Secure Outsourcing of Large-scale Linear Programming

Zhi-kuan WANG and Yang LIU
Nanjing University of Aeronautics and Astronautics, China
*Corresponding author

Keywords: Cloud computing, Outsource-secure algorithms, Linear programming, Data security.

Abstract. Linear programming (LP) is often used in reality production and life, such as the use of resources, human resource management, production arrangement. The client should pay out huge amounts of overhead to dispose huge data sets with its resource-constraint devices. Fortunately, Cloud computing can finish off this deficiency. The client outsources the computation to cloud servers. Then the servers accomplish the task and return the result to the client. But in the process, the correctness, verifiability and privacy should be emphasized. In our paper, based on the studies and analysis of previous protocols for secure outsourcing of linear programming in cloud computing, we find the obvious deficiencies in efficiency and safety. On the basis of the existing protocols, we propose a new protocol that combines the advantages of existing protocols and improves the shortcomings of existing protocols. We also analyze the security and efficiency of the new protocol and design a suitable simulation to verify the efficiency of this protocol.

Introduction

The scale of the data is exploding, so it needs a huge commitment of computing resources to store and deal with the data, which are not possessed for the resource-constrained clients. Then how to solve this problem causes a number of researches. Fortunately, cloud computing provides a suitable service to solve the problem with sufficiently low cost and relatively high efficiency. The cloud servers provide services to the clients in a multi-tenant model, dynamically partitioning or releasing different physical and virtual resources for clients. This means that the clients can obtain services at any time by any quantized way. Cloud outsourcing capacitates a client which has relatively limited computing resources to distribute a tough of accomplishment task to cloud servers who possess great computing power. Then the servers accomplish the task and return the result to the client.

As a client outsources the computation to cloud servers, there are still several essential conditions should be considered, such as correctness, verifiability and privacy [1, 2, 3, 4]. Correctness means that the cloud server accurately follows the protocol to fulfill the computation task. Verifiability is the correct of the returned result. The process of the computation in the cloud is opaque for clients. There are different motivations to give a spurious result to clients, such as finance consume, problem of hardware and software and attack from another. Therefore, the verification is obligate and the time consume should be much shorter than solving the outsourced problem. Privacy indicates that the information of input and output should not be revealed to the server.

In this paper, we focus on secure and efficient outsourcing of linear programming (LP) computations. Several protocols [5, 6, 7, 8] have been proposed for outsourcing linear programming computation to cloud server. The related contents in these protocols [5, 6, 7, 8] are similar at a certain extent. First, the client transforms the primal LP problem to a new LP problem, from which the client can get the solution of the primal problem, and outsources the new LP problem to the cloud. Then the cloud solves the new LP problem and its duality problem and returns the corresponding result to the client. Finally, the client can recover the solution of the primal problem by the returned result and verify its correctness. There are several problems in these protocols. First, in [5, 6, 7], the formulation of the LP problem is not the standard LP problem on engineering computing and optimization tasks. As will be seen later, it has been proved that the formulation of the LP problem in [5, 6, 7] is invalid by Cao's paper [9]. Second, although the protocol in [8] has
reduced the computation overhead of the client and the communication between the client and the server in the standard form, the security is reduced and some sensitive information can be revealed.

**System Model and Definition of LP Problem**

Figure 1 illustrates our system model, including two main entities: a client which has an inextricable linear programming problem by itself because of its limited computing power, and a cloud server which has the robust computation capacity. In consideration of the sensitive information of the client, first, the client needs to encrypt the primal problem to a new problem by a secret key. Then the client outsources the encrypted problem to the cloud server. Apparently, the encrypted problem must has the similar structure to the primal problem, and the results are relational. After solving the new problem, the cloud not only returns the solution, but proves the correctness of the solution. If it is correct, the client decrypts it and gets the result of the primal problem. If not, the client rejects the solution.

![Figure 1. System Model.](image)

We can concisely generalize the standard form [10] of linear programming as follows,

\[
\begin{align*}
\min & \quad c^T x \\
\text{subject to} & \quad Ax = b, \quad x \geq 0
\end{align*}
\]

Where \( c, x \) is an \( n \times 1 \) vector, \( b \) is an \( m \times 1 \) vector, \( A \) is an \( m \times n \) full row rank matrix. In practical applications, \( m \) is often much smaller than \( n \).

**Our Protocol**

We propose a new protocol for the secure LP outsourcing by using the standard from Eq.1. Based on the previous protocols, our modified protocol leads to a better performance. We also make the secret key size between the previous protocols [6, 8]. Finally, we analyze the security of our protocol and compare the efficiency of our protocol with the previous protocols [6, 8].

In our protocol, we assume \( x = My - r \) where \( M \) is an \( n \times n \) diagonal matrix with all positive entries to hide \( x \). We multiply a positive constant \( \gamma > 0 \) with \( f \) to confuse the cloud server, where \( f \) is the optimal value of the objection function. And the LP problem becomes

\[
\begin{align*}
\min & \quad \gamma(c^T My - c^T r) \\
\text{s.t.} & \quad A(My - r) = b, \\
& \quad My - r \geq 0
\end{align*}
\]

It's obvious \( c^T r \) is a constant, we can neglect it in the form. To protect \( A \) and \( b \), we multiply both sides of the equality constraint by an \( m \times m \) invertible matrix \( Q \). The further transformed form is as follows
\[
\begin{align*}
\begin{cases}
\min & \quad \gamma c^T My, \\
\text{s.t.} & \quad QAMy = Q(b + Ar), \\
& \quad My \geq 0,
\end{cases}
\end{align*}
\]

But the constraint $My \geq 0$ may expose $M, r$, we multiply both sides of $My \geq 0$ with an $n \times n$ diagonal matrix. Then we reform the transformed LP problem

\[
\begin{align*}
\begin{cases}
\min & \quad c^T y, \\
\text{s.t.} & \quad A' = QAM, \\
& \quad M' = PM, \\
& \quad r' = Pr, \\
& \quad b' = Q(b + Ar), \\
& \quad c' = \gamma M^T c,
\end{cases}
\end{align*}
\]

Then the transformed LP problem Eq.4 can be outsourced to the cloud server. As the cloud server return the solution, we can retrieve the primal result with the $\gamma, M, r$. Next we illustrate how to verify the correctness of a solution.

**Result Verification**

The basic idea of the verification is the LP duality theorem [10], as in [11] introduced. There are three possible cases for the LP problem solution: *Feasible and bounded, Infeasible and Unbounded.* First, if the Eq.1 has a feasible and bounded solution. The *duality of Eq.4* is

\[
\begin{align*}
\begin{cases}
\max & \quad b^\top s + r^\top t, \\
\text{s.t.} & \quad A^\top s + M^\top t = c', \\
& \quad t \geq 0,
\end{cases}
\end{align*}
\]

Where $s, t$ are the $m \times 1$ and $n \times 1$ vectors of dual decision variables. Based on the strong duality of the LP problems [12], While the objective values of the primal problem and dual problem are same, the primal feasible solution $y$ and a dual feasible solution $(s, t)$ are the optimal solutions. Therefore, the dual optimal solution should be returned and considered as a part of the proof $\Gamma$. Then, the following conditions,

\[
c^\top y = b^\top s + r^\top t, A' y = b', M' y \geq r', A^\top s + M^\top t = c', t \geq 0.
\]

Second, if the Eq.1 has the infeasible solution, considering the auxiliary problem,

\[
\begin{align*}
\begin{cases}
\min & \quad z, \\
\text{s.t.} & \quad z \leq A' x + b' \leq z, \\
& \quad M' y - r' \geq -z,
\end{cases}
\end{align*}
\]

While the Eq.7 has 0 as the optimal objective value if and only if the Eq.4 is feasible [13]. Then, if the Eq.4 is infeasible, the Eq.7 must have positive optimal value. Thus, the cloud server should return the solution of Eq.7 and its proof. Finally, the duality theory implies that the solution of transformed form Eq.4 is unbounded, which means the transformed form Eq.4 is feasible and its dual problem is infeasible. We can use the auxiliary problem to prove that the dual problem is infeasible, i.e.

\[
\begin{align*}
\begin{cases}
\min & \quad z, \\
\text{s.t.} & \quad z \leq A' x + M^\top t - c' \leq z, \\
& \quad t \geq -z,
\end{cases}
\end{align*}
\]

And proving it has a positive optimal objective value.
The complete protocol description for secure outsourcing of linear programming in the cloud is defined as follows.

KeyGen($1^\kappa$): Input a secret $K = (Q, M, r, P, \gamma)$, this algorithm generates a random $m \times n$ invertible matrix $Q$, a random $n \times n$ diagonal matrix $M$ with all positive entries, a random $n \times n$ diagonal matrix, a random $n \times 1$ vector and a constant $\gamma > 0$.

ProbTransform($\Phi$): Input an LP problem, which can be denoted by $\Phi = (A, b, c)$. The client uses the secret key $K$ to encrypt $\Phi$ to be $\Phi_K = (A', b', c', M', r')$, and sends $\Phi_K$ to the cloud server.

ProbSolve($\Phi_K$): Input a transformed problem, the cloud server solves the transformed LP problem and its dual problem using a public LP solver. The solution is $(y, s, t, \Gamma)$, where $\Gamma$ is some additional information if the LP is infeasible or unbounded.

ResultVerify($u, s, t, \Gamma$): Input the solution $(u, s, t, \Gamma)$, the client uses the duality theorem to verify whether the cloud cheats. If the solution is correct, the client can compute $x = My - r$ using the secret key, otherwise the client rejects the solution.

Security Analysis

Correctness: We assume $y$ is the optimal value of Eq.4, Then $x = My - r$ is the optimal value of Eq.1. If not, there is a $x^*$.Satisfying $c^T x^* \leq c^T x$. Let $x^* = My^* - r$. it holds that $c^T My^* - c^T r = c^T x^* \leq c^T x = c^T My - c^T r$ where $A'y^* = b', M' \geq r'$, then we find $y$ is not the optimal solution for Eq.4. It is a contradiction.

Soundness: The soundness of our protocol follows a fact [6], The LP problem $\Phi$ and $\Phi_K$ are equivalent to each other through the affine mapping. No cloud can cheat because of the duality theorem.

Privacy: For input privacy, $(A, b)$ is protected well by $(Q, M, r)$. The cloud server can only gain the rank and size information of $A$. For output privacy, $c' = \gamma M^T c$ can protects $c$ well. Because $(Q, M, r, \gamma)$ is random and uniform, our protocol can avoid the client’s information leaked.

Comparison with Precious Protocol

For the clients, the communication cost is the matrix multiplications. Transforming the primal LP problem $\Phi$ to new form $\Phi_K$, it takes 4 matrix-matrix multiplications and 3 matrix-vector multiplications. The verification needs 4 matrix-vector multiplications. Comparing with Wang’s [6] needs 5 matrix-matrix multiplications, 3 matrix-vector multiplications on transforming and 4 matrix-vector multiplications on verification. In addition, the particularity of random matrix $\lambda$ needs to be considered. In Chen’s protocol [8], the computation cost of client is greatly saved, which needs 1 matrix-matrix multiplication, 3 matrix-vector multiplications on transforming and 2 matrix-vector multiplications on verification, but its security is also greatly reduced.

Experimental Results

To compare the practical efficiency of our protocol with others, we simulate the experiment in a PC with Intel(R) Core(TM) i7-4770 CPU @ 3.4Ghz and 4GB memory using Matlab R2013a. The test-case is the COAP linear programming problem test set from university of Florida [14]. The detailed experimental results are shown in Table.1.

Our purpose is to verify whether the client has a computational gain by outsourcing the problem. In Table 1, all time is measured in seconds. $t_{\text{original}}$ denotes the time to solve the original LP problem; $t_{\text{cloud}}$ denotes the time for a cloud to solve the encrypted problem; $t_{\text{customer}}$ is the time a client spends to transform the original problem to a new one; $t_{\text{original}}/t_{\text{customer}}$ denotes the computational gain; and $t_{\text{original}}/t_{\text{cloud}}$ denotes the cloud’s efficiency. Because the similitude of verification method and the consumption of verification time are approximately same, we ignore the verification time.
consumption. From the result of experiment, we can find out that it's more economical and convenient to outsource the LP problem to the cloud.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Original problem $t_{\text{original}}$</th>
<th>Encrypted problem $t_{\text{cloud}}$</th>
<th>Computational gain $t_{\text{original}}/t_{\text{customer}}$</th>
<th>Cloud efficiency $t_{\text{original}}/t_{\text{cloud}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>27×51</td>
<td>0.007709</td>
<td>0.018397</td>
<td>6.169800</td>
<td>0.419040</td>
</tr>
<tr>
<td>56×138</td>
<td>0.010294</td>
<td>0.304900</td>
<td>5.401000</td>
<td>0.033762</td>
</tr>
<tr>
<td>173×295</td>
<td>0.017509</td>
<td>8.642800</td>
<td>4.664400</td>
<td>0.002025</td>
</tr>
<tr>
<td>305×472</td>
<td>0.027760</td>
<td>31.612000</td>
<td>1.983100</td>
<td>0.000878</td>
</tr>
<tr>
<td>821×1876</td>
<td>0.221130</td>
<td>58.839000</td>
<td>0.571640</td>
<td>0.000058</td>
</tr>
</tbody>
</table>

Summary

In this paper, we propose a new protocol for secure linear programming outsourcing problems. Compared with Chen's protocol [8], it is inefficient, but it's more secure and the efficiency is acceptable. Our protocol improves the existing problems of the previous two protocols, which ensures the efficiency and achieves the purpose of protecting the privacy of customers. It's well applicability and practicability.

Acknowledgement

This work is supported by Foundation of Graduate Innovation Center in NUAA (kfjj20161605).

References


