International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 4 | Issue 2 | ISSN : 2456-3307

Cloud Services Accessing with Privacy-Preserving Public Auditing

Asha.K [1] Mrs P.Madhura [2]
[1]Department Of Computer Applications , Riims College, Tirupati, Andhra Pradesh, India
[2]Head, Department Of Computer Applications,Riims,Tirupati, Andhra Pradesh, India

ABSTRACT

To protect outsourced data in cloud storage against corruptions, adding fault tolerance to cloud storage along with data integrity checking and failure reparation becomes critical. Recently, create codes have gained quality due to their lower repair information measure whereas providing fault tolerance. Existing remote checking strategies for regenerating-coded information only offer personal auditing, requiring data owners to forever stay on-line and handle auditing, also as repairing, which is sometimes impractical. During this paper, we tend to propose a public auditing theme for the regenerating-code-based cloud storage. To solve the regeneration problem of failing authenticators in the absence of information house owners, we introduce a proxy, which is privileged to regenerate the authenticators, into the normal public auditing system model. Moreover, we tend to style a novel public verifiable critic, which is generated by a few of keys and can be regenerated using partial keys. Thus, our theme will completely release data owners from on-line burden. Additionally, we randomize the inscribe coefficients with a pseudorandom function to preserve information privacy. in depth security analysis shows that our theme is obvious secure below random oracle model and experimental analysis indicates that our theme is highly economical and may 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 1969, Leonard Kleinrock, one of the chief scientists of the original Advanced Research Projects Agency Network (ARPANET) project which seeded the Internet, said: "As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread of „computer utilities" which, like present electric and telephone utilities, will service individual homes and offices across the country." This vision of computing utilities based on a service provisioning model anticipated the massive transformation of the entire computing industry in the 21st century whereby computing services will be readily available on demand, like other utility services available in today's society. Similarly, computing service users (consumers) need to pay providers only when they access computing services. In addition, consumers no longer need to invest heavily or encounter difficulties in building and maintaining complex IT infrastructure. In such a model, users access services based on their requirements without regard to where the services are hosted. This model has been referred to as utility computing, or recently as Cloud computing. The latter term denotes the infrastructure as a "Cloud" from which businesses and users are able to access application services from anywhere in the world on demand. Hence, Cloud computing can be classified as a new paradigm for the dynamic provisioning of computing services, typically supported by state-of-the-art data centers containing ensembles of networked Virtual Machines. Cloud computing delivers infrastructure, platform, and software (application) as services, which are made available as subscription-based services in a pay-asyou-go model to consumers. These services in industry are respectively referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

A Berkeley Report in Feb 2009 states "Cloud computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service". Clouds aim to power the next generation data centers by architecting them as a network of virtual services (hardware, database, user-interface, application logic) so that users are able to access and deploy applications from anywhere in the world on demand at competitive costs depending on users QoS (Quality of Service) requirements. Developers with innovative ideas for new Internet services no longer require large capital outlays in hardware to deploy their service or human expense to operate it. It offers significant benefit to IT companies by freeing them from the low level task of setting up basic hardware (servers) and software infrastructures and thus enabling more focus on innovation and creating business value for their services. The business potential of Cloud computing is recognized by several market research firms including IDC, which reports that worldwide spending on Cloud services will grow from $16 billion by 2008 to $42 billion in 2012. Furthermore, many applications making use of these utility-oriented computing systems such as clouds emerge simply as catalysts or market makers that bring buyers and sellers together. This creates several trillion dollars of worth to the utility/pervasive computing industry as noted by Sun Microsystems co-founder Bill Joy. He also indicated "It would take time until these markets to mature to generate this kind of value. Predicting now which companies will capture the value is impossible. Many of them have not even been created yet."

II. PROPOSED ALGORITHM

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.

Overview:-

Although introduced private remote data checking schemes for regenerating-code-based cloud storage, there are still some other challenges for us to design a public auditable version. First, although a direct extension of the techniques in can realize public verifiability in the multi-servers setting by viewing each block as a set of segments and performing spot checking on them, such a straightforward method makes the data owner generate tags for all segments independently, thus resulting in high computational overhead. Considering that data owners commonly maintain limited computation and memory capacity, it is quite significant for us to reduce those overheads. Second, unlike cloud storage based on traditional erasure code or replication, a fixed file layout does not exist in the regenerating-code-based cloud storage. During the repair phase, it computes out new blocks, which are totally different from the faulty ones, with high probability Thus, a problem arises when trying to determine how to regenerate authenticators for the repaired blocks. A direct solution, which is adopted is to make data owners handle the regeneration. However, this solution is not practical because the data owners will not always remain online through the life-cycle of their data in the cloud, more typically, it becomes off-line even after data uploading. Another challenge is brought in by the proxy in our system model (see Section II-C). The following part of this section shows our solution to the problems above. First, we construct a BLS-based authenticator, which consists of two parts for each segment of coded blocks. Utilizing its homomorphism property and the linearity relation amongst the coded blocks, the data owner is able to generate those authenticators in a new method, which is more efficient compared to the straightforward approach. Our authenticator contains the information of encoding coefficients to avoid data pollution in the reparation with wrong coefficients. To reduce the bandwidth cost during the audit phase, we perform batch verification over all α blocks at a certain server rather than checking the integrity of each block separately does. Moreover, to make our scheme secure against the replace attack and replay attack, information about the indexes of the server, blocks, and segments are all embedded into the authenticator. Besides, our primitive scheme can be easily improved to support privacy-preserving through the masking of the coding coefficients with a keyed PRF. All the algebraic operations in our scheme work over the field G F(p) of modulo p, where p is a large prime (at least 80 bits).

Construction of Our Auditing Scheme:-

Considering the regenerating-code-based cloud storage with parameters (n, k, , α, β), we assume β = 1 for simplicity. Let 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.

Setup: The audit scheme related parameters are initialized in this procedure.

KeyGen (1κ) → (pk,sk): The data owner generates a random signing key pair (spk,ssk), two random elements x, y R ←– Zp and computes pkx ← gx , pky ← gy. Then the secret parameter is sk = (x, y,ssk) and the public parameter is pk = (pkx , pky,spk).

Delegation (sk) → (x): The data owner sends encrypted x to the proxy using the proxy's public key, then the proxy decrypts and stores it locally upon receiving.

$SigAndBlockGen(sk, F) \rightarrow (\Phi, \Psi, t)$: The data owner uniformly chooses a random identifier $ID \overset{R}{\leftarrow} \{0,1\}^*$, a random symbol $u \overset{R}{\leftarrow} G$, one set $\Gamma = \{w_1, w_2, \dots w_m\}$

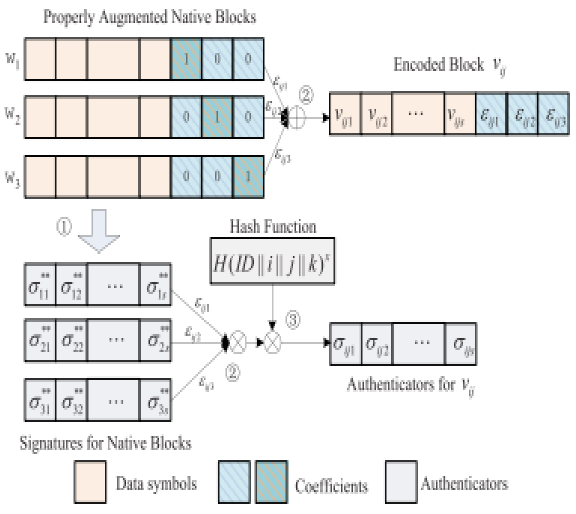

Figure 1. An example with m = 3 for the Sig And BlockGen(·) algorithm

With elements $w_i \overset{R}{\leftarrow} G$, and a file tag t = (I D||u||w1|| … ||wm)||Sigssk(I D||u||w1|| … ||wm) for F. Sig() means a standard signature scheme. Recall that the original file F is split into m blocks, {wi} m i=1; the client computes and stores nα coded blocks amongst n cloud servers. Viewing each segment of the blocks as a single symbol for simplicity, our signature is generated simultaneously with the encoding process as follows (Fig.1 shows an instance of this algorithm with m = 3.):

Augmentation: The data owner properly augments the native m blocks

Signing for Native Blocks: The data owner views the data parts of the augmented blocks as a set of segments and computes authenticator for each segment, i.e.,

$$\sigma^{**}_{j_o k_o} = \left(u^{\overline{w}_{j_o k_o}} \cdot \prod_{\lambda=1}^{m} w_{\lambda}^{\overline{w}_{j_o(s+\lambda)}} \right)^{y} = \left(u^{\overline{w}_{j_o k_o}} \cdot w_{j_o} \right)^{y} \quad (6)$$

where 1 ≤ jo ≤ m, 1 ≤ ko ≤ s

Combination and Aggregation: The data owner randomly chooses m elements {εijλ}m λ=1 from G F(p)

to be coefficients and linearly combines the augmented native blocks to generate
coded blocks vij(1 ≤ i ≤ n, 1 ≤ j ≤ α), as follows:

$$v_{ij} = \sum_{\lambda=1}^{m} \varepsilon_{ij\lambda} w_\lambda \in GF(p)^{s+m} \qquad (7)$$

and apparently each symbol can be obtained by

$$v_{ijk} = \sum_{\lambda=1}^{m} \varepsilon_{ij\lambda} \overline{w}_{\lambda k} \in GF(p) \qquad (8)$$

with 1 ≤ k ≤ s + m. Then we can get the aggregated tags as:

$$\sigma^*_{ijk} = \prod_{\lambda=1}^{m} (\sigma^{**}_{\lambda k})^{\varepsilon_{ij\lambda}} = \left(u^{v_{ijk}} \cdot \prod_{\lambda=1}^{m} w_\lambda^{\varepsilon_{ij\lambda}} \right)^y \qquad (9)$$

with 1 ≤ k ≤ s.

Authenticator Generation: The data owner generates an authenticator for vijk as:

$$\sigma_{ijk} = H(ID||i||j||k)^x \cdot \sigma^*_{ijk}$$
$$= H(ID||i||j||k)^x \left(u^{v_{ijk}} \prod_{\lambda=1}^{m} w_\lambda^{\varepsilon_{ij\lambda}} \right)^y \qquad (10)$$

where the symbols i, j, k denote the index of the server, the index of the block at a server and the index of a segment in a certain coded block.

Audit: For each of the n servers, TPA verifies the possession of α coded blocks by randomly checking samples of segments of every block and performing a batch verification.

Challenge$(\mathcal{F}_{info}) \rightarrow (\mathcal{C})$: Taking the information of file $\mathcal{F}_{info}$ as input, TPA randomly generates a c-element set $Q_i = \{(k^*_\xi, a^*_\xi)\}_{1 \le \xi \le c}$ for server i $(1 \le i \le n)$ with the sampled segment indexes $k^* \in_R [1, s]$ and corresponding randomness $a^* \xleftarrow{R} GF(p)$. Furthermore, to execute a batch auditing, TPA picks another random symbol set $\Lambda_i = \{a_\xi\}_{1 \le \xi \le \alpha}$ with each $a_\xi \xleftarrow{R} GF(p)$. TPA sends $\mathcal{C} = \{Q_i, \Lambda_i\}$ to the server i.

ProofGen$(\mathcal{C}, \Phi, \Psi) \rightarrow (\mathcal{P})$: Upon receiving the challenge $\mathcal{C} = \{Q_i, \Lambda_i\}$, server i first computes

$$\mu_{ij} = \sum_{\tau=1}^{c} a^*_\tau v_{ijk^*_\tau}, \quad \sigma_{ij} = \prod_{\tau=1}^{c} \sigma_{ijk^*_\tau}{}^{a^*_\tau} \qquad (11)$$

for coded block j ∈ [1, α] stored on itself, and then it generates an aggregated proof using i , i.e.,

$$\mu_i = \sum_{j=1}^{\alpha} a_j \mu_{ij}, \quad \sigma_i = \prod_{j=1}^{\alpha} \sigma_{ij}{}^{a_j} \qquad (12)$$

For the coefficients checking, server i generates m symbols as

$$\rho_{i\lambda} = \left(\sum_{j=1}^{\alpha} a_j \varepsilon_{ij\lambda} \right) \cdot \sum_{\tau=1}^{c} a^*_\tau \qquad (13)$$

where $1 \le \lambda \le m$. Let $\Theta_i = \{\rho_{i1}, \rho_{i2}, \ldots, \rho_{im}\}$.

Thus the ith server responds to the TPA with proof $\mathcal{P} = \{\mu_i, \sigma_i, \Theta_i, t\}$.

Verify$(\mathcal{P}, pk, \mathcal{C}) \rightarrow (0, 1)$:

Upon receiving a proof P from server i, TPA uses spk to verify the signature on t to check whether or not it is the delegated file that needs auditing, retrieves u and $\Gamma = \{w_1, w_2, \ldots w_m\}$ if successful and aborts if otherwise.

Then, TPA computes:

$$RHS = e\left(\prod_{j=1}^{\alpha} \prod_{\tau=1}^{c} H_{ijk^*_\tau}^{a_j a^*_\tau}, pk_x \right) \cdot e\left(u^{\mu_i}, pk_y\right) \qquad (14)$$

where $H_{ijk^*_\tau} = H(ID||i||j||k^*_\tau)$. Finally, TPA verify if the following equation holds:

$$e\left(\prod_{\lambda=1}^{m} w_\lambda^{-\rho_{i\lambda}}, pk_y \right) \cdot e(\sigma_i, g) \stackrel{?}{=} RHS \qquad (15)$$

if so, output 1; otherwise, output 0.

Repair: As previously introduced, the data owner empowers a proxy to take charge of faulty server reparation and moves off-line once it completes its upload procedure. When TPA detects a server corruption, an alert will be sent to proxy and then a repair procedure will be triggered. Moreover, to avoid pollution attack, the blocks downloaded from servers will be first verified before they are used for block regeneration. Without loss of generality, we assume that the TPA has identified a certain faulty server η.

ClaimForRep$(\mathcal{F}_{info}) \rightarrow (\mathcal{C}_r)$: The proxy randomly contacts ℓ healthy servers $\{i_\theta\}_{1 \le \theta \le \ell}$. For each server $i \in \{i_\theta\}_{1 \le \theta \le \ell}$, the proxy draws a coefficient set $\Lambda_i = \{a_\xi\}_{1 \le \xi \le \alpha}$ with $a_\xi \in GF(p)$. Then it sends claim $\mathcal{C}_r = \{\Lambda_i\}$ to server i.

GenForRep$(\mathcal{C}_r, \Phi, \Psi) \rightarrow (BA)$:

When server i receives the repair claim Cr, it linearly combines α local coded blocks into one single block and then generates tags for verification:

$$\tilde{v}_i = \sum_{j=1}^{\alpha} a_j v_{ij}, \quad \tilde{\sigma}_{ik} = \prod_{j=1}^{\alpha} {\sigma_{ijk}}^{a_j} \qquad (16)$$

where $(1 \leq k \leq s)$. Then server i responds with the block and authenticators set $BA_i = \{\tilde{v}_i, \{\tilde{\sigma}_{ik}\}_{1 \leq k \leq s}\}$.

$BlockAndSigReGen(C_r, BA) \rightarrow (\Phi', \Psi', \perp)$: Before regeneration, the proxy will execute batch verification for the received blocks. By viewing each $\tilde{v}_i$ as $(\tilde{v}_{i1}, \ldots, \tilde{v}_{is}, \tilde{\varepsilon}_{i1}, \ldots, \tilde{\varepsilon}_{im})$ consulting Eq.(5), the proxy verifies whether the k following equations hold,

$$e(\prod_{i \in \{i_\theta\}} \tilde{\sigma}_{ik}, g) \stackrel{?}{=} e\left(\prod_{i \in \{i_\theta\}} \prod_{j=1}^{\alpha} H_{ijk}^{a_j}, pk_x\right) \cdot e\left(u^{\sum_{i \in \{i_\theta\}} \tilde{v}_{ik}}, pk_y\right) \cdot e\left(\prod_{\lambda=1}^{m} w_\lambda^{\sum_{i \in \{i_\theta\}} \tilde{\varepsilon}_{i\lambda}}, pk_y\right) \qquad (17)$$

where 1 ≤ k ≤ s and Hijk = H(I D||i|| j||k). If verification fails, which means that some of the servers connected are malicious for repair, proxy aborts the reparation; otherwise, it continues to generate new coded blocks and authenticators.

Assuming that the repaired coded blocks would be stored at a new added server η , the proxy rebuilds α blocks as follows: The proxy picks up random coefficients $z_i \xleftarrow{R} GF(p)$ for each $i \in \{i_\theta\}_{1 \leq \theta \leq \ell}$ and then computes a linear combination.

$$v_{\eta' j} = \sum_{i \in \{i_\theta\}} z_i \tilde{v}_i \qquad (18)$$

and regenerate authenticators for each segment

$$\sigma_{\eta' jk} = T_{jk}^{x} \cdot \prod_{i \in \{i_\theta\}} (\tilde{\sigma}_{ik})^{z_i} \qquad (19)$$

where $1 \leq j \leq \alpha, 1 \leq k \leq s$ and the transform operator T_{jk} denotes

$$T_{jk} = \frac{H(ID||\eta'||j||k)}{\prod_{i \in \{i_\theta\}} \prod_{j^*=1}^{\alpha} H(ID||i||j^*||k)^{a_{j^*} z_i}} \qquad (20)$$

Example 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. After producing encoded blocks and authenticators, the staffs upload and distribute them to the cloud servers. Since that the staffs will be frequently off-line, the company employs a trust third party (the TPA) to interact with the cloud and perform periodical verification on the staffs' data blocks in a sampling mode. Once some data corruption is detected, the proxy is informed, it will act on behalf of the staffs to regenerate the data blocks as well as corresponding authenticators in a secure approach. So we could see that our scheme guarantees that the staffs can use the regenerating-code-based cloud in a practical and lightweight way, which completely releases the staffs from online burden for data auditing and reparation.

III. CONCLUSION

In this paper, we propose a public auditing theme for the regenerating-code-based cloud storage system, where the data owner's area unit privileged to delegate TPA for his or her data validity checking. To guard the initial data privacy against the TPA, we randomize the coefficients in the starting rather than applying the blind technique throughout the auditing method. Considering that the information owner cannot continually keep on-line in practice, so as to stay the storage accessible and verifiable after a malicious corruption, we introduce a semi-trusted proxy into the system model and supply a privilege for the proxy to handle the reparation of the coded blocks and authenticators. To better acceptable for the regenerating-code-scenario, we style our appraiser supported the BLS signature. This authenticator will be with efficiency generated by the data owner at the same time with the coding procedure. Extensive analysis shows that our theme is provable secure, and also the performance evaluation shows that our theme is very efficient and can be feasibly integrated into a regenerating-code-based cloud storage system.

IV. REFERENCES

[1]. Hamdi, M., "Security of cloud computing, storage, and networking," Collaboration Technologies and Systems (CTS), 2012 International Conference on , vol., no., pp.1,5, 21-25 May 2012 doi: 10.1109/CTS.2012.6261019

[2]. Dinesha, H.A.; Agrawal, V.K., "Multi-level authentication technique for accessing cloud services," Computing, Communication and Applications (ICCCA), 2012 International Conference on , vol., no., pp.1,4, 22-24 Feb. 2012 doi: 10.1109/ICCCA.2012.

[3]. Baliga, J.; Ayre, R.W.A.; Hinton, K.; Tucker, RodneyS. "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," Proceedings of the IEEE , vol.99, no.1, pp.149,167, Jan. 2011 doi:10.1109/JPROC.2010.2060451

[4]. Kumar, A.; Byung Gook Lee; HoonJae Lee; Kumari, A., "Secure storage and access of data in cloud computing," ICT Convergence (ICTC), 2012 International Conference on , vol., no., pp.336,339, 15-17 Oct. 2012 doi: 10.1109/ICTC.2012.6386854

[5]. IEEE - The Application of Cloud Computing in Education Informatization, Modern Educational Tech... center Bo Wang, HongYu Xing.

[6]. NISTDefinition http://www.au.af.mil/au/awc/awcgate/nist/cloud-def-v15.doc

[7]. CA Technologies cloud authentication system http://www.ca.com/us/authentication-system.aspx

[8]. X. Suo, Y. Zhu, G. S. Owen, "Graphical passwords: A survey," in Proc. 21st Annual Computer Security Application. Conf. Dec. 5–9, 2005, pp. 463–472.

[9]. S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "Authentication using graphical passwords: Basic results," in Proc. Human-Comput. Interaction Int., Las Vegas, NV, Jul. 25–27, 2005.

[10]. Barsoum, A.; Hasan, A.,"Enabling Dynamic Data and Indirect Mutual Trust for Cloud Computing Storage Systems," Parallel and Distributed Systems, IEEE Transactions on, vol.24, no.12,pp.2375,2385,Dec.2013doi:10.1109/TPDS.2012.337.

[11]. ZHANG GUOLI, LIU WANJUN," The Applied Research of Cloud Computing Platform Architecture in the E-Learning Area". IEEE 2010

[12]. Cheng Jiejing, Huang Jingjing, Liu Xiaoxiao, Le Huijin, "On the Application of Educational Information Resource in Jiangxi Province Based on Cloud Computing". IEEE 2010

[13]. Ye Conghuan, Chen xiaowen, "E-learning Support Service Based on Interaction Among Local Campus Clouds" IEEE 2011

[14]. Huibin Yin, lun Han, ling Liu, ling Dong, "The Application Research of GAE on E-Learning" IEEE 2011

[15]. Xin Tan, Yongbeom Kim, "Cloud Computing for Education: A Case of Using Google Docs in MBA Group Projects" IEEE 2011

[16]. Bo Wang, HongYu Xing, "The Application of Cloud Computing in Education Informatization" IEEE 2011

[17]. Zhihe Yang, "Study on an Interoperable Cloud framework for e-Education" IEEE 2011

[18]. Ruan Gaofeng, Cai ling, "Online Course Development Based On a Public Cloud Computing Infrastructure" IEEE 2010

[19]. Yuhua Liu, Lilong Chen, Kaihua Xu, Yi Zhang, "Application Modes, Architecture and Challenges for Cloud Educational System" IEEE 2010