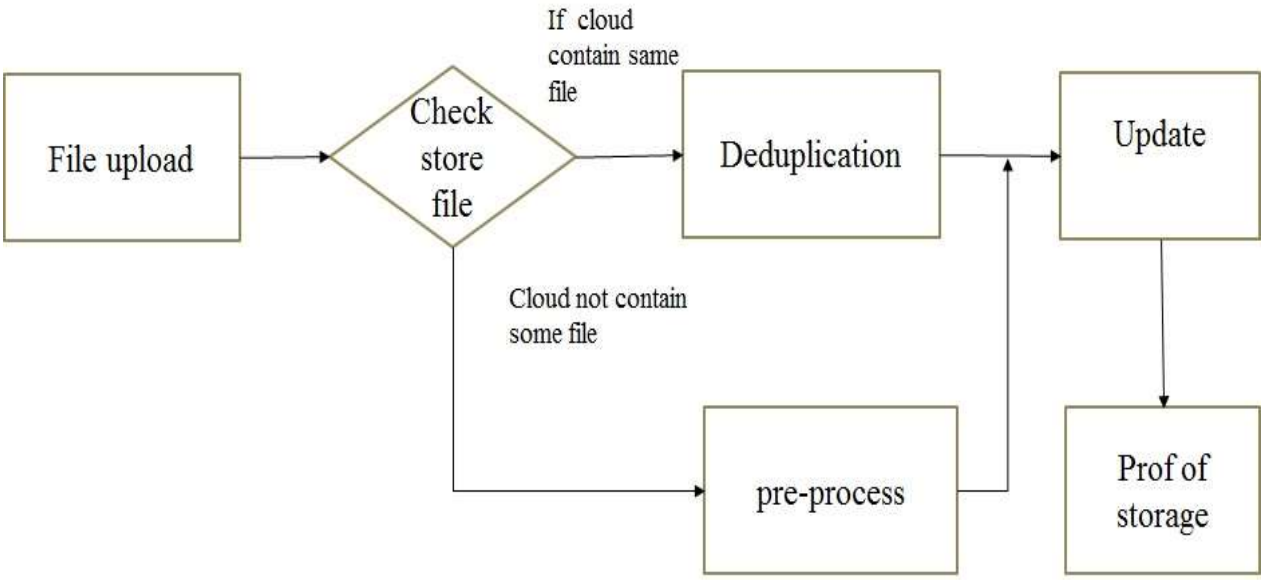


Figure 1. Architecture Diagram of Proposed System

3. MATHEMATICAL MODEL

Let S be the Whole system which consists,

S= {I, P, O}

Where,

I-Input,

P- procedure,

O- Output.

I-{F,Q}

F-Filesset of {f1,f2,…,fn}

Q- Users Query{q1,q2,…….,qN}

Procedure(P):

Where :

F = represents the file,

m1,m2,m3,m4=represents the i^{th} block of the file,

e=encryption key.

Step 1: Pre-process Phase

In the pre-process phase,

e ← H(F), id ← H(e).

Then, the user announces that it has a certain file via id. If the file does not exist, the user goes into the upload phase. Otherwise, the user goes into the deduplication phase.

Step 2 The Upload Phase

Let the file F = (m1, . . . ,mn).

The user first invokes the encoding according

(C, T) ← Encode(e, F)

Step 3. The Deduplication Phase

If a file announced by a user in the pre-process phase exists in the cloud server, the user goes into the deduplication phase and runs the deduplication protocol

res ∈ {0, 1} ← Deduplicate{U(e, F), S(T)}

Step: 4 The Update Phase

In this phase, a user can arbitrarily update the file, by invoking the update protocol

res ∈ {he∗, (C∗, T ∗)i,⊥} ← Update{U(e, ι,m,OP), S(C, T)}

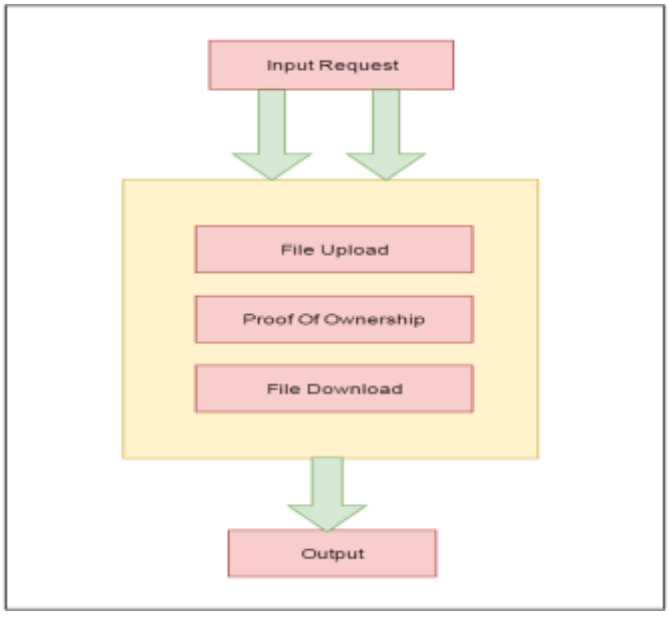
Step 5: The Proof of Storage Phase

At any time, users can go into the proof of storage phase if they have the ownerships of the files. The users and the cloud server run the checking protocol

res ∈ {0, 1} ←Check{S(C, T), U(e)}

Output(O):

User can upload, download update on cloud server and provide data dedupliaction.



I. MODELS USED IN SYSTEM

1.User Module:

2.Admin Module:

1]User Module:-

- New User Registration

- Give Attributes or Privilege When User register e. g. Student or Staff etc.

-User login in system

-user Upload file in system.

-User select privilege or attribute first e.g. student or staff

-Browse Text File to Upload and click on Upload button and generates tag file for it.

-If tag exist in server database then file is deduplicated & print message - file already exist, then give proof of ownership pointer to this user of existing file for accessing & this user is also owner of that existing file.

-If tag not exist in server database then file is unique then encrypt file and stored on cloud folder in drive.

-User also can download file from cloud.

-user shows all file that his own uploaded i.e. unique file & deduplicated file

-click on download link to download that file

4. Access File

-user shows all files for his attribute uploaded by owner of file.

-click on download link to download that file

5. Graph

1. Upload Time & Encrypt Time Graph

2. Download Time & Decrypt Time Graph

6. Logout

Admin Module:

1. **Admin Login**

-Account Activation-

-All users request for activation- if admin click on active then user account activated.

-show all users uploaded files

-show all users deduplicated files

-Logout.

.

4. CONCLUSIONS

We proposed the comprehensive requirements in multi-user cloud storage systems and introduced the model of deduplicatable dynamic PoS. We designed a novel tool called HAT which is an efficient authenticated structure. Based on HAT, we proposed the first practical deduplicatable dynamic PoS scheme called DeyPoS and proved its security in the random oracle model. The theoretical and experimental results show that our DeyPoS implementation is efficient, especially when the file size and the number of the challenged blocks are large.

5. ACKNOWLEDGEMENT

This research work is done under the guidance of Prof. Snehal Khartad.

6. REFERENCES

[1] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Proc. of FC*, pp. 136–149, 2010.

[2] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[3] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 843–859, 2013.

[4] C. A. Ardagna, R. Asal, E. Damiani, and Q. H. Vu, "From Security to Assurance in the Cloud: A Survey," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 2:1–2:50, 2015.