Scientific Journal of Impact Factor (SJIF): 4.14

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 3, Issue 9, September -2016

Reproducible Effective POS for Multiple User Environment.

Prof. A.A.Pundlik[1], BarvePriyanka Yashwant[2], TadaviHina Latif[3], KarmaseAtharva Rajendra[4]

LokneteGopinathjiMunde Institute of Engineering Education & Research

Abstract:*Dynamic Proof of Storage (PoS) could be a helpful scientific discipline primitive that allows a user to see the integrity of outsourced files and to with efficiency update the files in a very cloud server. though researchers have planned several dynamic PoS schemes in single user environments, the matter in multi-user environments has not been investigated sufficiently. A sensible multi-user cloud storage system wants the secure client-side cross-user deduplication technique, that permits a user to skip the uploading method and procure the possession of the files now, once alternative house owners of an equivalent files have uploaded them to the cloud server. To the simplest of our data, none of the present dynamic PoSs will support this system. during this paper, we have a tendency to introduce the conception of deduplicatable dynamic proof of storage associated propose an economical construction referred to as DeyPoS, to realize dynamic PoS and secure cross-user deduplication, at the same time. Considering the challenges of structure diversity and personal tag generation, we have a tendency to exploit a unique tool referred to as Homomorphicgenuine Tree (HAT). we have a tendency to prove the protection of our construction, and therefore the theoretical analysis and experimental results show that our construction is economical in follow.*

Keywords:*Deduplication ,Proof of ownership, Dynamic proof of storage, Cloud Computing.*

I. INTRODUCTION

Storage outsourcing is turning into additional and additional enticing to each trade and tutorial because of the benefits of low value, high accessibility, and straightforward sharing. collectively of the storage outsourcing forms, cloud storage gains wide attention in recent years. several firms, like Amazon, Google, and Microsoft, give their own cloud storage services, wherever users will transfer their files to the servers, access them from varied devices, and share them with the others. though cloud storage services ar wide adopted in current days, there still stay several security problems and potential threats .Data integrity is one among the foremost vital properties once a user outsources its files to cloud storage. Users ought to be convinced that the files keep within the server don't seem to be tampered. ancient techniques for safeguarding knowledge integrity, like message authentication codes (MACs)and digital signatures, need users to transfer all of the files from the cloud server for verification, that incurs a significant communication value. These techniques don't seem to be appropriate for cloud storage services wherever users could check the integrity oftentimes, like each hour. Thus, researchers introduced Proof of Storage (PoS) for checking the integrity while not downloading files from the cloud server. what is more, users may need many dynamic operations, like modification, insertion, and deletion, to update their files, whereas maintaining the potential of PoS. Dynamic PoS is projected for such dynamic operations. In distinction with PoS, dynamic PoSemploysecht structures, like the Merkle tree. Thus, once dynamic operations ar dead, users regenerate tags (which are used for integrity checking, like MACs and signatures) for the updated blocks solely, rather than create for all blocks. to raised perceive the subsequent contents, we tend to gift additional details concerning PoS and dynamic PoS. In these schemes, every block of a file is hooked up a(cryptographic) tag that is employed for substantiating the integrity of that block. once a champion desires to ascertain the integrity of a file, it every which way selects some block indexes of the file, and sends them to the cloud server. consistent with these challenged indexes, the cloud server returns the corresponding blocks beside their tags. The champion checks the block integrity and index correctness. the previous are often directly bonded by cryptanalytic tags. a way to affect the latter is that the major distinction between PoS and dynamic PoS In most of the PoS schemes, the block index is “encoded” into its tag, which implies the champion will check the block integrity and index correctness at the same time. However, dynamic PoS cannot cypher the block indexes into tags, since the dynamic operations could modification several indexes of non-updated blocks, that incurs reserve computation and communication value. as an example, there's a file consisting of one thousand blocks, and a replacement block is inserted behind the second block of the file. Then, 998 block indexes of the first file ar modified, which implies the user should generate and send 999 tags for this update. structures are introduced in dynamic PoSs to unravel this challenge. As a result, the tags are hooked up to the structure instead of the block indexes .However, dynamic PoS remains to be improved in an exceedingly multi-user atmosphere, because of the necessity of cross-user American state duplication on the client-side. this means that users will skip the uploading method and acquire the possession of files now, as long because the uploaded files exist already within the cloud server. this method will cut back space for storing for the cloud server, and save transmission information measure for users. To the simplest of our data, there's no dynamic PoS that may support secure cross-user American state duplication.

II. LITERATURE REVIEW

Z. Xia, X. Wang, X. Sun[1]efficient and dynamic search scheme is proposed, which supports not only the accurate multi-keyword ranked search but also the dynamic deletion and insertion of documents.**Prof.minquizhou, Prof. Rongzhang**[2]PraposeSecurity and Privacy in Cloud Computing becomes a buzzword More and more companies step into Cloud and provide services above on it. **Prof. J.Rao Prof. rasoolasal ,Prof. Claudio A. Ardagna**[3]In the public cloud vision, infrastructure, platform, and software services are provisioned to tenants (i.e.customers and service providers) on a pay-as-you-go basis.Cloud tenants can use cloud resources at lower prices, and higher performance and flexibility, than traditional on-premises resources.**Prof. Yun Kuanchang ,Prof. Narn yin lee**[4]
We can provable data possession in this model, which reduce the data block access, but also reduce the amount of computation on the server and client and server traffic.**Prof. Luigi V. Mancini prof. Roberto Di Rietro**[5]learned that a MAC-based variant of our first basic scheme was mentioned by Ari Juels during a presentation at CCS . This is an independent and concurrent discovery.**Prof. Dan Feng ,Prof.Wei Xu ,Prof.Jingning Liu**[6]propose the first lattice-based PoS protocol from our new construction of LHTVs. Our LPoS protocol is public verifiable and unforgeable assuming SIS is hard.

III. DEDUPLICATION AND DYNAMIC PROOF OF STORAGE

No Such system of Dynamic proof of storage will Achive cross user deduplication. To remove this drawbacks we implimentDeduplicatable dynamic proof of storage.

1.1 System Model

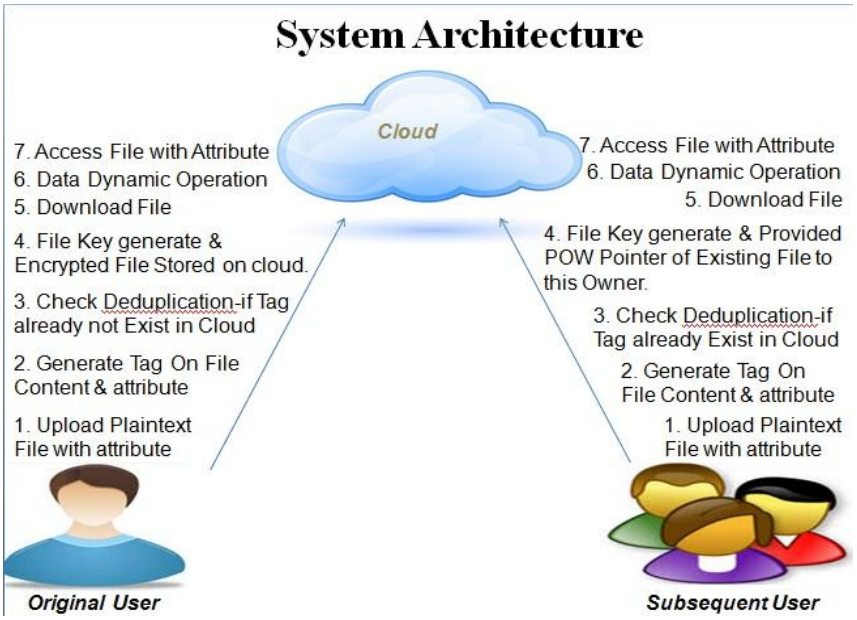
System Architecture
Cloud
7. Access File with Attribute
6. Data Dynamic Operation
5. Download File
4. File Key generate & Encrypted File Stored on cloud.
3. Check Deduplication-if Tag already not Exist in Cloud
2. Generate Tag On File Content & attribute
1. Upload Plaintext File with attribute
Original User
7. Access File with Attribute
6. Data Dynamic Operation
5. Download File
4. File Key generate & Provided POW Pointer of Existing File to this Owner.
3. Check Deduplication-if Tag already Exist in Cloud
2. Generate Tag On File Content & attribute
1. Upload Plaintext File with attribute
Subsequent User

Fig 2 System Architecture

As shown in Fig. 2. 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 deduplicatable dynamic PoS system: pre-process, upload, deduplication, update, and proof of storage.

1.1.1 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.

1.1.2 Upload 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**.**

3.1.3Deduplication Phase

The files to be uploaded alreadyexist in the cloud server. The subsequent users possessthe files locally and the cloud server stores the authenticatedstructures of the files. Subsequent users need to convince the cloud server that they own the files without uploading themto the cloud server.

If 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 deduplicationphase, we say that the users have the ownerships of the files.

3.1.4 Update Phase

Users may modify, insert, or delete some blocks of the files. Then, they update the corresponding parts of the encoded files and the authenticated structures in the cloud server, even the original files were not uploaded by themselves. Note that, users can update the files only if they have the ownerships of the files, which means that the users should upload the files in the upload phase or pass the verification in the deduplication.For each update, the cloud server has to reserve the original file and the authenticated structure if there exist other owners, and record the updated part of the file and the authenticated structure. This enables users to update a file concurrently in our model, since each update is only “attached” to the original file and authenticated structure.

3.1.5 Proof of Storage Phase

Users only possess a small constant size metadata locally and they want to check whether the files are faithfully stored in the cloud server without downloading them. The files may not be uploaded by these users, but they pass the deduplication phase and prove that they have the ownerships of the files. Note that, the update phase and the proof of storage phase can be executed multiple times in the life cycle of a file. Once the ownership is verified, the users can arbitrarily enter the update phase and the proof of storage phase without keeping the original files locally.

IV. CALCULATION

3.1 Pre-Process Phase

$$e \leftarrow H(F),\ id \leftarrow H(e).$$

Where,
id = File Identity.

3.2 Upload Phase

FileF = (m1, . . . ,mn).

The user first invokes the encoding according,

$$(C, T) \leftarrow Encode(e, F)$$

Where,
m1, . . . ,mn= Represents Ithblock of file.
e = Encryption key.

3.3 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

$$res \in \{0, 1\} \leftarrow Deduplicate\{U(e, F), S(T)\}$$

Where,
res = Current uploading file.
e = Encryption Key.
F= Uploaded File.

3.4 Step: 4 The Update Phase

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

$$res\in \{he*, (C*, T*)i, \perp\} \leftarrow Update\{U(e, \imath, m, OP), S(C, T)\}$$

Where,
res= Current updating file.
S(C,T)= Represent block to be uploaded.

3.5 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

$$res\in \{0, 1\} \leftarrow Check\{S(C, T), U(e)\}$$

Where,
res =Current file.
S(C,T)= Block of file.

V. CONCLUSION

We plannedthe great necessities in multi-user cloud storage systems and introduced the model of deduplicatable dynamic PoS. we had developed a unique tool known as HAT that is Associate in Nursing economical genuine structure. supported HAT, we had planned the primary sensible deduplicatable dynamic PoS theme known as DeyPoS and evidenced its security within the random oracle model. The theoretical and experimental results show that our DeyPoS implementation is economical, particularly once the file size and therefore the range of the challenged blocks area unit giant.

VI. REFRENCES

[1] 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.

[2] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," IEEE Communications Surveys Tutorials, vol. 15, no. 2, pp. 843859, 2013.

[3] 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

[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. of CCS, pp. 598–609, 2007.

[5] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," in Proc. Of SecureComm, pp. 1–10, 2008. G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in Proc. of ASIACRYPT, pp. 319–333, 2009.

[6] C. Erway, A. K¨upc ¨u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. of CCS, pp. 213–222, 2009.

[7] R. Tamassia, "Authenticated Data Structures," in Proc. of ESA, pp. 2–5, 2003.

[8] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS, pp. 355–370, 2009.

[9] F. Armknecht, J.-M. Bohli, G. O. Karame, Z. Liu, and C. A. Reuter, "Outsourced proofs of retrievability," in Proc. of CCS, pp. 831–843, 2014.

[10] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Journal of Cryptology, vol. 26, no. 3, pp. 442–483, 2013.