INTERNATIONAL JOURNAL OF COMPUTATIONAL MATHEMATICAL IDEAS

ISSN: 0974-8652 / www.sijcmi.weebly.com / VOL 6-NO3-PP 6323-6327 (2014)

Public Auditing With Privacy for Secure Public Cloud Storage

Seetha Maha Lakshmi.Thopuri [#1], A. Srinivasa Rao, Assistant Professor, M.Tech.[#2]

Department Of Computer Science and Engineering,
Buchepalli Venkayamma Subbareddy Engineering College, *Affiliated to JNTUK*, Chimakurthy, Andhra Pradesh, India.

[1*] seetha.thopuri@gmail.com

[2*] asr.bvsr@gmail.com

Abstract –**In this we propose a secure cloud storage system supporting privacy-preserving public auditing. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public audit ability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free.**

Keywords- **Data storage, privacy preserving, public auditability, cloud computing, delegation, batch verification, zero knowledge,Homomorphic Linear Authenticator, Third Party Auditor, Public Auditing, Zero knowledge, Data storage, Cloud computing.**

I. INTRODUCTION

A privacy-preserving public auditing system for data storage security in cloud computing in this the homo morphic linear authenticator and random masking to guarantee that the TPA[1] would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process. It not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Using cloud storage, usre can remotly store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact the user on longer have physical possession of the outsourced data makes the data integrity rotection in cloud computing a formidable task, especially for the users with constrained computing resource. Enabling public audit ability for cloud storage is of critical importance so that user can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. Here the secure cloud storage system supporting privacy preserving public auditing is proposed. The public auditability, i.e. "provable data possession" (PDP) is a model for ensuring possession of data files on untrusted storages. The scheme utilizes the RSA based homo morphic non-linear authenticators for auditing outsourced data and suggests randomly sampling a few blocks of the file. However, the public auditability in their scheme demands the linear combination of sampled blocks exposed to external auditor. When used directly, the protocol is not provably privacy preserving, and thus may leak user data information to the auditor. Juels et al. [11] describes a "proof of retrievability" (PoR) model, where spot-checking and error correcting codes are used to ensure both possession and retrievability of data files on remote archive service systems.

However, the number of audit challenges a user can perform is a fixed priori, and public auditability is not supported in their main scheme. Although they describe a straightforward Merkle-tree construction for public PoRs, this approach only works with encrypted data. Dodis et al. [5] give a study on different variants of PoR with private auditability. Shacham et al.[13] design an improved PoR scheme built with full proofs of security in the security model defined in [11]. Similar to the construction in [8], they use publicly verifiable homomorphic non-linear authenticators that are built from provably secure BLS signatures. Based on the elegant BLS construction, a compact and public verifiable scheme is obtained. Again, their approach does not support privacy preserving auditing for the same reason as [8]. The propose allowing a TPA to keep online storage honest by first encrypting the data then sending a number of pre computed

ISSN: 0974-8652 / www.sijcmi.weebly.com / VOL 6-NO3-PP 6323-6327 (2014)

symmetric-keyed hashes over the encrypted data to the auditor. The auditor verifies both the integrity of the data file and the server's possession of a previously committed decryption key. This scheme only works for encrypted files and it suffers from the auditor statefullness and bounded usage, which may potentially bring in online burden to users when the keyed hashes are used up. The dynamic version of the prior PDP scheme, using only symmetric key cryptography but with a bounded number of audits. Consider a similar support for partial dynamic data storage in a distributed scenario with additional feature of data error localization. In a subsequent work, Wang et al. [10] propose to combine BLS-based HLA with MHT to support both public auditability and full data dynamics. Almost simultaneously developed a skip lists based scheme to enable provable data possession with full dynamics support. However, the verification in these two protocols requires the linear combination of sampled blocks just as [8], [13], and thus does not support privacy preserving auditing. While all the above schemes provide methods for efficient auditing and provable assurance on the correctness of remotely stored data, none of them meet all the requirements for privacy preserving public auditing in cloud computing. More importantly, none of these schemes consider batch auditing, which can greatly reduce the computation cost on the TPA when coping with a large number of audit delegations.

A.MAC Based Solution

It is used to authenticate the data. In this, user upload data blocks and MAC to CS provide its secret key SK to TPA. The TPA will randomly retrieve data blocks & Mac uses secret key to check correctness of stored data on the cloud. Problems with this system are listed below as it introduces additional online burden to users due to limited use (i.e. Bounded usage) and stateful verification. Communication & computation complexity TPA requires knowledge of data blocks for verification Limitation on data files to be audited as secret keys are fixed After usages of all possible secret keys, the user has to download all the data to recomputed MAC & republish it on CS. TPA should maintain & update states for TPA which is very difficult it supports only for static data not for dynamic data.

B. HLA Based Solution

It supports efficient public auditing without retrieving data block. It is aggregated and required constant bandwidth. It is possible to compute an aggregate HLA which authenticates a linear combination of the individual data blocks.

C. Using Virtual Machine

Abhishek Mohta proposed Virtual machines which uses RSA algorithm, for client data/file encryption and decryptions [5]. It also uses SHA 512 algorithm which makes message digest and check the data integrity. The Digital signature is used as an identity measure for client or data owner. It solves the problem of integrity, unauthorized access, privacy and consistency.

D. Non Linear Authentication

Authenticator with random masking techniques to achieve Cloud security [7]. K. Gonvinda proposed digital signature Method to protect the privacy and integrity of data [8]. It uses RSA algorithm for encryption and decryption which follows the process of digital signatures for message authentication.

II. RELATED WORK

Consider a cloud data storage service involving three different entities, as illustrated in Fig. the cloud user (U), who has large amount of data files to be stored in the cloud; the cloud server (CS), which is managed by the cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources, the third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. To save the computation resource as well as the online burden, cloud users may resort to TPA for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from TPA. Assuming that the data integrity threats towards user data can come from both internal and external attacks at CS. These may include: software bugs, hardware failures, bugs in the network path, economically motivated hackers, malicious or accidental management errors, etc. Besides, CS can be self-interested. For their own benefits, such as to maintain reputation, CS might even decide to hide these data corruption incidents to users. Using third-party auditing service provides a cost-effective method for users to gain trust in cloud. Considering the TPA, who is in the business of auditing, is reliable and independent. However, it may harm the user if the TPA could learn the outsourced data after the audit. Note that in our project, beyond users' reluctance to leak data to TPA, it is also assumed that cloud servers have no incentives to reveal their hosted data to external parties. On the one hand, there are regulations, e.g., HIPAA [2], requesting CS to maintain users' data privacy. On the other hand, as users' data belong to their business asset [3], there also exist financial incentives for CS to protect it from any external parties. Therefore neither CS nor TPA has motivations to collide with each other during the auditing process. To authorize the CS to respond to the audit delegated to TPA's, the user can issue a certificate on TPA's public key, and all audits from the TPA are authenticated against such a certificate. To enable privacy-preserving public auditing for cloud data

Storage under the aforementioned model, our protocol design should achieve the following security and performance guarantees.

1. Public auditability: To allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users.

2. Storage correctness: To ensure that there exists no cheating cloud server that can pass the TPA's audit without indeed storing users' data intact.

3. Privacy-preserving: To ensure that the TPA cannot derive users' data content from the information collected during the

ISSN: 0974-8652 / www.sijcmi.weebly.com / VOL 6-NO3-PP 6323-6327 (2014)

auditing process.

The Privacy-Preserving Public Auditing Protocol

TPA		Cloud Server
1. Retrieve file tag t, verify its signature, and quit if fail;		
2. Generate a random challenge $chal = \{(i, \nu_i)\}_{i \in I}$;	$\xrightarrow{\{(i,\nu_i)\}_{i\in I}}$ challenge request $chal$	3. Compute $\mu' = \sum_{i\in I} \nu_i m_i$, and $\sigma = \prod_{i\in I} \sigma_i^{\nu_i}$;
		4. Randomly pick $r \leftarrow \mathbb{Z}_p$, and $R = e(u, v)^r$ and $\gamma = h(R)$;
	$\xleftarrow{\{\mu,\sigma,R\}}$ storage correctness proof	5. Compute $\mu = r + \gamma\mu' \mod p$;
6. Compute $\gamma = h(R)$, and then verify $\{\mu, \sigma, R\}$ via Equation 1.		

4. Lightweight: To allow TPA to perform auditing with minimum communication and computation overhead. Consider a cloud data storage service involving three

The Batch Auditing Protocol

TPA		Cloud Server
1. Verify file tag t_k for each user k, and quit if fail;		For each user k $(1 \le k \le K)$:
2. Generate a random challenge $chal = \{(i, \nu_i)\}_{i \in I}$;	$\xrightarrow{\{(i,\nu_i)\}_{i\in I}}$ challenge request $chal$	3. Compute μ'_k, σ_k, R_k as single user case;
		4. Compute $\mathcal{R} = R_1 \cdot R_2 \cdots R_K$, $\mathcal{L} = vk_1 \| vk_2 \| \cdots \| vk_K$
	$\xleftarrow{\{\{\sigma_k,\mu_k\}_{1\le k\le K},\mathcal{R}\}}$ storage correctness proof	and $\gamma_k = h(\mathcal{R} \| v_k \| \mathcal{L})$;
6. Compute $\gamma_k = h(\mathcal{R} \| v_k \| \mathcal{L})$ for each user k and do batch auditing via Equation 3.		5. Compute $\mu_k = r_k + \gamma_k \mu'_k \mod p$;

the cloud user (U), who has large amount of data files to be stored in the cloud; the cloud server (CS), which is managed by the cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources, the third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. To save the computation resource as well as the online burden, cloud users may resort to TPA for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from TPA.

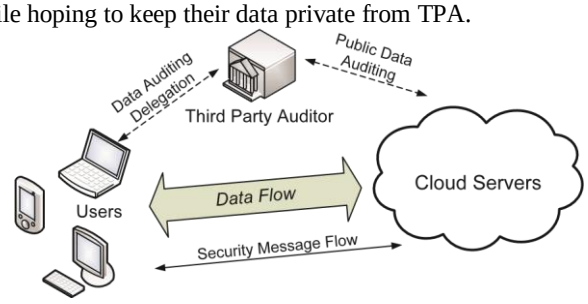


Fig. 1. The architecture of cloud data storage service.

Assuming that the data integrity threats towards user data can come from both internal and external attacks at CS. These may include: software bugs, hardware failures, bugs in the network path, economically motivated hackers, malicious or accidental management errors, etc. Besides, CS can be self-interested. For their own benefits, such as to maintain reputation, CS might even decide to hide these data corruption incidents to users. Using third-party auditing service provides a cost-effective method for users to gain trust in cloud. Considering the TPA, who is in the business of auditing, is reliable and independent. However, it may harm the user if the TPA could learn the outsourced data after the audit. Note that in our project, beyond users' reluctance to leak data to TPA, it is also assumed that cloud servers have no incentives to reveal their hosted data to external parties. On the one hand, there are regulations, e.g., HIPAA [2], requesting CS to maintain users' data privacy. On the other hand, as users' data belong to their business asset [3], there also exist financial incentives for CS to protect it from any external parties. Therefore neither CS nor TPA has motivations to collide with each other during the auditing process. To authorize the CS to respond to the audit delegated to TPA's, the user can issue a certificate on TPA's public key, and all audits from the TPA are authenticated against such a certificate. Communication and computation overhead Properties of the Protocol: It is easy to see that our protocol achieves public auditability. There is no secret keying material or states for the TPA to keep or maintain between audits, and the auditing protocol does not pose any potential online burden on users. This approach ensures the privacy of user data content during the auditing process by employing a random masking r to hideμ, a linear combination of the data blocks. Note that the value R in our protocol, which enables the privacy preserving guarantee, will not affect the validity of the equation, due to the circular relationship between R and in $\gamma = h(R)$ and the verification equation. Storage correctness thus follows from that of the underlying protocol [13]. Besides, the HLA helps achieve the constant communication overhead for server's response during the audit: the size of {μ, α, R} is independent of the number of sampled blocks.

III. EVALUATION

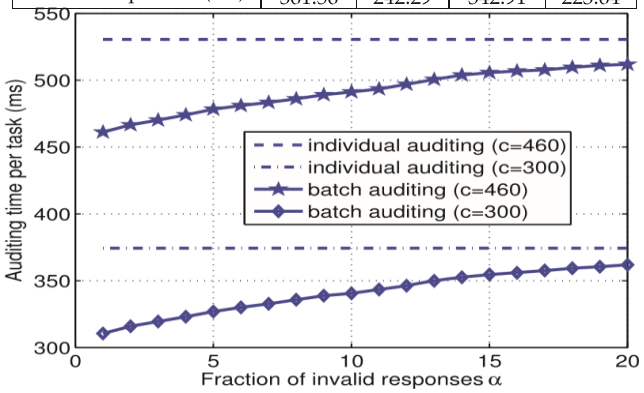
we evaluate the security of the proposed scheme by Analyzing its fulfillment of the security guarantee described Namely, the Storage correctness and privacy preserving property. We start from the single user case, where our main result is originated. Then, we show the security guarantee of batch auditing for the TPA in multiuser setting. The below theorem shows that TPA cannot derive users' data from the information collected during auditing. Theorem 2. From the server's response TPA cannot recover. Proof. We show the existence of a simulator that can produce a valid response even without the knowledge, in the random oracle model. Now, the TPA is treated as an adversary.

Performance under Different Number of Sampled Blocks c for High Assurance (95%) Auditing

s = 1	Our Scheme		[13]	
Sampled blocks c	460	300	460	300

ISSN: 0974-8652 / www.sijcmi.weebly.com / VOL 6-NO3-PP 6323-6327 (2014)

Sever comp. time (ms)	335.17	219.27	333.23	217.33
TPA comp. time (ms)	530.60	357.53	526.77	353.70
Comm. cost (Byte)	160	160	40	40
s = 10	Our Scheme		[13]	
Sampled blocks c	460	300	460	300
Sever comp. time (ms)	361.56	242.29	342.91	223.64



set since the simulator is controlling the random oracle. We remark that this back patching technique in the random oracle model is also used in the proof of the underlying scheme [13]. We now report some performance results of our experiments. We consider our auditing mechanism happens between a dedicated TPA and some cloud storage node, where user's data are outsourced to. In our experiment, the TPA/user side process is implemented on a workstation with an Intel Core 2 processor running at 1.86 GHz, 2,048 MB of RAM, and a 7,200 RPM Western Digital 250 GB Serial ATA drive. The cloud server side process is implemented on Amazon Elastic Computing Cloud (EC2) with a large instance type [27], which has 4 EC2 Compute Units, 7.5 GB memory, and 850 GB instance storage. The randomly generated test data is of 1 GB size. All algorithms are implemented using C language. Our code uses the Pairing-Based Cryptography (PBC) library version 0.4.21. The elliptic curve utilized in the experiment is an MNT curve, with base field size of 159 bits and the embedding degree 6. The security level is chosen to be 80 bit, which means ¼ 80 and ¼ 160. All experimental results represent the mean of 20 trials. Because the cloud is a pay-per-use model, users have to pay both the storage cost and the bandwidth cost (for data transfer) when using the cloud storage auditing. Thus, when implementing our mechanism, we have to take into consideration both factors. In particular, we conducts the experiment with two different sets of storage/communication tradeoff parameter s as introduced in Section 3.4. When s ¼ 1, the mechanism incurs extra storage cost as large

IV. CONCLUSION AND FUTURE WORK

Here proposed is a privacy-preserving public auditing system for data storage security in Cloud Computing. The homo morphic linear authenticator and random masking guarantees that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. With the establishment of privacy-preserving public auditing in cloud computing, TPA may concurrently handle multiple auditing delegations upon different user's requests. The individual auditing of these tasks for TPA can be tedious and inefficient. Batch auditing not only allows TPA to perform the multiple auditing tasks simultaneously, but also reduces the computation cost on TPA side. And also we can extend our work to support for the data dynamics which includes the block level operations of modification, deletion, insertion.

V. REFERENCES

[1] Krebs, "Payment Processor Breach May Be Largest Ever," Onlineat,http://voices.washingtonpost.com/securityfix/2009/01/payment processor breach may b.html, Jan. 2009.

[2] Q. Wang, C. Wang, J. Li, K. Ren, 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-Verlag, Sep. 2009, pp. 355–370. [3] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. of Asiacrypt 2008, vol. 5350, Dec 2008, pp. 90–107.

[4] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson, and D. X. Song. Provable data possession at untrusted stores. In Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, pages 598–609, 2007.

[5] Cloud Security Alliance, "Security guidance for critical Areas of focus in cloud computing," 2009,. cloudsecurityalliance.org.

[6] Dr. P. K. Deshmukh, Mrs. V. R. Desale, Prof. R. A. Deshmukh, "Investigation of TPA (Third Party Auditor Role) foe Cloud Data Security", International Journal of Scientific and Engineering Research, vo. 4,no. 2,ISSn 2229-5518, Feb 2013.

[7] Jachak K. B., Korde S. K., Ghorpade P. P. and Gagare G. J. ,"Homomorphic Authentication with Random Masking Technique Ensuring Privacy & Security in Cloud Computing", Bioinfo Security Informatics, vol. 2, no. 2, pp. 49-52, ISSN. 2249-9423, 12 April 2012 IEEE Trans. Mobile Computing, vol. 6, no. 8, pp. 872-887, Aug. 2007. Cloud Security Alliance.

[8] M. Arrington, "Gmail Disaster: Reports of Mass Email Deletions,"http://www.techcrunch.com/2006/12/28/gmail-isasterreportsof- mass-email-deletions/, 2006.

[9] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors," http:// www.techcrunch.com/2008/07/10/mediamaxthelinkup-closesits-doors/, July 2008.

[10] Amazon.com, "Amazon s3 Availability Event: July 20, 2008,"http://status.aws.amazon.com/s3-20080720.html,July2008.

[11] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

ISSN: 0974-8652 / www.sijcmi.weebly.com / VOL 6-NO3-PP 6323-6327 (2014)

[12] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-609, 2007.

[13] M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Cryptology ePrint Archive, Report 2008/186, 2008.

[14] A. Juels and J. Burton, S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, Oct. 2007.

[15] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop Hot Topics in Operating Systems (HotOS '07), pp. 1-6, 2007.

[16] 104th United States Congress, "Health Insurance Portability and Accountability Act of 1996 (HIPPA)," http://aspe.hhs.gov/ admnsimp/pl104191.htm, 1996.

[17] R. Curtmola, O. Khan, and R. Burns, "Robust Remote Data Checking," Proc. Fourth ACM Int'l Workshop Storage Security and Survivability (StorageSS '08), pp. 63-68, 2008.

[18] K.D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Proc. ACM Workshop Cloud Computing Security (CCSW '09), pp. 43-54, 2009.

[19] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," J. Cryptology, vol. 17, no. 4, pp. 297-319, 2004.

[20] A.L. Ferrara, M. Green, S. Hohenberger, and M. Pedersen, "Practical Short Signature Batch Verification," Proc. Cryptographers' Track at the RSA Conf. 2009 on Topics in Cryptology (CT-RSA), pp. 309-324, 2009.