ISSN:0975-9646

Practical Outsourcing of Linear Programming in Secured Cloud Computing

Lochan .B
Assistant Professor
Department of Computer Science
M.S. Engineering College, Bangalore

Abstract-Cloud Computing enables customers with limited computational resources to outsource their large computation workloads to cloud, and economically enjoy the massive computational power, bandwidth, storage, and even appropriate software that can be shared in a pay-per-use manner. 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. Treating the cloud as an intrinsically insecure computing platform from the viewpoint of the cloud customers, we must design mechanisms that not only protect sensitive information by enabling computations with encrypted data, but also protect customers from malicious behaviors by enabling the validation of the computation result.

In order to achieve practical efficiency, our mechanism design explicitly decomposes the Linear Programming(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 tradeoff via higher-level abstraction LP computations than the general circuit representation. 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 result 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.

I. INTRODUCTION

Cloud Computing enables 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 rapidly deployed with great efficiency and minimal management overhead[1]. 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. 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 that can be rapidly deployed with great efficiency and minimal management overhead[2]. The computation workloads often contain sensitive information, such as the business financial records, proprietary research data, or personally identifiable health information etc[3]. 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. The operational details inside the cloud are not transparent enough to customers[4]. As a result, there do exist various motivations for cloud server to behave unfaithfully and to return incorrect results, they may behave beyond the classical semi honest model.Fig.1 explains about architecture of secure outsourcing linear programming problem in cloud computing.

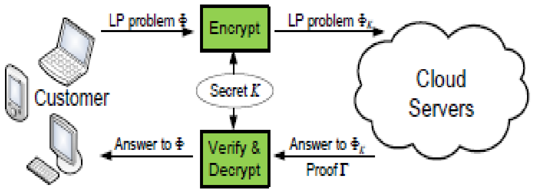

Fig. 1: Architecture of secure outsourcing linear programming problems in Cloud Computing

II. PROBLEM STATEMENT

The 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 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 key to map into some encrypted version and outsources problem to CS.

The Background on Linear Programming is an optimization problem is usually formulated as a mathematical programming problem that seeks the values for a set of decision variables to minimize (or maximize) an objective function representing the cost subject to a set of constraints. For linear programming, the objective function is an affine function of the decision variables, and the constraints are a system of linear equations and inequalities. Since a constraint in the form of a linear inequality can be expressed as a linear equation by introducing a non-negative slack variable, and a free decision variable can be expressed as the difference of two non-negative auxiliary variables.

III. The Proposed Schemes

The Linear Programming outsourcing scheme which provides a complete outsourcing solution for not only the privacy protection of problem input/output, but also its efficient result checking. It start from an overview of secure LP outsourcing design framework and discuss a few basic techniques and their demerits, which leads to a stronger problem transformation design utilizing affine mapping. 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). Note that our proposed mechanism provides us one-time pad types of flexibility. Namely, we shall never use the same secret key K to two different problems. Thus, when analyzing the security strength of the mechanism, we focus on the cipher text only attack.

III. Security Analysis

The analysis on correctness and soundness guarantee via the following two theorems.

Theorem 1: Our scheme is a correct verifiable linear programming outsourcing scheme.

Proof: The proof consists of two steps. First shows that for any problem and its encrypted version solution computed by honest cloud server will always be verified successfully. This follows directly from the duality theorem of linear programming. Namely, all conditions derived from duality theorem and auxiliary LP problem construction for result verification are necessary and sufficient. Next show that correctly verified solution always corresponds to the optimal solution of original problem . For space limit, we only focus on the normal case. The reasoning for infeasible cases follows similarly.

Theorem 2: Our scheme is a sound verifiable linear programming outsourcing scheme.

Proof: Similar to correctness argument, the soundness of the proposed mechanism follows from the facts that the LP problem are equivalent to each other through affine mapping, and all the conditions there after for result verification are necessary and sufficient.

IV. Performance Analysis

The theoretic Analysis involves two side overhead. They are Customer Side Overhead and Server Side Overhead.

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(n^2). Thus, it is straight-forward that the most time consuming operations are the matrix-matrix multiplications in problem encryption algorithm ProbEnc.
2. Server Side Overhead*:* For cloud server, its only computation overhead is to solve the encrypted LP problem as well as generating the result proof , both of which correspond to the algorithm ProofGen. If the encrypted LP problem belongs to normal case, cloud server just solves it with the dual optimal solution as the result proof , which is usually readily available in the current LP solving algorithms and incurs no additional cost for cloud . If the encrypted problem does not have an optimal solution, additional auxiliary LP problems can be solved to provide a proof. Because for general LP solvers, phase I method (solving the auxiliary LP) is always 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(n^3) time.

V. Related Work

The Related Work mainly deals with Work on Secure Computation Outsourcing , Work on Secure Multiparty Computation and Work on Delegating Computation and Cheating Detection.

1.Work on Secure Computation Outsourcing:

General secure computation outsourcing that fulfills all aforementioned requirements, such as input/output privacy and correctness/soundness guarantee has been shown feasible in theory. However, it is currently not practical due to its huge computation complexity. The customized

solutions are expected to be more efficient than the general way of constructing the circuits. A set of problem dependent disguising techniques are proposed for different scientific applications like linear algebra, sorting, string pattern matching, etc. However, these disguise techniques explicitly allow information disclosure to certain degree. Besides, they do not handle the important case of result verification, which in our work is bundled into the design and comes at close-to-zero additional cost.

However, both protocols use heavy cryptographic primitive such as homomorphic encryptions and/or oblivious transfer and do not scale well for large problem set. In addition, both designs are built upon the assumption of two non-colluding servers and thus vulnerable to colluding attacks. Based on the same assumption provide protocols for secure outsourcing of modular exponentiation, which is considered as prohibitively expensive most public-key cryptography operations.

2. Work on Secure Multiparty Computation: Another large existing list of work that relates to these is Secure Multi-party Computation (SMC), first introduced by Yao and later extended by Goldreich and many others. SMC allows

two or more parties to jointly compute some general function while hiding their inputs to each other. As general SMC can be very inefficient, have proposed a series of customized solutions under the SMC context to a spectrum of special computation problems, such as privacy-preserving cooperative statistical analysis, scientific computation, geometric computations, sequence comparisons,etc. However, directly applying these approaches to the cloud computing model for secure computation outsourcing would still be problematic. The major reason is that they did not address the asymmetry among the computational powers possessed by cloud and the customers which we specifically avoid in the mechanism design by shifting as much as possible computation burden to cloud only. Another reason is the asymmetric security requirement. In SMC no single involved party knows all the problem input information, making result verification a very difficult task. But in our model, we can explicitly exploit the fact that the customer knows all input information and thus design efficient result verification mechanism.

3. Work on Delegating Computation and Cheating Detection:

Detecting the unfaithful behaviors for computation outsourcing is not an easy task, even without consideration of input/output privacy. Verifiable computation delegation, where a computationally weak customer can verify the correctness of the delegated computation results from a powerful but untrusted server without investing too much resources, has found great interests in theoretical computer science community. In distributed computing and targeting the specific computation delegation of one-way function inversion. The customer can then use the commitment combined with a sampling approach to carry out the result verification.

VI CONCLUSION

It 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. It develops problem transformation techniques that enable customers to secretly transform the original LP into some arbitrary one while protecting sensitive input/output information. 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. The plan to investigate some interesting future work as follows: 1) devise robust algorithms to achieve numerical stability; 2) explore the sparsity structure of problem for further efficiency improvement; 3) establish formal security framework; 4) extend our result to non-linear programming computation outsourcing in cloud.

REFERENCE

[1] P. Mell and T. Grance, “Draft nist working definition of cloud computing,” Referenced Online http://csrc.nist.gov/groups/SNS/cloudcomputing/index. html, 2010.

[2] Cloud Security Alliance, “Security guidance for critical areas of focus in cloud computing” online at cloudsecurityalliance.org.

[3] M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, and E. H. Spafford, “Secure outsourcing of scientific computations”.

[4] S. Hohenberger and A. Lysyanskaya, “How to securely outsource cryptographic computations”.

[5] M. Atallah and K. Frikken, “Securely outsourcing linear algebra computations,”