ISSN:0975-9646

D.Srinivas / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (6) , 2011, 2691-2693.

Privacy-Preserving Public Auditing In Cloud Storage Security

D. Srinivas
Kakinada Institute of Engg & Tech, Kakinada

Abstract--The Cloud Computing is the new vision of computing utility, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or cloud provider interaction. Cloud computing technologies can be implemented in a wide variety of architectures, under different service and deployment models, and can coexist with other technologies and software design approaches. The security challenges cloud computing presents the burden of local data storage and maintenance. public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed. To securely introduce an effective third party auditor to check the integrity of outsourced data. 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. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We utilize the homomorphic non- linear authenticator and random masking to guarantee 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.

I. INTRODUCTION

The cloud computing has rapidly grown in recent years due to the advantages of greater flexibility and availability of computing resources at lower cost. Security and privacy, however, are a concern for agencies and organizations considering migrating applications to public cloud computing environments. Cloud Computing has been envisioned as the next generation architecture of IT enterprise, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk [1]. As a disruptive technology with profound implications, Cloud Computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced into the Cloud. The Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users' outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity.

II. PROBLEM STATEMENT

The Cloud and Threat Model

The Cloud security responsibilities can be taken on by the customer, if he is managing the cloud, but in the case of a public cloud, such responsibilities are more on the cloud provider and the customer can just try to assess if the cloud provider is able to provide security. cloud data storage service involving three different entities. 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 *cloud service provider* (CSP) to provide data storage service and has significant storage space and computation resources (we will not differentiate CS and CSP hereafter.); 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 security on behalf of the user upon request. Cloud users dynamically interact with the CS to access and update their stored data for various application purposes. The traditional cryptographic technologies for data integrity and availability, cannot work on the outsourced data without a local copy of data. it is not a practical solution for data validation by downloading them due to the expensive communications, especially for large size files. The ability to audit the correctness of the data in a cloud environment can be formidable and expensive for the cloud users. Therefore, it is crucial to realize public auditability for CSS, so that data owners may resort to a third party auditor (TPA), who has expertise and capabilities that a common user does not have, for periodically auditing the outsourced data. This audit service is significantly important for digital forensics and credibility in clouds. The users may resort to TPA for ensuring the storage security of their outsourced data, while hoping to keep their data private from TPA. We consider the existence of a semi-trusted CS as [11] does. Namely, in most of time it behaves properly and does not deviate from the prescribed protocol execution. However, during providing the cloud data storage based services, for their own benefits the CS might neglect to keep or deliberately delete rarely accessed data files which belong to ordinary cloud users. Moreover, the CS may decide to hide the data corruptions caused by server hacks or Byzantine failures to maintain reputation. We assume the TPA, who is in the business of auditing, is reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process. TPA should be able to

efficiently audit the cloud data storage without local copy of data and without bringing in additional on-line burden to cloud users. However, any possible leakage of user's outsourced data towards TPA through the auditing protocol should be prohibited. the audit delegation and authorize CS to respond to TPA's audits, the user can sign a certificate granting audit rights to the TPA's public key, and all audits from the TPA are authenticated against such a certificate.

Design Goals

The privacy-preserving public auditing for cloud data storage under the aforementioned model, our protocol design should follow the security and performance.

Public Audit: It allows TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data .

Storage Consistency: the data in cloud server that can pass the audit from TPA without indeed storing users' data intact.

Privacy-Preserving: to ensure that there exists no way for TPA to derive users' data content from the information collected during the auditing process.

Batch Auditing: It enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously.

Light Weight: It allow TPA to perform auditing with minimum communication and computation overhead.

III. PRIVACY-PRESERVING PUBLIC AUDITING

The privacy-preserving public auditing, we propose to uniquely integrate the homomorphic non-linear authenticator with random masking technique. In our protocol, the non-linear blocks in the server's response is masked with randomness generated the server. With random masking, the TPA no longer has all the necessary information to build up a correct group of non-linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected. On the other hand, the correctness validation of the block authenticator pairs can still be carried out in a new way which will be shown shortly, even with the presence of the randomness. Our design makes use of a public key based HLA, to equip the auditing protocol with public auditability. Specifically, we use the HLA proposed in [13], which is based on the short signature scheme.

Periodic Sample Audit

In the Cloud Server environment random "sampling" checking greatly reduces the workload of audit services, while still achieve an effective detection of misbehavior. Thus, the probabilistic audit on sampling checking is preferable to realize the abnormality detection in a timely manner, as well as rationally allocate resources. The fragment structure can provide the support of probabilistic audit as well: given a random chosen challenge (or query) Q = {(i, vi)}iεI , where I is a subset of the block indices and vi is a random coefficient, an efficient algorithm is used to produce a constant-size response(μ1, μ2, … , μs, _′), where μi comes from all {mk,i, vk}kεI and all {_k, vk}kεI . Generally, this algorithm relies on homomorphic properties to aggregate data and tags into a constant size response, which minimizes network communication. Since the single sampling checking may overlook a very small number of data abnormality, we propose a periodic sampling approach to audit outsourcing data, which is called as Periodic Sampling Audit. In this way, the audit activities are efficiently scheduled in an audit period, and a TPA needs merely access small portions of file to perform audit in each activity. Therefore, this method can detect the exceptions in time, and reduce the sampling numbers in each audit.

Security Consistency for Batch Auditing

The way to describe the result to a multi-user setting will not affect the aforementioned security insurance, as shown in the Theorem.

Theorem: The batch auditing protocol achieves the same storage correctness and privacy preserving guarantee as in the single-user case.

Solution*:* The privacy-preserving guarantee in the multi-user setting. The storage correctness guarantee, we are going to reduce it to the single-user case. We use the forking technique for the verification equation for the batch audits involves K challenges from the random block. This time we need to ensure that all the other K − 1 challenges are determined before the forking of the concerned random oracle response. This can be done using the idea in [4]. As soon as the adversary issues the very first random oracle query for i = h(R||vi||L) for any i ε [1,K], the simulator immediately determines the values j = h(R||vj ||L) for all j ε [1,K]. This is possible since they are all using the same R and L. Now, all but one of the k's are equal, so a valid response can be extracted similar to the single-user case.

IV. RELATED WORK

The public auditability in their defined "provable data possession" (PDP) model for ensuring possession of data files on untrusted storages. Their scheme utilizes the RSAbased homomorphic 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, their protocol is not provably privacy preserving, and thus may leak user data information to the auditor. Juels *et al.* [11] describe 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 fixed a 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 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.

V. CONCLUSION

We propose a privacy-preserving public auditing system for data storage security in Cloud Computing. We utilize the homomorphic non-linear authenticator and random masking to guarantee 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 security. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are probably secure and highly efficient.

REFERENCES

[1] M. Arrington, "Gmail disaster: Reports of mass email deletions," Online at http://www.techcrunch.com/2006/ 12/28/gmail-disasterreports-of-mass-email-deletions/, December 2006.

[2] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors," Online at http://www.techcrunch.com/2008/07/10/ mediamaxthelinkup-closes-its-doors/, July 2008.

[3] Amazon.com, "Amazon s3 availability event: July 20, 2008," Online at http://status.aws.amazon.com/s3-20080720.html,2008.

[4] S. Wilson, "Appengine outage," Online at http://www. cio-weblog.com/50226711/appengine outage.php, June 2008.

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

[6] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. of CCS'07*, Alexandria, VA, October 2007, pp. 598–609.

[7] M. A. Shah, R. Swaminathan, and M. Baker, "Privacypreserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.

[8] 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.

[9] A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in *Proc. of CCS'07*, Alexandria, VA, October 2007, pp. 584–597.

[10] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, http://www. cloudsecurityalliance.org.

[11] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. of Asiacrypt 2008*, vol. 5350, Dec 2008, pp. 90–107.

[12] 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.

[13] 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.

[14] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In In proceedings of CRYPTO'04, volume 3152 of LNCS, pages 41–55. Springer-Verlag, 2004.

[15] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In Advances in Cryptology (CRYPTO'01), volume 2139 of LNCS, pages 213–229, 2001.

D. Srinivas Academic Carier start with DCME (thandra paparaya polytechnic, bobbli), Receved Batchlor Degree B.Tech from kakinada institute of eng and technology, kakinada, and Masters Degree M.tech from (GIET) Godavari Institute of Engg and Technology, Rajahamandry. Having 4 years of extensive experience in (KIET), presently working as a Assistant Professor. Having Research interest include in cloud computing.