International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 6, Nov - Dec. 2013 pp-3582-3586 ISSN: 2249-6645

Hiding Data Transmission with High Security in Cloud Computing with Cloud Server

Fahmida Begum[1], K. Bhargavi[2], Venkateswarlu Maninti[3]
[1]*Asso.Professor, Dept. of MCA, Dr. K.V Subba Reddy college of MCA, Kurnool. (Dt), A.P,*
[2]*Asst. Professor in MCA Dept. Palamuru University,*
[3]*Asst. Professor, Dept. of IT, Fishermen Training Institute, Salalah, Sultante of Oman,*

ABSTRACT : *Cloud computing economically enables the paradigm of data service outsourcing. However, to protect data privacy, sensitive cloud data has to be encrypted before outsourced to the commercial public cloud, which makes effective data utilization service a very challenging task. Focusing on engineering computing and optimization tasks, this paper investigates secure outsourcing of widely applicable linear programming 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. The resulting flexibility allows us to explore appropriate security/ efficiency tradeoff via higher-level abstraction of LP computations than the general circuit representation. The result verification mechanism is extremely efficient and incurs close-to-zero additional cost on both cloud server and customers. Extensive security analysis and experiment results show the immediate practicability of our mechanism design.*

Keywords: *Confidential data, Computation Outsourcing, Optimization, Cloud Computing.*

I. INTRODUCTION

Cloud Computing provides convenient on-demand network access to a shared pool of configurable computing resources that can be rapidly deployed with great efficiency and minimal management overhead. The 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 both hardware and software and/or the operational overhead therein. 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 so as to provide end to- end data confidentiality assurance in the cloud and beyond. On the other hand, the operational details inside the cloud are not transparent enough to customers .For example, for the computations that require a large amount of computing resources, there are huge financial incentives for the cloud to be "lazy" if the customers cannot tell the correctness of the output. Besides, possible software bugs, hardware failures, or even 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 workloads and to validate the integrity of the computation result, it would be hard to expect cloud customers to turn over control of their workloads from local machines to cloud solely based on its economic savings and resource flexibility. For practical consideration, such a design should further ensure that customers perform less amount of operations following the mechanism than completing the computations by themselves directly.

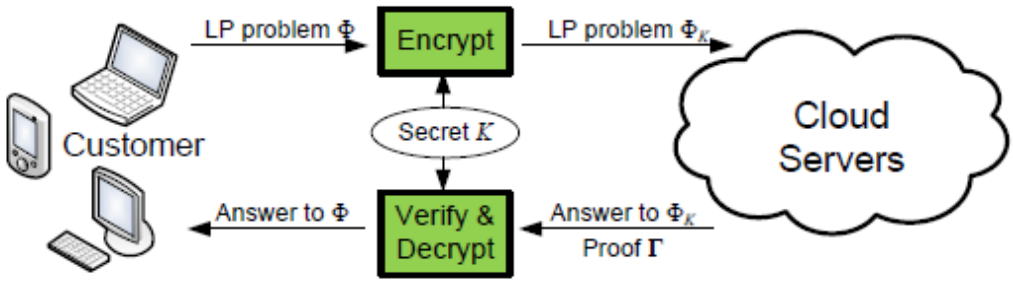

Fig1. Architecture of secure outsourcing linear programming problems in Cloud Computing

In this paper, we study practically efficient mechanisms for secure outsourcing of linear programming (LP) computations. Linear programming is an algorithmic and computational tool which captures the first order effects of various system parameters that should be optimized, and is essential to engineering optimization. It has been widely used in various engineering disciplines that analyze and optimize real-world systems, such as packet routing, flow control, power management of data centers, etc. Because LP computations require a substantial amount of computational power and usually involve confidential data, we propose to explicitly decompose the LP computation outsourcing into public LP solvers running on the cloud and private LP parameters owned by the customer. The flexibility of such a decomposition allows us to explore higher-level abstraction of LP computations than the general circuit representation for the practical efficiency.

II. PROBLEM DEFINITION

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 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, making the computation over encrypted data a very hard problem. On the other hand, the operational details inside the cloud are not transparent enough to customers. As a result, there do exist various motivations for cloud server to behave unfaithfully and to return incorrect results, i.e., they may behave beyond the classical semi honest model.

Fully homomorphic encryption (FHE) scheme, 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.

III. EXISTING SYSTEM:

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. 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. 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, making the computation over encrypted data a very hard problem. On the other hand, the operational details inside the cloud are not transparent enough to customers. As a result, there do exist various motivations for cloud server to behave unfaithfully and to return incorrect results, i.e., they may behave beyond the classical semi hones model. Thus, we argue that the cloud is intrinsically not secure from the viewpoint of customers.

Disadvantages of Existing System:

1) Fully homomorphic encryption (FHE) scheme, a general result of secure computation outsourcing has been shown viable in theory.
2) The cryptography and the theoretical computer science communities have made steady advances in "secure outsourcing expensive computations".
3) It is Semi-Honest Model.

IV. PROPOSED SYSTEM

Cloud computing economically enables the paradigm of data service outsourcing. However, to protect data privacy, sensitive cloud data has to be encrypted before outsourced to the commercial public cloud, which makes effective data utilization service a very challenging task. Focusing on engineering computing and optimization tasks, this paper investigates secure outsourcing of widely applicable linear programming 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. The resulting flexibility allows us to explore appropriate security/ efficiency tradeoff via higher-level abstraction of LP computations than the general circuit representation. The result verification mechanism is extremely efficient and incurs close-to-zero additional cost on both cloud server and customers. Extensive security analysis and experiment results show the immediate practicability of our mechanism design.

ADVANTAGES:

1) To combat against unauthorized information leakage, sensitive data have to be encrypted before outsourcing.
2) To provide end to- end data confidentiality assurance in the cloud and beyond.
3) The operational details inside the cloud are not transparent enough to customers.

V. MODULE DESCRIPTION

A) Mechanism Design Framework:

We propose to apply problem transformation for mechanism design. The general framework is adopted from a generic approach, while our instantiation is completely different and novel. 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,ø) → {øk}. This algorithm encrypts the input tuple ø into øk with the secret key K. According to problem transformation, the encrypted input øk has the same form as ø, and thus defines the problem to be solved in the cloud.

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

• ResultDec(K, ø, y, Γ) → {x,⊥}. This algorithm may choose to verify either y or x via the proof Γ. 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.

B) Basic Techniques:

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. Note that to simplify the presentation, we assume that the cloud server honestly performs the computation, and defer the discussion on soundness to a later section.

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 to Eq.

(2) for the following constraints transformation, $Ax = b \rightarrow A'x = b'$

where $A' = QA$ and $b' = Qb$.

C) Enhanced Techniques via Affine Mapping:

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 ø 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 ø 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.

D) RESULT VERIFICATION:

Till now, we have been assuming the server is honestly performing the computation, while being interested learning information of original LP problem.

However, such semi honest model is not strong enough to capture the adversary behaviors in the real world. In many cases, especially when the computation on the cloud requires a huge amount of computing resources, there exists strong financial incentives for the cloud server to be "lazy". They might either be not willing to commit service-level-agreed computing resources to save cost, or even be malicious just to sabotage any following up computation at the customers. Since the cloud server promises to solve the LP problem øk = (A′,B′, b′, c′), we propose to solve the result verification problem by designing a method to verify the correctness of the solution y of øk. The soundness condition would be a corollary thereafter when we present the whole mechanism in the next section. Note that in our design, the workload required for customers on the result verification is substantially cheaper than solving the LP problem on their own, which ensures the great computation savings for secure LP outsourcing.

The LP problem does not necessarily have an optimal solution. There are three cases as follows.

• Normal: There is an optimal solution with finite objective value.

• Infeasible: The constraints cannot be all satisfied at the same time.

• Unbounded: For the standard form, the objective function can be arbitrarily small while the constraints are all satisfied.

VI. PERFORMANCE ANALYSIS

A) THEORETIC ANALYSIS:

1) CUSTOMER SIDE OVERHEAD:

According to our mechanism, customer side computation overhead consists of key generation, problem encryption operation, and result verification, which corresponds to the three algorithms KeyGen, ProbEnc, and ResultDec, respectively. Because KeyGen and Result-Dec only require a set of random matrix generation as well as vector-vector and matrix-vector multiplication, the computation complexity of these two algorithms are upper bounded via O(n2). Thus, it is straight-forward that the most time consuming operations are the matrix-matrix multiplications in problem encryption algorithm ProbEnc. Since $m \leq n$, the time complexity for the customer local computation is thus asymptotically the same as matrix-matrix multiplication, i.e., $O(n\rho)$ for some $2 < \rho \leq 3$. In our experiment, the matrix multiplication is implemented via standard cubic-time method, thus the overall computation overhead is O(n3). However, other more efficient matrix multiplication algorithms can also be adopted, such as the Strassen's algorithm with time complexity O(n2.81) . In either case, the over all customer side efficiency can be further improved.

2) SERVER SIDE OVERHEAD:

For cloud server, its only computation overhead is to solve the encrypted LP problem Øk as well as generating the result proof Γ , both of which correspond to the algorithm ProofGen. If the encrypted LP problem ø belongs to normal case, cloud server just solves it with the dual optimal solution as the result proof Γ , which is usually readily available in the current LP solving algorithms and incurs no additional cost for cloud . If the encrypted problem øk does not have an optimal solution, additional auxiliary LP problems can be solved to provide a proof. Because for general LP solvers, executed at first to determine the initial feasible solution , proving the auxiliary LP with optimal solutions also introduces little additional overhead. Thus, in all the cases, the computation complexity of the cloud server is asymptotically the same as to solve a normal LP problem, which usually requires more than O(n3) time . Obviously, the customer will not spend more time to

encrypt the problem and solve the problem in the cloud than to solve the problem on his own. Therefore, in theory, the proposed mechanism would allow the customer to outsource their LP problems to the cloud and gain great computation savings.

3) EXPERIMENT RESULTS:

We now assess the practical efficiency of the proposed secure and verifiable LP outsourcing scheme with experiments. We implement the proposed mechanism including both the customer and the cloud side processes in Matlab and utilize the MOSEK optimization through its MATLAB interface to solve the original LP problem and encrypted LP problem øk. Both customer and cloud server computations in our experiment are conducted on the same workstation with an Intel Core 2 Duo processor running at 1.86 GHz with 4 GB RAM. In this way, the practical efficiency of the proposed mechanism can be assessed without a real cloud environment. We also ignore the communication latency between the customers and the cloud for this application since the computation dominates the running time as evidenced by our experiments.

VII. CONCLUSIONS

We formalize 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. We develop problem transformation techniques that enable customers to secretly transform the original LP into some arbitrary one while protecting sensitive input/output information. We also investigate duality theorem and derive a set of necessary and sufficient condition for result verification. Such a cheating resilience design can be bundled in the overall mechanism with close-to-zero additional overhead. Both security analysis and experiment results demonstrates the immediate practicality of the proposed mechanism.

LITERATURE REFERENCES

[1] P. Mell and T. Grance, "Draft nist working definition of cloud computing," Referenced on Jan. 23rd, 2010 Online at http://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 at 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 at https://www.sun.com/offers/ details/sun transparency.xml.
[5] M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, and E. H. Spafford, "Secure outsourcing of scientific computations," Advances in Computers, vol. 54, pp. 216–272, 2001.
[6] S. Hohenberger and A. Lysyanskaya, "How to securely outsource cryptographic computations," in Proc. of TCC, 2005, pp. 264–282.
[7] M. J. Atallah and J. Li, "Secure outsourcing of sequence comparisons," Int. J. Inf. Sec., vol. 4, no. 4, pp. 277–287, 2005.
[8] D. Benjamin and M. J. Atallah, "Private and cheating-free outsourcing of algebraic computations," in Proc. of 6th Conf. on Privacy, Security, and Trust (PST), 2008, pp. 240–245.
[9] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in Proc. Of CRYPTO'10, Aug. 2010.
[10] M. Atallah and K. Frikken, "Securely outsourcing linear algebra computations," in Proc. of ASIACCS, 2010, pp. 48–59.
[11] A. C.-C. Yao, "Protocols for secure computations (extended abstract)," in Proc. of FOCS'82, 1982, pp. 160–164.
[12] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proc of STOC, 2009, pp. 169–178.
[13] D. Luenberger and Y. Ye, Linear and Nonlinear Programming, 3rd ed. Springer, 2008.
[14] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in Proc. of ICDCS'10, 2010.
[15] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained access control in cloud computing," in Proc. of IEEE INFOCOM'10, San Diego, CA, USA, March 2010.
[16] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge University Press, 2004.
[17] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," in Proc. of STOC'87, 1987, pp. 1–6.
[18] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in Proc. Of EUROCRYPT'99, 1999, pp. 223–238.
[19] J. Li and M. J. Atallah, "Secure and private collaborative linear programming," in Proc. of CollaborateCom, Nov. 2006.
[20] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, "Delegating computation: interactive proofs for muggles," in Proc. of STOC, 2008.

AUTHOR PROFILES

1). Fahmida Begum did his MCA from Osmania University, and pursuing Ph.D from MJPRU , U. P. Her interested areas are mobile computing and cloud computing . I have 9 years experience of Teaching in various colleges. At present she is working as an Associate Professor in Dr. K.V Subba Reddy college of MCA, Kurnool.(Dt).

2). K.Bhargavi working as Asst.prof in MCA Dept. Palamuru University, she Completed M.Tech from NU. Her Research interests in security Issues in cloud Computing.

3). Mr.Venkateswarlu Maninti has received Master‟ s Degree from Jawaharlal Nehru Technological University,Anantapur, AP (India). He is currently Lecturer in Information Technology, Fishermen Training Institute, Salalah, Sultante of Oman. He served different levels as a Lecturer at SGPR Govt. Polytechnic, Kurnool, AP (India), Asst. Professor, Associate Professor and Head of the Department of Computer Science and Engineering of Dr.K.V.Subba Reddy College of Engineering for Women, Kurnool, AP (India). His research interest includes Denial of Services in Cloud Computing and Cloud Storage, Grid Computing Network and Wireless Networks & Database applications. He has published 5 peer reviewed national journal papers, 2 International journal papers. Published IEEE publications and organized national conferences/workshops and also having Microsoft Certification and IBM certification etc.