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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Regenerating Code Based Secure Cloud Storage

Vivek Kr Anand, Prashant Tripath, Amit Waje, Vivek kumar

Student, Department of Computer Engineering, Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, Savitribai Phule Pune University, Pune, India

ABSTRACT: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. A public auditing scheme for the regenerating-code-based cloud storage is proposed. To solve the regeneration problem of failed authenticators in the absence of data owners, a proxy is introduced, which is privileged to regenerate the authenticators, into the traditional public auditing system model. Moreover, a novel public variable authenticator is designed, which is generated by a couple of keys and can be regenerated using partial keys. Thus, it completely releases data owners from online burden. In addition, the encode coefficients are randomized with a pseudo- random function to preserve data privacy. Extensive security analysis shows that it is proved secure under random oracle model and experimental evaluation indicates that it 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, privileged, provable secure.

I. INTRODUCTION

In this paper, we focus on the integrity verification problem in regenerating-code-based cloud storage, especially withthe functional repair strategy. Similar studies have been performed by Bo Chen et al. and H. Chen el alseparately and independently. Extended 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. The overhead of using cloud storage should be minimized as much as possible such that a user does not need to perform too many operations to their outsourced data(in additional to retrieving it) In particular, users may not want to go through the complexity in verifying and reparation. The auditing schemes in, imply the problem that users need to always stay online, which may impede its adoption inpractice, especially for long-term archival storage.

Background:
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 enterprisers still feel hesitant.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Motivation:

1. 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.
2. Existing remote checking methods for regenerating- coded data only provide private auditing, requiring data owners to al- ways stay online and handle auditing, as well as repairing, which is sometimes impractical. We propose a public auditing scheme for the regenerating-code-based cloud storage.
3. 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.

Objective and Goal:

1. The main objective of this project is to Authenticator Regeneration.
2. To achieve the Privacy-preserving.
3. Public auditability.
4. Storage correctness.
5. Error Location.

II. LITERATURE SURVEY

[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.

[5] HAIL: A high-availability and integrity layer for cloud storage
From This Paper, we Referred-
In this paper author 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-

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

tolerant multiple-cloud storage called NCCloud, which achieves cost-effective repair for a permanent single-cloud failure.

III. SOFTWARE REQUIREMENT SPECIFICATION

User Classes and Characteristics

To design products that satisfy their target users, a deeper understanding is needed of their user characteristics and product properties in development related to unexpected problems that the user's faces every now and then while developing a project. The study will lead to an interaction model that provides an overview of the interaction between user characters and the classes. It discovers both positive and negative patterns in text documents as higher level features and deploys them over low-level features (terms).

In proposed work is designed to implement above software requirement. To implement this design following software requirements are used. Operating system: Windows XP/7.

1. Coding Language : JAVA/J2EE
2. Database : MYSQL
3. Tool : Eclipse Luna

IV. IMPLEMENTATION STATUS

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.

V. COMPARISON BETWEEN EXISTING SYSTEM AND PROPOSED SYSTEM

Item	Existing System	Proposed System
Algorithms	1. AES Encryption Algorithms	2. SHA Algorithms 3. AES Encryption Algorithms
Accuracy	Low	High
Complexity	Low	High
Explanation	In the existing system the regeneration problem of failed authenticators in the absence of data owners, We only use data Content.Proxy Agent are not work in existing system,	The proposed system is to propose we propose a publicauditing scheme for the regenerating-code-based cloud storage.To solve the regeneration problem of failed authenticators in

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

	Disadvantages: 1. Required more time	the absence of data owners, we introduce a proxy, and Generate the Hash Value of the file,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 keysand can be regenerated using partial keys. Thus, our scheme cancompletely release data owners from online burden. In addition,we randomize the encode coefficients with a pseudorandom functionto preserve data privacy. Extensive security analysis showsthat our scheme is provable secure under random oracle modeland experimental evaluation indicates that our scheme is highly efficient and can be feasibly integrated into the regeneratingcode-based cloud storage.

VI. ALGORITHM FOR RELEVANT FEATURE DISCOVERY

Efficient Algorithms play important role in the relevant feature discovery from text document using text mining. The following steps explain the relevance feature of text documents:

1. Start.
2. Upload File, One copy save on Cloud Server and Another save on Authenticator
3. TPA Check the file is Hacked or not
5. If hack then replaced the Hash Value of the Original file
7. Stop

VII. SYSTEM ARCHITECTURE

System consists of two main modules which are further divided into sub modules as follows:

I] Cloud Server:

1. Dealing with receive data from Data Owner
2. Store the Data in encrypted form
3. Give the Data read permissions to authorized User
4. Accept and Replacement of Data through Proxy Agent

II] Cloud Client Use or Data Owner able to outsource their Data.

1. Encrypt the Data while Outsourcing of it.
2. Delegation between Data Owner and Proxy Agent
3. Generate a sk secrete key and Assign to the corresponding Authenticators present in PA

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

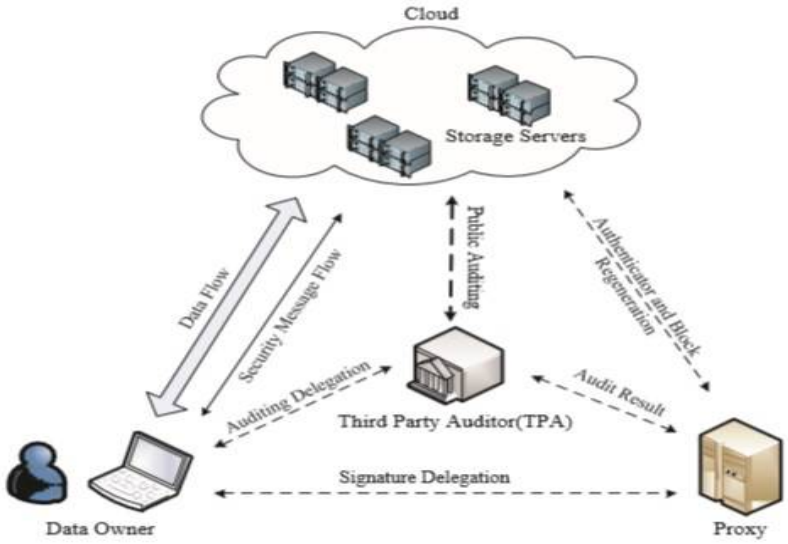

Figure1: System Architechture with TPA Services

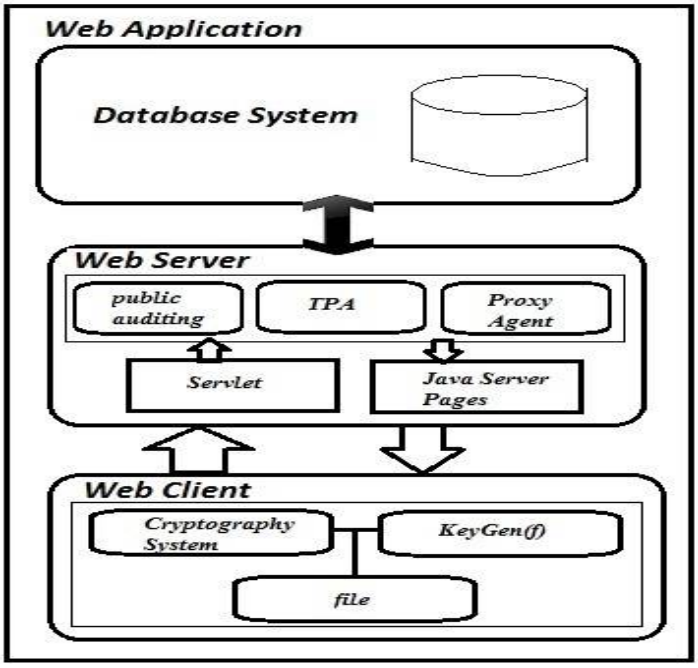

Figure2: System Architechture

Explanation

1. Data User able see data Stored on cloud Server and can make request to the Data or file

Third Party Auditor(TPA)

2. Examining the outsourced data and data owner Data to ensure the Data Integrity.
3. Public Auditing by checking h(.) code
4. Send Acknowledgment to the Proxy for decision making.

Proxy Agent(PA)

5. Accept the sk key from the Data Owner

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

6. Accept the Acknowledgment from TPA and perform Re-generation of code behalf of

Data Owner

7. Authenticator replaces or repair the block of data and solve integrity issue.

VIII. MATHEMATICAL MODULE

Let us consider,
System Description: Let S be the Whole System that consists of S = {P,T,I,C,F,X}.
1. Input: P
where P is any kind of outsourced data.
2. Let T be the process of encryption.
T={t1, t2, ...tm}
3. Let I be the process of Data decryption
C={c1, c2.....cn}
4. Let F be the case of Failure
F=Partial/Unsuccessful encryption and decryption constraints.
5. Let X be the case of Success
X= Successful encryption and decryption of data following the constraints. 1.7 Goals and Objectives
• Public auditability.
• Storage correctness.
• Privacy-preserving.
• Authenticator Regeneration.
• Error Location.

IX. EXPERIMENTAL SET UP AND RESULT

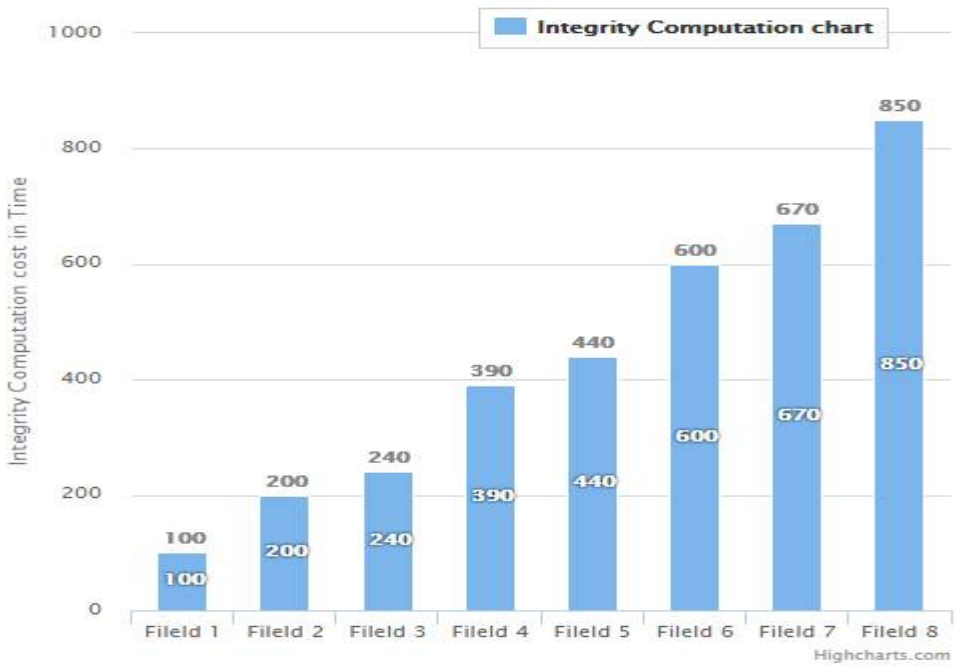

Figure 1: Integrity Computation Chart of Existing System

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

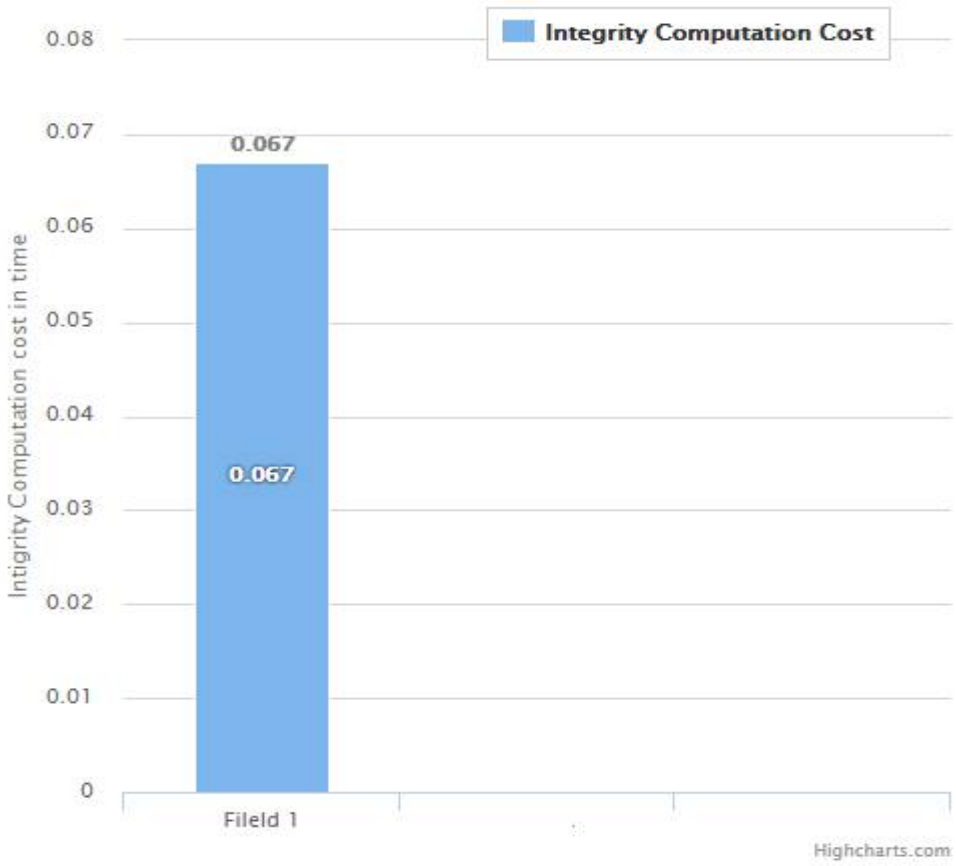

Figure 2:Integrity Computation Chart of Proposed System

Above diagram shows that the comparison of Integrity Computation of our proposed system with existing system. Figure 1 show thatintegrity computations chart of existing system and Figure 2 show that integrity computation chart of proposed system. X- Axis represents file id and Y-axis represents integrity computation cost in time. In the existing system integrity computation is vary according to files but with the help of our proposed system integrity computation is same for all files. The result shows that the performance of our system is best as compare to other state of art systems.

X. CONCLUSION

A privacy-preserving public auditing system for data storage security in Cloud Computing is proposed. The homomorphic linear authenticator and random masking are utilized 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.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

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.