ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3 , Issue 4 , April 2014

Security in Cloud Audit Using the Key Based Homomorphic Linear Authenticator

S.Muralitharan[1], M.Vijayaraj[2]

Assistant Professor, Department of Information Technology, Mohamed Sathak Engineering College, Kilakarai, TamilNadu, India [1]

Associate Professor &Head, Department of Information Technology, Mohamed Sathak Engineering College, Kilakarai, TamilNadu, India[2]

Abstract: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 auditability 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. 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, I propose a secure cloud storage system supporting privacy-preserving public auditing. I 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.

Keywords:Cloud Computing, Cryptographic Protocols, Data Storage, , Privacy-Preserving, Public Auditability

I. INTRODUCTION

CLOUD Computing has been envisioned as the next-generation information technology (IT) architecture for enterprises, 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 to the Cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc [2].

While 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. Examples of outages and security breaches of noteworthy cloud services appear from time to time [3]–[7]. Secondly, there do exist various motivations for CSP to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For examples, CSP might reclaim storage for monetary reasons by discarding data that

has not been or is rarely accessed, or even hide data loss incidents so as to maintain a reputation [8]–[10].
In short, although outsourcing data to the cloud is economically attractive for long-term large-scale data storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the successful deployment of the cloud architecture.

As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted [11]. In particular, simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the network. Besides, it is often insufficient to detect the data corruption only when accessing the data, as it does not give users correctness assurance for those unaccessed data and might be too late to recover the data loss or damage. Considering the large size of the outsourced data and the user's constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for the cloud users [10], [12]. Moreover, the overhead of using cloud storage should be minimized as much as possible, such that user does not need to perform too many operations to use the data (in additional to retrieving the data).

For example, it is desirable that users do not need to worry about the need to verify the integrity of the data before or after the data retrieval. Besides, there may be more than one user accesses the same cloud storage, say in an enterprise setting. For easier management, it is desirable that the cloud server only entertains verification request from a single designated party.

To fully ensure the data integrity and save the cloud users' computation resources as well as online burden, it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third party auditor (TPA) to audit the outsourced data when needed. The TPA, who has expertise and capabilities that users do not, can periodically check the integrity of all the data stored in the cloud on behalf of the users, which provides a much more easier and affordable way for the users to ensure their storage correctness in the cloud. Moreover, in addition to help users to evaluate the risk of their subscribed cloud data services, the audit result from TPA would also be beneficial for the cloud service providers to improve their cloud based service platform, and even serve for independent arbitration purposes [9]. In a word, enabling public auditing services will play an important role for this nascent cloud economy to become fully established, where users will need ways to assess risk and gain trust in the cloud.

Recently, the notion of public auditability has been proposed in the context of ensuring remotely stored data integrity under different system and security models [8], [10], [11], [13]. Public auditability allows an external party, in addition to the user himself, to verify the correctness of remotely stored data. However, most of these schemes [8], [10], [13] do not consider the privacy protection of users' data against external auditors. Indeed, they may potentially reveal user data information to the auditors. This severe drawback greatly affects the security of these protocols in Cloud Computing. From the perspective of protecting data privacy, the users, who own the data and rely on TPA just for the storage security of their data, do not want this auditing process introducing new vulnerabilities of unauthorized information leakage towards their data security [14]. Moreover, there are legal regulations, such as the US Health Insurance Portability and Accountability Act (HIPAA) [15], further demanding the outsourced data not to be leaked to external parties [9]. Exploiting data encryption before outsourcing [11] is one way to mitigate this privacy concern, but it is only complementary to the privacypreserving public auditing scheme to be proposed in this paper. Without a properly designed auditing protocol, encryption itself cannot prevent data from "flowing away" towards external parties during the auditing process. Thus, it does not completely solve the problem of protecting data privacy but just reduces it to the key management. Unauthorized data leakage still remains a problem due to the potential exposure of decryption keys.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3 , Issue 4 , April 2014

II. BACKGROUND

I consider a cloud data storage service involving three different entities, as illustrated in Fig. 1: 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 (I 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 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.

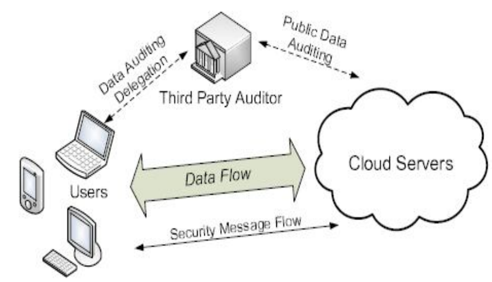

Fig. 1: The architecture of cloud data storage service

I consider the existence of a semi-trusted CS as [16] does. Namely, in most of time it behaves properly and does not deviate from the prescribed protocol execution. However, 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.

I 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. However, it harms the user if the TPA could learn the outsourced data after the audit. To authorize the CS to respond to the audit delegated to TPA's, 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. These authentication handshakes are omitted in the following presentation.

III. OBSERVATION

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.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3 , Issue 4 , April 2014

3) Privacy-preserving: to ensure that the TPA cannot derive users' data content from the information collected during the auditing process.
4) Batch auditing: to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously.
5) Lightweight: to allow TPA to perform auditing with minimum communication and computation overhead.

IV. PUBLIC AUDITING SYSTEM

A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof). KeyGen is a key generation algorithm that is run by the user to setup thescheme. SigGen is used by the user to generate verification metadata, which may consist of MAC,signatures, or other related information that will be used for auditing. GenProof is run by the cloud server to generate a proof of data storage correctness,while VerifyProof is run by the TPA to audit the proof from the cloud server.

Running a public auditing system consists of two phases, Setup and Audit:
• Setup: The user initializes the public and secret parameters of the system by executing KeyGen, and pre-processes the data file F by using SigGen to generate the verification metadata.The user then stores the data file F and the verification metadata at the cloud server, and deletes its local copy. As part of pre-processing, the user may alter the data file F by expanding it or including additional metadata to be stored at server.
• Audit: The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will derive a response message from a function of the stored data file F and its verification metadata by executing GenProof. The TPA then verifies the response via VerifyProof.

V. HOMOMORPHIC LINEAR AUTHENTICATOR

To effectively support public auditability without having to retrieve the data blocks themselves, the HLA technique [8], [10], [13] can be used. HLAs, like MACs, are also some unforgeable verification metadata that authenticate the integrity of a data block. The difference is that HLAs can be
aggregated. It is possible to compute an aggregated HLA which authenticates a linear combination of the individual data blocks.

At a high level, an HLA-based proof of storage system works as follow. The user still authenticates each element of F = (m1, · · · ,mn) by a set of HLAs ϕ. The cloud server stores {F, ϕ }. The TPA verifies the
cloud storage by sending a random set of challenge {i}. (More precisely, F, ϕ and {i} are all vectors, so {i} is an ordered set or {i, i} should be sent). The cloud server then returns μ = Pi i · mi and an aggregated authenticator (both are computed from F, ϕ and {i}) that is supposed to authenticate μ. Though allowing efficient data auditing and consuming only constant bandwidth, the direct adoption of these HLA-based techniques is still not suitable for our purposes. This is because the linear combination of blocks, μ = Pi i · mi, may potentially reveal user data information to TPA, and violates the privacypreserving guarantee. Specifically, if an enough number
of the linear combinations of the same blocks are collected, the TPA can simply derive the user's data content by solving a system of linear equations.

1. Retrieve file tag t, verify its signature, and quit if fail;
2. Generate a random challenge
chal = $\{(i,V_i)\}i\in I$;
3. Compute $\mu' = \sum_{i\in I} v_i m_i$, and also

$$\sigma = \prod_{i\in I} \sigma_i^{v_i}$$

4. Randomly pick r ← Zp, and compute

$R = e(u, v)^r$ and $\gamma = h(R)$;
5. Compute $\mu = r + \gamma\ \mu' \bmod p$;
6. Compute $\gamma = h(R)$, and then
verify $\{\mu, \sigma, R\}$ via Equation 1.

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 hide μ, 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 γ 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 $\{\sigma, \mu, R\}$ is independent of the number of sampled blocks c.

VI. CONCLUSION

In this paper, we propose a privacy-preserving public auditing system for data storage security in Cloud Computing. We utilize the homomorphic linear authenticatorand 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. 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 ourschemes are provably secure and highly efficient.

REFERENCES

[1] P. Mell and T. Grance, "Draft NIST working definition of cloud computing," Referenced on June. 3rd, 2009 Online at http://csrc.nist.gov/groups/SNS/cloudcomputing/index.html, 2009.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz,A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. UCB-EECS-2009-28, Feb 2009.

[3] 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.

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

[5] Amazon.com, "Amazon s3 availability event: July 20, 2008," Online at http://status.aws.amazon.com/s320080720.html,2008.

[6] S. Wilson, "Appengine outage," Online at http://www.cioweblog.com/50226711/Appengineoutage.php, June 2008.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing,"2009..

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

[14] 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.

[15] 104th United States Congress, "Health Insurance Portability and Accountability Act of 1996 (HIPPA)," Online at http:// aspe.hhs.gov/admnsimp/pl104191.htm, 1996.

[16] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained access control in cloud computing," in Proc. of IEEE INFOCOM'10, San Diego, CA, USA, March2010.

[17] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," J. Cryptology, vol. 17, no. 4, pp. 297–319, 2004.

[18] A. L. Ferrara, M. Greeny, S. Hohenberger, and M. Pedersen, "Practical short signature batch verification," in Proceedings of CT-RSA, volume 5473 of LNCS. Springer-Verlag, 2009, pp. 309–324.

[19] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc.Of SecureComm'08, 2008, pp. 1–10.

[20] C.Wang, Q.Wang, K. Ren, andW. Lou, "Ensuring data storage security in cloud computing," in Proc. of IWQoS'09, July 2009, pp. 1–9.

BIOGRAPHY

S.Muralitharan, Assistant professor, Department of Information Technology, Mohamed Sathak Engineering College, Kilakarai. Graduated in Master of Engineering in computer science and engineering.

M.Vijayaraj, Associate Professor & Head, Department of Information Technology, Mohamed Sathak Engineering College, Kilakarai. Research Scholar, Doing research in the area of V2I(Vehicle to Infrastructure)Networks.