IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727,
PP 19-21
www.iosrjournals.org

Reasonable Successful POS for Multi-User Surroundings.

Priyanka Y. Barve, Hina L. Tadvi, Atharva R. Karmase., Prof.A.A.Pundlik.
LokneteGopinathji Munde Institute of Engineering Education & Research
Computer Department, Savitribai Phule Pune Univercity Pune, India.

Abstract: *Data de- duplication is nothing but data compression method which is used to eliminate the duplicate copies of repeating data. This approach is frequently used for reducing the storage space and save bandwidth under cloud server. Dynamic Proof of Storage (POS) is a useful cryptographic primitive that enables a user to check the integrity of outsourced files 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 de-duplication 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. In this introduce the concept of de-duplicatable dynamic proof of storage and propose an efficient construction, to achieve dynamic POS and secure cross-user de-duplication, simultaneously. In this project we are presenting the authorized data de-duplication to protect the data security by including differential privileges or attribute of users in the duplicate check. Different new de-duplication constructions presented for supporting authorized duplicate check.*

Keywords - *Cloud storage, dynamic proof of storage, de-duplication, authorized duplicate check, confidentiality.*

I. Introduction

Today's cloud service suppliers provide each extremely offered storage and massively parallel computing resources at comparatively low prices. One important challenge of cloud storage services is that the management of the ever-increasing volume of knowledge. To make knowledge management ascendable in cloud computing, de-duplication has been a widely known technique and has attracted additional and additional attention recently. Knowledge de-duplication could be a specialized knowledge compression technique for eliminating duplicate copies of continuance knowledge in storage. The technique is employed to enhance storage utilization and might even be applied to network knowledge transfer to cut back the amount of bytes that has to be sent rather than keeping multiple knowledge copies with an equivalent content, de-duplication eliminates redundant knowledge by keeping just one physical copy and referring different redundant knowledge to it copy. De-duplication will take place at either the file level or the block level. For file level de-duplication, it eliminates duplicate copies of an equivalent file. De-duplication can even occur at the block level, that eliminates duplicate blocks of information that occur in non-identical files.

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 such an authorized de-duplication system, each user is issued attribute during system initialization .Each file uploaded to the cloud is also bounded by attribute to specify which kind of users is allowed to perform the duplicate check and access the files. Before submitting his duplicate check request for some file, the user needs to take this file and attributes with file as inputs. The user is able to find a duplicate for this file if and only if there is a copy of this file and a matched attribute stored in cloud.

II. Literature Review

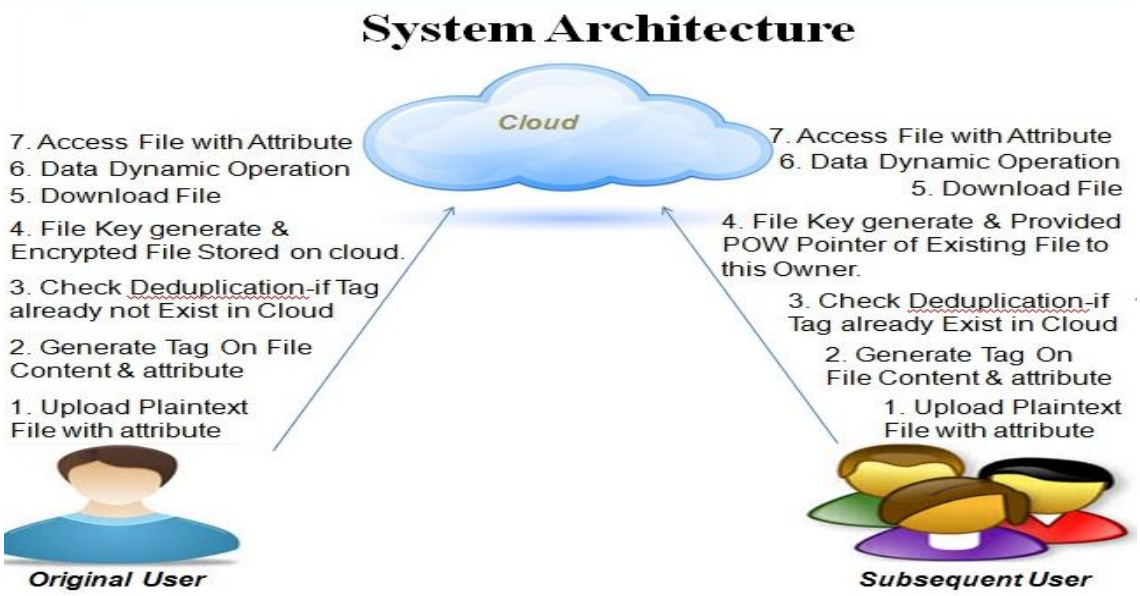
Cloud computing can organized lot of resource of computing, storage and application with great efficiency and minimal economic overhead. Cloud computing presents a secured multi keyword ranked search schemes over encripted cloud which supports dynamic update operation like delete and insertion of document[2]. Security and privacy issue present a strong barriers for users to adapt into cloud computing system ,multi located data storage and services make privacy issues more bad[3]. More security strategies should be deployed in the cloud environment to achieve availability ,confidentiality ,data integrity, control and audit[4]. Dynamic proof of storage is a useful cryptographic primitive which enable user to check the integrity of out-source file and to efficiently update the file to cloud server. Techniques used in PoS and Dynamic PoS are homo-morphic message authentication code and homo-morphic signature which reduces the communication cost[5]. To further reduce the cost and to achieve security we designed a homo-morphic authenticated tree (HAT).

HAT will support dynamic operation cross user de-duplication integrity verification.

III. Proposed System

These system consist of two types of entities cloud server and users like original and subsequent. Original user uploaded the file to the cloud server and subsequent user who proved the ownership of the file .there are five phases in a de-duplicatable dynamic Pos system-Preprocess, upload, de-duplication, update and proof of storage.

3.1 System Architecture



3.2 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-File set of {f1,f2,…,fn}

Q- Users Query{q1,q2,……,qN}

Procedure(P):

Where :

F = represents the file,

e=encryption key.

3.2.1: 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 de-duplication phase.
File Tag(File, Attribute) - It computes SHA-1 hash of the File and Attribute as File Tag.

3.2.2 The Upload Phase

Let the file F = (m1, . . . ,mn).
The user first invokes the encoding according
(C, T)←Encode(e, F)
File Encrypt(File) - It encrypts the File with Convergent Encryption using Hybrid AES-DES algorithm.

File Upload Req(FileID, File, Tag) – It uploads the File Data to the Storage Server if the file is Unique and updates the File Tag stored.

3.2.3 The De-duplication Phase

If a file announced by a user in the pre-process phase exists in the cloud server, the user goes into the de-duplication phase and runs the de-duplication protocol res ∈ {0, 1} ← De-duplicate{U(e, F), S(T)}
DupCheckReq(Tag) - It requests the Storage Server for Duplicate Check of the File by sending the file Tag

3.2.4 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)}
Insertion(Tag, File) – It request for insert the new data.
Modification(Tag, File) – It request for modify the file that he is uploaded into file that already available in cloud.
Deletion(Tag, File) – It request for delete the file that already available in cloud.

3.2.5 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)}

3.2.6 Access Phase

When each file uploaded to the cloud is also bounded by a privilege or attribute to specify which kind of users is allowed to perform the duplicate check and access the files.
ShareFileReq(File, Attribute) - It requests the Storage server to generate the Share File with the File Tag and Target Sharing Attribute.

Output(O):

User can upload, download, update and access file with attribute on cloud server and provide data deduplication.

IV. Performance Evaluation

Evaluation consist of cost in upload phase, cost in the deduplication phase and cost in proof of storage phase. In upload phase first mentioned initialization time for contructing HAT with different sizes of files and blocks . When the block size is 4kb, the authenticator size is less than 3% of the file size . The communication cost considers the data send from users and data send from the cloud server .

V. Conclusion

This system proposed the comprehensive requirements in multi-user cloud storage systems and introduced the model of de-duplicatable dynamic POS. De-duplicatable Dynamic Proof of Storage implementation is efficient, when the file size is large. Each file uploaded to the cloud is bounded by a privilege or attribute to specify which kind of users is allowed to perform the duplicate check and access the files. We developed a novel tool HAT.

References

[1] Kun He, Jing Chen, Ruiying Du, Qianhong Wu, GuoliangXue, and Xiang Zhang" DeyPoS: Deduplicatable Dynamic Proof of Storage for Multi-User Environments" IEEE Transactions on Computer, Volume: 65, Issue: 12, pp. 3631 - 3645, 2016.
[2] Zhihua Xia, X. sun ,Qian Wang " A secure and dynamic multi-keyword ranked search scheme over Encrypted cloud data".
[3] Minqui zhoul,Rong Zhang, Wei xie,Weining Qian "security and privacy in cloud computing :A survey"
[4] R.Gennaro and D. Wichs ," Fully Homomorphic message authenticators, " in proc. of ASIACRYPT, pp.
[5] D.Boneh and D.M.Freeman, "Homomorphic signatures for polynomial functions" in proc. of EUROCRYPT, pp.
[6] D.Catalano , D.Fiore, and B.Warinschi, "Homomorphic signatures with efficient verification for polynomial functions" in proc. of CRYPTO, pp.