International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 4| ISSN: 2456-3307

Sanctioning Assestation of Retrievability in Cloud

M. Jangamreddy[1], **K. Sunitha**[2]

[12]Department of Computer Science, RIIMS College, Tirupathi, India

ABSTRACT

Cloud computing plays an important role in data storing and accessing now-a-days. Cloud Service Brokers are acting as important meditators for transferring the data between cloud Service Providers and Cloud Consumers. Cloud Computing moves the appliance code and informationbases to the centralized massive data centers, where the management of the info and services might not be totally trustworthy. during this work, we tend to study the matter of ensuring the integrity of information storage in Cloud Computing. To reduce the procedure value at user aspect throughout the integrity verification of their information, the notion of public verifiability has been planned. However, the challenge is that the procedure burden is simply too immense for the users with resource-constrained devices to cypher the general public authentication tags of file blocks. To tackle the challenge, we propose a replacement cloud storage theme involving a cloud storage server and a cloud audit server, where the latter is assumed to be semi-honest. especially, we consider the task of permitting the cloud audit server, on behalf of the cloud users, to pre-process the info before uploading to the cloud storage server and later corroborative the data integrity. It outsources the significant computation of the tag generation to the cloud audit server and eliminates the involvement of user within the auditing and within the preprocessing phases. what is more, we tend to strengthen the Proof of Retrievabiliy model to support dynamic information operations, yet as guarantee security against reset attacks launched by the cloud storage server within the transfer part.
Keywords : Cloud Storage Server , Cloud Audit Server, Cloud Service Provider, Cloud Consumer

I. INTRODUCTION

Cloud Computing has been visualised because the nextgeneration design of the IT enterprise attributable to itslong list of unexampled advantages: on-demand selfservice,ubiquitous network access, location-independentresource pooling, speedy resource physical property, and usagebasedpricing. especially, the ever cheaper and a lot ofpowerful processors, at the side of the "software as aservice" (SaaS) computing design, square measure remodelingdata centers into pools of computing service on an enormousscale.Although having appealing benefits as a promisingservice platform for the net, this new knowledge storageparadigm in "Cloud" brings several difficult problemswhich have profound influence on the usability, responsibleness,scalability, security, and performance of thesystem. one amongst the most important considerations with remote knowledgestorage is that of knowledge integrity verification at untrusted servers. for example, the storage service supplier mightdecide to hide such knowledge loss incidents because the Byzantinefailure from the shoppers to keep up a name. What ismore serious is that for saving cash and space for storing the service supplier may deliberately discard seldomaccessed knowledge files that belong to a standard consumer.Considering the big size of the outsourced electronicdata and therefore the client's strained resource capability, thecore of the matter may be generalized as however will theclient realize an economical thanks to perform periodical integrityverification while not the native copy of knowledge files.In order to beat this drawback, several schemeshave been planned underneath totally different system and security models. all told these works, nice efforts have been created to style solutions that meet numerous requirements: High theme potency, homeless verification,unbounded use of queries and retrievability of knowledge, etc.According to the role of the admirer within the model, allthe schemes on the market fall under 2 categories: personal verifiability and public verifiability. though achievinghigher potency, schemes with personal verifiability imposecomputational burden on shoppers. On the opposite hand, public verifiability alleviates shoppers from performing arts alot of computation for making certain the integrity of knowledge storage. To be specific, shoppers square measure ready to delegate athird party to perform the verification while not devotion of their computation resources. within the cloud, the shoppersmay crash unexpectedly or cannot afford the overload offrequent integrity checks. Thus, it looks a lot of rationaland sensible to equip the verification protocol withpublic verifiability, that is predicted to play a a lot ofimportant role in achieving higher potency for Cloud Computing.

We propose a new PoR scheme with two independent cloud servers. Particularly, one server is for auditing and the other for storage of data. The cloud audit server is not required to have high storage capacity. Different from the previous work with auditing server and storage server, the user is relieved from the computation of the tags for files, which is moved and outsourced to the cloud audit server. Furthermore, the cloud audit server also plays the role of auditing for the files remotelystored in the cloud storage server.

We develop a strengthened security model by considering the reset attack against the storage server in the upload phase of an integrity verification scheme.

We present an efficient verification scheme for ensuring remote data integrity in cloud storage.The proposed scheme is proved secure against reset attacks in the strengthened security model while supporting efficient public verifiability and dynamic data operations simultaneously.

II. POR SCHEME

The basic goal of PoR model is to achieve proof of retrievability. Informally, this property ensures that if an adversary can generate valid integrity proofs of any file F for a non-negligible fraction of challenges, we can construct a PPT machine to extract F with overwhelming probability.

Setup: The cloud audit server chooses a random $\alpha \leftarrow Z_p, u_1, u_2, \cdots, u_s \leftarrow G$, and computes $v \leftarrow g\_$. The secret key is $sk = (\alpha)$ and the public key is $pk = (v, \{u_j\}_{1\le j\le s})$.

Upload (Phase 1: Client→ Cloud Audit Server):

The client uploads $F = (M_1, \ldots, M_n)$ to the cloud audit server. Given the file F, the cloud audit server generates a root R based on the construction of Merkle Hash Tree (MHT), where the leave nodes of the tree are an ordered set of hashes of file blocks $H(M_i)$ $(i = 1, \ldots, n)$. Next, he signs the root R under his private key α as $h(R)\_ \leftarrow sig_{sk}(R)$. The file tag $t = sig_{sk}(R)$ is sent back to the client as a receipt. (Phase 2: Cloud Audit Server→ Cloud Storage Server): The homomorphic authenticators together with metadata are produced as follows: for each block $M_i = (M_{i1}, M_{i2}, \cdots, M_{is})$, the cloud audit server computes a signature σ_i as

$$\sigma_i \leftarrow \left(H(M_i) \cdot \prod_{j=1}^{s} u_j^{M_{ij}} \right)^{\alpha}.$$

Denote the set of signatures by $\Phi = \{\sigma_i\}_{1\le i\le n}$. The cloud audit server sends $F_\Phi = \{F,\Phi\}$ to the cloud storage server. Then, the cloud audit server keeps the receipt t and deletes $F_\Phi$from its local storage. Integrity Verification: Either the client or the cloud audit server can verify the integrity of the outsourced data by challenging the cloud storage server. To generate the challenge query, the cloud audit server (verifier) picks a random c-element subset I of set [1, n] that denote the positions of the blocks to be checked.

- **Data Modification**

Suppose a client intends to modify the i-th block M_i to M'_i , then the following procedures have to been performed:

1) The client sends an update request message “update = (M, i, M'_i)” to the cloud audit server, where M denotes the modification operation.

2) Upon receiving the request, the cloud audit server generates the corresponding signature , and sends update′ = (update, σ'_i) to the storage server.

$$\sigma'_i = \left(H(M'_i) \cdot \prod_{j=1}^{s} u_j^{\hat{M}'_{ij}} \right)^{\alpha}.$$

3) Upon receiving update′, the storage serverperforms the following operations.– He replaces the block M_i with M_i and outputs F′.

– Replaces the σi with σ′i and outputs Φ′.
– Replaces H(Mi) with H(M′i) in theMerkle hash tree construction and generates the new root R′.
– For the modification operation, replies the client with a proof Pupdate =(Ωi,H(Mi),R′), where Ωi is the AAI of Mi.
4) After receiving the proof Pupdate from thestorage server, the cloud audit server operates as follows.
– He generates root R using {Ωi,H(Mi)}.
– Authenticates R by checking if e(t, g) =e(h(R), v).
– Computes the new root value ˆR using
{Ωi,H(M′i)} and checks if ˆR = R′– Signs the new root metadata R′ by t′ =sigsk(R′) and sends it to the server for storage.

- **Data Insertion**

Suppose the data owner wants toinsert block M∗ after the i-th block Mi. The protocolprocedures are similar to the data modification case.

1) After receiving the proof for insert operationfrom the storage server, the client first generatesroot R using {Ωi,H(Mi)} and authenticatesR by checking if e(t, g) = e(h(R), v).
2) If it is not true, output FALSE, otherwise theclient can now check whether the server hasperform the insertion as required or not, byfurther computing the new root value using{Ωi,H(H(Mi)‖H(M∗))} and comparing itwith R′.
3) If not, output FALSE, otherwise output TRUE.
4) The cloud auditor server signs the new rootmetadata R′ by sigsk(R′) and sends it to theserver for storage.

III. CONCLUSION

This paper proposes a brand new proof of retrievability for cloud storage, during which a trustworthy audit server is introduced to preprocess and transfer the info on behalfof the purchasers. In this, the computation overhead for tag generation on the consumer aspect is reduced considerably. The cloud audit server conjointly performs the info integrity verification or change the outsourced knowledge upon theclients' request. Besides, we have a tendency to construct another new scheme proved secure underneath a PoR model with increasedsecurity against reset attack within the transfer section. Thescheme conjointly supports public verifiability and dynamicdata operation at the same time.

IV. REFERENCES

[1]. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in CCS '07: Proceedings of the 14th ACM conference on Computer and communications security. New York, NY, USA:ACM, 2007, pp. 598–609.

[2]. A. Juels and B. S. K. Jr., "Pors: proofs of retrievability for large files," in CCS '07: Proceedings of the 14th ACM conference on Computer and communications security. New York, NY, USA: ACM, 2007, pp. 584–597.

[3]. H. Shacham and B. Waters, "Compact proofs of retrievability," in ASIACRYPT '08: Proceedings of the 14th International

[4]. Conference on the Theory and Application of Cryptology and Information Security. Berlin, Heidelberg: Springer-Verlag, 2008,pp. 90–107.

[5]. K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: theory and implementation," in Proceedings of CCSW 2009. ACM, 2009, pp. 43–54.

[6]. M. Naor and G. N. Rothblum, "The complexity of online memory checking," J. ACM, vol. 56, no. 1, pp. 2:1–2:46, Feb. 2009. [Online]. Available: http://doi.acm.org/10.1145/1462153.1462155

[7]. E.-C. Chang and J. Xu, "Remote integrity check with dishonest storage server," in Proceedings of ESORICS 2008, volume 5283of LNCS. Springer-Verlag, 2008, pp. 223–237.

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

[9]. A. Oprea, M. K. Reiter, and K. Yang, "Space-efficient block storage integrity," in In Proc. of NDSS 2005, 2005.

[10]. T. S. J. Schwarz and E. L. Miller, "Store, forget, and check: Using algebraic signatures to check remotely administered storage," in ICDCS '06: Proceedings of the 26th IEEE InternationalConference on Distributed Computing Systems. Washington, DC, USA: IEEE Computer Society, 2006.

[11]. Q. Wang, K. Ren, S. Yu, and W. Lou, "Dependable and secure sensor data storage with dynamic integrity assurance," ACM Transactions on Sensor Networks, vol. 8, no. 1, pp. 9:1–9:24,

Aug. 2011. [Online]. Available: http: //doi.acm.org/10.1145/1993042.1993051

[12]. L. V. M. Giuseppe Ateniese, Roberto Di Pietro and G. Tsudik, "Scalable and efficient provable data possession," in InternationalConference on Security and Privacy in Communication Networks(SecureComm 2008), 2008.

[13]. C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in INFOCOM, 2010, pp. 525–533.

[14]. Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Dynamic audit services for integrity verification of outsourced storages in clouds," in SAC, 2011, pp. 1550–1557.

[15]. Q. Zheng and S. Xu, "Fair and dynamic proofs of retrievability," in CODASPY, 2011, pp. 237–248.

[16]. J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou, "Fine-grained access control system based on attribute-based encryption," ESORICS, 2013.

[17]. J. Li and K. Kim, "Hidden attribute-based signatures without anonymity revocation," Information Sciences, vol. 180, no. 9, pp. 1681–1689, 2010.

[18]. J. Li, C. Jia, J. Li, and X. Chen, "Outsourcing encryption of attribute-based encryption with mapreduce," ICICS, 2012.

[19]. X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms of outsourcing modular exponentiations," ESORICS, pp. 541–556, 2012.

[20]. Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," IEEETrans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231–2244, 2012.

[21]. H. Xiong, X. Zhang, D. Yao, X. Wu, and Y. Wen, "Towards endto- end secure content storage and delivery with public cloud," in CODASPY, 2012, pp. 257–266.

[22]. Q. Zheng and S. Xu, "Secure and efficient proof of storage with deduplication," in CODASPY, 2012, pp. 1–12.

[23]. C. Wang, Q. Wang, and K. Ren, "Ensuring data storage security in cloud computing," in Proceedings of IWQoS 2009, Charleston, South Carolina, USA, 2009.

[24]. C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," Cryptology ePrint Archive, Report 2008/432, 2008, http://eprint.iacr.org/.

[25]. X. Lei, X. Liao, T. Huang, H. Li, and C. Hu, "Outsourcinglarge matrix inversion computation to a public cloud,," in IEEETransactions on Cloud Computing, 2013, pp. vol. 1, no. 1.

[26]. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling publicverifiability and data dynamics for storage security in cloud computing," in ESORICS, 2009, pp. 355370.