ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Crucial Aggregate Cryptosystem Regarding Scalable Facts Giving In Foreign Safe-Keeping

Chennuru Uttam Kumar
M.Tech Student,
Department of CSE,
VITAM.

Jayalakshmi Nemana
Guide
Department of CSE,
VITAM.

Abstract:

In the cloud server, cloud users can store their data and high quality of services and applications.in the cloud environment they were used configurable computing resources, without the problem of local data storage and maintenance problems.Cloud users no longer have physical possession of the outsourced data makes the data integrity protection in cloud computing a formidable task.if we not providing the integrity protection should be able to just use the cloud storage as if it is local. Public auditability is mandatory for cloud storage.so users can approach to a third-party auditor (TPA) to check the integrity of outsourced data than they are not worry about the cloud protection. The auditing process should bring in no new vulnerabilities toward clouduser data and we can reduce the additional online burden to the 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.

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 data are 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, While cloud computing makes these advantages more appealing than ever, it also brings new and challenging security threats toward 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 [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 data status. For examples, CSP might reclaim storage for monetary reasons by discarding data that have not been or are rarely accessed, or even hide data loss incidents to maintain a reputation [8], [9], [10]. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on data 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 data and thus drastically reduces the communication and computation overhead as compared to the straightforward data auditing approaches. By integrating the HLA with random masking, our protocol guarantees that the TPA could not learn any knowledge about the data 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 data storage security in cloud computing and provide a privacy-preserving auditing protocol. Our scheme enables an external auditor to audit user's cloud data without learning the data 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.

LITERATURE SURVEY

Ateniese et al. [9] are the first to consider public auditability in their "provable data possession" PDP) model for ensuring possession of data files on untrusted storages. They utilize the RSA-based homomorphic linear authenticators for auditing outsourced data 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 data information to the external auditor.

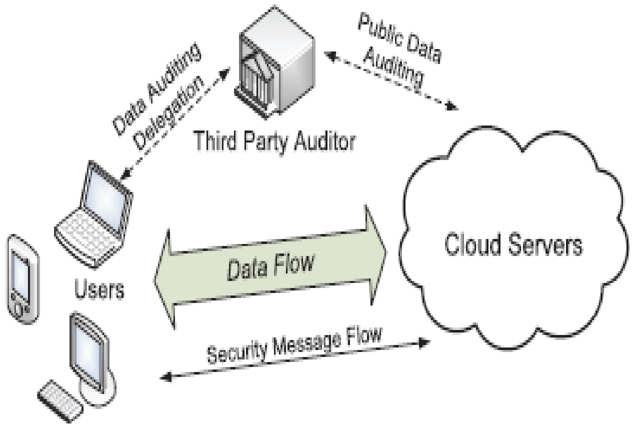
Juels et al. [11] describe a "proof of retrievability" (PoR) model, where spot-checking and error-correcting codes are used to ensure both "possession" and "retrievability" of data files on remote archive service systems. However, the number of audit challenges a user can perform is fixed a priori, and 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.

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 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 [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 data (in additional to retrieving the data).

ARCHITECTURE:



Existing System:

Cloud users no longer physically possess the storage of their data .traditional cryptographic primitives for the data security protection cannot appilicable directly .In general, 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.

The huge amount of outsourced data and the user's constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be problem and expensive for the cloud users. 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 data (in additional to retrieving the data).Cloudusers may not want to go through the complexity in verifying the data integrity. Besides, there may be more than one user accesses the same cloud storage. For easier management, it is desirable that cloud only entertains verification request from a single designated party.

Disadvantages:

1. In the cloudserver there is no long term security
2. In the existing there is no perfect integrity verification process.
3. Adversary issues are generated here
4. It takes more amount of time for recover the file.

Proposed System:

Cloudusers may approach 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, TPA provides much more security for storage correctness. 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. Public auditing services will play an important role for this cloud economy to become fully established; where users will need ways to assess risk and gain trust in the cloud.

Advantages:

1. Identifies the integrity file of information
2. Public auditing and data auditing gives the good advantages related to integrity here in implementation process.
3. In less amount of time recover the file the integrity file very easily here.

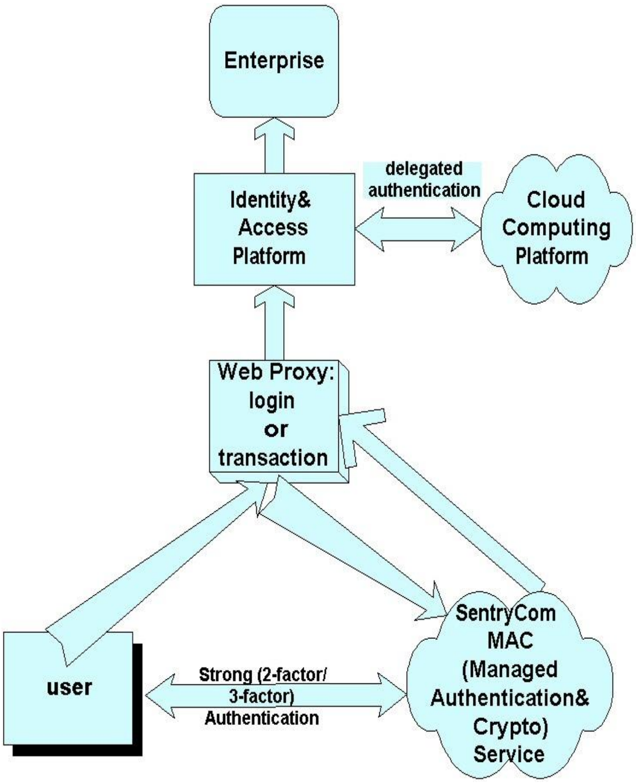
A public auditing scheme consists of four algorithms

A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof).

- KeyGen: key generation algorithm that is run by the user to setup the scheme
- SigGen: used by the user to generate verification metadata, which may consist of MAC, signatures or other information used for auditing

- GenProof: run by the cloud server to generate a proof of data storage correctness
- VerifyProof: run by the TPA to audit the proof from the cloud server

Flowchart:


Enterprise
Identity& Access Platform
delegated authentication
Cloud Computing Platform
Web Proxy: login or transaction
SentryCom MAC (Managed Authentication& Crypto) Service
user
Strong (2-factor/ 3-factor) Authentication

Setup and Audit:

Setup: The user initializes the public and secret parameters of the system by executing KeyGen, and preprocesses 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 preprocessing, 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 by executing Reproof using F and its verification metadata as inputs. The TPA then verifies the response via Verify Proof. Our framework assumes that the TPA is stateless; i.e., TPA does not need to maintain and update state between audits, which is a desirable property especially in the public auditing system. Note that it is easy to extend the framework above to capture a state ful auditing system, essentially by splitting the verification metadata into two parts which are stored by the TPA and the cloud server, respectively. Our design does not assume any additional property on the data file. If the user wants to have more error resilience, he can first redundantly encodes the data file and then uses our system with the data that has error correcting codes integrated.

Module Description:

Owner

Data owner primary responsibility is uploading the collection of files into cloudserver and cloud data auditing is the additional work for the data owner. Dataowner does not give correctness assurance for unaccessed data and might be too late to recover the data loss or damage. Considering the large size of the outsourced data and the owner’s constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for data owners.

Cloud server

Using homo morphic authenticators helps achieve a constant communication overhead for public audit ability. However, the direct extension of the approach to support data dynamics may have security and efficiency problems. Take block insertion, for example. In the original homomorphism authenticator schemes, to prevent a cloud server using the same authenticator to S

TPA

To fully ensure data security and save data owners’ computation resources, we propose to enable publicly auditable cloud storage services, where data owners can resort to an external third party auditor (TPA) to verify the outsourced data when needed. Third party auditing provides a transparent yet cost-effective method for establishing trust between data owner and cloud server. In fact, based on the audit result from a TPA, the released audit report would not only help owners to evaluate the risk of their subscribed cloud data services, but also be beneficial for the cloud service provider to improve their cloud based service platform

User

Cloud data storage provides dynamic and scalable storage services to users but also allows easy ondemand file sharing.Tpa will generate the Security message flow environment between the users and cloudserver. Tpa will always correctness assurance to the users. Achieving the same data dynamics support for public auditing services while maintaining file consistency is another future challenge.

CONCLUSION

In this paper, we propose a privacy-preserving public auditing system for data storage security in cloud comput- ing. We utilize the homomorphic 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.

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 multiuser setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly efficient. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of our design on both the cloud and the auditor side. We leave the full-fledged implementation of the mechanism on commercial public cloud as an important future extension, which is expected to robustly cope with very large scale data and thus encourage users to adopt cloud storage services more confidently.

REFERENCES

[1] 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.

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

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

[4] Cloud Security Alliance, "Top Threats to Cloud Computing," http://www.cloudsecurityalliance.org, 2010.

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

[6] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors," http:// www.techcrunch.com/2008/07/10/mediamaxthelinkup -closes- its-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 Data 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 Data 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 Data 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 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 Comput- ing 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. Cryptogra- phers' Track at the RSA Conf. 2009 on Topics in Cryptology (CT-RSA), pp. 309-324, 2009.

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

[22] 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.

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

[24] R.C. Merkle, “Protocols for Public Key Cryptosystems,” Proc. IEEE Symp. Security and Privacy, 1980.

[25] 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), pp. 319-333, 2009.