Volume 4, Issue 10, October 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Privacy-Process Public Auditing & Documents Purity for Secure Cloud Storage

[1]K. Nagendra Nath*, [2]V. V. R. Manoj
[1]Student, M.Tech, CSEAssistant Professor, CSE, Dhanekula Institute of Engineering & Technology, Ganguru, Vijayawada, Andhrapradesh, India
[2]Dhanekula Institute of Engineering & Technology, Ganguru, Vijayawada, Andhrapradesh, India

Abstract -- Using Cloud Storage, users can remotely store their documents and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local documents storage and maintenance. However, the fact that users no longer have physical possession of the outsourced documents makes the documents 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 documents and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user documents 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 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. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.

Keywords - - privacy preserving, Documents Integrity,Cloud Storage.

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 [2]. 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 documents are being centralized or outsourced to the cloud. From users' perspective, including both individuals and IT enterprises, storing documents remotely to thecloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, While cloud computing makes these advantages more appealing than ever, it also brings new and challenging security threats toward users' outsourced documents. Since cloud service providers (CSP) are separate administrative entities, documents outsourcing is actually relinquishing user's ultimate control over the fate of their documents. As a result, the correctness of the documents 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 documents integrity [4]. Examples of outages and security breaches of noteworthy cloud services appear from time to time [5], [6], [7]. Second, there do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced documents status. For examples, CSP might reclaim storage for monetary reasons by discarding documents that have not been or are rarely accessed, or even hide documents loss incidents to maintain a reputation [8], [9], [10]. In short, although outsourcing documents to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on documents integrity and availability.

To address these problems, our work utilizes the technique of public key-based homomorphism linear authenticator (or HLA for short) [9], [13], [8], which enables TPA to perform the auditing without demanding the local copy of documents and thus drastically reduces the communication and computation overhead as compared to the straightforward documents auditing approaches. By integrating the HLA with random masking, our protocol guarantees that the TPA could not learn any knowledge about the documents content stored in the cloud server (CS) during the efficient auditing process. The aggregation and algebraic properties of the authenticator further benefit our design for the batch auditing. Specifically, our contribution can be summarized as the following three aspects:

1) We motivate the public auditing system of documents storage security in cloud computing and provide a privacy-preserving auditing protocol. Our scheme enables an external auditor to audit user's cloud documents without learning the documents content.

2) To the best of our knowledge, our scheme is the first to support scalable and efficient privacy-preserving public

storage auditing in cloud. Specifically, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA in a privacy-preserving manner.

3) We prove the security and justify the performance of our proposed schemes through concrete experiments and comparisons with the state of the art.

II. LITERATURE SURVEY

Ateniese et al. [9] are the first to consider public auditability in their "provable documents possession" PDP) model for ensuring possession of documents files on untrusted storages. They utilize the RSA-based homomorphic linear authenticators for auditing outsourced documents and suggest randomly sampling a few blocks of the file. However, among their two proposed schemes, the one with public auditability exposes the linear combination of sampled blocks to external auditor. When used directly, their protocol is not provably privacy preserving, and thus may leak user documents information to the external 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 documents 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.

Later, Bowers et al. [18] propose an improved framework for POR protocols that generalizes Juels' work. Dodis et al. [29] also give a study on different variants of PoR with private auditability. Shacham and Waters [13] design an improved PoR scheme built from BLS signatures [19] with proofs of security in the security model defined in [11]. Similar to the construction in [9], they use publicly verifiable homomorphic 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 is not privacy preserving due to the same reason as [9].

This problem, if not properly addressed, may impede the success of cloud architecture. As users no longer physically possess the storage of their documents, traditional cryptographic primitives for the purpose of documents security protection cannot be directly adopted [11]. In particular, simply downloading all the documents for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across thenetwork. Besides it is often insufficient to detect the documents corruption only when accessing the documents, as it does not give users correctness assurance for those unaccessed documents and might be too late to recover the documents loss or damage. Considering the large size of the outsourced documents and the user's constrained resource capability, the tasks of auditing the documents correctness in a cloud environment can be formidable and expensive for the cloud users [12], [8]. Moreover, the overhead of using cloud storage should be minimized as much as possible, such that a user does not need to perform too many operations to use the documents (in additional to retrieving the documents).

ARCHITECTURE

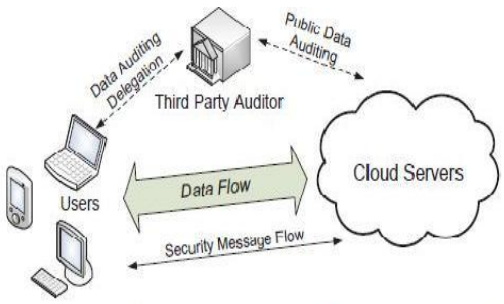


Fig. 1: The architecture of cloud data storage service

III. EXISTING SYSTEM

Since cloud service providers (CSP) are separate administrative entities, documents outsourcing is actually relinquishing user's ultimate control over the fate of their documents. As a result, the correctness of the documents 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 documents integrity.

DISADVANTAGES OF EXISTING SYSTEM

Although outsourcing documents to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on documents integrity and availability. This problem, if not properly addressed, may impede the success of cloud architecture.

As users no longer physically possess the storage of their documents, traditional cryptographic primitives for the purpose of documents security protection cannot be directly adopted. In particular, simply downloading all the documents 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 documents corruption only when accessing the documents, as it does not give users correctness assurance for those unaccessed documents and might be too late to recover the documents loss or damage.

IV. PROPOSED SYSTEM

To fully ensure the documents 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 documents storage, so that users may resort to an independent third party auditor (TPA) to audit the outsourced documents when needed. The TPA, who has expertise and capabilities that users do not, can periodically check the integrity of all the documents stored in the cloud on behalf of the users, which provides a much more easier and affordable way forthe users to ensure their storage correctness in the cloud. Moreover, in addition to help users to evaluate the risk of their subscribed cloud documents services, the audit result from TPA would also be beneficial for the cloud serviceproviders to improve their cloud based service platform, and evenserve for independent arbitration purposes. In a word, enabling public auditing services will play an important role for this nascent cloud economy to becomefully established, where users will need ways to assess risk and gain trust in the cloud.

A. ADVANTAGES OF PROPOSED SYSTEM:

1) We motivate the public auditing system of documents storage security in Cloud Computing and provide a privacy-preserving auditing protocol. Our scheme enables an external auditor to audit user's cloud documents without learning the documents content.

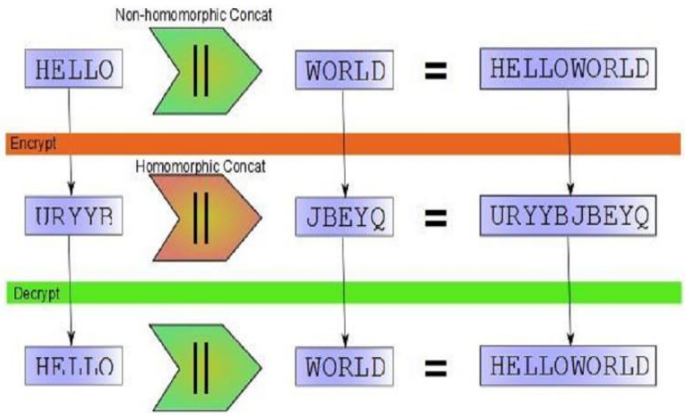
2) To the best of our knowledge, our scheme is the first to support scalable and efficient privacy preserving public storage auditing in Cloud. Specifically, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA in a privacy-preserving manner.

3) We prove the security and justify the performance of our proposed schemes through concrete experiments and comparisons with the state-of-the-art.

B. OTHER ADVANTAGES:

- PublicAuditability
- StorageCorrectness
- PrivacyPreserving
- Batchauditing
- Lightweight
- Setup Phase
- Audit Phase

Homomorphic CryptosystemsThey are ones here mathematical operations on the ciphertext have regulareffects on the plaintext. A very simple demonstration of the mathematicalconsistency required: A user sends a request to add the numbers 1 and 2, which areencrypted to become the numbers 33 and 54, respectively. The server in the cloudprocesses the sum as 87, which is downloaded from the cloud and decrypted tothe final answer, 3.A normal symmetric cipher -- DES, AES is not homomorphic.The RSA algorithm is homomorphic but only with respect to multiplication.



At first, the notion of processing data without having access to it may seemparadoxical, even logically impossible.To convince you that there is no fallacy , andto give you some intuition about the solution, let us consider an analogous problemin the physical world. Sita owns a jewelry store. She has raw precious materials gold, diamonds, silver, etc. She wants her workers to assemble into intricately designedrings and necklaces. But she distrusts her workers and assumes that they will stealher jewels if given the opportunity. In other words, she wants her workers toprocess the materials into finished pieces, without giving them access to thematerials. For that she uses a transparent impenetrable glove box, secured by a lockfor which only she has the key. She puts the raw precious materials inside the box, locks it, and gives it to a worker. Using the gloves, the worker assembles the ringor necklace inside the box. Since the box is impenetrable, the worker cannot get tothe precious materials, and users he might as well return the box to Sita, with thefinished piece inside. Sita unlocks the box with her key and extracts the ring ornecklace. In short, the worker processes the raw materials into a finished piece, without having true access to the materials. Of course, Sita's jewelry store is only ananalogy.

Implementation of FHE

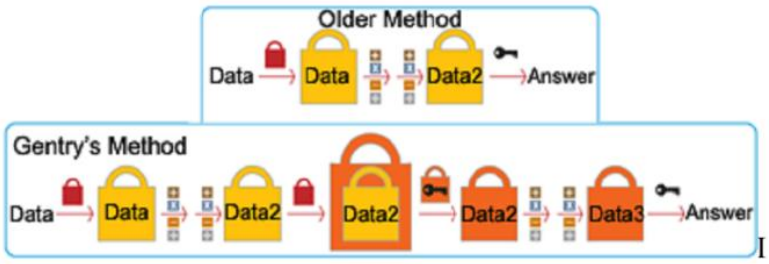


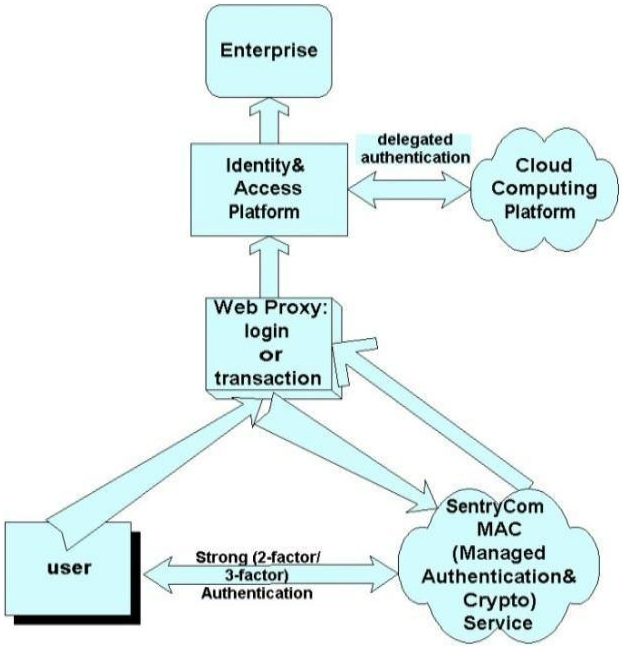
Figure . Craig Gentry implementation of FHE

In 2009 Craig Gentry of IBM has proposed the first encryption system "Fully Homomorphic" that evaluates an arbitrary number of additions and multiplications and thus calculate any type of function on encrypted data. The internal working of this adds another layer of encryption every few steps and uses an encrypted key to unlock the inner layer of scrambling. This decryption "refreshes" the data without exposing it, allowing an infinite number of computations on the same.

V. ALGORITHM

A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, and VerifyProof).
KeyGen: Key generation algorithm that is run by the user tosetup the scheme.
*SigGen:*Used by the user to generate verification Metadocuments,which may consist of MAC, signatures or other information used for auditing.
*GenProof:*run by the cloud server to generate a proof ofdocuments storage correctness.
*VerifyProof:*run by the TPA to audit the proof from thecloud server.

VI. FLOWCHART



VII. MODULES

A. Privacy-Preserving Public Auditing Module:

Homomorphic authenticators are unforgivable verification Metadocuments generated from individual documents blocks, which can be securely aggregated in such a way to assure an auditor that a linear combination of documents blocks is correctly computed by verifying only the aggregated authenticator. Overview to achieve privacy-preserving public auditing, we propose to uniquely integrate the homomorphic authenticator with random mask technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by a pseudo random function (PRF).The proposed scheme is as follows:

B. Batch Auditing Module:

With the establishment of privacy-preserving public auditing in Cloud Computing, TPA may concurrently handle multiple auditing delegations upon different users' requests. The individual auditing of these tasks for TPA can be tedious and very inefficient. Batch auditing not only allows TPA to perform the multiple auditing tasks simultaneously, but also greatly reduces the computation cost on the TPA side.

C. Documents Dynamics Module:

Supporting documents dynamics for privacy-preserving public risk auditing is also of paramount importance. Now we show how our main scheme can be adapted to build upon the existing work to support documents dynamics, including block leveloperations of modification, deletion and insertion. We can adopt this technique in our design to achieve privacy-preservingpublic risk auditing with support of documents dynamics.

VIII. CONCLUSION

In this paper, we utilize the homomorphic (Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on ciphertext and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintext. This is a desirable feature in modern communication system architectures. Homomorphic encryption would allow the chaining together of different services without exposing the data to each of those services, for example a chain of different services from different companies could 1) calculate the tax 2) the currency exchange rate 3) shipping, on a transaction without exposing the unencrypted data to each of those services.) 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.

REFERENCES

[1] CongWang ; Chow,S.S.M. ; QianWang ; KuiRen ;WenjingLou "Privacy_preserving Public Auditing for Secure CloudStorage", IEEE Transactions on Computers Volume: 62 , Issue: 2 2013 ,PP no : 362 - 375

[2] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM '10, Mar. 2010.

[3] P. Mell and T. Grance, "Draft NIST Working Definition of Cloud Computing," http://csrc.nist.gov/groups/SNS/cloudcomputing/ index.html, June 2009.

[4] 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," Technical Report UCB-EECS-2009-28, Univ. of California, Berkeley, Feb. 2009.

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

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

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

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

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

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

[11] 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.

[12] Cloud Security Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing," http://www.cloudsecurityalliance. org, 2009.

[13] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (Asiacrypt), vol. 5350, pp. 90-107, Dec. 2008.

[14] C. Wang, K. Ren, W. Lou, and J. Li, "Towards Publicly Auditable Secure Cloud Documents Storage Services," IEEE Network Magazine, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

[15] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop 104th United States Congress, "Health Insurance Portability and Accountability Act of 1996 (HIPPA)," http://aspe.hhs.gov/ admnsimp/pl104191.htm, 1996.

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

[17] 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.

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

[19] 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.

[20] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Documents Possession," Proc. Int'l Conf. Security and Privacy in Comm. Networks (SecureComm '08), pp. 1-10, 2008.

[21] C. Wang, Q. Wang, K. Ren, and W. Lou, "Towards Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Service Computing, vol. 5, no. 2, 220-232, Apr.-June 2012.

[22] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic Provable Documents Possession," Proc. ACM Conf. Computer and Comm. Security (CCS '09), pp. 213-222, 2009.

[23] R.C. Merkle, "Protocols for Public Key Cryptosystems," Proc. IEEE Symp. Security and Privacy, 1980.

[24] G. Ateniese, S. Kamara, and J. Katz, "Proofs of Storage from Homomorphic Identification Protocols," Proc. 15th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT),

[25] M. Bellare and G. Neven, "Multi-Signatures in the Plain Public- Key Model and a General Forking Lemma," Proc. ACM Conf. Computer and Comm. Security (CCS), pp. 390-399, 2006.

[26] Amazon.com, "Amazon Elastic Compute Cloud," http://aws. amazon.com/ec2/, 2009.

[27] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. Yau, "Efficient Provable Documents Possession for Hybrid Clouds," Cryptology ePrint Archive, Report 2010/234, 2010.

