International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue
1st International Conference on Advent Trends in Engineering, Science and Technology
"ICATEST 2015", 08 March 2015

Privacy-preserving public auditing for secure cloud storage

Kiran G,Dhapodkar , Prof.Kahkashan Siddavatam
[1] *Designation, Complete address, email-id, phone number (Font Size 10, Times New Roman, Italic)*
[2] *Designation, Complete address, email-id, phone number (Font Size 10, Times New Roman, Italic)*
[3] *Designation, Complete address, email-id, phone number* author[3] *(Font Size 10, Times New Roman, Italic)*

Abstract: Cloud computing is a relatively new concept that offers the potential to deliver scalable elastic services to many. The notion of pay-per use is attractive and in the current global recession hit economy it offers an economic solution to an organization's IT need. Cloud computing is on demand high quality services and application which can store the users data remotely. Users are able to use huge storage and the processing capabilities of the cloud.This type of services will reduce the burden of local data storage and maintenance. Many auditing schemes came into existence for data integrity verification which can ensure the storage inconsistencies if any known to end users. Users can use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud data storage security is of critical importance so encouraged people to use cloud storage services by providing secure environment. Along with the widespread benefits on cloud computing, however, security issues with cloud data storage are arising in terms of confidentiality, integrity and reliability.So that users can resort to an external audit party to check the integrity of outsourced data when needed. We have developed a TPA which can verify and maintain the storage correctness. In this paper we introduced the encryption and decryption of data to protect the data and also the TPA to perform audits for multiple users simultaneously and efficiently.

Keywords – Data storage, privacy-preserving, public auditability, cryptographic protocols, cloud computing.

1. INTRODUCTION

When you store your photos online instead of on your home computer, or use webmail or a social networking site, you are using a "cloud computing" service. If you are an organization, and you want to use, for example, an online invoicing service instead of updating the in house one you have been using for many years, that online invoicing service is a "cloud computing" service.

Cloud computing refers to the delivery of computing resources over the Internet. Instead of keeping data on your own hard drive or updating applications for your needs, you use a service over the Internet, at another location, to store your information or use its applications. Doing so may give rise to certain privacy Implications.

The present availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing have led to a growth in cloud computing.With the Development of Cloud Computing, Data Security becomes, more and more important in cloud computing. This paper analyzes the basic problem of cloud computing data security.

As the user of cloud computing can access and use tools or applications through a web browser without having to install them on their computers. Users can access data anywhere location and at any time. So 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. To detect the data corruption is often insufficient only when accessing the data , as it does not give any type assurity to user that the data is correctly stored. Data encryption exploiting before outsourcing. In this paper privacy preserving public auditing scheme to be proposed but it is only complementary. Without designed a properly auditing protocol, encryption itself cannot prevent data from "flowing away" towards external parties during the auditing process.

In this paper how to enable a privacy-preserving third-party auditing protocol, independent to data encryption. Without focusing on data storage, our first aim is to support privacy-preserving public auditing in Cloud Computing. Based on the audit result, TPA could release an audit report, which would not only help users 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. Public auditability allows an

external party, in addition to the user himself, to verify the correctness of remotely stored data. Exploiting data encryption before outsourcing. In this we utilizes the technique of public key based homomorphic authenticator which enables TPA to perform the auditing without demanding the local copy of data and it reduces the communication and computation overhead.

No.	Item	Particulars	Result	Remarks
1.				
2.				
3.				

Table 1: Name of the table

2. proposed system

To fully ensure the privacy preserving and data integrity, We consider a cloud data storage service involving three different entities as illustrated in Fig. 1: 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 cloud user (U), who has large amount of data files to be stored in the cloud; the cloud server (CS), which is managed by cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources (we will not differentiate CS and CSP hereafter.); the third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service security 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. The users may resort to TPA for ensuring the storage security of their outsourced data, while hoping to keep their data private from TPA. We consider the existence of a semi-trusted CS as [14] does. Namely, in most of time it behaves properly and does not deviate from the prescribed protocol execution. However, during providing the cloud data storage based services, for their own benefits the CS might neglect to keep or deliberately delete rarely accessed data files which belong to ordinary cloud users. Moreover, the CS may decide to hide the data corruptions caused by server hacks or Byzantine failures to maintain reputation. We assume the TPA, who is in the business of auditing, is reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process. TPA should be able to efficiently audit the cloud data storage without local copy of data and without bringing in additional on-line burden to cloud users.

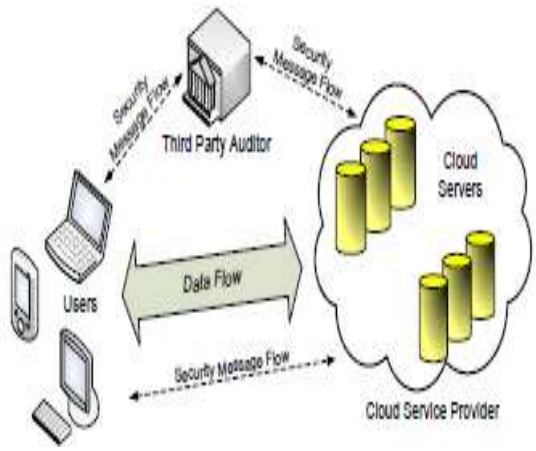

Fig. 1: The architecture of cloud data storage services

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

3. design goals

To enable privacy-preserving public auditing for cloud data storage under the aforementioned model, our protocol design should achieve the following security and performance guarantee:

1) Public auditability: to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional on-line burden to the cloud users; 2) Storage correctness: to ensure that there exists no cheating cloud server that can pass the audit from TPA without indeed storing users' data intact; 3) Privacy-preserving: to ensure that there exists no way for TPA to derive users' data content from the information collected during the auditing process; 4) Batch auditing: to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously; 5) Lightweight: to allow TPA to perform auditing with minimum communication and computation overhead.

Fig. 1 : Name of the figure

4. the proposed scheme

In this section presents 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.

A. Notation and Preliminaries :

• F – the data file to be outsourced, denoted as a sequence of n blocks m1,...,mn ∈ Zp for some large prime p.

• fkey(·) – pseudorandom function (PRF), defined as: {0,1}∗×key → Zp.

• πkey(·) – pseudorandom permutation (PRP), defined as: {0,1}log2(n) ×key →{ 0,1}log2(n).

• MACkey(·) – message authentication code (MAC) function, defined as: {0,1}∗×key →{ 0,1}l.

• H(·), h(·) – map-to-point hash functions, defined as: {0,1}∗ → G, where G is some group.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar.

We now introduce some necessary cryptographic background for our proposed scheme.

Bilinear Map Let G1, G2 and GT be multiplicative cyclic groups of prime order p. Let g1 and g2 be generators of G1 and G2, respectively. A bilinear map is a map e : G1 ×G2 → GT with the following properties [15]: 1) Computable: there exists an efficiently computable algorithm for computing e; 2) Bilinear: for all u ∈ G1, v ∈ G2 and a, b ∈ Zp, e(ua, vb) = e(u, v)ab; 3) Non-degenerate: e(g1, g2) = 1; 4) for any u1, u2 ∈ G1, v ∈ G2, e(u1u2, v) = e(u1, v) · e(u2, v).

B. Definitions and Framework

A similar definition of previously proposed schemes in the context of remote data integrity checking [9], [11], [13] and adapt the framework for our privacy preserving public auditing system.

In public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof). KeyGen is a key generation algorithm that is run by the user to setup the scheme. SigGen is used by the user to generate verification metadata, which may consist of digital signatures. GenProof is run by the cloud server to generate a proof of data storage correctness, while VerifyProof is run by the TPA to audit the proof. This are the all working of algorithm which is done their job.

Running a public auditing system consists of two phases, Setup and Audit:

Setup: The user initializes the public and secret parameters of the system by executing KeyGen, and 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 GenProof using F and its verification metadata as inputs. The TPA then verifies the response via VerifyProof.

Note that in our design, we do not assume any additional property on the data file, and thus regard error-correcting codes as orthogonal to our system.

In this framework we 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.

C. If the user wants to have more error-resiliency, he/she can first redundantly encode the data file and then provide us with the data file that has error correcting codes integrated. The Basic Schemes

- This is privacy-preserving as long as it is impossible Before giving our main result, in this we study two classes of schemes as a warm-up. The first one is a MAC-based which suffers from undesirable systematic demerits – bounded usage and stateful verification, which may pose additional online burden to users, in a public auditing setting. This somehow also shows that the auditing problem is still not easy to solve even we have introduced a TPA. The second one is a system based on homomorphic linear authenticators (HLA), which covers many recent proof of storage systems. We will identify the reason why all existing HLA-based systems are not privacy-preserving. The analysis of these basic schemes leads to our main result, which overcomes all these drawbacks. Our main scheme to be presented is based on a specific HLA scheme.

- MAC-based Solution : To make use of MAC to authenticate the data their are two possible ways. The important way is just uploading the data blocks with their MACs to the server, and sends the corresponding secret key sk to the TPA. After that the TPA can randomly retrieve blocks with their MACs and check the correctness via sk. Other than the high (linear in the sampled data size) communication and computation complexities, the TPA requires the knowledge of the data blocks for verification. To overcome the requirement of the data in TPA verification, one may restrict the verification to just consist of equality checking. The idea is as follows: Before data outsourcing, the cloud user chooses s random message authentication code keys $\{sk\tau\}1\leqslant \tau \leqslant s$, pre-computes s (deterministic) MACs, $\{MACsk\tau(F)\}1\leqslant \tau \leqslant s$ for the whole data file F, and publishes these verification metadata (the keys and the MACs) to TPA. The TPA can reveal a secret key $sk\tau$ to the cloud server and ask for a fresh keyed MAC for comparison in each to recover F in full given $MACsk\tau$ (F) and $sk\tau$.
- HLA-based Solution: To effectively support public auditability without having to retrieve the data blocks themselves, the HLA technique [9], [13], [8] can be used.
- HLAs, like MACs, are also some unable to be forged verification metadata that authenticate the integrity of a data block. The difference is that HLAs can be aggregated. It is possible to compute an aggregated HLA which authenticates a linear combination of the individual data blocks.
- Our design makes use of a public key based HLA, to equip the auditing protocol with public auditability.

D. 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 the 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. From another point of view, 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 occurring of the randomness. Our design makes use of a public key based HLA, to equip the auditing protocol with public auditability.

5. evaluation

Security Proofs

We evaluate the security of the proposed scheme by analyzing its fulfillment of the security guarantee, namely, the storage correctness and privacy preserving. We start from the single user case, where our main result is originated. Then we show the security guarantee of batch auditing for the TPA in multi-user setting.

1) Storage Correctness Guarantee: We need to prove that the cloud server can not generate valid response toward TPA without faithfully storing the data ,as captured by Theorem 1.

Theorem 1: If the cloud server passes the Audit phase, then it must indeed possess the specified data intact as it is.

Proof: The proof consists of two steps. First, we show that there exists an extractor of μ' in the random oracle model. Once a valid response $\{\sigma,\mu'\}$ are obtained, the correctness of this statement follows from Theorem 4.2 in [13]. The extractor controls the random oracle $h(\cdot)$ and answers the hash query issued by the cloud server, which is treated as an adversary here. For a challenge $\gamma = h(R)$ returned by the extractor, the cloud server outputs $\{\sigma, \mu, R\}$ such that the following equation holds.

Next, we show that if the response$\{\sigma,\mu,R\}$is valid, where $\mu = \mu+rh(R)$ and $R =(v2)r$, then the underlying μmust be valid too. This can be derived immediately from the collision free property of hash function $h(\cdot)$ and determinism of discrete exponentiation.

Now, the cloud server is treated as an adversary. The extractor controls the random oracle $h(\cdot)$ and answers the hash query issued by the cloud server. For a challenge $\gamma = h(R)$ returned by the extractor, the cloud server outputs $\{\sigma,\mu,R\}$ such that the following equation holds.

$$R \cdot e(\sigma\gamma,g) = e((sc\ Y\ i=s1H(Wi)vi)\gamma \cdot u\mu,v). \quad (1)$$

Suppose that our extractor can rewind a cloud server in the execution of the protocol to the point just before the challenge h(R) is given. Now the extractor sets h(R) to be $\gamma^* \_= \gamma$. The cloud server outputs $\{\sigma, \mu^*,R\}$ such that the following equation holds.

$$R \cdot e(\sigma\gamma*,g) = e((sc\ Y\ i=s1H(Wi)vi)\gamma* \cdot u\mu*,v). \quad (4)$$

Finally, we show that the validity of μimplies the correctness of $\{mi\}i\in I$ where $\mu = i\in I\ \nu imi$. Here we utilize the small exponent (SE) test technique of batch verification in [17]. Because $\{\nu i\}$ are picked up randomly by the TPA in each Audit phase, $\{\nu i\}$ can be viewed similarly as the random chosen exponents in the SE test [17]. Therefore, the correctness of individual sampled blocks is ensured. All above sums up to the storage correctness guarantee.

Privacy Preserving Guarantee:

We want to make sure that TPA cannot derive users' data content from the information collected during auditing process. This is equivalent to prove the

Theorem 2. Note that if μcan be derived by TPA, then {mi}i∈I can be easily obtained by solving a group of linear equations when enough combinations of the same blocks are collected.

Proof: We show the existence of a simulator that can produce a valid response even without the knowledge of μ′, in the random oracle model. Now, the TPA is treated as an adversary. Given a valid σ from the cloud server, firstly, randomly pick γ,μ from Zp, set R ← e((Qsc i=s1 H(Wi)vi)γ·uμ,v)/e(σγ,g). Finally, backpatch γ = h(R) since the simulator is controlling the random oracle h(·). We remark that this backpatching technique in the random oracle model is also used in the proof of the underlying scheme [13].

REFERENCES

[1].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.

[2].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.

[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," University of California, Berkeley, Tech. Rep. UCB-EECS-2009-28, Feb 2009.

[4].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.

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

[6]. Amazon.com, "Amazon s3 availability event: July 20, 2008," Online at http://status.aws.amazon.com/s3-20080720.html, 2008.

[7]. S. Wilson, "Appengine outage," Online at http://www.cio-weblog.com/50226711/appengine outage.php, June 2008.

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

[9].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.

[10]. Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, http://www. cloudsecurityalliance.org.

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

[12]. K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A high-availability and integrity layer for cloud storage," in Proc. of CCS'09, 2009.

[13]. T. Schwarz and E. L. Miller, "Store, forget, and check: Using algebraic signatures to check remotely administered storage," in Proc. of ICDCS'06, 2006.

[14]. M. Bellare and G. Neven, "Multi-signatures in the plain publickey model and a general forking lemma," in Proc. of CCS, 2006, pp. 390– 399.

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

[16]. Bethencourt A. Sahai, and B. Waters, "Ciphertext-policy attribute based encryption," in Proc. IEEE Symp. Security and Privacy, Oakland, CA, 2007.

[17]. Cong 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, 2010.

[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]. M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in Proc. of HotOS'07, 2007, pp. 1–6.

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

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

[22]. http://www.wikipedia.org.