Realistic and Safe Outsourcing of Linear Programming in Cloud Computing

Mohd Zeeshan, E.Venkata Ramana, Dr. J Sasi Kiran

***Abstract*— Cloud computing makes customers to outsource large-scale computational tasks to the cloud, where massive computational power can be easily utilized in a pay-per-use manner with limited computational resources. However, security is the major concern especially when end-user's confidential data are processed and produced during the computation. Thus,there must be a mechanism which not only protect sensitive information by enabling computations with encrypted data, but also protect customers from malicious behaviors by validating the computation result.To achieve realistic 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.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.**

***Index Terms*— cloud computing, outsourcing , end-user's confidential data, linear programming(LP), Encryption, public LP solvers, private LP parameters.**

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.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 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.

To validate the computation result, we utilize the fact that the result is from cloud server solving the transformed LP problem.

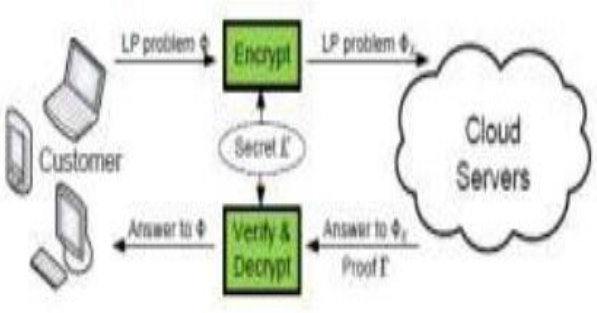

Fig 1: Architecture of secure outsourcing linear programming problems in Cloud Computing

II. MODULES DESCRIPTION

A. Mechanism Design Framework
B. Basic Techniques
C. Enhanced Techniques via Affine Mapping
D. Result Verification

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.

• KeyGen $(1^k) \rightarrow \{K\}$. 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.

Manuscript received October 05, 2014.
Mohd Zeeshan, Department of Computer Science and Engineering, Vidya Vikas Institute of Technology, Hyderabad,India
E.Venkata Ramana, Department of Computer Science and Engineering, Vidya Vikas Institute of Technology, Hyderabad,India
Dr. J Sasi Kiran, Department of Computer Science and Engineering, Vidya Vikas Institute of Technology, Hyderabad,India

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

• ProofGen (ϕK) → {(y,Γ)}. This algorithm augments a generic solver that solves the problem ϕK 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.

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.

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.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 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.

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 _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 _K. 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 in Eq,the objective function can be arbitrarily small while the constraints are all satisfied.

III. Preliminary Performance Results

Benchmark		Original Problem	Encrypted Problem		Asymmetric Speedup	Cloud Efficiency
#	size	$t_{original}$ (sec)	t_{cloud} (sec)	$t_{customer}$ (sec)	$\frac{t_{original}}{t_{customer}}$	$\frac{t_{original}}{t_{cloud}}$
1	m = 50, n = 60	0.167	0.170	0.007	26.5 ×	0.981
2	m = 100, n = 120	0.227	0.239	0.005	46.7 ×	0.956
3	m = 200, n = 240	0.630	0.613	0.017	37.3 ×	1.037
4	m = 400, n = 480	3.033	3.671	0.090	33.5 ×	0.835
5	m = 800, n = 960	19.838	23.527	0.569	34.9 ×	0.851
6	m = 1600, n = 1920	171.862	254.012	4.015	42.6 ×	0.690
7	m = 3200, n = 3840	1757.570	2661.360	47.602	36.4 ×	0.745

Table 1: Preliminary Performance Results

Here $t_{original}$, t_{cloud} , and $t_{customer}$ denotes the cloud-side original problem solving time, cloud-side encrypted problem solving time, and customer-side computation time, respectively. The asymmetric speedup captures the customer efficiency gain via

LP outsourcing. The cloud efficiency captures the overall computation cost on cloud introduced by solving encrypted LP problem, which should ideally be as closer to 1 as possible.

IV. CONCLUSION

In this paper, for the first time, 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 aset 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.

REFERENCES

Good Teachers are worth more than thousand books, we have them in Our Department.

[1] B. Chen and H. H. Cheng, "A Review of the Applications of Agent Technology in Traffic and Transportation Systems," IEEE Trans. Intelligent Transportation Systems, vol. 11, no. 2, 2010, pp. 485–497.
[2] B.P. Gokulan and D. Srinivasan, "Distributed Geometric Fuzzy Multiagent Urban Traffic Signal Control," IEEE Trans. Intelligent TransportationSystems, vol. 11, no. 3, 2010, pp. 714–727.
[3] D.C. Gazis, "Traffic Control: From Hand Signals to Computers," Proc. IEEE, vol. 59, no. 7, 1971, pp. 1090–1099.
[4] F.-Y. Wang, "Parallel System Methods for Management and Control of Complex Systems," Control and Decision, vol. 19, no. 5, 2004, pp. 485–489.
[5] F.-Y. Wang, "Toward a Revolution in Transportation Operations: AI for Complex Systems," IEEE IntelligentSystems, vol. 23, no. 6, 2008, pp. 8–13.
[6] F.-Y. Wang, "Parallel Control and Management for Intelligent Transportation Systems: Concepts, Architectures, and Applications," IEEE Trans. Intelligent Transportation Systems, vol. 11, no. 3,2010, pp. 1–9.
[7] F.-Y. Wang, "Agent-Based Control for Networked Traffic Management Systems," IEEE IntelligentSystems, vol. 20, no. 5, 2005, pp. 92–96.
[8] F.-Y. Wang and S. Tang, "Artificial Societies for Integrated and Sustainable Development of Metropolitan Systems," IEEE Intelligent Systems, vol. 19, no. 4, 2004, pp. 82–87.
[9] I. Foster et al., "Cloud Computing and Grid Computing 360-Degree Compared," Proc. Grid ComputingEnvironments Workshop, IEEE Press, 2008, pp. 1–10.
[10] M.C.Choy, D.Srinivasan and R.L.Cheu, "Cooperative, Hybrid Agent Architecture for Real-Time Traffic Signal Control," IEEE Trans. Systems, Man and Cybernetics, Part A: Systemsand Humans, vol. 33, no. 5, 2003, pp. 597–607.
[11] M.C.Choy, D.Srinivasan, and R.L.Cheu, "Neural Networks for Continuous Online Learning and Control," IEEE Trans. Neural Networks, vo1. 7, no. 6, 2006, pp. 1511–1531.
[12] N. Suri, K.M. Ford, and A.J. Cafias, "An Architecture for Smart Internet Agents," Proc. 11th Int'l FLAIRS
[13] Conf., AAAI Press, 1998, pp. 116–120.