Effective Auditing Services for Data Security in Clouds

S.Jayanthi[1], S.Sivagami[2]

[1]PG Scholar, Dhanalakshmi srinivasan engineering college, perambalur, Tamil Nadu.

[2]Asst.Professor, Dept. of Information Technology, Dhanalakshmi srinivasan engineering college, perambalur , Tamil Nadu.

jaii.selva@gmail.com, ssivagami186@gmail.com

Abstract: Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. The introduction of effective TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing. Our Audit system can support dynamic data operations and timely anomaly detection with the help of several effective techniques, such as fragment structure, random sampling, and index hash table. We propose an efficient approach based on probabilistic query and periodic verification for improving the performance of audit services. Extensive security and performance analysis show the audit system verifies the integrity with the lower communication cost, minimum storage, and less computation overhead. The main technical contribution is that the proxy re-encryption scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. Our method fully integrates encrypting, encoding, and forwarding.

Key Terms—Storage security, provable data possession, audit service, cloud storage

I. INTRODUCTION

In cloud computing, one of the core design principles is dynamic scalability, which guarantees cloud storage service to handle growing amounts of application data in a flexible manner or to be readily enlarged. In cloud computing, cloud data storage contains two entities as cloud user and cloud service provider/ cloud server. Cloud user is a person who stores large amount of data on cloud server which is managed by the cloud service provider. User can upload their data on cloud without worrying about storage and maintenance. A cloud service provider will provide services to cloud user. The major issue in cloud data storage is to obtain correctness and integrity of data stored on the cloud. Cloud Service Provider (CSP) has to provide some form of mechanism through which user will get the confirmation that cloud data is secure or is stored as it is. No data loss or modification is done.

While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users' outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually 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, there do exist various motivations for CSP to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For examples, CSP might reclaim storage for monetary reasons by discarding data that has not been or even hide data loss incidents so as to maintain a reputation.

Security audit is an important solution enabling trace-back and analysis of any activities including data accesses, security breaches, application activities, and so on. The traditional cryptographic technologies, based on hash functions and signature schemes, cannot support for data integrity verification without a local copy of data. In addition, it is evidently impractical for audit services to download the whole data for checking data validation due to the communication cost, especially for large-size files. Therefore, following security and performance objectives should be addressed to achieve an efficient audit for outsourced storage in clouds:

Public auditability: To allow a third party auditor (TPA) or clients with the help of TPA to verify the correctness of cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to cloud services;

Dynamic operations: To ensure there is no attack to compromise the security of verification protocol or cryptosystem by using dynamic data operations;

Timely detection: To detect data errors or losses in outsourced storage, as well as anomalous behaviors of data operations in a timely manner;

Effective forensic: To allow TPA to exercise strict audit and supervision for outsourced data, and offer efficient evidences for anomalies;

Lightweight: To allow TPA to perform audit tasks with the minimum storage, lower communication cost, and less computation overhead.

In this paper, we introduce a dynamic audit service for integrity verification of untrusted and outsourced storages. Constructed on interactive proof system (IPS) with the zero- knowledge property, our audit service can provide public auditability without downloading raw data and protect privacy of the data. Also, our audit system can support dynamic data operations and timely anomaly detection with the help of several effective techniques, such as fragment structure, random sampling, and index-hash table (IHT). We also propose an efficient approach based on probabilistic query and periodic verification for improving the performance of audit services. A proof-of-concept prototype is also implemented to evaluate the feasibility and viability of our proposed approaches. Our experimental results not only validate the effectiveness of our approaches, but also show that our system does not create any significant computation cost and require less extra storage for integrity verification.

II. EXISTING SYSTEM

Traditional cryptographic technologies for data integrity and availability, based on hash functions and signature schemes cannot work on the outsourced data without a local copy of data. In addition, it is not a practical solution for data validation by downloading them due to the expensive communications, especially for large-size files.

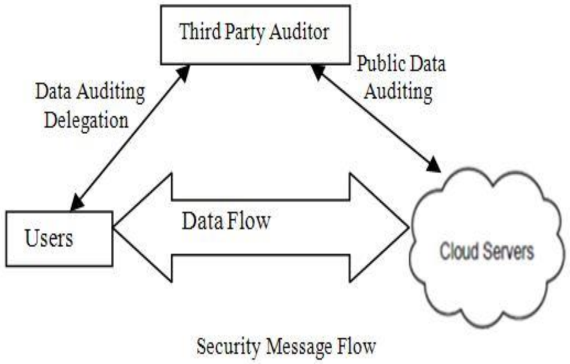


Fig 1: Architecture of Cloud Data storage service

Moreover, the ability to audit the correctness of data in a cloud environment can be formidable and expensive for cloud users. Therefore, it is crucial to realize public auditability for CSS, so that data owners (DOs) may resort to a TPA, who has expertise and capabilities that a common user does not have, for periodically auditing the outsourced data. This audit service is significantly important for digital forensics and data assurance in clouds.

III. LITERATURE SURVEY

A.Provable Data Possession*:*

Provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The goal of a PDP scheme that achieves probabilistic proof of data possession is to detect server misbehavior when the server has deleted a fraction of the file.

To meet these performance goals, our PDP schemes sample the server's storage, accessing a random subset of blocks. In doing so, the PDP schemes provide a probabilistic guarantee of possession; a deterministic guarantee cannot be provided without accessing all blocks. In fact, as a special case of our PDP scheme, the client may ask proof for all the file blocks, making the data possession guarantee deterministic. Sampling proves data possession with high probability based on accessing few blocks in the file, which radically alters the performance of proving data possession.

B. Pors: Proofs Of Retrivability:

The goal of a POR is to accomplish these checks without users having to download the files themselves. A POR can also provide quality of service guarantees, it show that a file is retrievable within a certain time bound. Setup phase: Verifier V encrypts the file F. It then embeds sentinels in random positions in F, sentinels being randomly constructed check values. Let F denote the file F with its embedded sentinels.Verification phase: In this paper Cryptographic techniques help users ensure the privacy and integrity of files they retrieve. It is also natural, however, for users to want to verify that archives do not delete or modify files prior to retrieval.

C. Data Possession for Hybrid Clouds*:*

It support multiple CSPs to cooperatively store and maintain the clients' data and a publicly verifiable PDP is used to verify the integrity and availability of their stored data in CSPs. The clients are allowed to dynamically access and update their data for various applications, and the verification process of PDP is seamlessly performed for the clients in hybrid clouds. Hence, it is a challenging problem to design a PDP scheme for supporting dynamic scalability.

In this work, we focus on the construction of PDP scheme for hybrid clouds, supporting privacy protection and dynamic scalability. We first provide an effective construction of Cooperative Provable Data Possession (CPDP) using Homomorphic Verifiable Responses (HVR) and Hash Index Hierarchy (HIH). This construction uses homomorphic property, such that the responses of the client's challenge computed from multiple CSPs can be combined into a single response as the final result of hybrid clouds.

Fragment Structure of CPDP: This structure relies on homomorphic properties to aggregate the data and tags into a constant size response, which minimizes network communication overheads.

D.Public Auditability:

It allow a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud is indeed intact, which achieving economies of scale for Cloud Computing.

Design Goals : Our design goals can be summarized as the following: 1) Public auditability for storage correctness assurance: to allow anyone, not just the clients who originally stored the file on cloud servers, to have the capability to verify the correctness of the stored data on demand; 2) Dynamic data operation support: to allow the clients to perform block-level operations on the data files while maintaining the same level of data correctness assurance.

E. Privacy-Preserving:

In cloud, data is stored in a centralized form and managing this data and providing security is a difficult task. TPA can read the contents of data owner hence can modify. The reliability is increased as data is handled by TPA but data integrity is not achieved. It uses encryption technique to encrypt the contents of the file. TPA checks the integrity of the data stored on a cloud but if the TPA itself leaks the user's data. Hence the new concept comes as auditing with zero knowledge privacy where TPA will audit the users' data without seeing the contents.

It uses public key based homomorphic linear authentication (HLA) which allows TPA to perform auditing without requesting for user data. It reduces communication & computation overhead. In this, HLA with random masking protocol is used which does not allow TPA to learn data content.

Goals:

- It allows TPA to audit users' data without knowing data content
- It supports batch auditing where multiple user requests for data auditing will be handled simultaneously.
- It provides security and increases performance

IV. PROPOSED SCHEME

The audit system, based on novel audit system architecture, can support dynamic data operations and timely abnormal detection with the help of several effective techniques, such as fragment structure, random sampling, and index-hash table. Furthermore, we propose an efficient approach based on probabilistic query and periodic verification for improving the performance of audit services. A proof-of-concept prototype is also implemented to evaluate the feasibility and viability of our proposed approaches.

The main technical contribution is that the proxy re-encryption scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. Our method fully integrates encrypting, encoding, and forwarding.

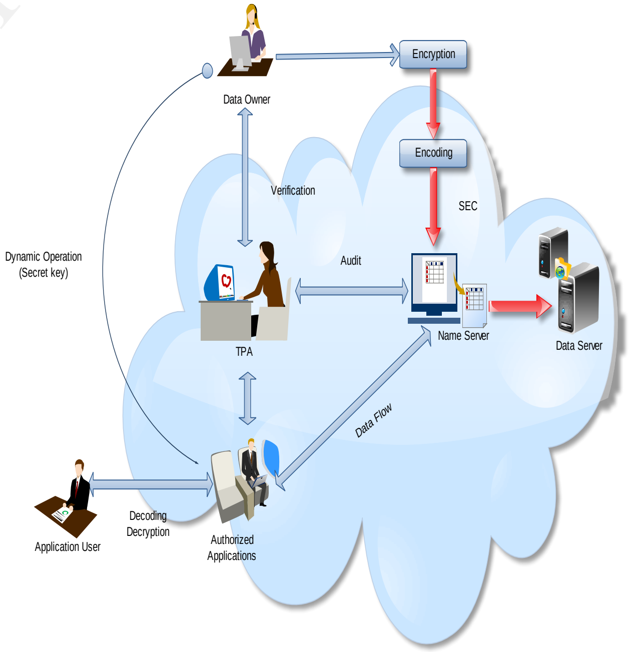


Fig.2 Audit system architecture

. The tight integration of encoding, encryption, and forwarding makes the storage system efficiently meet the requirements of data robustness, data confidentiality, and data forwarding.

V. MODULES

A. Authorization:

Data owner is register to cloud. . Then the cloud server allocates the memory and authorized id for Data Owner for Data Owner. The Data Owner register their details in cloud such as name, address, port no and buying size ..

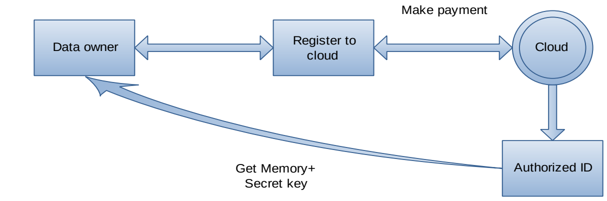


Fig: Registration

Then server send the amout details according to buying size. Data owner make payment to cloud server fter registration the server sends the user id and it allocates the memory for Data Owner. And select the type of plan chose in cloud system.

b.Tag generation*:*

The client (DO) uses an ID to preprocess a Outsourced file, which consists of a collection of n Blocks, generates a set of public verification parameters (PVPs) and IHT that are stored in TPA, transmits the file and some verification tags to CSP, and may delete its local copy.

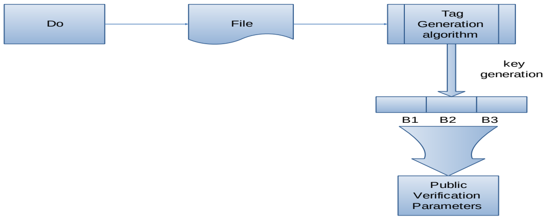
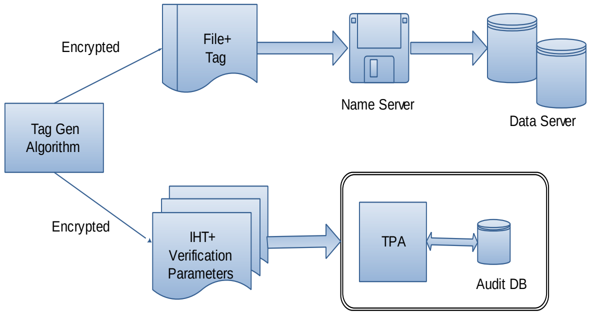


Fig:Tag Generation process

All operations are based on data blocks. The tag generation algorithm generate no of tags. The fragment framework consists of n block-tag pair with a signature tag of a block B generated by some secrets Keys. The aforementioned processes involve some procedures: KeyGen, TagGen, Update, Delete, Insert algorithms, as well as an Interactive Proof Protocol of Retrievability. TagGen (sk;F) takes a secret key sk and a file F, and returns a triple, where denotes the secret used to generate verification tags, is a set of *PVPs and IHT.*

C. Tag storage process:

Generates a set of public verification parameters (PVPs) and IHT that are stored in TPA, transmits the file and some verification tags to CSP, and may delete its local copy to maximize the storage efficiency and audit performance, our audit system introduces a general fragment structure for outsourced storages.



To store a file in a storage server, and maintains a corresponding authenticated index structure at a TPA. Index-Hash Table used to support dynamic data operations, we introduce a simple IHT to record the changes of file blocks, as well as generate the hash value of each block in the verification process. The structure of our IHT is similar to that of file block allocation table in file systems. Generally, the IHT consists of serial number, block number, version number, and random integer.

d. Probabilistic query based audit:

In contrast with "whole" checking, random "sampling" checking greatly reduces the workload of audit services, while still achieves an effective detection of misbehaviors.

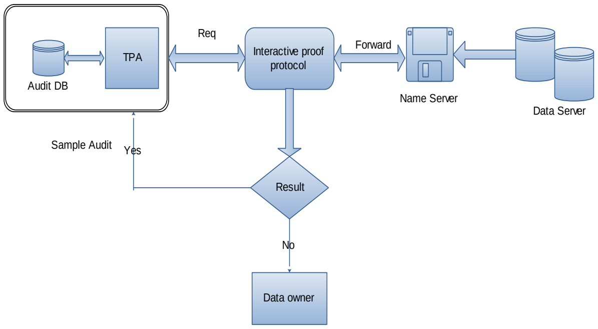


Fig: Probabilistic query based audit:

e. Periodic verification*:*

Too frequent audits may waste the network bandwidth and computing resources of TPA and CSPs. However, less frequent audits would not be conducive to detect the exceptions in a timely manner. Thus, it is necessary to disperse the audit tasks throughout the entire audit cycle so as to balance the overload and increase the difficulty of attacks in a relatively short period of time.

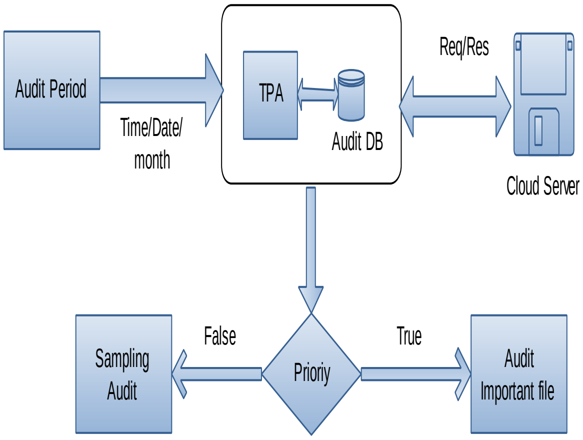


Fig: . *Periodic verification*

For example, a common audit period may be assigned as one week or one month, and the audit Period for important files may be set as one day.

f. Dynamic operations*:*

To ensure the security, dynamic data operations are available only to DOs or AAs, who hold the secret key sk. Definitions for dynamic data operations:

Update- is an algorithm run by AA to update the block of a file m0 i at the index i by using sk, and it returns a new verification metadata.

Delete- is an algorithm run by AA to delete the block mi of a file mi at the index i by using sk, and it returns a new verification metadata.

Insert- is an algorithm run by AA to insert the block of a file mi at the index i by using sk,and it returns a new verification metadata.

g. Secure erasure code (encoding):

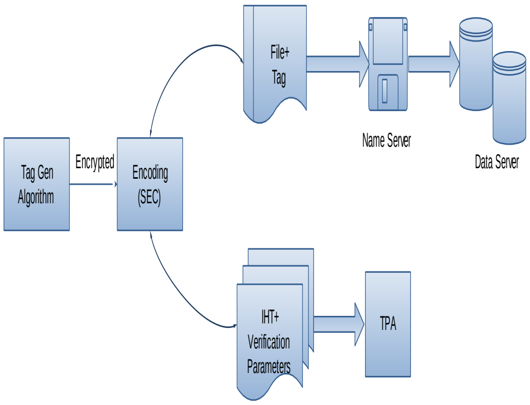


Fig: *Secure erasure code*

In the data storage phase, Data owner encrypts his message M and Upon receiving cipher texts from a user, the storage server performs Encode on the set of k cipher texts and stores the encoded result (codeword symbol) dispatches it to Cloud servers and TPA. One way to provide data robustness is to replicate a message such that each storage server stores a copy of the message.

h. Data retrieval:

In the data retrieval phase, Application user requests to retrieve a data from storage servers. The message is either stored by him or forwarded to him. User sends a retrieval request to Authorized Application.

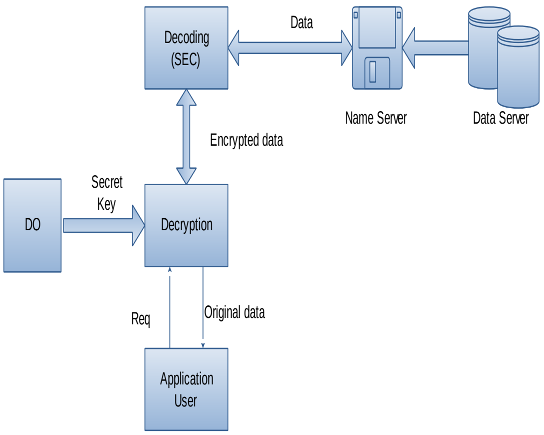


Fig: Data retrival

Upon receiving the retrieval request and executing a proper authentication process with a data owner, codeword symbols and does partial Decoding on the received codeword symbols by using the key share DO. Finally, Application user combines the partially decrypted codeword symbols to obtain the original data.

V. CONCLUSIONS

In this paper, we presented a construction of dynamic audit services for untrusted and outsourced storages. We also presented an efficient method for periodic sampling audit to enhance the performance of TPAs and storage service providers. Our experiments showed that our solution has a small, constant amount of overhead, which minimizes computation and communication costs.

REFERENCES

[1] Amazon Web Services, "Amazon S3 Availability Event: July 20, 2008," http://status.aws.amazon.com/s3-20080720.html, July 2008.

[2] A. Juels and B.S. Kaliski Jr., "PORs: Proofs of Retrievability for Large Files," Proc. ACM Conf. Computer and Communications Security (CCS '07), pp. 584-597, 2007.

[3] M. Mowbray, "The Fog over the Grimpen Mire: Cloud Computing and the Law," Technical Report HPL-2009-99, HP Lab., 2009.

[4] A.A. Yavuz and P. Ning, "BAF: An Efficient Publicly Verifiable Secure Audit Logging Scheme for Distributed Systems," Proc. Ann. Computer Security Applications Conf. (ACSAC), pp. 219-228, 2009.

[5] G. Ateniese, R.C. Burns, R. Curtmola, J. Herring, L. Kissner, Z.N.J. Peterson, and D.X. Song, "Provable Data

Possession at Untrusted Stores,” Proc. 14th ACM Conf. Computer and Comm. Security, pp. 598-609, 2007.

[6] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, “Scalable and Efficient Provable Data Possession,” Proc. Fourth Int’l Conf. Security and Privacy in Comm. Netowrks (SecureComm), pp. 1-10, 2008.

[7] C.C. Erway, A. Ku ¨pc¸u ¨, C. Papamanthou, and R. Tamassia, “Dynamic Provable Data Possession,” Proc. 16th ACM Conf. Computer and Comm. Security, pp. 213-222, 2009.

[8] H. Shacham and B. Waters, “Compact Proofs of Retrievability,” Proc. 14th Int’l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology Advances in Cryptology (ASIACRYPT ’08), J. Pieprzyk, ed., pp. 90-107, 2008.

[9] H.-C. Hsiao, Y.-H. Lin, A. Studer, C. Studer, K.-H. Wang, H. Kikuchi, A. Perrig, H.-M. Sun, and B.-Y. Yang, “A Study of User- Friendly Hash Comparison Schemes,” Proc. Ann. Computer Security Applications Conf. (ACSAC), pp. 105-114, 2009.

[10] A.R. Yumerefendi and J.S. Chase, “Strong Accountability for Network Storage,” Proc. Sixth USENIX Conf. File and Storage Technologies (FAST), pp. 77-92, 2007.

[11] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S. Yau, “Efficient Provable Data Possession for Hybrid Clouds,” Proc. 17th ACM Conf. Computer and Comm. Security, pp. 756-758, 2010.

[12] M. Xie, H. Wang, J. Yin, and X. Meng, “Integrity Auditing of Outsourced Data,” Proc. 33rd Int’l Conf. Very Large Databases (VLDB), pp. 782-793, 2007.

[13] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,” Proc. IEEE INFOCOM, pp. 1-9, 2010.

[14] B. Sotomayor, R.S. Montero, I.M. Llorente, and I.T. Foster, “Virtual Infrastructure Management in Private and Hybrid Clouds,” IEEE Internet Computing, vol. 13, no. 5, pp. 14-22, Sept./Oct. 2009.

[15] A. Bialecki, M. Cafarella, D. Cutting, and O. O’Malley, “Hadoop: A Framework for Running Applications on Large Clusters Built of Commodity Hardware,” technical report, http://lucene.apache. org/hadoop/, 2005.