International Journal Of Engineering Research & Management Technology ISSN: 2348-4039

Email: editor@ijermt.org May- 2014 Volume 1, Issue 3 Website: ijermt.org

CLOUD DATA STORAGE PROTECTION FOR THE MASSES

C.Hari Har
M. Tech (Computer Engineering)
Veermata Jijabai Technological Institute
Mumbai, India

Prof. Mrs. Varshapriya J.N.
Veermata Jijabai Technological Institute
Mumbai, India

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 toward 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 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:Data storage, privacy preserving, public auditability, cloud computing.

INTRODUCTION

Cloud computing promises lower costs, rapid scaling, easier maintenance, and service availability anywhere, anytime, a key challenge is how to ensure and build confidence that the cloud can handle user data securely. A recent Microsoft survey found that "58 percent of the public and 86 percent of business leaders are excited about the possibilities of cloud computing. But more than 90 percent of them are worried about security, availability, and privacy of their data as it rests in the cloud. "This tension makes sense: users want to maintain control of their data, but they also want to benefit from the rich services that application developers can provide using that data. So far, the cloud offers little platform-level support or standardization for user data protection beyond data encryption at rest, most likely because doing so is nontrivial. Protecting user data while enabling rich computation requires

both specialized expertise and resources that might not be readily available to most application developers. Building in data-protection solutions at the platform layer is an
attractive option: the platform can achieve economies of scale by amortizing expertise costs and distributing sophisticated security solutions across different applications and their developers. 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 multiuser setting, where the TPA can perform multipleauditing 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

SERVICE ARCHITECTURE.

We consider a cloud data storage service involving three different entities, as illustrated in Fig. 1: the cloud user, who has large amount of data files to be stored in the cloud; the cloud server, which is managed by the cloud service provider to provide data storage service and has significant storage space and computation resources the third-party auditor, 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. As users no longer possess their data locally, it is of critical importance for users to ensure that their data are being correctly stored and maintained. To save the computation resource as well as the online burden potentially brought by the periodic storage correctness
verification, cloud users may resort to TPA for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from TPA.

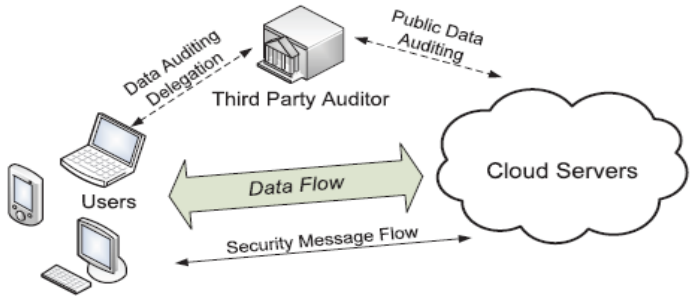


Fig. 1. The architecture of cloud data storage service.

DATA PROTECTION AS A SERVICE: Currently, users must rely primarily on legal agreements and implied economic and reputational harm as a proxy platform could help achieve a robust technical solution by making it easy for developers to write maintainable applications that protect user data in the cloud, there by providing the same economies of scale for
security and privacy as for computation and storage; and enabling independent verification both of the platform's operation and the runtime state of applications on it, so users can gain confidence that their data is being handled properly.

Much as an operating system provides isolation between processes but allows substantial freedom inside a process, cloud platforms could offer transparently verifiable partitions for applications that compute on data units, while still allowing broad computational latitude within those partitions.

DPaaS enforces fine-grained access control policies on data units through application confinement and information flow checking. It employs cryptographic protections at rest and offers robust logging and auditing to provide accountability. Crucially, DPaaS also directly addresses the issues of rapid development and maintenance.

ENCRYPTION. Traditional searchable encryption has been widely studied in the context of cryptography. In the realm of data protection, developers often view encryption as a kind of a silver bullet, but in reality, it's just a

tool albeit a powerful one to help achieve data protection properties. Although full-disk encryption (FDE)and computing on encrypted data have recently gained attention, these techniques have fallen short of answering all of the security and maintenance challenges mentioned earlier. FDE encrypts entire physical disks with a symmetric key, often in disk firmware, for simplicity and speed. Although FDE is effective in protecting private data in certain scenarios such as stolen laptops and backup tapes, the

concern is that it can't fulfill data protection goals in the cloud, where physical theft isn't the main threat. At the other end of the spectrum, Craig Gentry recently proposed the first realization of fully homo morphic encryption (FHE),which offers the promise of general computation on cipher texts. Basically, any function in plaintext can be transformed into an

work, but it doesn't know the data it's computing. Naturally, this property gives strong privacy guarantees when computing on private data, but the question of its practicality for general cloud applications still remains.

THREAT MODEL

In this paper, the data integrity threats toward users' datacan come from both internal and external attacks at CS. These may include: software bugs, hardware failures, bugs in the network path, economically motivated hackers, malicious or accidental management errors, etc. Besides, CS can be self-interested. For their own benefits, such as to maintain reputation, CS might even decide to hide these data corruption incidents to users. Using third-party auditing service provides a cost-effective method for usersto gain trust in cloud. We assume the TPA, who is in the business of auditing, is reliable and independent. However, it may harm the user if the TPA could learn the outsourced data after the audit.

DESIGN MODEL

In this paper, 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:. 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. Storage correctness: to ensure that there exists no cheating cloud server that can pass the TPA's audit
without indeed storing users' data intact. Privacy preserving: to ensure that the TPA cannot derive users' data content from the information collected during the auditing process. 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. Lightweight: to allow TPA to perform auditing with minimum communication and computation overhead.

PRELIMINARIES

We now introduce some necessary cryptographic backgroundfor our proposed scheme. Bilinear Map. Let GG1, GG2, and GGT be multiplicative cyclic groups of prime order p. Let g1 and g2 be generators of GG1and GG2, respectively. A bilinear map is a map e : GG1 _ GG2 ! GGT such that for all u 2 GG1, v 2 GG2 and a; b 2 ZZp,eðua; vbÞ ¼ eðu; vÞab. This bilinearity implies that for any u1, u2 2 GG1, v 2 GG2, eðu1 _ u2; vÞ ¼ eðu1; vÞ _ eðu2; vÞ. Of course, there exists an efficiently computable algorithm for computing e and the map should be nontrivial, i.e., e is non degenerate: e(g1,g2) not equal to 1.

ENSURING CLOUD DATA STORAGE

In cloud data storage system, users store their data in the cloud and no longer possess the data locally. Thus, the correctness and availability of the data files being stored on the distributed
cloud servers must be guaranteed. One of the key issues is to effectively detect any unauthorized data modification and corruption, possibly due to server compromise and/or random Byzantine failures. Besides, in the distributed case when such inconsistencies are successfully detected, to find which server the data error lies in is

also of great significance, since it can be the first step to fast recover the storage errors. To address these problems, our main scheme for ensuring cloud data storage is presented in this section. The first part of the section is devoted to a review of basic tools from coding theory that are needed in our scheme for file distribution across cloud servers. Then, the homomorphic token is introduced. The token computation function we are considering belongs to a family of universal hash function, chosen to preserve the homomorphic properties, which can be perfectly integrated with the verification of erasure-coded data Subsequently, it is also shown how to derive a challenge response protocol for verifying the storage correctness as well as identifying misbehaving servers. Finally, the procedure for file retrieval and error recovery based on erasure-correcting code is outlined.

A. FILE DISTRIBUTION PREPARATION

It is well known that erasure-correcting code may be usedto tolerate multiple failures in distributed storage systems. Incloud data storage, we rely on this technique to disperse thedata file F redundantly across a set of n = m+ k distributedservers. A (m + k, k) Reed-Solomon erasure-correcting codeis used to create k redundancy parity vectors from m datavectors in such a way that the original m data vectors can bereconstructed from any m out of the m + k data and parityvectors. By placing each of the m + k vectors on a differentserver, the original data file can survive the failure of any k ofthe m+k servers without any data loss, with a space overheadof k/m. For support of efficient sequential I/O to the originalfile, our file layout is systematic, i.e., the unmodified m datafilevectors together with k parity vectors is distributed acrossm+ k different servers.

Let **F** = (F1, F2, . . . , Fm) and Fi = (f1i, f2i, . . . , fli)T
(i ∈ {1, . . . ,m}), where l ≤ 2p − 1. Note all these blocks
are elements of GF(2p). The systematic layout with parity
vectors is achieved with the information dispersal matrix A,
derived from an m × (m+ k).

PRIVACY-PRESERVING PUBLIC AUDITING SCHEME

To achieve privacy-preserving public auditing, 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 by 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 datamcontent, 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 anew 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) Scheme details. Let GG1, GG2, and GGT be multiplicative cyclic groups of prime order p, and e : GG1.GG2 ! GGT be a bilinear map as introduced in preliminaries. Let g be a generator of GG2. H is a secure map-to-point hash function: f 0; 1g, GG1, which maps strings uniformly to GG1. Another hash function GGT ! Z Zp maps group element of GGT uniformly to Z Zp. Our scheme is as follows: Setup Phase: The cloud user runs Key Gen to generate the public and secret parameters. Specifically, the user chooses a random signing key pair spk; ssk, a random x ZZp, a random element u GG1, and computes v gx. The secret parameter is sk x; ssk and the public parameters .

.

SECURITY AND PRIVACY CHALLENGES

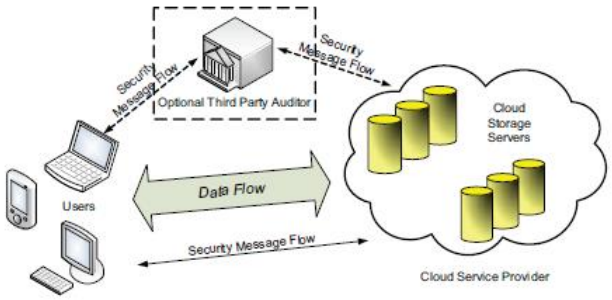
It's impossible to develop a single data-protection solutionfor the cloud because the term means too many different things. Any progress must first occur in a particular domain accordingly, our work focuses on an important class of widely used applications that includes e-mail, personal financial management, social networks, and business tools such as word processors and spreadsheets. The following criteria define this class of applications: provide services to a large number of distinct end users, as opposed to bulk data processing or workflow management for a single entity; use a data model consisting mostly of sharable units, where all data

objects have access control lists (ACLs) with one or more users; and developers could run the applications on a separate,computing platform that encompasses the physical infrastructure, job scheduling, user authentication, and the base software environment, rather than implementing the platform themselves. Overly rigid security is as detrimental to cloud service value as inadequate security. A primary challenge in designing a platform-layer solution useful to many applications is ensuring that it enables rapid development and
maintenance. To ensure a practical solution, we considered the following goals relating to data protection as well as ease of development and maintenance: Integrity. The user's stored data won't be corrupted. Privacy. Private data won't be leaked to any unauthorized entity. Access transparency. Logs will clearly indicate who or what accessed any data. Ease of verification. Users will be able to easily verify what platform or application code is running, as well as whether the cloud has strictly enforced their data's privacy policies. Rich computation. The platform will allow efficient rich computations on sensitive user data. Development and maintenance support. Because they face a long list bugs to find and fix, frequent software upgrades, continuoususage pattern changes, and user demand for high performance developers will receive both development and maintenance support.

PROPOSED SYSTEM:

In this ,our public auditing scheme which provides a complete outsourcing solution of data not only the data itself, but also its integrity checking. After introducing notations and brief preliminaries, we start from an overview of our public auditing system and discuss two straightforward schemes and their demerits. Then, we present our main scheme and show how to extent our main scheme to support batch auditing for the TPA upon delegations from multiple users. Finally, we discuss how to generalize our privacy-preserving public auditing scheme and its support of data dynamics.

ALGORITHM 1 Token Pre-computation



1: procedure
2: Choose parameters l, n and function f, _;
3: Choose the number t of tokens;
4: Choose the number r of indices per verification;
5: Generate master key K p rp and challenge k chal;
6: for vector G(j), j ← 1, n do
7: for round i← 1, t do
8: Derive _i = fk chal (i) and k(i) p rp from KPRP .
9: Compute v(j)i =Pr q=1 _qi ∗ G(j)[_k(i)pr p (q)]
10: end for
11: end for
12: Store all the vi.s locally.
13: end procedure

ALGORITHM2 Correctness Verification and Error Localization

1: procedure CHALLENGE(i)
2: Recompute _i = fkchal (i) and k(i)
prp from KPRP ;
3: Send {_i, k(i)
prp} to all the cloud servers;
4: Receive from servers:
{R(j)i =Prq=1 _qi ∗ G(j)[_k(i)prp(q)]|1 ≤ j ≤ n}
5: for (j ← m + 1, n) do
6: R(j) ← R(j)−Prq=1 fkj (sIq,j)·_qi , Iq = _k(i)prp(q)
7: end for
8: if ((R(1)i , . . . ,R(mi) ·P==(R(m+1)i , . . . ,R(n)i)) then
9: Accept and ready for the next challenge.
10: else
11: for (j ← 1, n) do
12: if (R(j)i ! =v(j)i) then
13: return server j is misbehaving.
14: end if
15: end for
16: end if
17: end procedure

ALGORITHM 3 Error Recovery

1: procedure% Assume the block corruptions have been detected among % the specified r rows; % Assume s ≤ k servers have been identified misbehaving
2: Download r rows of blocks from servers;
3: Treat s servers as erasures and recover the blocks.
4: Resend the recovered blocks to corresponding servers.
5: end procedure.

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 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.

REFERENCE

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 CloudComputing,"http://csrc.nist.gov/groups/SNS/cloudcomputing/,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 Computing," 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-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 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 AuditableSecure 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, "Auditingto Keep Online Storage Services Honest," Proc. 11th USENIXWorkshop 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