Hybrid Cloud Approach For Secure Authorised Deduplication

[1] V. Ramakrishna, [2] P. Sai Sathvik, [3]K. Pravalika, [4]B. Prabhunandhan
[1]Assistant Professor, [2,3,4]B.Tech Student
[1,2,3,4]Department of CSE
[1,2,3,4]Anurag Group ofInstitutions, Venkatapur, Ghatkesar, Hyderabad, Telangana500038
Email: [1]ramakrishnacse@cvsr.ac.in, [2]16h61a05g1@cvsr.ac.in, [3]16h61a05e6@cvsr.ac.in, [4]16h61a05c9@cvsr.ac.in

***Abstract* – Data deduplication is one of essential data compression strategies for putting off replica copies of repeating data, and has been broadly used in cloud storage to reduce the amount of storage area and retailer bandwidth. To shield the confidentiality of sensitive information. whilst assisting deduplication, the convergent encryption approach has been proposed to encrypt the records earlier than outsourcing. To better shield data security, this paper makes the first strive to formally tackle the trouble of authorized data deduplication. Different from normal deduplication systems, the differential privileges of customers are further considered in duplicate check besides the data itself. We also present several new deduplication constructions supporting authorized duplicate check in a hybrid cloud architecture. Security analysis demonstrates that our scheme is invulnerable in phrases of the definitions specified in the proposed safety model. As a proof of concept, we enforce a prototype of our proposed approved reproduction test scheme and conduct testbed experiments the use of our prototype. We exhibit that our proposed approved reproduction take a look at scheme incurs minimal overhead compared to ordinary operations.**

I. INTRODUCTION

Cloud computing offers reputedly unlimited “virtualized” assets to customers as services throughout the total Internet, while hiding platform and implementation details. Today’s cloud carrier vendors provide each incredibly accessible storage and hugely parallel computing assets at pretty low costs. As cloud computing will become prevalent, an increasing amount of records is being stored in the cloud and shared through users with designated privileges, which define the get admission to rights of the stored data. One essential assignment of cloud storage services is the administration of the ever-increasing extent of data. To make facts administration scalable in cloud computing, deduplication has been a ordinary technique and has attracted greater and greater attention recently. Data deduplication is a specialized records compression technique for getting rid of duplicate copies of repeating data in storage. The technique is used to improve storage utilization and can also be applied to network data transfers to decrease the wide variety of bytes that have to be sent. Instead of maintaining a couple of records copies with the equal content, deduplication eliminates redundant facts by means of preserving only one bodily replica and referring different redundant information to that copy. Deduplication can take vicinity at both the file degree or the block level. For file level deduplication, it eliminates duplicate copies of the identical file. Deduplication can additionally take region at the block level, which eliminates replica blocks of facts that take place in non-identical files.

II. LITERATURESURVEY

Automated Certification for Compliant Cloud based Business Processes A key problem in the deployment of large-scale, reliable cloud computing concerns the difficulty to certify the compliance of business processes operating in the cloud. Standard audit procedures such as SAS-70 and SAS- 117 are hard to conduct for cloud based processes. The paper proposes a novel approach to certify the compliance of business processes with regulatory requirements. The approach translates process models into their corresponding Petri net representations and checks them against requirements also expressed in this formalism. Being Based on Petri nets, the approach provides well founded evidence on adherence and, in case of noncompliance, indicates the possible vulnerabilities. Keywords: Business process models, Cloud computing, Compliance certification, Audit, Petri nets. Automatic protocol blocker for privacy preserving public auditing in cloud computing Cloud Computing has been envisioned as the next generation architecture of IT enterprise, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk . As a disruptive technology with profound implications, Cloud Computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced into the Cloud. From users’ perspective, including both individuals and IT enterprises, storing data remotely into the cloud in a flexible on demand manner brings appealing benefits: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. While these advantages of using clouds are unarguable, due to the opaqueness of the Cloud—as separate administrative entities, the internal operation details of Cloud Service Providers (CSP) may not be known by cloud users—data outsourcing is also relinquishing user’s ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Secondly, for the benefits of their own, there do exist various motivations for cloud service providers to behave unfaithfully towards the cloud users regarding the status of their outsourced data. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. Thus, enabling public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed.

To securely introduce an effective third party auditor (TPA), the following two fundamental requirements have to be met:
TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user. Specifically, our contribution in this work can be summarized as the following three aspects:
1) We motivate the public auditing system of data storage security in Cloud Computing and provide a privacy preserving auditing protocol, i.e., our scheme supports an external auditor to audit user’s outsourced data in the cloud without learning knowledge on the data content.
2) To the best of our knowledge, our scheme is the first to support scalable and efficient public auditing in the Cloud Computing. In particular, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.
3) We prove the security and justify the performance of our proposed schemes through concrete experiments and comparisons with the state-of-the-art.
To enable privacy-preserving public auditing for cloud data storage under the aforementioned model, our protocol design should achieve the following security and performance guarantee:
1) Public auditability: to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional on-line burden to the cloud users.
2) Storage correctness: to ensure that there exists no cheating cloud server that can pass the audit from TPA without indeed storing users’ data intact.
3) Privacy-preserving: to ensure that there exists no way for TPA to derive users’ data content from the information collected during the auditing process.
4) Batch auditing: to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously.
5) Lightweight: to allow TPA to perform auditing with minimum communication and computation overhead.

III. EXISTING SYSTEM

In data deduplication systems, the private cloud is involved as proxy to allow data owners/users to securely perform duplicate check with differential privileges. The data owners only outsource their data storage by utilizing public cloud while data operation is managed in private cloud. In such system, each user is issued a set of privileges during system initialization and each file uploaded to cloud also bounded by a set of privileges to specify which kind of users is allowed to perform the duplicate check and access the file.

Disadvantages of the existing System:

Computing file token for sharing with other users having other privileges is not possible Private keys sharing among users is possible. Private keys sharing among users is possible.

IV. PROPOSED SYSTEM

In this new deduplication system, a hybrid cloud architecture is introduced to users directly, which will be kept and managed by the private cloud server instead. To perform duplicate check for some file, the user needs to get the file token from private cloud server. The private cloud server will also check the user's identity before issuing the corresponding token to user. The authorized duplicate check for this file can be performed by the user with the public cloud before uploading this file.

Advantages of the proposed system:

Users cannot share private keys of privileges in this proposed construction. Share file token can be generated to share file with users having different.

V. SYSTEM ARCHITECTURE

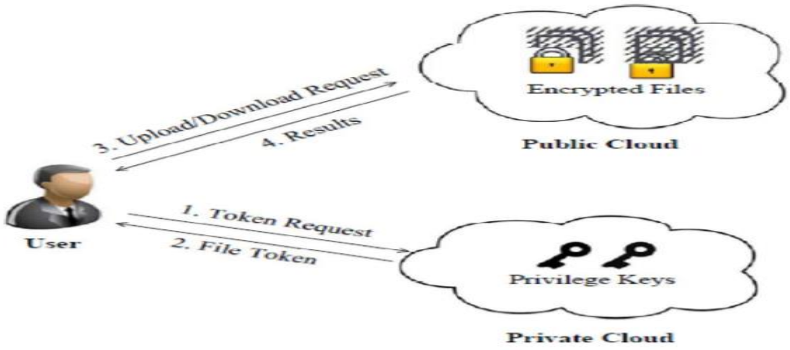


Fig1: System architecture

VI. ALGORITHM USED

Convergent Encryption: Convergent encryption provides data confidentiality in de-duplication. A user (or data owner) derives a convergent key from each original data copy and encrypts the data copy with the convergent key. In addition, the user also derives a tag for the data copy, such that the tag will be used to detect duplicates. Here, we assume that the tag for the data copy, such that the tag will be used to detect duplicates. Here, we assume that the tag correctness property holds, i.e., if two data copies are the same, then their tags are the same. To detect duplicates, the user first sends the tag to the server side to check if the identical copy has been already stored. Note that both the convergent key and the tag are independently derived and the tag cannot be used to deduce the convergent key and compromise data confidentiality. Both the encrypted data copy and its corresponding tag will be stored on the server side.

A convergent encryption scheme can be defined with four primitive functions:

1. **KeyGenCE**(M)!K is the key generation algorithm that maps a data copy M to a convergent key K.
2. **EncCE(K, M)!**C is the symmetric encryption algorithm that takes both the convergent key K and the data copy M as inputs and then outputs a ciphertextC;
3. **DecCE(K, C)!**M is the decryption algorithm that takes both the ciphertextC and the convergent key K as inputs and then outputs the original data copy M; and
4. **TagGen(M)!T (M**) is the tag generation algorithm that maps the original data copy M and outputs a tag T (M).

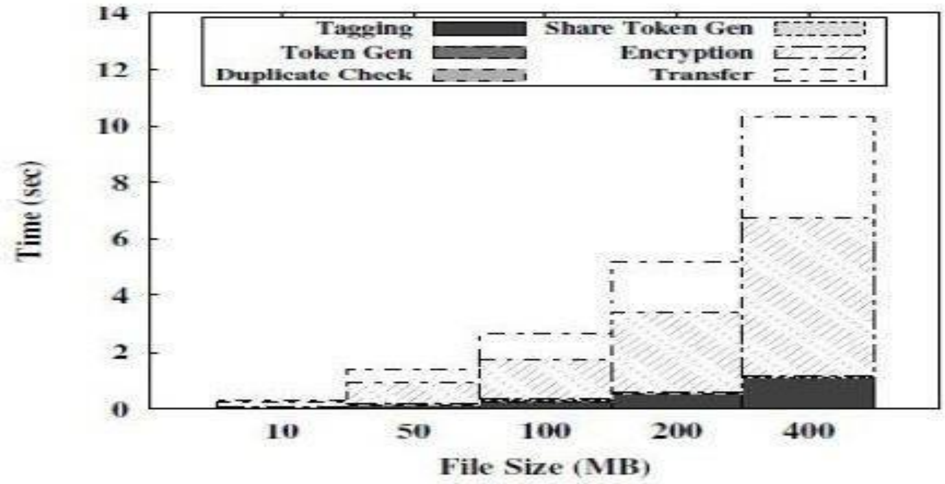
PSEUDO CODE:

```
Step1:Calculate the two convergent key values.
Step2: Compare the two keys and files get accessed.
Step3: Apply de-duplication to eradicated the duplicated
values.
Step4: If any other than the duplicates it will be checked
once again
Step5: That data will be unique and also more confidential the authorized can access and data is stored.
```

VII. IMPLEMENTATION RESULT

Our implementation of the Client provides the following function calls to support token generation and deduplication along the file upload process.

1. FileTag(File) - It computes SHA-1 hash of the File as File Tag;
2. TokenReq(Tag, UserID) - It requests the Private Server for File Token generation with the File Tag and User ID;
3. DupCheckReq(Token) - It requests the Storage Server for Duplicate Check of the File by sending the file token receivedfrom private server;
4. ShareTokenReq(Tag, {Priv.}) - It requests the Private Server to generate the Share File Token with the File Tag andTarget Sharing Privilege Set;
5. FileEncrypt(File) - It encrypts the File with Convergent Encryption using 256-bit AES algorithm in cipher block chaining
6. (CBC) mode, where the convergent key is from SHA-256 Hashing of the file;
7. FileUploadReq(FileID, File, Token) – It uploads the File Data to the Storage Server if the file is Unique and updates the
8. File Token stored. Our implementation of the Private Server includes corresponding request handlers for the tokengeneration and maintains a key storage with Hash Map.
9. TokenGen(Tag, UserID) - It loads the associated privilege keys of the user and generate the token with HMAC-SHA-1



VIII. CONCLUSION

The notion of authorized data deduplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new deduplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that these schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct testbed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

REFERENCES

1) Jingyi Li, Jigang Wu, Member, IEEE, Long Chen, Jiaxing Li, "Secure and Reliable Distributed Deduplication with Blockchain", IEEE, 6^{TH} CCF Conference on Big Data, Oct, 2018.

2) Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P.C. Lee, and Wenjing Lou, "A Hybrid Cloud Approach for Secure

Authorized Deduplication", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 26, NO. 5, MAY ,2015

3) S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.

4) J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. In ICDCS, pages 617–624, 2002.

5) OpenSSL Project. http://www.openssl.org/.

6) P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In Proc. of USENIX LISA, 2010.

7) M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013.