IJDCST @Sep-Oct, Issue- V-2, I-8, SW-28
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Improvised Protection in Secure Cloud Storage Using Public Appraisal Mechanism

Sudheer Mundra, G. Prasuna,
M-tech Student Scholar, Department of Computer Science Engineering, VRS & YRN College of Engineering & Technology, Chirala;Prakasam (Dt); Andhra Pradesh, India.
Assistant Professor, Department of Computer Science Engineering, VRS & YRN College of Engineering &Technology, Chirala; Prakasam (Dt), Andhra Pradesh, India.

Abstract — Using Cloud Storage, users can remotely store their data an d enjoy the on-demand high quality applications and services from a shared pool of configurable computing resour ces, 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, 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 per formance analysis show the proposed schemes are provably secure and highly efficient.

We aim to scale up the existing system by allowing the TPA to perform auditing for multiple users in a batch manner for better efficiency. This auditing by the TPA is done in a secured way and does not bring any new vulnerability into the existing system.

Index Terms — Data storage, privacy-preserving, public auditability, cryptographic protocols, cloud computing.

I. INTRODUCTION

CLOUD Computing has been envisioned as the next-generation information technology (IT) ar-chitecture for enterprises, due to its long list of un-precedented advantages in the IT history: on-demand self-service, ubiquitous network access, location in-dependent 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 chal-lenging 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 follow-ing reasons. First of all, although the infrastructures under the cloud are much more powerful and reli-able 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 trans-mission 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 privacy-preserving 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.

Therefore, how to enable a privacy-preserving third-party auditing protocol, independent to data encryption, is the problem we are going to tackle in this paper. Our work is among the first few ones to support privacy-preserving public auditing in Cloud Computing, with a focus on data storage. Besides, with the prevalence of Cloud Computing, a foreseeable increase of auditing tasks from different users may be delegated to TPA. As the individual auditing of these growing tasks can be tedious and cumbersome, a natural demand is then how to enable the TPA to efficiently perform multiple auditing tasks in a batch manner, i.e., simultaneously.

To address these problems, our work utilizes the technique of public key based homomorphic linear authenticator (or HLA for short) [8], [10], [13], which enables TPA to perform the auditing without demand-ing the local copy of data and thus drastically re-duces 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 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 pro-vide a privacy-preserving auditing protocol, i.e., our scheme enables an external auditor to audit user's outsourced data in the cloud without learning the data content.

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

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

In the present system, the TPA is able to audit only one user at a time. As cloud servers may concurrently handle multiple verification sessions from different clients, given K signatures on K distinct data files from K clients, it is more advantageous to aggregate all these signatures into a single short one and verify it at one time. To achieve this goal, we extend our scheme to allow for provable data updates and verification in a multi-client system. The signature scheme allows the creation of signatures on arbitrary distinct messages. Moreover, it supports the aggregation of multiple signatures by distinct signers on distinct messages into a single short signature, and thus greatly reduces the communication cost while providing efficient verification for the authenticity of all messages.

II. RELATED WORK

Ateniese *et al.* [8] are the first to consider public auditability in their defined "provable data possession" (PDP) model for ensuring possession of data files on untrusted storages. Their scheme utilizes the RSA-based homomorphic 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 re-mote 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 straight forward Merkle-tree construction for public PoRs, this approach only works with encrypted data. Dodis *et al.* [20] give a study on different variants of PoR with private auditability. Shacham *et al.* [13] design an improved PoR scheme built from BLS signatures [16] with full proofs of security in the security model defined in [11]. Similar

IJDCST @Sep-Oct, Issue- V-2, I-8, SW-28
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

to the construction in [8], they use publicly verifiable homomorphic linear authenticators that are built from provably secure BLS signatures. Based on the elegant BLS construction, a com-pact and public verifiable scheme is obtained. Again, their approach does not support privacy-preserving auditing for the same reason as [8]. Shah *et al.* [9], [14] 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 statefulness and bounded usage, which may potentially bring in online burden to users when the keyed hashes are used up.

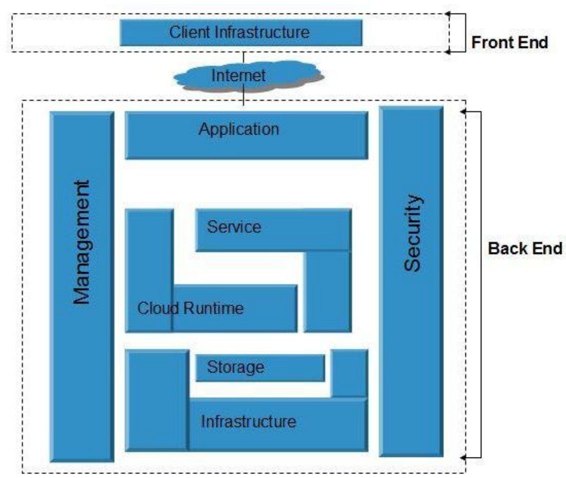
In other related work, Ateniese *et al.* [17] propose a partially dynamic version of the prior PDP scheme, using only symmetric key cryptography but with a bounded number of audits. In [18], Wang *et al.* consider a similar support for partial dynamic data storage in a distributed scenario with additional 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, Erway *et al.* [19] developed a skip lists based scheme to enable provable data possession with full dynamics support. However, the verification in these two proto-cols 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.

The Cloud Computing architecture

The Cloud Computing architecture comprises of many cloud components, each of them are loosely coupled. We can broadly divide the cloud architecture into two parts:

- Front End
- Back End

Each of the ends are connected through a network, usually via Internet. The following diagram shows the graphical view of cloud computing architecture:



FRONT END

Front End refers to the client part of cloud computing system. It consists of interfaces and applications that are required to access the cloud computing platforms, e.g., Web Browser.

BACK END

Back End refers to the cloud itself. It consists of all the resources required to provide cloud computing services. It comprises of huge data storage, virtual machines, security mechanism, services, deployment models, servers, etc.

IMPORTANT POINTS

It is the responsibility of the back end to provide built-in security mechanism, traffic control and protocols.

ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

The server employs certain protocols, known as middleware, helps the connected devices to communicate with each other.

Cloud based delivery

Software as a service (SaaS)

The software-as-a-service (SaaS) service-model involves the cloud provider installing and maintaining software in the cloud and users running the software from their cloud clients over the Internet (or Intranet). The users' client machines require no installation of any application-specific software - cloud applications run on the server (in the cloud). SaaS is scalable, and system administration may load the applications on several servers. In the past, each customer would purchase and load their own copy of the application to each of their own servers, but with the SaaS the customer can access the application without installing the software locally. SaaS typically involves a monthly or annual fee

Software as a service provides the equivalent of installed applications in the traditional (non-cloud computing) delivery of applications.

Software as a service has four common approaches:

1. single instance
2. multi instance
3. multi-tenant
4. flex tenancy

Development as a service (DaaS)

Development as a service is web based, community shared development tools. This is the equivalent to locally installed development tools in the traditional (non-cloud computing) delivery of development tools.

Platform as a service (PaaS)

Platform as a service is cloud computing service which provides the users with application platforms and databases as a service. This is equivalent to middleware in the traditional (non-cloud computing) delivery of application platforms and databases.

Infrastructure as a service (IaaS)

Infrastructure as a service is taking the physical hardware and going completely virtual (e.g. all servers, networks, storage, and system management all existing in the cloud). This is the equivalent to infrastructure and hardware in the traditional (non-cloud computing) method running in the cloud. In other words, businesses pay a fee (monthly or annually) to run virtual servers, networks, storage from the cloud. This will mitigate the need for a data center, heating, cooling, and maintaining hardware at the local level.

Cloud networking

Generally, the cloud network layer should offer:

- High bandwidth (low latency)
  Allowing users to have uninterrupted access to their data and applications.
- Agile network
  On-demand access to resources requires the ability to move quickly and efficiently between servers and possibly even clouds.
- Network security
  Security is always important, but when you are dealing with multi-tenancy, it becomes much more important because you're dealing with segregating multiple customers.

III. EXISTING SYSTEM

Although the existing schemes aim at providing integrity verification for different data storage systems, the problem of supporting both public auditability and data dynamics has not been fully addressed. How to achieve a secure and efficient design to seamlessly integrate these two important components for data storage service remains an open challenging task in Cloud Computing.

IV. PROPOSED SYSTEM

To achieve privacy-preserving public auditing[19], we propose to uniquely integrate the homomorphic linear authenticator with random masking technique. In our protocol, the linear combination of sampled 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 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 proposed by Boneh, Lynn and Shacham (hereinafter referred as BLS signature) [16].

In the existing system, the TPA is able to audit only one user at a time. As cloud servers may concurrently handle multiple verification sessions from different clients, given K signatures on K distinct data files from K clients, it is more advantageous to aggregate all these signatures into a single short one and verify it at one time. To achieve this goal, we extend our scheme to allow for provable data updates and verification in a multi-client system. The signature scheme allows the creation of signatures on arbitrary distinct messages. Moreover, it supports the aggregation of multiple signatures by distinct signers on distinct messages into a single short signature, and thus greatly reduces the communication cost while providing efficient verification for the authenticity of all messages.

V. CONCLUSION

In this paper, we propose a privacy-preserving public auditing system for data storage security in Cloud Computing. 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 multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. By implementing the above discussed scalability mechanism, we are able to efficiently and effectively performing the auditing process on the cloud Extensive analysis shows that our schemes 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/cloud-computing/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.

[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/s3-20080720.html, 2008.

[6] S. Wilson, "Appengine outage," Online at http://www. cio-weblog.com/50226711/appengine outage.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 un-trusted 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," Cryptol-ogy 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, http://www. cloudsecurityalliance.org.

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

[16] 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.

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

[17] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. of CCS'09, 2009, pp.213–222.

[18] Y. Dodis, S. P. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in TCC, 2009, pp. 109–127.

[19] Privacy-Preserving Public Auditing for Secure Cloud Storage, IEEE Transactions on Computers, 2013

AUTHOR'S PROFILE

SUDHEER MUNDRA received M.C.A from C.M.R College of Engineering &Technology, Kandlakoya(v), Hyderabad, Andhra Pradesh. and currently pursuing M.Tech in Computer Science Engineering at VRS & YRN College of Engineering &Technology, Chirala; Prakasam (Dt), Andhra Pradesh. His areas of interest incude Cloud Computing.

Mrs. G. PRASUNA is currently working as Associate professor in VRS & YRN College of Engineering & Technology, Chirala, AP, India. She completed her Master of Computer Applications from Andhra University, Vishakapatnam . And then completed her M.Tech in Computer Science and Engineering as her specialization from JNTUK, Kakinada. She is pursuing her Ph.D Degree in Computer Science and Engineering from Acharya Nagarjuna University, Guntur. Her research area includes Software Engineering. She has a teaching experience of 14 years. She published papers in 1 International Journals and 2 National Conference.