IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 5, Ver. I (Sep.- Oct. 2017), PP 22-26
www.iosrjournals.org

Secure and Practical Outsourcing of Linear Programming in Cloud Computing

*Naveen M, G Hemanth Kumar
Department of Studies in Computer Science, University of Mysore Manasagangotri-570006, Mysore, INDIA.
Corresponding Author: Naveen M

Abstract: *Cloud Computing has great potential of providing robust computational power to the society at reduced cost. It enables customers with limited computational resources to outsource their large computation workloads to the cloud, and economically enjoy the massive computational power, bandwidth, storage, and even appropriate software that can be shared in a pay-per-use manner. Despite the tremendous benefits, security is the primary obstacle that prevents the wide adoption of this promising computing model, especially for customers when their confidential data are consumed and produced during the computation. 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 fight 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. As a result, there do exist various motivations for cloud server to behave unfaithfully and to return incorrect results.*

Date of Submission: 24-08-2017 Date of acceptance: 08-09-2017

I. . Introduction

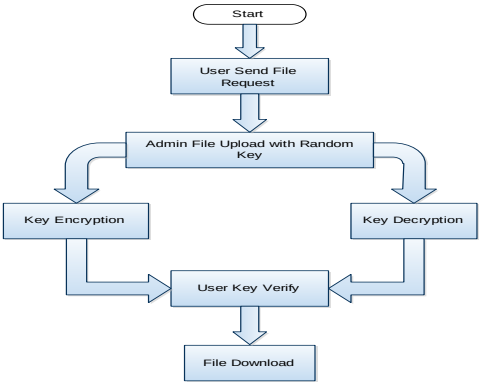
Cloud Computing has great potential of providing robust computational power to the society at reduced cost. It enables customers with limited computational resources to outsource their large computation workloads to the cloud, and economically enjoy the massive computational power, bandwidth, storage, and even appropriate software that can be shared in a pay-per-use manner. Despite the tremendous benefits, security is the primary obstacle that prevents the wide adoption of this promising computing model, especially for customers when their confidential data are consumed and produced during the computation.

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 fight 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.

This paper investigates secure outsourcing of widely applicable linear programming computations. In order to achieve practical efficiency, our mechanism design explicitly decomposes the linear programming computation outsourcing into public linear programming solvers running on the cloud and private linear programming parameters owned by the customer. The resulting flexibility allows us to explore appropriate security/ efficiency tradeoff via higher-level abstraction of linear programming computations than the general circuit representation. In particular, by formulating private data owned by the customer for linear programming 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 linear programming problem into some arbitrary one while protecting sensitive input/output information. To validate the computation result, we further explore the fundamental duality theorem of linear programming 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. Extensive security analysis and experiment results show the immediate practicability of our mechanism design.

II. Proposed Model

In this proposed model, the outsourced computation workloads often contain sensitive information, such as the business financial records, proprietary research data, or personally identifiable health information etc. To fight 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. So In our experiment we use Fully Homomorphism Encryption (FHE) scheme. i.e., 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.



A. Module Description:

There are 4 modules in designing of secure and practical outsourcing of linear programming computation. Those are:

1. Mechanism Design Framework
2. Basic Techniques
3. Enhanced Techniques via Affine Mapping
4. Result Verification

B. 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.

Fig1*:*Schematic representation of the Data flow diagram

• KeyGen(1^k) → {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 Linear program problem.

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

• ResultDec(K,ϕ, 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.

Table1. All the test cases mentioned above passed successfully. No defects encountered.

C. **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.

Table1. *All the test cases mentioned above passed successfully. No defects encountered.*

Test ID	Test case	Expected Result	Observed Result	Output
1	User login	It gives authority to authorized user	Authorized user can log in	Pass
2	File request	Requested file should be given	File which is requested is sent	Pass
3	To view file	File which is been saved can be seen	The saved file name would be present	Pass
4	Same file name not taken	Error message has to be displayed	Error message displayed	Pass
5	Execute/run the application	Application should run without any interruption	Application is executing properly	Pass

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 $b' = Qb$.

D. Enhanced Techniques via Affine Mapping:

To enhance the security strength of Linear Programming outsourcing, we must be able to change the feasible region of original Linear Programming 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.

E. Result Verification:

Till now, we have been assuming the server is honestly performing the computation, while being interested learning information of original Linear Programming 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 a strong financial incentive 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 Linear Programming problem $\phi_K = (A', B', b', c')$, we propose to solve the result verification problem by designing a method to verify the correctness of the solution y of ϕ_K. The workload required for customers on the result verification is substantially cheaper than solving the Linear Programming problem on their own, which ensures the great computation savings for secure Linear Programming outsourcing.

The Linear Programming 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 in Eq. (1), the objective function can be arbitrarily small while the constraints are all satisfied.

III. Experimental Results

A. Key Generation

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 Linear programming problem.

B. Problem Encryption

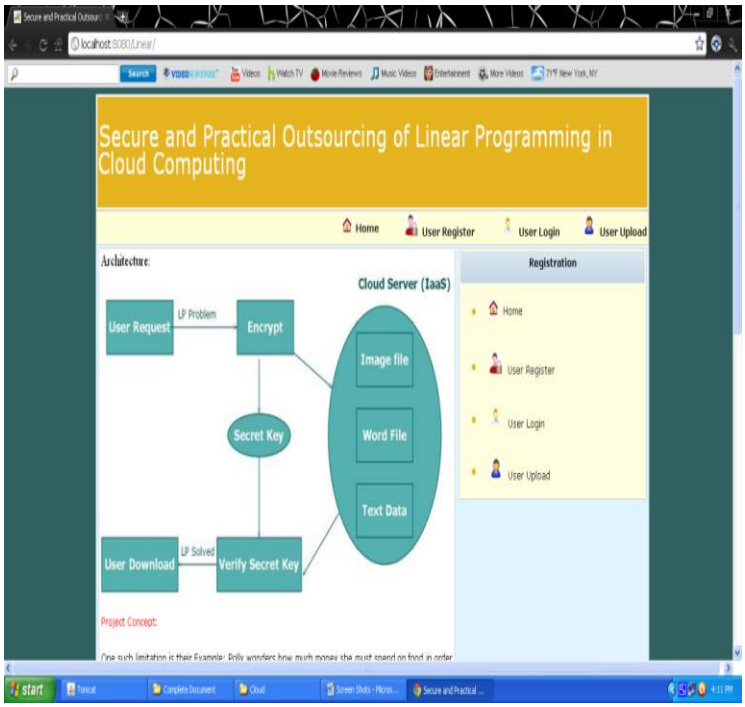
This algorithm encrypts the input tuple Φ into Φ_k with the secret key K. According to problem transformation, the encrypted input Φ_k has the same form as Φ, and thus defines the problem to be solved in the cloud.

C. Proof Generation

This algorithm augments a generic solver that solves the problem Φ K to produce both the output y and a proof Γ. The output y later decrypts to x, and Γ is used later by the customer to verify the correctness of y or x.

D. Key Description

The mechanism must produce an output that can be decrypted and verified successfully by the customer.



IV. Conclusion

In this paper, we formalize the problem of securely outsourcing Linear Programming computations in cloud computing and provide practical design which fulfills input/output privacy and efficiency. By explicitly decomposing Linear Programming computation outsourcing into public Linear Programming solvers and private data, our mechanism design is able to explore appropriate security/efficiency tradeoffs via higher level Linear Programming computation than the general circuit representation. We develop problem transformation techniques that enable customers to secretly transform the original Linear Programming into some arbitrary one while protecting sensitive input/output information. Also investigate linear programming and derive a set of necessary and sufficient condition for result verification. Security analysis and experiment results demonstrate the immediate practicality of the proposed mechanism.

Future Works:

We plan to investigate some interesting future work as follows:

1) Devise robust algorithms to achieve numerical stability.
2) Explore the growing structure of problem for further efficiency improvement.
3) Establish formal security framework using some techniques.
4) Extend our result to non-linear programming computation outsourcing in cloud.

References

[1] M. Atallah and K. Frikken, "Securely outsourcing linear algebra computations," in Proc. of ASIACCS, 2010, pp. 48–59.g," in Proc. Of ICDCS, 2004, pp. 4–11.
[2] Sun Microsystems, Inc., "Building customer trust in cloud computing with transparent security," 2009, online at https://www.sun.com/offers/details/sun transparency.xml.
[3] Cong Wang, Kui Ren and Jia Wang, "Secure and Practical Outsourcing of Linear Programming in Cloud Computing", IEEE INFOCOM 2011.
[4] M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-PreservingAudit and Extraction of Digital Contents," Cryptology ePrint Archive, Report 2008/186, 2008.
[5] G. Ateniese, S. Kamara, and J. Katz, "Proofs of Storage fromHomomorphic Identification Protocols,"Proc. 15th Int'l Conf.Theory and Application of Cryptology and Information Security:
[6] Advances in Cryptology (ASIACRYPT),pp. 319-333, 2009.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Naveen M. "Secure and Practical Outsourcing of Linear Programming in Cloud Computing." IOSR Journal of Computer Engineering (IOSR-JCE), vol. 19, no. 5, 2017, pp. 22–26.