Security Architecture Implemented in Cloud Computing using Linear Programming

[1]K. Ravindra, [2]K. Saritha
[1,2]Dept. of CSE, Avanthi Institute of Engg. & Tech., Cherukupally, Bhogapuram, Vizianagaram, AP, India

Abstract
Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (example:networks, servers , storage ,applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. To protect a customer confidential data in a cloud is become a major concern In this paper, we present a secure outsourcing mechanism for solving large-scale systems of linear programming computation in cloud. 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. In particular, by formulating private data owned by the customer for LP problem as a set of matrices and vectors, we are able to develop a set of efficient privacy-preserving problem transformation techniques, which allow customers to transform original LP problem into some arbitrary one while protecting sensitive input/output information. To validate the computation result,we further explore the fundamental duality theorem of LP computation and derive the necessary and sufficient conditions that correct result must satisfy. Such result verification mechanism is extremely efficient and incurs close-to-zero additional cost on both cloud server and customers.

Keywords
Security Architecture, Cloud Computing, Linear Programming

I. Introduction

Cloud Computing provides on-demand network access to a shared pool of configurable computing resources that can be rapidly deployed with great efficiency and minimal management overhead [1]. Under such unprecedented computing model, customers with computationally weak devices are no longer limited by the slow processing speed, memory, and other hardware constraints, but can enjoy the literally unlimited computing resources in the cloud through the convenient yet flexible pay-per-use manners [2].
Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the "software as a service" (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. Meanwhile, the increasing network bandwidth and reliable yet flexible network connections make it even possible that clients can now subscribe high-quality services from data and software that reside solely on remote data centers. Although envisioned as a promising service platform for the Internet, this new data storage paradigm in "Cloud" brings about many challenging design issues which have profound influence on the security and performance of the overall system. One of the biggest concerns with cloud data storage is that of data integrity verification at untrusted servers. For example, the storage service provider, which experiences Byzantine failures occasionally, may decide to hide the data errors from the clients for the benefit of their own. What is more serious is that for saving money and storage space the service provider might neglect to keep or deliberately delete rarely accessed data files which belong to an ordinary client. Consider the large size of the outsourced electronic data and the client's constrained resource capability, the core of the problem can be generalized as how can the client find an efficient way to perform periodical integrity verifications without the local copy of data files In all these works, great efforts are made to design solutions that meet various requirements: highscheme efficiency, stateless verification, unbounded use of queries and irretrievability of data, etc.
Cloud computing promises a cost effective enabling technology to outsource storage and massively parallel computations. However, existing approaches for provably secure outsourcing of data and arbitrary computations are either based on tamper-proof hardware or fully homomorphic encryption. The former approaches are not scaleable, while the latter ones are currently not efficient enough to be used in practice.
Cloud computing is "a model for enabling convenient, on-demand network access to a shared pool of computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."
Inherent to this model are privacy problems. By using services in the cloud clients lose control over their data. Current security mechanisms focus on protecting the data transfer to and from the service provider. But the threat of insider attacks keeps many potential customers from using cloud computing for critical applications.

II. Rellated Works

Despite the tremendous benefits, the fact that customers and cloud are not necessarily in the same trusted domain brings many security concerns and challenges towards this promising computation outsourcing model [3]. Firstly, customer's data that are processed and generated during the computation in cloud are often sensitive in nature, such as business financial records, proprietary research data, and personally identifiable health information etc [4]. While applying ordinary encryption techniques to these sensitive information before outsourcing could be one way to combat the security concern, it also makes the task of computation over encrypted data in general a very difficult problem [5]. Secondly, since the operational details inside the cloud are not transparent enough to customers [4], no guarantee is provided on the quality of the computed results from the cloud. For example, for computations demanding a large amount of resources, there are huge financial incentives for the cloud server to be "lazy" if the customer cannot tell the correctness of the answer. Besides, possible software/ hardware malfunctions and/ or outsider attacks might also affect the quality of the computed results. Thus, we argue that the cloud is intrinsically not secure from the viewpoint of customers. Without providing a mechanism for secure computation outsourcing, i.e., to protect the sensitive input and output information of the outsourced computing needs and to validate the integrity of the computation result, it would

be hard to expect cloud customers to turn over control of their computing needs from local machines to cloud solely based on its economic savings and resource flexibility.

Focusing on the engineering and scientific computing problems, this paper investigates secure outsourcing for widely applicable large-scale systems of Linear Equations (LE), which are among the most popular algorithmic and computational tools in various engineering disciplines that analyze and optimize real-world systems. By "large", we mean the storage requirements of the system coefficient matrix may easily exceed the available memory of the customer's computing device [6], like a modern portable laptop. In practice, there are many real world problems that would lead to very large scale and even dense systems of linear equations with up to hundreds of thousands [7-8] or a few million unknowns [9]. For example, a typical double-precision 50,000X50, 000 system matrix resulted from electromagnetic application would easily occupy up to 20 GBytes storage space, seriously challenging the computational power of these low-end computing devices. Because the execution time of a computer program depends not only on the number of operations it must execute, but also on the location of the data in the memory hierarchy of the computer [6], solving such large-scale problems on customer's weak computing devices can be practically impossible, due to the inevitably involved huge IO cost. Therefore, resorting to cloud for such computation intensive tasks can be arguably the only choice for customers with weak computational power, especially when the solution is demanded in a timely fashion.

III. Problem Statement

we consider a computation outsourcing architecture involving two different entities, as illustrated in Fig. 1: the cloud customer, who wants to solve some computationally expensive LE problems with his low-end devices; the Cloud Server (CS), which has significant computation resources to provide utility computing services.

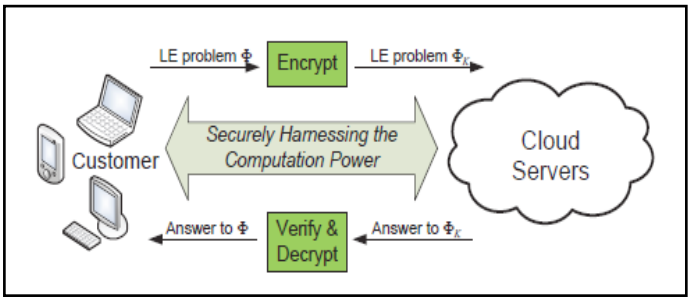


Fig. 1: Architecture of Secure Outsourcing Problems of LargeScale System of Liner Eqations in Cloud Computing

The customer has a large-scale system of LE problem Ax = b, denoted as ¢ = (A, b), to be solved. However, due to the lack of computing resources, like processing power, memory, and storage etc., he cannot carry out such expensive (O (n^P) (2 < P < 3)) computation locally. Thus, the customer resorts to Cloud Server (CS) for solving the LE computation and leverages its computation power in a pay-per-use manner. Instead of directly sending original problem ¢, the customer first uses a secret key K to map ¢ into some encrypted version ¢ K. Then, based on ¢ K, the customer starts the computation outsourcing protocol with CS, and harnesses the powerful resources of cloud in a privacy-preserving manner. The CS is expected to help the customer finding the answer of ¢K, but supposed to learn as little as possible on the sensitive information contained in ¢. After receiving the solution of encrypted problem ¢ K, the customer should be able to first verify the answer. If it's correct, he then uses the secret K to map the output into the desired answer for the original problem ¢.

IV. Existing System

There have been several existing approaches provided in order to overcome the secure outsourcing of the programming in cloud computing. Based on Yao's garbled circuits and Gentry's breakthrough work on Fully Homomorphic Encryption (FHE) [2-3], a general result of secure computation outsourcing has been shown viable in theory, where the computation is represented by an

Encrypted combinational Boolean circuit that allows to be evaluated with encrypted private inputs [7]. However, it's far from practical to apply this general mechanism to our daily computations, because the FHE operation is highly complex and also the pessimistic circuit sizes cannot be handled in practice when constructing original and encrypted circuits. In general this overhead motivates us to seek efficient solutions at higher abstraction levels when compared to the circuit representations for specific computation outsourcing problems. Even though some elegant designs have been proposed in the literature, on secure outsourcing of sequence comparisons, scientific computations and matrix multiplication [6] etc. still it is hardly possible to apply them directly in a practically efficient manner, especially for large problems.

In those approaches, either heavy cloud-side cryptographic computations, or multi-round interactive protocol executions, or huge communication complexities, are involved. In short, practically efficient mechanisms with immediate practices for secure computation outsourcing in cloud are still missing. There have been several existing approaches provided in order to overcome the secure outsourcing of the programming in cloud computing. Based on Yao's garbled circuits and Gentry's breakthrough work on Fully Homomorphic Encryption (FHE), in theory a general result of secure computation outsourcing has been shown viable, where the computation is represented by an encrypted combinational Boolean circuit that allows to be evaluated with encrypted private inputs. However, practical application of this general mechanism to our daily computations would be far from practical, due to the extremely high complexity of FHE operation as well as the pessimistic circuit sizes that cannot be handled in practice when constructing original and encrypted circuits. This overhead in general solutions motivates us to seek efficient solutions at higher abstraction levels than the circuit representations for specific computation outsourcing problems. Although some elegant designs on secure outsourcing of scientific computations, sequence comparisons, and matrix multiplication etc. have been proposed in the literature, it is still hardly possible to apply them directly in a practically efficient manner, especially for large problems.

This mechanism brings cloud customer great computation savings from secure LP outsourcing as it only incurs O(N^P) for some local computation 2<P<3 overhead on the customer, while solving a normal LP problem usually requires more than O(n^3) time.

V. Proposed System

To enable secure and practical outsourcing of LP under the aforementioned model, our mechanism design should achieve the following security and performance guarantees.

A. Correctness
Any cloud server that faithfully follows the mechanism must produce an output that can be decrypted and verified successfully by the customer.

B. Soundness
No cloud server can generate an incorrect output that can be decrypted and verified successfully by the customer with non-negligible probability.

C. Input/Output Privacy
No sensitive information from the customer's private data can be derived by the cloud server during performing the LP computation.

D. Efficiency
The local computations done by customer should be substantially less than solving the original LP on his own. The computation burden on the cloud server should be within the comparable time complexity of existing practical algorithms solving LP problems

The LP outsourcing scheme which provides a complete outsourcing solution for – not only the privacy protection of problem input/ output, but also its efficient result checking.

In this framework, the process on cloud server can be represented by algorithm ProofGen and the process on customer can be organized into three algorithms (KeyGen, ProbEnc, ResultDec). These four algorithms are summarized below and will be instantiated later.

KeyGen(1k) → {K}. This is a randomized key generation algorithm which takes a system security parameter k, and returns a secret key K that is used later by customer to encrypt the target LP problem.

ProbEnc(K,φ) → [φ K}. This algorithm encrypts the input tuple ,φ into, φK with the secret key K. According to problem transformation, the encrypted input ,φ K has the same form as,φ, and thus defines the problem to be solved in the cloud.

ProofGen(φ K) → {(y, Ļ)}. This algorithm augments a generic solver that solves the problem K to produce both the output y and a proof Ļ . The output y later decrypts to x, and Ļ is used later by the customer to verify the correctness of y or x.

ResultDec(K, φ, y,T) → {x,⊥}. This algorithm may choose to verify either y or x via the proof T. In any case, a correct output x is produced by decrypting y using the secret K. The algorithm outputs ⊥ when the validation fails, indicating the cloud server was not performing the computation faithfully.

Before presenting the details of our proposed mechanism, we study in this subsection a few basic techniques and show that the input encryption based on these techniques along may result in an unsatisfactory mechanism. However, the analysis will give insights on how a stronger mechanism should be designed.

1. Hiding Equality Constraints (A, b)
First of all, a randomly generated m × m non-singular matrix Q can be part of the secret key K. The customer can apply the matrix for the following constraints transformation,

Ax=b A'x=b' (1)

Where A'=QA B'=QB

Since we have assumed that A has full row rank, A′ must have full row rank. Without knowing Q, it is not possible for one to determine the exact elements of A. However, the null spaces of A and A′ remains the same, which may violate the security requirement of some applications. The vector b is encrypted in a perfect way since it can be mapped to an arbitrary b′ with a proper choice of Q.

2. Hiding Inequality Constraints (B)
The customer cannot transform the inequality constraints in the similar way as used for the equality constraints. This is because for an arbitrary invertible matrix Q, Bx ≥ 0 is not equivalent to QBx ≥ 0 in general.

To hide B, we can leverage the fact that a feasible solution to must satisfy the equality constraints. To be more specific, the feasible regions defined by the following two groups of constraints are the same.

$$\begin{cases} \mathbf{Ax} = \mathbf{b} \\ \mathbf{Bx} \geq 0 \end{cases} \Rightarrow \begin{cases} \mathbf{Ax} = \mathbf{b} \\ (\mathbf{B} - \lambda \mathbf{A})\mathbf{x} = \mathbf{B}'\mathbf{x} \geq 0 \end{cases} \quad (2)$$

λ where is a randomly generated n×m matrix in K satisfying that |B′| = |B − λ A| 6= 0 and _b = 0. Since the condition λ b = 0 is largely underdetermined, it leaves great flexibility to choose _ in order to satisfy the above conditions.

To enhance the security strength of LP outsourcing, we must be able to change the feasible region of original LP and at the same time hide output vector x during the problem input encryption. We propose to encrypt the feasible region of φ by applying an affine mapping on the decision variables x. This design principle is based on the following observation: ideally, if we can arbitrarily transform the feasible area of problem φ from one vector space to another and keep the mapping function as the secret key, there is no way for cloud server to learn the original feasible area information. Further, such a linear mapping also serves the important purpose of output hiding, as illustrated below.

Let M be an n × n non-singular matrix and r be an n × 1 vector. The affine mapping defined by M and r transforms x into y = $M^{-1}(x + r)$. Since this mapping is an one-to-one mapping, the LP problem φ in Eq. (2) can be expressed as the following LP problem of the decision variables y.

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{My} - \mathbf{c}^T\mathbf{r} \\ \text{subject to} \quad & \mathbf{AMy} = \mathbf{b} + \mathbf{Ar} \\ & \mathbf{BMy} \geq \mathbf{Br}. \end{aligned}$$

Using the basic techniques, this LP problem can be further transformed to

$$\begin{aligned} \text{minimize} \quad & \gamma \mathbf{c}^T\mathbf{My} \\ \text{subject to} \quad & \mathbf{QAMy} = \mathbf{Q}(\mathbf{b} + \mathbf{Ar}), \\ & \mathbf{BMy} - \lambda\mathbf{QAMy} \geq \mathbf{Br} - \lambda\mathbf{Q}(\mathbf{b} + \mathbf{Ar}). \end{aligned}$$

One can denote the constraints of above LP via Eq. (3):

$$\begin{cases} \mathbf{A}' = \mathbf{QAM} \\ \mathbf{B}' = (\mathbf{B} - \lambda\mathbf{QA})\mathbf{M} \\ \mathbf{b}' = \mathbf{Q}(\mathbf{b} + \mathbf{Ar}) \\ \mathbf{c}' = \gamma\mathbf{M}^T\mathbf{c} \end{cases} \quad (3)$$

If the following condition is hold

$$|\mathbf{B}'| \neq 0, \quad \lambda\mathbf{b}' = \mathbf{Br}, \quad \text{and} \quad \mathbf{b} + \mathbf{Ar} \neq 0, \quad (4)$$

then the LP problem φK = (A′,B′, b′, c′) can be formulated via Eq. (5),

$$\text{minimize} \quad \mathbf{c}'^T\mathbf{y} \quad \text{subject to} \quad \mathbf{A}'\mathbf{y} = \mathbf{b}', \mathbf{B}'\mathbf{y} \geq 0. \quad (5)$$

V. Conclusion
Cloud computing refers to the use and access of multiple server-based computational resources via a digital network (WAN, Internet connection using the World Wide Web, etc.). Cloud users may access the server resources using a computer or other device.

In cloud computing, applications are provided and managed by the cloud server and data is also stored remotely in the cloud configuration. Users do not download and install applications on their own device or computer; all processing and storage is maintained by the cloud server.

Security is the major problem in Cloud as its open to many users. In order to maintain the integrity of data in cloud we investigated the problem of securely outsourcing large-scale LP in cloud computing. It will 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. We develop problem transformation techniques that enable customers to secretly transform the original LP into some arbitrary one while protecting sensitive input/output information

References

[1] P. Mell, T. Grance (2010),"Draft nist working definition of cloud computing", [Online] Available: http://csrc.nist.gov/groups/SNS/cloud computing /index.html, 2010.

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

[3] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia,"A view of cloud computing", Communications of the ACM, Vol. 53, No. 4, pp. 50–58, April 2010.

[4] R. Cramer and I. Damg°ard, "Secure distributed linear algebra in a constant number of rounds," in Proc. of CRYPTO, 2001, pp. 119–136.

[5] V. Prakash, S. Kwon, R. Mittra,"An efficient solution of a dense system of linear equations arising in the method-ofmoments formulation", Microwave and Optical Technology Letters, Vol. 33, No. 3, pp. 196–200, 2002.

[6] K. Forsman, W. Gropp, L. Kettunen, D. Levine, J. Salonen,"Solution of dense systems of linear equations arising fromintegral-equation formulations", IEEE Antennas and PropagationMagazine, Vol. 37, No. 6, pp. 96–100, 1995.

[7] K. Nissim and E. Weinreb,"Communication efficient secure linear algebra", in Proc. of TCC, 2006, pp. 522–541

[8] J. R. Troncoso-Pastoriza, P. Comesa ˜ na, F. P´erez-Gonz´alez,"Secure direct and iterative protocols for solving systems of linear equations", in Proc. of 1st International Workshop on Signal Processing in the EncryptEd Domain (SPEED), 2009, pp. 122–141.

[9] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, W. Lou,"Fuzzy keyword search over encrypted data in cloud computing", in Proc. of IEEE INFOCOM'10 Mini-Conference, San Diego, CA, USA, March 2010.

[10] Y. Saad,"Iterative Methods for Sparse Linear Systems", 2nd ed. Society for Industrial and Applied Mathematics, 2003

[11] P. Mohassel, E. Weinreb,"Efficient secure linear algebra in thepresence of covert or computationally unbounded adversaries", in Proc. of CRYPTO, 2008, pp. 481–496.57, No. 3, pp. 240-282, 2006.

[12] C. Wang, N. Cao, J. Li, K. Ren, W. Lou,"Secure ranked keyword search over encrypted cloud data", in Proc. of ICDCS, 2010.

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



K. Ravindra was Working as Assistant Professor at Avanthi Institute of Engineering and Technology, Cherukupally, Tagarapuvalasa from August 2009 to till date. Worked as Programmer in TEQIP (Technical Education Quality Improvement Program) at Andhra University College of Engineering from December 2005 to August 2009. Worked as Lecturer at Bapuji Institute of Science and Technology P.G. College, Hyd. from July 2002 to November 2005. Presented a paper called "The Development Process of the Semantic Web and Web Ontology" in IJACSA International Journal for Advanced Computer Science Applications



K. Saritha was perusing MTech in Avanthi Institute of Engineering and Technology, Cherukupally, Bhogapuram, Vizianagaram, AP, India. She completed her BTech at PYDAH Emgineering College from 2006 to 2010.