Deduplication of Dynamic Proof of Storage for Multi-User Environments

Rahul Hulsure
Final Year Diploma in Computer Technology
Vishweshwarayya Polytechnic College, Almala
LATUR

Amol Datal
Final Year Diploma in Computer Technology
Vishweshwarayya Polytechnic College, Almala
LATUR

Ambulge S. S.
Head of Department
Computer Technology
Vishweshwarayya Polytechnic College, Almala
LATUR

ABSTRACT

Now a days Dynamic Proof of Storage (DeyPoS) could be a helpful crypto graphical primitive that permits a user to survey the integrity of outsourced files and to with efficiency update the files during a cloud server. Though researchers have planned several DeyPoS schemes in single user environments, the matter in multi-user environments has not been investigated sufficiently. A sensible multi-user cloud storage system requirement the secure client-side cross-user deduplication technique, that permits a user to skip the uploading method and obtain the possession of the files now, once different house owners of identical files have uploaded them to the cloud server. To the best of our data, none of the fundamental DeyPoS will support this system. during this paper, we tend to introduce the construct of deduplicatable dynamic proof of storage Associate in Nursing d propose an economical creation referred to as DeyPoS, to realize dynamic DeyPoS and secure cross-user deduplication, at the same time. Considering the challenges of structure diversity and personal tag generation, we tend to exploit a novel tool referred to as Homo morphic documented Tree (HAT). We tend to prove the protection of our construction, and therefore the theoretical analysis and experimental results show that our construction is economical in observe.

Keywords

Deduplication, Cloud Storage, encryption, Proof of Ownership, Revocation

1. INTRODUCTION

STORAGE outsourcing is becoming more and more attractive to both industry and academia 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 [1] [2]. 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 [3] [4]. 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.

2. LITERATURE REVIEW

1]. Towards E_cient Proofs of Retrievability in Cloud Storage. Author: JiaXu and Ee-Chien Chang in year 2012

In this paper author focused on Proofs of Retrievability (POR) is a cryptographic method for remotely auditing the integrity of les stored in the cloud, without keeping a copy of the original les in local storage. In a POR scheme, a user Alice backup her data le together with some authentication data to a potentially dishonest cloud storage server Bob. Later, Alice can sometimes and remotely verify the integrity of her data stored with Bob using the authentication data, without retrieving back the data le during a variation. Besides security, performances in statement, storage overhead and computation are most important considerations. Sachems and Waters [1] gave a fast scheme with O(s) communication bits and a factor of 1=s le size expansion. Although Ateniese et al. [2] achieves constant communication requirement with the same 1=s storage overhead, it requires rigorous computation in the setup and variation. In this paper, we incorporate a recent creation of constant size polynomial commitment scheme into Sachems and Waters [1] scheme. The resulting scheme requires constant communication bits (particularly, 720 bits if elliptic curve is used or 3312 bits if a modulo group is used) per verication and a factor of 1=s le size expansion, and its computation in the setup and verication is signicantly reduced compared to Ateniese et al.

2] Dynamic Provable Data Possession Author: C. Chris ErwayAlptekinK¨upc¸ ¨u In November 29, 2009

In this paper author proposed as storage-outsourcing services and resource-sharing networks have become widespread, the problem of resourcefully proving the integrity of data stored at untrusted servers has received increased consideration. In the provable data possession (PDP) model, the client preprocesses the data and then sends it to an untrusted server for storage, while possession a small amount of meta-data. The client later asks the server to prove that the stored data has not been tampered with or deleted (without downloading the actual data). However, the original PDP scheme applies only to static (or append-only) files. We present a definitional framework and efficient constructions for dynamic provable data possession (DPDP), which extends the PDP model to support provable updates to stored data. We use a new version of valid dictionaries based on rank information. The price of dynamic updates is a performance change from O(1) to O(log n) (or O(nǫ log n)), for a file containing of n blocks, while maintaining the same (or better, respectively) probability of misbehavior detection. Our experiments show that this slowdown is very low in practice (e.g., 415KB proof size and

30ms computational overhead for a 1GB file). We also show how to apply our DPDP scheme to outsourced file systems and version control systems (e.g., CVS).

3] Leakage Resilient Proofs of Ownership in Cloud Storage, Revisited Author: JiaXu and Jianying Zhou in year of 2009

Cloud storage service (e.g. Dropbox, Skydrive, Google Drive, iCloud, Amazon S3) is becoming more and more popular in recent years [1]. The volume of personal or business data stored in cloud storage keeps increasing [2,3,4]. In face to the challenge of rapidly growing volume of data in cloud, deduplication technique is highly demanded to save disk space by removing duplicated copies of the same le (Single Instance Storage). SNIA white paper [5] reported that the deduplication technique can save up to 90% storage, dependent on applications.

4] **A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data, Author: Zhihua *Member, IEEE,* Xinhui Wang Xia,in year of 2015**

In recent year the increasing popularity of cloud computing, more and more data owners are motivated to outsource their data to cloud servers for great convenience and reduced cost in data management. However, sensitive data should be encrypted before outsourcing for privacy requirements, which obsoletes data utilization like keyword-based document retrieval. In this paper, we present a secure multi-keyword ranked search scheme over encrypted cloud data, which concurrently supports dynamic update operations like deletion and insertion of documents. Specifically, the vector space model and the widely-used TFIDF model are combined in the index construction and query generation. We construct a special tree-based index structure and propose a "Greedy Depth-first Search" algorithm to provide efficient multi-keyword ranked search. The secure kNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. In order to resist statistical attacks, phantom terms are added to the index vector for blinding search results. Due to the use of our special tree-based index structure, the proposed scheme can achieve sub-linear search time and deal with the deletion and insertion of documents flexibly. Wide experiments are conducted to demonstrate the efficiency of the proposed scheme.

3. EXISTING SYSTEM

When user uploads data that already exist in the cloud storage, the user should be deterred from accessing the data that were stored before obtained the ownership by uploading it (backward secrecy) this dynamic ownership changes may occur very frequently in a practical cloud system, and thus, it should be properly managed in order to avoid the security degradation of the cloud service with multiple user handling. In the formal approach, most of the existing scheme have been proposed in order to perform a POW process in an efficient and robust manner, since the hash of the file, which is treated as "proof" for the entire file, is vulnerable to being leaked to outside adversaries because of its relatively small size if data owner uploads data unique and most first type of data in the cloud storage he is an initial uploader, if the data already exist called as subsequent uploader since this implies that others owner may have uploaded same data previously, he is called as subsequent uploader as per the multiple users are involved.

4. PROPOSED SYSTEM

Our system model considers two types of entities: the cloud server and users, as shown in Fig. 4.1. 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 deduplicative dynamic DeyPoS system: pre-process, upload, avoid the redundant data, 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 deduplication 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.

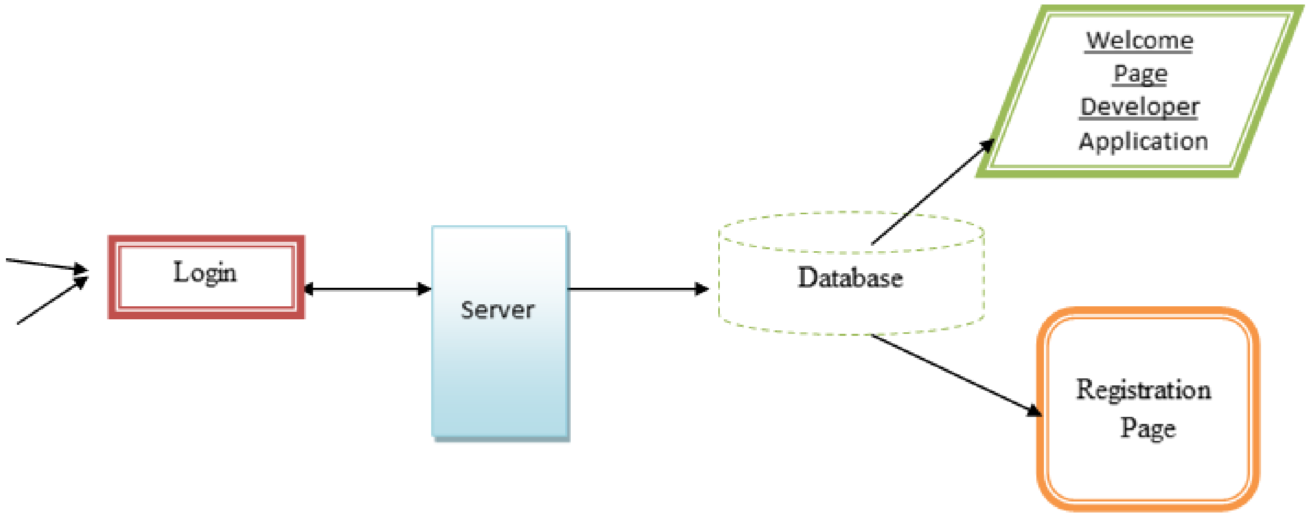

Fig 4.1: Proposed Architecture of Data Deduplication

5. PROPOSED SYSTEM TECHNIQUE

Approach	Cost	Throughput	Used Bandwidth	Deduplication ratio	Required Storage
directory Level Deduplication	Low	High	Low	Low	Medium
segment level Deduplication	High	Low	Low	High	Less
Parent level Deduplication	Relatively Low	Medium	Low	Medium	Medium
Location based Deduplication	High	Medium	High	Medium	Medium

5.1 ALGORITHM: The Deduplication Proving Algorithm

```
procedure DEDUPPROVE (αs, kc, αc, {c1, .. . , cn}, I,Q)
c ← 0, t ← ∅, ζ ← 1, l ← 1
while ζ ≤ n do
    δ ← 0
    while ζ <ɩjl do
        δ ← δ + c_ , ζ ← ζ + 1
    pop the first element in Q
    t ← t ∪ {fkc (iklikvi) + αcαsδ}
    c ← c + c_
    l ← l + 1, ζ ← ζ + 1
return c, t
```

6. FUTURE ENHANCEMENT

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.

7. CONCLUSION

We proposed the complete requirements in multi-user cloud storage systems and introduced the model of deduplicable DeyPoS. We designed a novel tool called HAT which is an efficient authenticated structure. Based on HAT, we proposed the first practical de duplicable 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.

8. 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.

[5] 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.

[6] 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.

[7] G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in Proc. of ASIACRYPT, pp. 319–333, 2009.

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

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

[10] 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.