Preserving Privacy for Secure and Outsourcing for Linear Programming in Cloud Computing

[1]A. Thirupathaiah, [2]K. VamsiRam, [3]B. A. Chakravarthy

[1,2,3]Dept. of CSE, St. Ann's College of Engineering & Technology, Andhra Pradesh, India

Abstract

Cloud computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. we utilize the public key based homomorphism authenticator and uniquely integrate it with random mask technique to achieve a privacy-preserving public auditing system for cloud data storage security while keeping all above requirements in mind. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously along with investigates secure outsourcing of widely applicable linear programming (LP) computations. In order to achieve practical efficiency, our mechanism design explicitly decomposes the LP computation outsourcing into public LP solvers running on the cloud and private LP parameters owned by the customer Extensive security and performance analysis shows the proposed schemes are provably secure and highly efficient.

Keywords

TAP, LP.OS

I. Introduction

Cloud Computing has been envisioned as the next-generation architecture of IT enterprise, 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 [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 into the Cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely into 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]. One fundamental advantage of the cloud paradigm is computation outsourcing, where the computational power of cloud customers is no longer limited by their resource-constraint devices. By outsourcing the workloads into the cloud, customers could enjoy the literally unlimited computing resources in a pay-per-use manner without committing any large capital outlays in the purchase of hardware and software and/or the operational overhead there in. Despite the tremendous benefits, outsourcing computation to the commercial public cloud is also depriving customers' direct control over the systems that consume and produce their data during the computation, which inevitably brings in new security concerns and challenges towards this promising computing model [3]. On the one hand, the outsourced computation workloads often contain sensitive information, such as the business financial records, proprietary research data, or personally identifiable health information etc. To combat against unauthorized information leakage, sensitive data have to be encrypted before outsourcing [4] so as to provide end to end data confidentiality assurance in the cloud and beyond. However, ordinary data encryption techniques in essence prevent cloud from performing any meaningful operation of the Underlying plaintext data [5], Examples include cloud service providers, for monetary reasons, reclaiming storage by discarding data that has not been or is rarely accessed, or even hiding data loss incidents so as to maintain a reputation [6]. In short, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, it does not 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 can not be directly adopted. Thus, how to efficiently verify the correctness of outsourced cloud data without the local copy of data files becomes a big challenge for data storage security in Cloud Computing. Note that simply downloading the data for its integrity verification is not a practical solution due to the expensiveness in I/O cost and transmitting the file across the network. Besides, it is often insufficient to detect the data corruption when accessing the data, as it might be too late for recover the data loss or damage. Considering the large size of the outsourced data and the user's constrained resource capability, the ability to audit the correctness of the data in a cloud environment can be formidable and expensive for the cloud users [7]. Therefore, to fully ensure the data security and save the cloud users computation resources, it is of critical importance to enable public audit ability for cloud data storage so that the users may resort to a third party auditor (TPA), who has expertise and capabilities that the users do not, to audit the outsourced data when needed. 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 [8]. In a word, enabling public risk auditing protocols 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 Cloud. Recently, the notion of public audit ability has been proposed in the context of ensuring remotely stored data integrity under different systems and security models [9]. Public audit ability allows an external party, in addition to the user himself, to verify the correctness of remotely stored data. However, most of these schemes [10] do not support the privacy protection of users' data against external auditors, i.e., they may potentially reveal user data information to the auditors. This 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 [11]. Moreover, there are legal regulations, such as the US Health

Insurance Portability and Accountability Act (HIPAA), further demanding the outsourced data not to be leaked to external parties [8]. Exploiting data encryption before outsourcing [12] 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 can not 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 one of managing the encryption keys. Unauthorized data leakage still remains a problem due to the potential exposure of encryption keys.

Specifically, we first formulate private data owned by the customer for LP problem as a set of matrices and vectors. This higher level representation allows us to apply a set of efficient Privacy-preserving problem transformation techniques, including matrix multiplication and affine mapping, to transform the original LP problem into some arbitrary one while protecting the sensitive input/output information. One crucial benefit of this higher level problem transformation method is that existing algorithms and tools for LP solvers can be directly reused by the cloud server. Although the generic mechanism defined at circuit level, e.g. [13], can even allow the customer to hide the fact that the outsourced computation is LP, we believe imposing this more stringent security measure than necessary would greatly affect the efficiency. To validate the computation result, we utilize the fact that the result is from cloud server solving the transformed LP problem. In particular, we explore the fundamental duality theorem together with the piece-wise construction of auxiliary LP problem to derive a set of necessary and sufficient conditions that the correct Customer LP problem Φ.

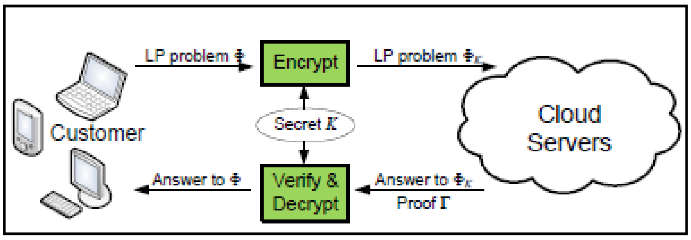

Fig. 1: Architecture of Secure Outsourcing Linear Programming Problems in Cloud Computing

Result must satisfy. Such a method of result validation can be very efficient and incurs close-to-zero additional overhead on both customer and cloud server. With correctly verified result, Customer can use the secret transformation to map back the desired solution for his original LP problem. We summarize our contributions as follows:

1. For the first time, we formalize the problem of securely outsourcing LP computations, and provide such a secure and practical mechanism design which fulfills input/output privacy, cheating resilience, and efficiency.
2. Our mechanism brings cloud customer great computation savings from secure LP outsourcing as it only incurs O(n_) for some 2 < _ ≤ 3 local computation overhead on the customer, while solving a normal LP problem usually requires more than O(n3) time [13].
3. The computations done by the cloud server shares the same time complexity of currently practical algorithms for solving the linear programming problems, which ensures that the use of cloud is economically viable.
4. The experiment evaluation further demonstrates the immediate practicality: our mechanism can always help customers achieve more than 30× savings when the sizes of the original LP problems are not too small, while introducing no substantial overhead on the cloud.

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 TPA to efficiently perform the multiple auditing tasks in a batch manner, i.e., simultaneously. To address these problems, our work utilizes the technique of public key based homomorphism authenticator [14], 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 homomorphism authenticator with random mask technique, our protocol guarantees that TPA could not learn any knowledge about the data content stored in the cloud server during the efficient auditing process.

II. Problem Statement

A. System and Threat Model

We consider a computation outsourcing architecture involving two different entities, as illustrated in fig. 1: the cloud customer, who has large amount of computationally expensive LP problems to be outsourced to the cloud; the cloud server (CS), which has significant computation resources and provides utility computing services, such as hosting the public LP solvers in a pay-per-use manner. The customer has a large-scale linear programming problem (to be formally defined later) to be solved. However, due to the lack of computing resources, like processing power, memory, and storage etc., he cannot carry out such expensive Computation locally. Thus, the customer resorts to CS for solving the LP computation and leverages its computation capacity in a pay-per-use manner. Instead of directly sending original problem, the customer first uses a secret K to map into some encrypted version K and outsources problem K to CS. CS then uses its public LP solver to get the answer of K and provides a correctness proof, but it is supposed to learn nothing or little of the sensitive information contained in the original problem description. After receiving the solution of encrypted problem K, the customer should be able to first verify the answer via the appended proof . If it's correct, he then uses the secret K to map the output into the desired answer for the original problem.

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 in the sense that in most of time it behaves properly and does not deviate from the prescribed protocol execution. While 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. However, any possible leakage of user's outsourced data towards TPA through the auditing protocol should be prohibited.

Note that to achieve the audit delegation and authorize CS to respond to TPA's audits, the user can sign a certificate granting audit rights to the TPA's public key, and all audits from the TPA are authenticated against such a certificate. These authentication handshakes are omitted in the following presentation.

B. 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 audit ability: 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.

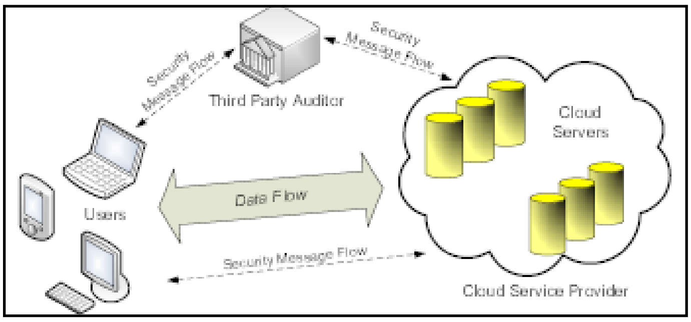

Fig. 2: Architecture of Cloud Storage Data Service

C. Notation and Preliminaries

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.

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

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, 16]: 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) 6= 1; 4) for any u1, u2 ∈ G1, v ∈ G2, e(u1u2, v) = e(u1, v)* e(u2, v).

III. Proposed Schemes.

In the introduction we motivated the public audit ability with achieving economies of scale for cloud computing. This section presents our public auditing scheme for cloud data storage security. We start from the overview of our public auditing system and discuss two straightforward schemes and their demerits. Then we present our main result for privacy-preserving public auditing to achieve the aforementioned design goals. We also show how to extent our main scheme to support batch auditing for TPA upon delegations from multi-users. Finally, we discuss how to adapt our main result to support data dynamics.

A. Definitions and Framework of Public Auditing System

We follow the similar definition of previously proposed schemes in the context of remote data integrity checking and adapt the framework for our privacy-preserving public auditing system.A 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 MAC, signatures, or other related information that will be used for auditing. 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 from the cloud server. Our public auditing system can be constructed from the above auditing scheme in two phases, Setup and Audit:

– Setup: The user initializes the public and secret parameters of the system by executing KeyGen, and pre-processes the data file F by using SigGen to generate the verification metadata. The user then stores the data file F at the cloud server, deletes its local copy, and publishes the verification metadata to TPA for later audit. As part of pre-processing, the user may alter the data file F by expanding it or including additional metadata to be stored at server.

singular matrix Q can be part of the secret key K. The customer can apply the matrix.

IV. Conclusion

In this paper, we propose a privacy-preserving public auditing system for data storage security in Cloud Computing, where TPA can perform the storage auditing without demanding the local copy of data. We utilize the homomorphic authenticator and random mask technique to guarantee that 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 along with the problem of securely outsourcing LP computations in cloud computing, and provide such a practical mechanism design which

fulfills input/output privacy, cheating resilience, and efficiency. By explicitly decomposing LP computation outsourcing into public LP solvers and private data, our mechanism design is able to explore appropriate security/efficiency tradeoffs via higher level LP computation than the general circuit
representation.

References

[1] P. Mell, T. Grance,"Draft nist working definition of cloud computing", Referenced on Jan. 23rd, 2010, [Online] Available: http://www.csrc.nist.gov/ groups/SNS/cloud-computing/index.html, 2010.

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

[3] C. Gentry,"Computing arbitrary functions of encrypted data", Commun. ACM, Vol. 53, No. 3, pp. 97–105, 2010.

[4] Sun Microsystems, Inc.,"Building customer trust in cloud computing with transparent security," 2009, [Online] Available: https://www.sun.com/offers/ details/sun transparency.xml.

[5] M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, E. H. Spafford,"Secure outsourcing of scientific computations", Advances in Computers, Vol. 54, pp. 216–272, 2001

[6] M. A. Shah, R. Swaminathan, M. Baker,"Privacy-preserving audit and extraction of digital contents", Cryptology ePrint Archive, Report 2008/186, 2008, [Online] Available: http:// eprint.iacr.org/.

[7] Q. Wang, C. Wang, J. Li, K. Ren, W. Lou,"Enabling public verifiability and data dynamics for storage security in cloud computing", in Proc. of ESORICS'09, Saint Malo, France, Sep. 2009.

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

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

[10] A. Juels, J. Burton S. Kaliski,"Pors: Proofs of retrievability for large files", in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 584–597.

[11] M. A. Shah, M. Baker, J. C. Mogul, R. Swaminathan, "Auditing to keep online storage services honest", in Proc. of HotOS'07. Berkeley, CA, USA: USENIX Association, 2007, pp1–6.

[12] 104th United States Congress,"Health Insurance Portability and Accountability Act of 1996, (HIPPA)", [Online] Available: http://aspe.hhs.gov/admnsimp/pl104191.htm, 1996, last access: July 16, 2009.

[13] D. Boneh, C. Gentry, B. Lynn, H. Shacham,"Aggregate and verifiably encrypted signatures from bilinear maps", in Proc. of Eurocrypt 2003, volume 2656 of LNCS. Springer-Verlag, 2003, pp. 416–432.

A.Thirupathaiah, received M.tech(IT) from Punjabi University in Nov 2003,Patiala.He worked as a Assoc. Professor in St.Anns college of engineering & Technology, in the period of (2008-2010).He is currently pursuing M.tech in computer science & engineering at St.Anns college of Engg. & technology which is affiliated under the JNTU, Kakinada. He is member of CSI and MISTE .He has total of 11 years of experience in Teaching field. His area of interests are Computer Networks , Network Security and Cloud Computing.

K.VamsiRam received the B.Tech from St.Anns college of engineering& Technology in INFORMATION TECHNOLOGY 2009.neering & Technology which is affiliated under JNTU Kakinada. His research interest includes Data Mining, Cloud omputing.

B.A.Chakravarthy received the B.Tech from St.Anns college of engineering& Technology in INFORMATION TECHNOLOGY 2010. His research interest includes Network Security, Cloud omputing.