SRCBT: Secure Regeneration of Corrupted Blocks by TPA in Cloud

Geeta C M*, Tejashwinishivaram K S*, Shreyas Raju R G*, Raghavendra S*, Rajkumar Buyya†, Venugopal K R‡ S S Iyengar§ L M Patnaik¶

*Department of Computer Science and Engineering, UVCE, Bangalore University, Bengaluru, India.
geetacmara@gmail.com, tejushivaram96@gmail.com, shreyasrajurg@gmail.com, raghush86@gmail.com
†CLOUDS Lab, School of Computing and Information Systems, The University of Melbourne, Australia.
raj@csse.unimelb.edu.au
‡Bangalore University, Bengaluru, India. venugopalkr@gmail.com
§Department of Computer Science and Engineering, Florida International University, USA.
iyengar@csc.lsu.edu
¶INSA, National Institute of Advanced Studies, Indian Institute of Science Campus, Bengaluru, India.
patnaiklm@yahoo.com

***Abstract*—To conserve the deployed information in cloud repository contrary to adulterations, including fault toleration to cloud repository together with information integrity verification and failure restoration becomes important. Belatedly, reconstructing codes acquire recognition because of their reduced reparation bandwidth while ensuring fault toleration. Prevailing distant auditing schemes for reconstructing-coded information authorizes an intermediary to reconstruct authenticators and information blocks on the suspended servers in the course of the reparation process. To address this issue, we propose Secure Regeneration of Corrupted Blocks by TPA in Cloud ($SRCBT$). We eliminate the semi-trusted proxy and allow the TPA to reconstruct authenticators and information chunks on the suspended servers in the course of the reparation process. Thus, our mechanism can totally relinquish information proprietors from online burden. The proposed scheme preserves privacy, TPA efficiently reconstructs authenticators and information chunks on the suspended servers in the course of the reparation process and also effectively performs batch auditing. The performance analysis shows that our mechanism is extremely effective and can be conceivably blended into the reconstructing-code-based distributed repository.**

***Index Terms*—Cloud repository, regenerating codes, public audit, privacy conserving, authenticator reconstruction, session-based auditing.**

I. Introduction

Cloud repository provides an on-demand information deploying utility framework, and is procuring recognition because of its adaptability and reasonable preservation cost. Anyhow, security concerns emerge when information repository is deployed to third-party cloud depository suppliers. Currently, optical networks [1] are utilized for effective data communication.

Assume that we deploy repository to a distributed server. If we identify manipulations in our deployed information, then we have to reconstruct the manipulated information and reconstruct the primary information. Reconstructing codes [2] have been introduced to reduce reparation traffic.

To completely guarantee the information sincerity and preserve the customers reckoning utilities, we present Secure Regeneration of Corrupted Blocks by TPA in Cloud ($SRCBT$) mechanism.

A. Motivation

The reconstructing codes have acquired recognition due to their reduced reparation bandwidth while ensuring fault toleration. In the prevailing scheme, the proxy is introduced that regenerates the corrupted blocks. In order to address this issue, we are motivated to eliminate the proxy, and allow the TPA to perform auditing and regeneration of corrupted blocks.

B. Contributions

In this paper, we present Secure Regeneration of Corrupted Blocks by TPA in Cloud ($SRCBT$) scheme. Our contributions are organized as follows:

(i) The dataowner empowers TPA to reconstruct authenticators and information chunks on the faulty servers.

(ii) TPA efficiently regenerates authenticators and information chunks on the faulty servers during the reparation process and saves in the temporary server which will not affect the permanent server.

C. Organisation

The list of the paper is compiled as follows: Related works are discussed in Section 2. In Section 3, existing frameworks and their limitations are presented and several preliminaries are discussed in Section 4. Problem statement and System framework demonstrates the operation of the framework and

978-1-7281-7366-5/20/ $31.00 ©2020 IEEE

furnishes the technicalities regarding the design goals, in Section 5. In Section 6, scheme details of Secure Regeneration of Corrupted Blocks by TPA in Cloud ($SRCBT$) has been discussed. Section 7, presents Security analysis while Performance evaluation results are analysed in Section 8. Conclusions are discussed in Section 9.

II. Related Works

Shen *et al.,* [3] outlined a distant information integrity verification mechanism that achieves information distribution with sensitive information hiding. The disadvantage of the scheme is that the computation cost of TPA in proof verification is more. Venugopal *et al.,* [4] illustrates soft computation techniques for data mining applications for depository.

Chen *et al.,* [5] designed a distant information verification procedure for network coding-based distributed repository frameworks. The advantage of the mechanism is that it has minimal transmission cost. The limitation is that the dataowner needs to stay online always. Chen *et al.,* [6] designed an information integrity preservation mechanism for reconstructing-code-based cloud storage. The scheme has low reparation traffic. The disdvantage is that the mechanism is not designed to handle public auditing. Geeta *et al.,* [7] have carried out comprehensive survey on the latest mechanisms in information verification and security in distributed computing.

Wang *et al.,* [8] designed indegenously minimal repository codes to repair different faulty nodes at the same period. The advantage of these codes are that they reduce the reparation bandwidth completely. The limitation is that the $LMSR$ codes possess the corresponding repository overhead as MSR codes. Neha *et al.,* [9] introduced network coding based collective fault tolerant mechanism. The mechanism has low repository cost and reparation bandwidth. The limitation is that the scheme need to be implemented for varying size-block of the document.

III. Background Work

Liu *et al.,* [2] designed a public verification mechanism for the reconstructing-code-based distributed repository. A proxy is introduced that regenerates the corrupted blocks and also again performs auditing, where the TPA has already carried out the auditing. The advantage of the scheme is that it efficiently regenerates the corrupted blocks and totally relieve information proprietor from online burden. The limitation is that the auditing is performed repeatedly by the TPA and the proxy, hence increasing the auditing time cost.

IV. Preliminary

The preliminaries forms the foundations of Secure Regeneration of Corrupted Blocks by TPA in Cloud ($SRCBT$) scheme and are discussed below.

A. Regenerating Codes:

Reconstructing codes [2] are introduced for cloud repository to minimize the reparation bandwidth. They have achieved recognition due to their reduced reparation bandwidth contrarily furnishing fault tolerance.

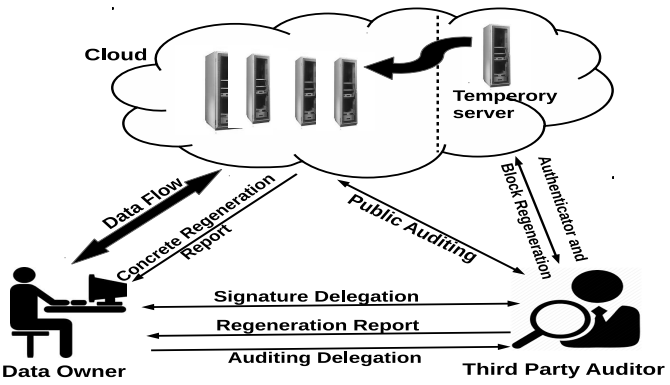

Fig. 1: Auditing System Model for Reconstructing-Code-based Cloud Repository

B. Bilinear Pairing Map:

$\mathcal{G}$ and $\mathcal{G}_T$ are two cyclic multiplicative clusters of large prime order p. A bilinear pairing is a map $e : \mathcal{G} * \mathcal{G} \rightarrow \mathcal{G}_T$ with the subsequent properties:

- Bilinear: $e(g_1^c, g_2^d)=e(g_1, g_2)^{cd}$ and c, d $\in_R Z_p$;
- Non-degenerate: There exists g_1, $g_2 \in \mathcal{G}$ such that $e(g_1, g_2) \neq 1$;
- Computability: An effective algorithm prevails to estimate $e(g_1, g_2)$ for all $(g_1, g_2) \in_R \mathcal{G}$.

V. Problem Definition and System Model

A. Problem Definition

Given the cloud storage auditing model based on reconstructing code based on cloud repository with Dataowner, TPA and CSP the main objectives are:

- The Dataowner empowers TPA to reconstruct authenticators and information chunks on the faulty servers.
- TPA efficiently regenerates authenticators and information chunks on the faulty servers during the reparation process and stores in the temporary server which will not affect the permanent server.

B. System Model

As demonstrated in Fig. 1., the system framework comprises of three objects: the Cloud Service Provider (CSP), the TPA and the Data Owner (DO). The DO constructs authenticator set and coded block set and deploys over n servers in cloud. The CSP provides information repository and distribution services to the customers. The Third Party Auditor (TPA) sends challenge based on the session given by the DO to the CSP (Session based auditing). Further, the TPA is empowered by the DO to reconstruct authenticators and information chunks on the faulty servers at the time of reparation process and stockpiles in the temporary server which will not affect the permanent server.

VI. The Algorithm

Considering the reconstructing-code-based cloud repository with parameters (n, k, l, α, β), and assume β = 1. Let $\mathcal{G}$ and $\mathcal{G}_{\mathrm{T}}$ be multiplicative cyclic clusters of the equivalent large prime order p, and $e : \mathcal{G} * \mathcal{G} \rightarrow \mathcal{G}_{\mathrm{T}}$ be a bilinear pairing map. Let g be a generator of $\mathcal{G}$ and $H(\cdot) : \{ 0, 1 \}^* \rightarrow \mathcal{G}$ be a secure hash function that maps strings consistently into cluster $\mathcal{G}$. Table I list the notations used in our mechanism description.

Our proposed mechanism Secure Regeneration of Corrupted Blocks by TPA in Cloud ($SRCBT$) comprises of two procedures: $Setup$ phase and $Repair$ phase.

A. Setup phase:

The DO executes this process to initialize the auditing mechanism. The DO executes the $KeyGeneration$ algorithm [2] and generates the secret parameter sk and public parameter pk. Further, the DO authorizes the TPA to audit and repair the corrupted blocks.

By utilizing the secret key (sk), the DO computes the $Re-key$ (τ_{Re-key}) using regression method [10]. (τ_{Re-key})=2[(Δ+4)/(2$\sqrt{sigX^2 sigY^2}$ + 4)]
where, Δ=$\delta(X^2)$+$\delta(Y^2)$+$\delta(Z^2)$
The DO delegates the $Re-key$ (τ_{Re-key}) to the TPA.

Next, by taking the sk and F as input the DO runs the $SigAndBlockGen$ algorithm. The DO outputs a coded chunk set (ψ_i), a validator set (ϕ_i) and a document label (t). Finally, the DO allocates these two sets over n cloud servers, particularly sending (ϕ_i, ψ_i, t) to server i and remove them from local repository. Next, The DO constructs TQ and PQ [See Algorithm 1, Phase 1].

B. Repair phase:

In this phase TPA maintains two sessions, session I for audit and session II for repair. When the DO sends audit request to the TPA, the TPA initiates session I and efficiently performs sector level batch auditing on the coded chunks cached in the distributed server. During the process of auditing, TPA might detect a server corruption. TPA makes a note of it, after completing the auditing process, TPA switches to session II ie., Repair phase. The procedure for Repair phase is explained in Algorithm 1, Phase 2.

VII. Security Analysis

Theorem 1: The proposed mechanism accomplishes confidentiality and integrity of the deployed data in the concrete regeneration phase.

Proof: In the proposed scheme, TPA securely and efficiently regenerates the corrupted blocks in the $Repair$ phase. TPA executes the $BlockAndSigReGen$ algorithm and outputs the regenerated block set ψ_i and authenticator set ϕ_i and further transmits (ψ_i, ϕ_i) to the TS (ζ'). Next, the regeneration report is sent to the DO and DO saves this report in the PQ. Now, the DO checks the PQ entries, and sends Regeneration Request ($RReq$) to the CSP. After accepting ($RReq$), the CSP securely transfers the regenerated blocks in the TS and saves in the PS and sends the concrete regeneration report

Algorithm 1: SRCBT: Secure Regeneration of Corrupted Blocks by TPA in Cloud

Input: $F1$ = (κ_1, κ_2,.....κ_n), $\kappa_{\mathrm{k}} \in \mathrm{Z_p}$, id_{k} where $k \in$ [1,n], sk

Output: ψ_i, ϕ_i, t, TQ and PQ

Phase 1: System Setup

Step 1: KeyGeneration

The DO creates an arbitrary signing key pair (spk, ssk), two random elements (x, y) $\xleftarrow{R}$ Z_p and estimates pk_x $\longleftarrow$ g^x, pk_y $\longleftarrow$ g^y. Then, secret key sk =(x, y, ssk) and public key pk = (pk_x, pk_y, spk).

Step 2: Delegation

By utilizing the secret key (sk), the DO computes the $Re-key$ (τ_{Re-key}) using regression method. (τ_{Re-key})=2[(Δ+4)/(2$\sqrt{sigX^2 sigY^2}$ + 4)] where, Δ=$\delta(X^2)$+$\delta(Y^2)$+$\delta(Z^2)$

The DO delegates the $Re-key$ (τ_{Re-key}) to the TPA.

Step 3: SigAndBlockGen

The DO takes sk, F as input, and then outputs ψ_i, ϕ_i and t.

Finally, the DO allocates these two sets over n cloud servers, particularly transmitting (ϕ_i, ψ_i, t) to server i and delete them from local repository.

The DO constructs TQ and PQ.

Phase 2: Repair Phase:

The DO empowers the TPA to repair the faulty server and goes off-line after completing the deploying process.

When TPA locates a server corruption, then it starts the repair procedure.

Step 1: BlockAndSigReGen Presuming that the restored coded chunks would be cached at a new TS (ζ'), the TPA reconstructs α chunks as follows:

The TPA chooses l arbitrary coefficients $z_i \leftarrow GF(p)$ for every $i \in i_\theta$ where 1$\leq \theta \leq$ l and then estimates a linear combination $c_{\eta' j}$= $\sum_{i \in i_o} z_i \bar{c}_i$ and reconstructs validators for every segment,

$\rho_{\eta jk}$= $\top^x{}_{jk}$. $\prod_{i \in i_o} (\bar{\rho}_{ik})^{z_i}$

where $1 \leq j \leq \alpha$, $1 \leq k \leq s$ and $\top^x{}_{jk}$ denotes the transform operator.

Eventually, the reconstructed chunk set ψ_i = $v_{\eta' j}$, 1$\leq j \leq \alpha$ and validator set ϕ_i = $\rho_{\eta' jk}$, 1$\leq k \leq \alpha$ are transmitted to TS (ζ').

Next, the regeneration report is sent to the DO and the DO saves this report in the PQ.

Thus, the repair process is terminated.

Step 2: PermanentReGen

The DO checks for the PQ entries. If it is not empty, then it requests the CSP to transfer the regenerated coded blocks from TS to PS.

Further, the CSP sends the success reports of regeneration to DO and the DO stores it in PQ. Next, this particular entry in the TQ is deleted after successful concrete regeneration process.

TABLE I: Summary of Notations

Notation	Description
n	The number of native information chunks.
d	The number of segment in a native information chunks.
b_{ik}	k^{th} segment of native chunk b_i
c_{ij}	The j^{th} coded chunk at server i
c_{ijk}	The k^{th} segment of coded block b_{ij}
t	File tag.
ϕ_i	The validator set for chunks in server i.
ψ_i	The coded chunk set for server i
$\mathcal{C}_r$	The claim for reconstruction.
$\mathbb{BA}$	The response from distributed server for reconstruction.
$\mathcal{PQ}$	Permanant queue
$\mathcal{TQ}$	Temperory queue
$\mathcal{PS}$	Permanant server
$\mathcal{TS}$	Temperory server
$\mathcal{CR}$	Concrete regeneration
$Sig()$	Standard signature scheme

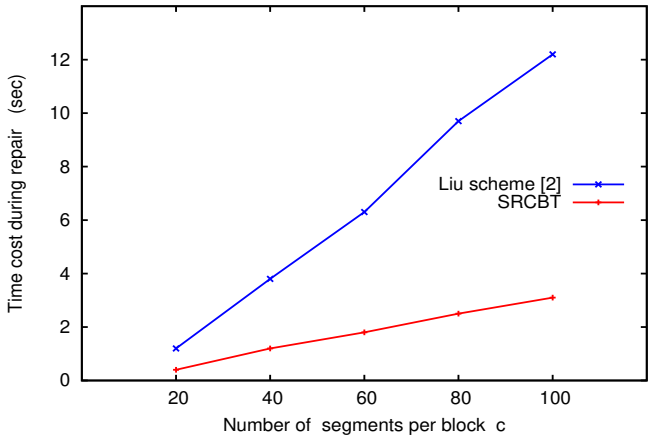

Fig. 2: Time for Repair considering k servers with different s

$(CRegen)$ to the DO and DO saves this report in the PQ. Thus, in the proposed scheme, it is not possible for the TPA to know the sensitive data deployed in the PS. Hence, the proposed scheme achieves confidentiality and integrity of the deployed data in the concrete regeneration phase.

VIII. PERFORMANCE ANALYSIS

In this section, we present an experimental evaluation of our proposed mechanism. All objects in our prototype are represented by PCs with Intel Core Intel(R) Core(TM) $i5$-6500U CPU @ 2.50GHz 2.59GHz and 8GB RAM Memory. The implementation of our algorithms utilizes open source Pairing-Based Cryptography (PBC) Library version 0.5.14, GMP version 5.13 and $Openssl$ version 1.0.1e. The security level is chosen to be 80 bits and thus |p| = 160. The choice of parameters (n, k, l, α, β) for reconstructing codes is in reference [5].

The time cost during repair considering k servers with different segments s is as shown in Fig. 2. In the Liu scheme [2], the repair phase includes verification for repair, authenticator regeneration and block regeneration. The proxy performs repeatedly auditing for the blocks to be repaired, and then performs authenticator regeneration and block regeneration. Hence, the time taken to repair by the proxy is more. In $SRCBT$ scheme, the repair phase performed by TPA includes only authenticator regeneration and block regeneration. Hence, the time taken in repair phase in $SRCBT$ is less compared to the Liu scheme [2].

IX. CONCLUSIONS

In the paper we propose Secure Regeneration of Corrupted Blocks by TPA in Cloud ($SRCBT$) mechanism. We have eliminated the semi-trusted proxy and allowed the TPA to reconstruct validators and information blocks on the faulty servers at the time of reparation phase. Thus, our mechanism totally releases the information proprietor from online burden. The proposed scheme achieves confidentiality and integrity of the deployed data in the concrete regeneration phase. TPA efficiently regenerates validators and information blocks on the faulty servers at the time of reparation phase and also effectively performs batch auditing. The security analysis of our mechanism demonstrates that our mechanism can accomplish the desired security goals. Performance analysis shows that our mechanism is effective and can be potentially integrated into the reconstructing-code-based cloud repository.

REFERENCES

[1] K. R. Venugopal, E. E. Rajan, and P. S. Kumar, "Impact of Wavelength Converters in Wavelength Routed All-Optical Networks," *Computer Communications*, vol. 22, no. 3, pp. 244–257, 1999.

[2] J. Liu, K. Huang, H. Rong, H. Wang, and M. Xian, "Privacy-Preserving Public Auditing for Regenerating-Code-Based Cloud Storage," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1513–1528, 2015.

[3] W. Shen, J. Qin, J. Yu, R. Hao, and J. Hu, "Enabling Identity-Based Integrity Auditing and Data Sharing with Sensitive Information Hiding for Secure Cloud Storage," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 331–346, 2019.

[4] K. R. Venugopal, K. G. Srinivasa, and L. M. Patnaik, "Soft Computing for Data Mining Applications," Springer, 2009.

[5] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote Data Checking for Network Coding-based Distributed Storage Systems," pp. 31–42, 2010.

[6] H. C. H. Chen and P. P. C. Lee, "Enabling Data Integrity Protection in Regenerating-Coding-based Cloud Storage: Theory and Implementation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 407–416, 2013.

[7] C. M. Geeta, S. Raghavendra, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Data Auditing and Security in Cloud Computing: Issues, Challenges and Future Directions," *International Journal of Computer (IJC)*, vol. 28, no. 1, pp. 8–57, 2018.

[8] J. Wang, W. Luo, W. Liang, X. Liu, and X. Dong, "Locally Minimum Storage Regenerating Codes in Distributed Cloud Storage Systems," *China Communications*, vol. 14, no. 11, pp. 82–91, 2017.

[9] N. Arya, R. R. Rout, and G. Lingam, "Network Coding Based Multiple Fault Tolerance Scheme in P2P Cloud Storage System," pp. 223–228, 2018.

[10] C. M. Geeta, S. Raghavendra, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "CRUPA: Collusion Resistant User Revocable Public Auditing of Shared Data in Cloud," *Accepted for publication in Journal of Cloud Computing (JoCC) Springer, 2020.*