ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 12, December 2015**

# A Survey on Privacy-Preserving Public Auditing For Regenerating-Code-Based Cloud Storage Using Attribute Based Approach

Meera Hanumant Ranadive[1], Prof.Bhavana Pansare[2]

Student, Dept. of Computer Engineering, Nutan Maharashtra Institute of Engineering & Technology, Talegaon Dabhade, Savitribai Phule Pune University, Pune, India

Professor, Dept. of Computer Engineering, Nutan Maharashtra Institute of Engineering & Technology, Talegaon Dabhade, Savitribai Phule Pune University, Pune, India

**ABSTRACT*:*** Cloud computing, is an emerging computing paradigm, enabling users to remotely store their data in a server and provide services on-demand. In cloud computing cloud users and cloud service providers are almost certain to be from different trust domains. Data security and privacy are the critical issues for remote data storage. A secure user enforced data access control mechanism must be provided before cloud users have the liberty to outsource sensitive data to the cloud for storage. With the emergence of sharing confidential corporate data on cloud servers, it is imperative to adopt an efficient encryption system with a fine-grained access control to encrypt outsourced data. Attribute-based encryption is a public key based encryption that enables access control over encrypted data using access policies and ascribed attributes. In this paper, we are going to analysis various schemes for encryption and possible solutions for their limitations, that consist of Attribute based encryption (ABE),KP-ABE, CP-ABE, Attribute-based Encryption Scheme with Non-Monotonic Access Structures. HABE,.To protect outsourced data in cloud storage against corruptions, adding fault tolerance to cloud storage together with data integrity checking and failure reparation becomes critical. Recently, regenerating codes have gained popularity due to their lower repair bandwidth while providing fault tolerance. Existing remote checking methods for regenerating-coded data only provide private auditing, requiring data owners to always stay online and handle auditing, as well as repairing, which is sometimes impractical. In this paper, we propose a public auditing scheme for the regenerating-code-based cloud storage. To solve the regeneration problem of failed authenticators in the absence of data owners, we introduce a proxy, which is privileged to regenerate the authenticators, into the traditional public auditing system model. Moreover, we design a novel public verifiable authenticator, which is generated by a couple of keys and can be regenerated using partial keys. Thus, our scheme can completely release data owners from online burden. In addition, we randomize the encode coefficients with a pseudorandom function to preserve data privacy. Extensive security analysis shows that our scheme is provable secure under random oracle model and experimental evaluation indicates that our scheme is highly efficient and can be feasibly integrated into the regenerating- code-based cloud storage.

**KEYWORDS:** Cloud storage, regenerating codes, public audit, privacy preserving, authenticator regeneration, proxy, and privileged, provable secure.

## I. INTRODUCTION

Cloud storage is now gaining popularity because it offers a flexible on-demand data outsourcing service with appealing benefits: relief of the burden for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personal maintenances,etc., Nevertheless, this new paradigm of data hosting service also brings new security threats toward users data, thus making individuals or enter prisers still feel hesitant. It is noted that data owners lose ultimate control over the fate of their outsourced data; thus, the correctness, availability and integrity of the data are being put at risk. On the one hand, the cloud service is usually faced with a broad range of internal/external adversaries, who would maliciously delete or corrupt users' data; on the other hand, the cloud serviceproviders may act dishonestly, attempting to hide data loss or corruption and claiming that the files are still correctly stored in the cloud for reputation or monetary reasons. Thus it makes great sense for users to implement an efficient protocol to perform periodical verifications of their outsourced data to ensure that the cloud indeed maintains their data correctly.

## II. RELATED WORK

**[1] "Above the clouds: A Berkeley view of cloud computing,"**
**From This Paper we Referred-**
The IT organizations have expresses concerns about critical issues (such as security) that exist with the widespread implementation of cloud computing. These types of concerns originate from the fact that data is stored remotely from the customer's location; in fact, it can be stored at any location. Security is one of the most argued-about issues in the cloud computing field; several enterprises look at cloud computing warily due to projected security risks.

**[2] "Provable data possession at untrusted stores,"**
**From This Paper we Referred-**
This keynote paper: In Cloud Computing moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new security challenges, which have not been well understood. This paper addressed the problem of ensuring the integrity of data storage in Cloud Computing.

**[3] PORs: Proofs of Retrievability for large files**
**From This Paper we Referred-**

The distributed storage systems apply redundancy coding techniques to stored data. One form of redundancy is based on regenerating codes, which can minimize the repair bandwidth, i.e., the amount of data transferred when repairing a failed storage node. Existing regenerating codes mainly require surviving storage nodes encode data during repair.

**[4] Multiple-replica provable data possession**
**From This Paper we Referred-**
In this approach, cloud computing is to avail all the resources at one place in the form a cluster and to perform the resource allocation based on request performed by different users. They defined the user request in the form of requirement query. Cloud Computing devices being able to exchange data such as text files as well as business information with the help of internet. Technically, it is completely distinct from an infrared. Using new models Iaas, Paas and Saas.

**[5] HAIL: A high-availability and integrity layer for cloud storage**
**From This Paper we Referred-**
In this paper to provide fault tolerance for cloud storage to stripe data across multiple cloud vendors. However, if a cloud suffers from a permanent failure and loses all its data, it is necessary to repair the lost data with the help of the other surviving clouds to preserve data redundancy. This paper presented a proxy-based storage system for fault-tolerant multiple-cloud storage called NCCloud, which achieves cost-effective repair for a permanent single-cloud failure.

## III. SCOPE OF RESEARCH

1. We design a novel homomorphic authenticator based on BLS signature, which can be generated by a couple of secret keys and verified publicly. Utilizing the linear subspace of the regenerating codes, the authenticators can be computed efficiently. Besides, it can be adapted for data owners equipped with low end computation devices (e.g. Tablet PC etc.) in which they only need to sign the native blocks.

2. To the best of our knowledge, our scheme is the first to allow privacy-preserving public auditing for regenerating code-based cloud storage. The coefficients are masked by a PRF(Pseudorandom Function) during the Setup phase to avoid leakage of the original data. This method is lightweight and does not introduce any computational overhead to the cloud servers or TPA.

3. Our scheme completely releases data owners from online burden for the regeneration of blocks and authenticators at faulty servers and it provides the privilege to a proxy for the reparation. Optimization measures are taken to improve the flexibility and efficiency of our auditing scheme; thus, the storage overhead of servers, the computational overhead of the data owner and communication overhead during the audit phase can be effectively reduced.

## IV. PROPOSED METHODOLOGY AND DISCUSSION

In this paper, we focus on the integrity verification problem in regenerating-code-based cloud storage, especially with the functional repair strategy. Similar studies have been performed by Bo Chen et al. and H. Chen el al. separately and independently. Extend the single-server CPOR scheme (private version in) to the regenerating- code-scenario; designed and implemented a data integrity protection (DIP) scheme for FMSR based cloud storage and the scheme is adapted to the thin-cloud setting1. However, both of them are designed for private audit, only the data owner is allowed to verify the integrity and repair the faulty servers. Considering the large size of the outsourced data and the user's constrained resource capability, the tasks of auditing and reparation in the cloud can be formidable and expensive for the users.

**Advantages of Proposed System**

1. Public Auditability: to allow TPA to verify the intactness of the data in the cloud on demand without introducing additional online burden to the data owner.
2. Storage Soundness: to ensure that the cloud server can never pass the auditing procedure except when it indeed manages the owner's data intact.
3. Privacy Preserving: to ensure that neither the auditor nor the proxy can derive users' data content from the auditing and reparation process.
4. Authenticator Regeneration: the authenticator of the re- paired blocks can be correctly regenerated in the absence of the data owner.
5. Error Location: to ensure that the wrong server can be quickly indicated when data corruption is detected.

DOI: 10.15680/IJIRCCE.2015. 0312131

## V. ARCHITECTURE

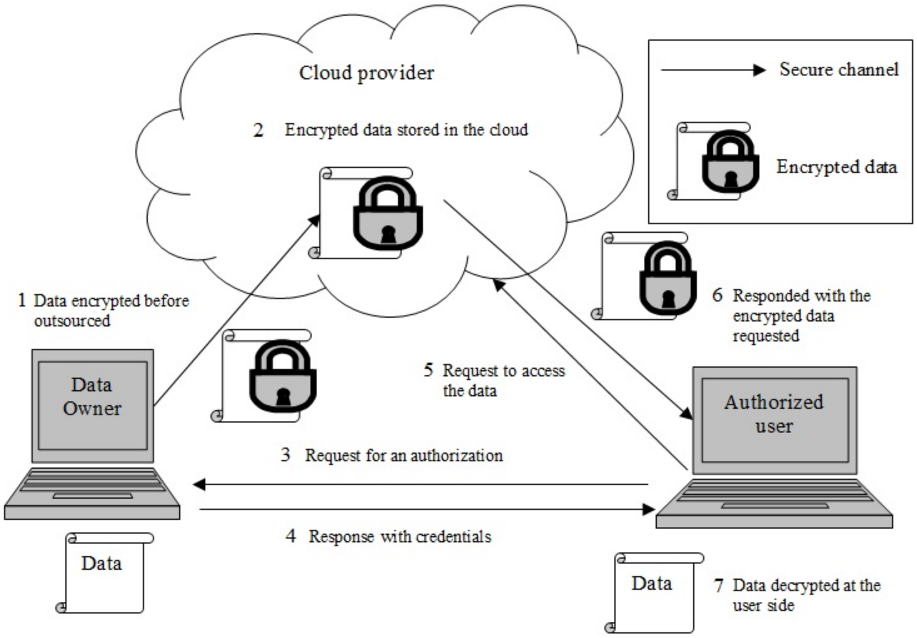
Cloud provider
2 Encrypted data stored in the cloud
Secure channel
Encrypted data
1 Data encrypted before outsourced
6 Responded with the encrypted data requested
5 Request to access the data
Data Owner
Authorized user
3 Request for an authorization
4 Response with credentials
Data
Data
7 Data decrypted at the user side

**Fig No 01. Attribute Based Encryption**

***Methodology Used:***

1. **Setup**: The data owner maintains this procedure to initialize the auditing scheme.

***KeyGen*(1*κ*) → (*pk*, *sk*):** This polynomial-time algorithm is run by the data owner to initialize its public and secret parameters by taking a security parameter *κ* as input.

***Degelation*(*sk*) → (*x*):** This algorithm represents the interaction between the data owner and proxy. The data owner delivers partial secret key *x* to the proxy through a secure approach.

***SigAndBlockGen*(*sk*, *F*):** This polynomial time algorithm is run by the data owner and takes the secret parameter *sk* and the original file *F* as input, and then outputs a coded block set, an authenticator set and a file tag *t*.

2. **Audit**: The cloud servers and TPA interact with one another to take a random sample on the blocks and check the data intactness in this procedure.

***Challenge*(*Finfo*) → (*C*):** This algorithm is performed by the TPA with the information of the file *Finfo* as input and a challenge *C* as output.

***ProofGen*→ (*P*):** This algorithm is run by each cloud server with input challenge *C*, coded block set and authenticator set, then it outputs a proof *P*.

***V erify*(*P*, *pk*, *C*) → (0, 1):** This algorithm is run by TPA immediately after a proof is received. Taking the proof *P*, public parameter *pk* and the corresponding challenge *C* as input, it outputs 1 if the verification passed and 0 otherwise.

3. **Repair:** In the absence of the data owner, the proxy interacts with the cloud servers during this procedure to repair the wrong server detected by the auditing process.

***ClaimForRep*(*Finfo*) → (*Cr*):** This algorithm is similar with the *Challenge*() algorithm in the Audit phase, but outputs a claim for repair *Cr*.

***GenForRep*→ (*BA*):**The cloud servers run this algorithm upon receiving the *Cr* and finally output the block and authenticators set *BA* with another two inputs.
***BlockAndSigReGen*(*Cr*,*BA*):** The proxy implements this algorithm with the claim *Cr* and responses *BA* from each server as input, and outputs a new coded block set and authenticator set if successful, outputting $\perp$ if otherwise.

## VI.CONCLUSION

In this paper, we analyse different attribute-based encryption schemes: ABE, KP-ABE, CP-ABE, ABE with non-monotonic access structure, HABE and MA-ABE .The main access polices are KP-ABE and CP-ABE, further schemes are obtained based on these policies. In this paper, we propose a privacy-preserving public auditing system for data storage security in Cloud Computing. We utilize the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly efficient.

## REFERENCES

[1] M. Armbrust et al., “Above the clouds: A Berkeley view of cloud computing,” Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, 2009.
[2] G. Ateniese et al., “Provable data possession at untrusted stores,” in Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2007, pp. 598–609.
[3] A. Juels and B. S. Kaliski, Jr., “PORs: Proofs of retrievability for large files,” in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 584–597.
[4] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, “MR-PDP: Multiple-replica provable data possession,” in Proc. 28th Int. Conf. Distrib. Comput. Syst. (ICDCS), Jun. 2008, pp. 411–420.
[5] K. D. Bowers, A. Juels, and A. Oprea, “HAIL: A high-availability and integrity layer for cloud storage,” in Proc. 16th ACM Conf. Comput. Commun. Secur., 2009, pp. 187–198.