A Regenerating Cloud Storage In Privacy Authantication System

Manigandan.S.K[1] Vaibalan.S[2]
[1]Assistant Professor [2]P.G Scholar
[1,2]Department of Master of Computer Application
[1,2]Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Avadi, Chennai-62

Abstract— Cloud storage enable users to remotely store up their data and enjoy the on-demand high value cloud applications the local hardware and software management. The benefits are plain, such a service is also relinquish users physical possess of their outsourced data, which certainly pose new security risks towards the rightness of the data in cloud The scheme has chops encode text security in the random oracle model presumptuous a variant of the computational Die-Hellman trouble. Our system is based on bilinear map between group Existing remote inspection methods for regenerating-coded data only give personal auditing, require data owners to always wait online and grip auditing, as well as repair, which is from time to time impractical. We suggest a public auditing scheme for the regenerating-code-based cloud storage. allowing for the cloud data are dynamic in nature, the future design advance supports secure and resourceful dynamic operations on out sourced data, as well as lump alteration, deletion, and append Extensive security analysis shows that our scheme is verifiable secure under random oracle mode land trial evaluation indicates that our scheme is highly competent and can be feasibly incorporated into the regenerating code-based cloud storage.

Key words: Cloud storage, Regenerating codes, Public audit, Personal preserving, Authenticator regeneration, Proxy, Advantaged, Provable secure

I. INTRODUCTION

The cloud storage is now fast status because it offers a supple on-demand data outsourcing service with attractive benefits: that relief of the storage management, worldwide data access with location independence, and evading of capital expenditures on hardware and software maintenance etc., [1]. however, this new pattern of data hosting service also bring new security threats toward users data, this making persons or enterprisers still feel hesitant. Provider may act dishonestly; attempt to hide data loss or corruption and claiming that the files are still properly stored in the cloud for repute or fiscal reasons. Thus it makes great sense for users to apply an efficient protocol to perform journal verifications of their outsourced data tonsure that the cloud indeed maintains their data properly. Many mechanisms dealing with integrity of outsourced data without a local copy have been future under different system and safety model up to nowhere is no need for Alice to get hold of Bob's public key corticated. When Bob receives the encrypted mail associates a third revelry, which we call the Private Key Generator (PKG). Bob authenticates himself to the PKG in the same way would authenticate himself to a CA and obtain his personal key from the PKG. though, both of them are intended for private audit, only the data owner is permitted to verify the honesty and repair the faulty servers. Bearing in mind the big size of the outsource data and the user's unnatural resource capacity, the tasks of auditing and reparation in clouds can be formidable and luxurious for the users [14]. The overhead of using cloud storage should be minimize such as possible such that a user does not need to do too many operation to the outsourced data(in supplementary to retrieve) [15]. In demanding users may not want to go through the difficulty in verifying and recompense. The auditing schemes in [7], [8] entail the complexity that users need to always stay online, this is may hinder its adoption in preparation, especially for long-term archival storage. Moving data into the cloud offers great convenience to users since don't have to care relating to the complexity of direct hardware management. The Cloud Computing venders, Amazon Simple Storage Service (S3) and Amazon Adaptable Compute Cloud (EC2) [2] are both well known examples. While the so internet-based online services do supply huge amount of storage space and customize computing properties, this computing platform shift, however, is eliminate the responsibility of local machines for data maintenance at the same time since users may not retain a limited copy of outsourced data, there exist different incentive for cloud service providers (CSP) to behave falsely towards the cloud users regarding the status of their outsourced data. For instance to add to the earnings margin by reducing cost, it is possible for CSP to throw away rarely accessed data without being detected in a timely style [9]. Similarly, CSP may even attempt to leather data loss instances so as to maintain a reputation [10]–[12]. Consequently, although outsourcing data into the cloud is carefully attractive for the cost and complexity of long-standing large-scale data storage, its missing of donation sturdy assurance of data integrity and availability may obstruct its wide adoption by both enterprise and person cloud users. Besides, it can be adapted for data owners ready with low end totaling devices (e.g. Tablet PC etc.) In THIS only need tossing the native blocks. To the best of our information, our scheme is the first to allow the privacy-preserving public auditing for regenerating code based cloud storage. The coefficient is masked by a PRF (Pseudo random Function) during the Situation phase to avoid escape of the original data. This method is trivial and does not introduce any commutate overhead of the cloud servers or TPA. Our scheme completely release data owners from online burden for the renaissance of blocks and authenticator sat defective servers and it provides the freedom to a proxy for the reparation our scheme is provable secure under random oracle model against adversary illustrated in Section II -C. Furthermore, we make an appraisal with the state of the art and experiment evaluates the presentation of our scheme. The rest of this broadsheet is prearranged as follows: Section II bring in some preliminary, the system copy, threat model, design goals and formal definition of our auditing scheme. Then we provide the complete description of our scheme in Segment III; Segment IV analyzes its security and Section Evaluates its presentation. Section VI presents a review of the related occupation on the auditing schemes in cloud storage.

II. PRELIMINARIES AND PROBLEM STATEMENT

In cloud data storage, a user stores his data through a CSP into a set of cloud servers, which are running in as simultaneous, cooperated and distributed manner. Data idleness can be employed with technique of erasure correcting code to additional tolerate fault or server crashes user's data grows in size and significance. After that, for application purposes, the user interacts with the cloud servers via CSP to access or get back his data. In some cases, the user may need to carry out block level operations on his data. The most universal forms of this operation we are allowing for are block update, delete, insert and append. Note that, wept more center on the support of file-oriented cloud application extra than non-file application data, such as common networking data. In other words, the cloud data we are manner in mind is not expected to be rapidly changing in a relative short period analyze the original exchange between the storage cost α' and the repair bandwidth′, then presented two extreme and almost pertinent points on the optimal tradeoff curve: the minimum band width regenerating(MBR) point, which represent the working point with the least likely repair band width, and the minimum storage regenerating(MSR) point, which communicate to the least likely storage cost on the servers. Denote by the parameter tipple (n, k, ℓ, α', γ'), we obtain

$$(\alpha'_{MSR}, \gamma'_{MSR}) = \left(\frac{|F|}{k}, \frac{|F|\ell}{k(\ell - k + 1)}\right) \quad (1)$$

$$(\alpha'_{MBR}, \gamma'_{MBR}) = \left(\frac{2|F|\ell}{2k\ell - k^2 + k}, \frac{2|F|\ell}{2k\ell - k^2 + k}\right) \quad (2)$$

As one basis of our work, the practical repair regenerating codes are non-systematic and do not perform as well for read operation as methodical codes but they actually make sense for the scenario in which data repair occurs much more often than read, such as narrow storage, data escrow and long-term archival storage [7]. As users no longer acquire here data locally, it is of critical significance to ensure users that their data are being properly stored and maintained. That is, users should be ready with security means so that they can make continuous rightness assurance (to enforce cloud storage service-level conformity) of their stored data even without the existence of local copies. In case that users do not essentially have the time, possibility or non fusibility Resources to monitor their data online, they can delegate the data auditing tasks to optional trusted TPA of their respective choices. However, to securely bring in such a TPA, any possible leakage of user's out sourced data towards TPA through the audit protocol should be forbidden. In our model, we take for granted that the point-to-point communication channels between each cloud server and the user is authenticated and dependable, which can be achieved in practice with little slide. These verification handshakes are omitted in the following presentation. When the irregular auditing process detects data corruption at one server, the repair process will contact ℓ servers and obtain β blocks from each to renew the tainted blocks. Fig.1 shows an instance of functional repair regenerating code. The strategy for choosing limitation tipple (n, k, ℓ, α, β) can be found in [7], [18].

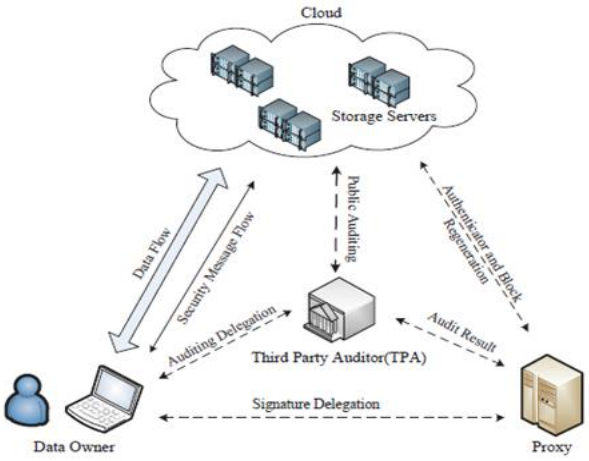

Fig. 1:

Compared with the traditional public audit system model, our system model involves an supplementary proxy agent. In order to reveal the level-headedness of our design and make our following account in Section III to be more clear and concrete, we consider such a reference scenario: A company employs a gainful regenerating-code-based public cloud and delivers long-term archival storage service for its staff, the staff are prepared with low end calculation devices (e.g., Laptop PC, Tablet PC, etc.) and will be regularly off-line. For public data auditing, the business relies on a trust third party association to check the data honesty; Similarly, to release the staffs from serious online burden for data and authenticator regeneration, the business supply a influential office (or cluster) as the proxy and provide proxy reparation service forth staff' data.

III. APPLICATIONS USE FOR IDENTITY-BASED ENCRYPTION

The original incentive for identity-based encryption is to help the operation of a public key transportation. In this piece, we show several other unconnected applications. Public key corticated contain a preset expiration date. In an IBE system key ending can be done by having Alice encrypt email sent to Bob using the public key Hence, we get the of annual private key expiration. Note that unlike the obtainable PKI, Alice does not need to obtain a new corticated from Bob every time Bob refresh his private key. In terms of compromised servers, we adopt a mobile adversary under the multi servers.

Algorithm 1: Setup phase

begin
  Choose parameters c, l, k, L and functions f, g;
  Choose the number t of tokens;
  Choose the number r of indices per verification;
  Generate randomly master keys $W, Z, K \in \{0,1\}^k$.
  for $(i \leftarrow 1$ **to** $t)$ **do**
    begin *Round i*
1   Generate $k_i = f_W(i)$ and $c_i = f_Z(i)$
2   Compute $v_i = H(c_i, D[g_{k_i}(1)], \ldots, D[g_{k_i}(r)])$
3   Compute $v'_i = AE_K(i, v_i)$
    end
  Send to $\mathcal{SRV}$: $(D, \{[i, v'_i]$ for $1 \leq i \leq t\})$
end

Fig. 2:

Setting, similar with [5], who can compromise almost n−k out of the n servers in any epoch, subject to the

(n, k)- MDS fault lenience condition. To avoid creeping-dishonesty which may lead to the unrecoverable of the stored data, the repair modus operandi will be triggered at the end of each epoch once some corruption is detected. There is some difference in our model compared with the one in [5]: First, the adversary can corrupt not only the data blocks but also the coding coefficients stored in the compromised servers; and second, the compromise server may act honestly for auditing but maliciously for reparation. We assume that some blocks stored in server Si are tainted at some time, the adversary may launch the following attack in order to prevent the auditor from detect the corruption in the worst case scenario, and the opponent can cooperation all the storage servers so that he can purposely modify the data files as long as they are inside consistent. In fact, this is equal to internal attack case where all servers are unspoken colluding together from the early stages of application or service deployment To perform the it proof of possession confirmation, OWN begins by re-generating the it token key kin assign step 1 of Algorithm 1. Note that OWN only wants to amass the master keys W; Z; and K, plus the current .For the proof to hold, the return integrity check must match the corresponding value precompiled by OWN. Though, in our scheme OWN has the choice of either keeping the pre-computed token locally or out sourcing them – in encrypted form –to SRV. Markedly, in the latter case, OWN's storage overhead is constant despite of the size of the out sourced data. Our scheme is also very proficient in provisos of computation and band width. Token index I. It also re-computes ice as above. Hide a data loss or bribery incident.

SOME PRIMARY NOTATIONS IN OUR SCHEME DESCRIPTION

Notation	Description
m	the number of native data blocks
s	the number of segment in a native data block
$\overline{w}_{ik}$	the kth segment of native block w_i
v_{ij}	the jth coded block at server i
v_{ijk}	the kth segment of coded block v_{ij}
$\varepsilon_{ij\lambda}$	the λth coefficient for coded block v_{ij}
t	file tag which contains a identifier ID and random symbols $u, w_1, w_2, ..., w_m$
σ_{ijk}	the authenticator for segment v_{ijk}
Φ_i	the authenticator set for blocks in server i
Ψ_i	the coded block set for server i
$\mathcal{C}$	the challenge for audit which contains an index-coefficient pair set Q_i and a random symbol set Λ_i for server i
$\mathcal{P}$	the proof for audit which contains: μ_i-the aggregated segment, σ_i-the aggregated authenticator and $\{\rho_{i1}, \rho_{i2}, ..., \rho_{im}\}$-the auxiliary values for coefficients checking
$\mathcal{C}_r$	the claim for reparation which contains a random coefficient set Λ_i for server i
BA	the response from cloud server for reparation which contains combined block $\widetilde{v}_i$,and s aggregated authenticators $\{\widetilde{\sigma}_{ik}\}_{1\leq k\leq s}$

Fig. 3:

IV. DELEGATION OF DECRYPTION

Another application for IBE systems is allocation of decryption capability. We give two instance applications. In both applications the user Bob plays the task of the PKG. Bob runs the net algorithm to generate his possess IBE system parameters prams and his own master-key. Here we view prams as Bob's public key. Bob obtains a corticated from a CA for his public key prams. When Alice wishes to send mail to Bob she obtains Bob's public key prams from Bob's public key corticated. Note that Bob is the simply one who know his master-key and therefore there is no key-escrow with this setup. To make our contribution easier to follow, we for a short time bring in our above scheme under the orientation scenario in Section II-B: The staffs (i.e., cloud users) first generate their public and private keys and then delegate the authenticator regeneration to a proxy (a cluster or powerful workplace providing by the business) by sharing partial private key. After produce agreed blocks and authenticators, the staff uploads and hand out to them to the cloud servers. Since that the staffs will be regularly off-line, the company employs a trust third party (the TPA) to interact with the cloud and do periodical verification on the staffs 'data blocks in an example mode. Once some data dishonesty is detected, the proxy is informed, it will act on behalf of the staffs to regenerate the data blocks as well as matching authenticators in a secure approach.

$$v_{\eta' j} = \sum_{i\in\{i_\theta\}} z_i \widetilde{v}_i$$

and regenerate authenticators for each segment,

$$\sigma_{\eta' jk} = \mathcal{T}_{jk}^{x} \cdot \prod_{i\in\{i_\theta\}} (\widetilde{\sigma}_{ik})^{z_i}$$

where $1 \leq j \leq \alpha, 1 \leq k \leq s$ and the transform operator denotes

$$\mathcal{T}_{jk} = \frac{H(ID||\eta'||j||k)}{\prod_{i\in\{i_\theta\}} \prod_{j^*=1}^{\alpha} H(ID||i||j^*||k)^{a_{j^*} z_i}}$$

Suppose Alice encrypts mail to Bob with the theme line as the IBE encryption key. Bob can decrypt mail use his master-key. Now, suppose Bob has several assistants each responsible for a divergent ta SK (e.g. one is `purchasing', a new is `human-resources', etc.). Bob gives one private key to each of his assistant's equivalent to the assistant's responsibility. Each as instant can then decrypt messages whose subject line falls within its farm duties, but it cannot decrypt letters intended for other assistant.

Algorithm 2: Verification phase

begin *Challenge i*

1. $\mathcal{OWN}$ computes $k_i = f_W(i)$ and $c_i = f_Z(i)$
2. $\mathcal{OWN}$ sends $\{k_i, c_i\}$ to $\mathcal{SRV}$
3. $\mathcal{SRV}$ computes $z = H(c_i, D[g_{k_i}(1)], \ldots, D[g_{k_i}(r)])$
4. $\mathcal{SRV}$ sends $\{z, v_i'\}$ to $\mathcal{OWN}$
5. $\mathcal{OWN}$ extracts v from v_i'. If decryption fails or $v \neq (i, z)$ then REJECT.

end

We point out that there is roughly no cost for OWN to do a corroboration. It only needs to re-generate the appropriate [ki; ci] pair (two PRF-s in vocations) and carry out one decryption in order to check the reply from SRV. Furthermore, the bandwidth consumed by the corroboration phase is steady (in both step 2 and 4 of Algorithm 2). This represents truly smallest overhead. The totaling cost for SRV, though slightly senior (r PRP-s on short inputs, and one hash), is still very practical.

V. AUDITING SYSTEM

Our auditing scheme consists of three events: Setup, Audit and mend. Each procedure contains positive polymorphism time algorithms as follows: Setup: The data owner maintains this process to initialize the auditing scheme. This segment presents our public auditing system which provides a complete out source solution of data– not only the data itself, but also its integrity examination. We start from an indication of our public auditing system and discuss two directly forward schemes and their demerits. Then we at hand our main scheme and show how to degree our main scheme to support group auditing for the TPA upon delegation from multiple users. Finally, we discuss how to simplify our privacy-preserving public auditing system and its hold of data active. The cloud servers and TPA interact with one one more to take a random sample on the blocks and check the data intactness in this procedure. Challenge (Info) → (C): This algorithm is performed byte TPA with the information of the file Info as input and challenge C as output. Proof Gen(C,_,) → (P): This algorithm is run by each cloud server with input challenge C, coded block set and authenticator set _, then it outputs a proof Pour framework assumes the TPA is state less, which is a attractive property achieved by our planned solution. It is easy to extend the structure above to capture a disgraceful auditing system, essentially by splitting the confirmation metadata into two parts which are stored by the TPA and the cloud server in that order. Our design does not assume any additional goods on the data file. If the user needs to have extra error-resiliency, he/she can always first redundantly program the data file and then uses our structure with the data file that has error-correcting codes integrated. In erasure coding, given a file F of k blocks, the client uses an(n; k) utmost distance divisible erasure code to create n oblique blocks out of the original k file blocks, and stores them at n servers (One coded block per server). Thus, the innovative file can be recovered from any k out of the n servers. When on earth the client detects bribery of one of the coded blocks, it can use the residual healthy blocks to regenerate the dishonored coded block. The storage Cost is jFj n k across all servers (jfj k per server). This is optimal in terms of redundancy-reliability storage tradeoff1. We define the HAIL system to available if the experiment from Figure 2 output 0; otherwise we say that the HAIL system is engaged. HAIL becomes engaged if the file cannot be improved either when a reorder is called or at the end of the research. In this section, we give limits for HAIL availability and show how to choose parameter in HAIL for given accessibility targets. There are several factors that donate to HAIL ease of use.

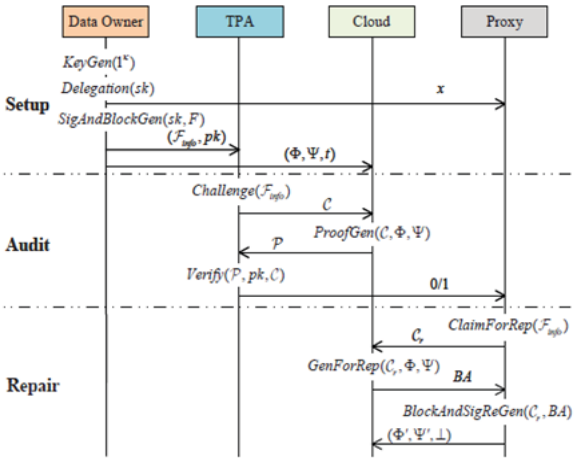

Fig. 5: Block And Sig ReGen (Cr,BA) → (_′, ′,⊥)

The proxy implements this algorithm with the claim Cr and responses from each server as input, and outputs a new coded block set ′ and authenticator set _′ if successful, outputting ⊥ if otherwise.

VI. THE PROPOSED SYSTEM

In this section we start from an overview of our auditing scheme, and then describe the main scheme and discuss how-to generalize our privacy-preserving public auditing scheme. Furthermore, we illustrate some optimized methods to improve its performance's – outsourced data. We assume that D can be represent as a single adjacent file of d equalized Blocks: D[1]; : : : ;D[d]. The actual bit-length of a block is not germane to the scheme. Although [7], [8] introduced private remote data checking schemes for regenerating-code-based cloud storeroom; there is still some other challenge for us to intend a public auditable version. First, although a direct extension of the techniques in [2],[12], [15] can realize public verifiability in the multi-servers setting by presentation each block as a set of segments and the theater spot checking on them, such a simple method makes the data owner generate tags for all segment independently, thus resulting in high computational overhead. Considering that data owners commonly maintain limited calculation and recall capacity, it is quite significant fours to reduce those expenses. Second, unlike cloud storage based on customary erasure code or replication, a fixed file layout does not exist in the regenerating-code-based cloud storage. Through the repair phase, it compute out new blocks, which are totally different from the faulty ones, with high probability. We first give an overview of our provable data possession scheme that supports sampling. In the Setup phase, the client computes a homomorphism verifiable tag (Timmy, Wi) for each block mi of the file. In order to uphold constant storage, the client generate the random values Wily concatenating the index i to a secret value v; thus, Tag Block has an extra parameter, i.

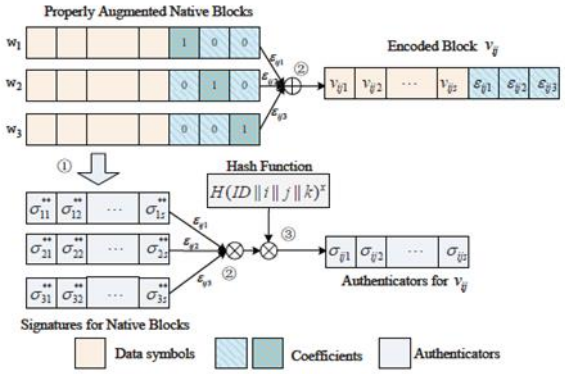

Fig. 6:

G and GT be multiplicative cyclic groups of the same large prime order p, and e :G × G → GT be a bilinear pairing map as introduced in the preliminaries. Let g be a generator of G and H(•) : {0, 1}∗ → G be a secure hash function that maps strings uniformly into group G. Table I list the primary notations and terminologies used in our scheme description. To make our contribution easier to follow, we briefly introduce our above scheme under the reference scenario in Section II-B: The staffs (i.e., cloud users) first generate their public and private keys, and then delegate the authenticator regeneration to a proxy(a cluster or powerful workstation provided by the company) by sharing partial private key.

VII. Related Work

The problem of remote data checking for integrity was first introduced in [26], [27]. Then Attendees et al. [2] and Jules et al. [3] gave rise to the similar notions provable data possession (PDP) and proof of irretrievability (POR), respectively. Attendees et al. [2] proposed a formal definition of the PDP model for ensuring possession of files on unfrosted storage, introduced the concept of RSA-based homomorphism tags and suggested randomly sampling a few blocks of the file. Similarly, Saba et al. [28] give a protocol for remote file integrity checking, based on the Diffuse-Hellman problem in composite-order groups. However, the server must access the complete file and in adding the client is forced to store several bits per file block, so storage at the client is linear with respect to the figure of file blocks as opposed to constant. The file is encoded in a single pass to diminish the cost of disk access. Since we use Reed-Solomon codes for implementing the dispersal and server code, calculation of parity blocks involves as associative operations in a Galois field. For an incremental encoding of the file (i.e., in a lone pass), we store the equality blocks into main memory, and update them when we process a file chunk in memory.

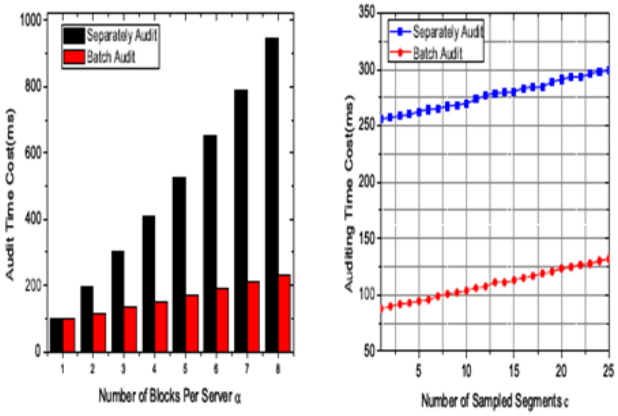

Fig. 7:

Presented by Sachem and Waters [12] with full proofs of security in the security model defined in [3]. They utilize the publicly verifiable homomorphism linear authenticator built from BLS signatures to achieve public auditing. However, their come near is not privacy preserving due to the same reason as [2].

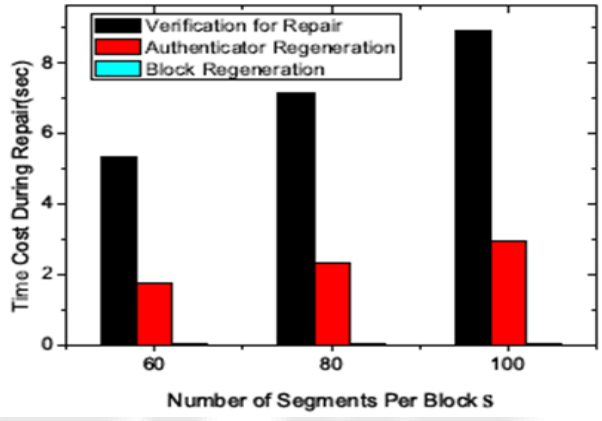

Fig. 8:

On the right graph in Figure 5 we keep the total number of servers constant at 20 and vary the number of primary servers between6 and 12.To further compare the two schemes we assess the irrespective tamper detection probabilities. In [8], the outsourced data is composed of d blocks and there are sentinels. We compute the probability that, m being the number of corrupted blocks, the system consumes all sentinels and bribery is undetected. Furthermore, Zhu et al. [10] proposed an efficient construction of cooperative provable data possession (CPDP) which can bus in multi-clouds, and [9] extend their primitive auditing protocol to sustain batch auditing for both multiple owners and numerous clouds. Portions of the work obtainable in this paper have before appeared as an extended abstract in [1]. Weave revised the piece a lot and add more scientific particulars as compared to [1]. The primary improvement is as follows: Firstly, we provide the protocol addition for privacy-preserving third-party auditing, and argue the application scenario for cloud storage service.

VIII. Conclusion

A public auditing scheme for their generating-code-based cloud storage system, where the data owners are privileged to hand over TPA for their data validity inspection. To protect the original data privacy against the TPA, we randomize the coefficients in the opening rather than apply

the blind technique during the auditing process. To summarize, the work describe in this paper represent an imperative step forward towards practical PDP technique. We expect that the relevant skin tone of our system (very low cost and support for dynamic outsourced data) make it attractive for realistic applications. Consequently, a challenger that observes adjustment of a file block on one server can, when poignant to other servers, simply position and attack blocks belong to the same dispersal password. Such an adversary can shady a file blocks in a surgically accurate, making such dishonesty hard to detect. Extensive analysis shows that our system is verifiable secure, and the performance evaluation shows that our scheme is highly efficient and cane feasibly integrated into a regenerating-code-based cloud storage system.

REFERENCES

[1] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Kaminski, G. Lend. Patterson, A. Rankin, and I. Stoical, "Above the clouds: A Berkeley view of cloud computing," Dept. Electrical Eng. and Compute. Sciences, University of California, Berkeley, Rep. UCB/EECS, vol. 28, p. 13, 2009.

[2] G. Attendees, R. Burns, R. Carmela, J. Herring, L. Kisser, Z. Peterson, and D. Song, "Provable data possession at entrusted stores," in Proceedings of the 14th ACM Conference on Computer and Communications Security, ser. CCS '07. New York, NY, USA: ACM, 2007, pp. 598–609.

[3] A. Jules and B. S. Kaminski Jr, "Pores: Proofs of irretrievability for large files," in Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007, pp. 584–597.

[4] R. Carmela, O. Khan, R. Burns, and G. Attendees, "Mr-pdp: Multiplereplicaprovable data possession," in Distributed Computing Systems,2008. ICDCS'08. The 28th International Conference on. IEEE, 2008,pp. 411–420.

[5] K. D. Bowers, A. Jules, and A. Opera, "Hail: a high-availability and integrity layer for cloud storage," in Proceedings of the 16th ACMconference on Computer and communications security. ACM, 2009,pp. 187–198.

[6] P. Galle, J. Stardom, and B. Waters, "Secure conjunctive keyword search over encrypted data," in ACNS'04,2004.

[7] G. Attendees, R. Burns, R. Carmela, J. Herring, L. Kisser. Peterson, and D. Song, "Provable data possession at entrusted stores," in ACM CCS'07, Full paper available on e-print (2007/202), 2007.

[8] A. Jules and B. Kalikow, "PORs: Proofs of irretrievability for large files," in ACM CCS'07, Full paper available on e-print (2007/243), 2007.

[9] John F. Gants, David Riesel, Christopher Chute, Wolfgang Schlichting, John McArthur, Stephen Minton, Iliad Hemet, Anna To cheval, and Alex Manfrediz,"The expanding digital universe: A forecast of worldwide information growth through 2010.IDC white paper—sponsored by EMC," Tech. Rep. March 2007, http://www.emc.com/about/Destination/digital universe/.

[10] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong, "Freeing: a distributed anonymous information storage and retrieval system," in International workshop on Designing privacy enhancing technologies, New York, NY, USA, 2001, pp. 46–66,Springer-Verlag New York, Inc.

[11] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in Proc. of HotOS'07. Berkeley, CA, USA: USENIX Association, 2007, pp. 1–6.

[12] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology reprint Archive, Report 2008/186, 2008, http://eprint.iacr.org/.

[13] G. Attendees, R. D. Petro, L. V. Mancini, and G. Studio, "Scalable and efficient provable data possession," in Proc. of SecureComm'08, 2008, pp. 1–10.

[14] Q. Wang, C. Wang, J. Li, K. Rend, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing, "in Proc. of ESORICS'09, volume 5789 of LNCS. Springer-Vela, Sep. 2009, pp. 355–370.

[15] C. Elway, A. Kusch, C. Papamanthou, and R. Tamassia, "Dynamic provable data ownership," in Proc. of CCS'09, 2009, pp. 213–222.

[16] H. Sachem and B. Waters, "Compact proofs of irretrievability," in Proc. of Asiacrypt'08, volume 5350 of LNCS, 2008, pp. 90–107.

[17] K. D. Bowers, A. Jules, and A. Opera, "Proofs of irretrievability: Theory and implementation," in Proc. of ACM workshop on Cloud Computing security (CCSW'09), 2009, pp. 43–54.

[18] R. Carmela, O. Khan, R. Burns, and G. Attendees, "Mr-pdp: Multiple-replica provable data possession," in Proc. of ICDCS'08. IEEE Computer Society, 2008, pp. 411–420.14

[19] Y. Dodos, S. Vashon, and D. Wicks, "Proofs of irretrievability via hardness amplification," in Proc. of the 6th Theory of Cryptography Conference (TCC'09), San Francisco, CA, USA, March 2009.

[20] K. D. Bowers, A. Jules, and A. Opera, "Hail: A high-availability and integrity layer for cloud storage," in Proc. of CCS'09, 2009, pp. 187–198.

[21] D. Bone and M. Franklin, "Identity-based encryption from the wail pairing," in Advances in Cryptology CRYPTO 2001. Springer, 2001,pp. 213–229.

[22] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for fr-reduction," IEICE transactions on fundamentals of electronics, communications and computer sciences, vol. 84,no. 5, pp. 1234–1243, 2001.

[23] R. Genera, J. Katz, H. Krawczyk, and T. Rabin, "Secure network coding over the integers," in Public Key Cryptography–PKC 2010.Springer, 2010, pp. 142–160.

[24] S. Goldwasser, S. Mycale, and R. Rivets, "A digital signature scheme secure against adaptive chosen message attacks," SIAM Journal of Computing, vol. 17, no. 2, pp. 281–308, 1988.

[25] P. S. Barito and M. Nearing, "Pairing-friendly elliptic curves of prime order," in Selected areas in cryptography. Springer, 2006, pp. 319–331.

[26] Y. Descartes, J.-J. Quisquater, and A. Sa¨ıdane, "Remote integrity checks, "in Integrity and Internal

Control in Information Systems VI. Springer, 2004, pp. 1–11.

[27] D. L. Gazonas Filch and P. S. L. M. Barrett, "Demonstrating data possession and untreatable data transfer." IACR Cryptology e Printer chive, vol. 2006, p. 150, 2006.

