# A Note on "Privacy-Preserving and Secure Cloud Computing: A Case of Large-Scale Nonlinear Programming"

Zhengjun Cao, Lihua Liu

**Abstract**. We show that the outsourcing algorithm for the case of linear constraints [IEEE Trans. Cloud Comput., 2023, 11(1), 484-498] cannot keep output privacy, due to the simple translation transformation. We also suggest a remedy method by adopting a hybrid transformation which combines the usual translation transformation and resizing transformation so as to protect the output privacy.

**Keywords**: Nonlinear programming, linear constraints, output privacy, translation transformation, resizing transformation

#### 1 Introduction

The secure outsourcing of large-scale nonlinear programming problem has many applications. Recently, Du et al. [1] have presented two outsourcing algorithms for the case of linear constraints and the case of nonlinear constraints. In the considered scenario, there are two entities, the user  $\mathcal{U}$  and the server  $\mathcal{S}$ . To keep the input and output privacies, the user needs to encrypt his input data by using some encryption techniques. After receiving the returned solution by the server, the user retrieves the true solution by the corresponding decryption. Though the algorithms are interesting, we find the outsourcing algorithm for the case of linear constraints cannot protect the output privacy, due to the simple translation transformation. We will suggest a method to fix this flaw.

### 2 Review of the two algorithms

The Algorithm I is designed for the case of linear constraints. The original problem is

P1 : Minimize  $f(\boldsymbol{x})$ subject to  $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \ \boldsymbol{x} \ge 0$ 

where  $f(\boldsymbol{x})$  is the nonlinear objective function,  $\boldsymbol{A} \in R^{m \times n}$  and  $\boldsymbol{b} \in R^m$  are linear constraints,  $\boldsymbol{x} \in R^{n \times 1}$  is an *n*-dimension variable vector  $(x_1, x_2, \cdots, x_n)^{\top}$ .

Z. Cao, Department of Mathematics, Shanghai University, Shanghai, 200444, China.

L. Liu, Department of Mathematics, Shanghai Maritime University, Shanghai, 201306, China.

Email: liulh@shmtu.edu.cn

In order to keep the input privacy and output privacy, the problem is converted into

P4 : Minimize 
$$f(\boldsymbol{z})$$
  
subject to  $\boldsymbol{A}'\boldsymbol{z} = \boldsymbol{b}', \ \boldsymbol{z} \ge \boldsymbol{r}$ 

where z = x + r, A' = LPQAR, b' = LPQb, r is a random vector, P, Q are two random diagonal matrixes, and L, R are two random permutation matrixes (see Algorithm 1, Algorithm 2, page 488, Ref.[1]).

The Algorithm II is designed for the case of nonlinear constraints. The original problem is

P2: Minimize 
$$f(\boldsymbol{x})$$
  
subject to  $g_i(\boldsymbol{x}) = 0, i = 1, \cdots, m$   
 $h_j(\boldsymbol{x}) \le b_j, j = 1, \cdots, l$   
 $a_k \le x_k \le u_k, k = 1, \cdots, n$ 

where the coefficient matrix and vector of the equality constraints are denoted as  $\boldsymbol{G} \in R^{m \times n}, \boldsymbol{b} \in R^{l \times 1}$ , and the inequality constraints are denoted by  $\boldsymbol{H} \in R^{l \times n}$ .

The problem is converted into

P5: Minimize 
$$f(\boldsymbol{z})$$
  
subject to  $\hat{g}_i(\boldsymbol{z}) = 0, i = 1, \cdots, m$   
 $\hat{h}_j(\boldsymbol{z}) \le \hat{b}_j, j = 1, \cdots, l$   
 $\hat{a}_k \le z_k \le \hat{u}_k, k = 1, \cdots, n$ 

where  $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{r}, \hat{a}_k = a_k + r_k, \hat{u}_k = u_k + r_k, r_k$  is the *k*th element of  $\boldsymbol{r}$ , the corresponding matrixes and vectors are

$$\hat{G} = XPQGY, \ \hat{H} = XTSHY, \ \hat{b} = XPQb.$$

We refer to  $\S3.2$  in Ref.[1] for the notation descriptions.

#### 3 Some typos

As we see, the new problem P4 is specified with the equality constraints

$$A'z = LPQAR(x + r) = LPQARx + LPQARr \neq LPQb$$

So, the new vector b' must be specified as

$$b' = LPQARx + LPQARr$$

Likewise, the new problem P5 is wrongly specified. In fact,

$$\hat{G}z = XPQGY(x+r) = XPQGYx + XPQGYr = XPQGYr \neq 0$$

Besides,

$$\hat{H}z = XTSHY(x+r) = XTSHYx + XTSHYr \nleq XPQb$$

These notations should be restated.

## 4 The loss of output privacy in Algorithm I

The outsourcing process of Algorithm I can be depicted as follows.

Table 1: The Algorithm I	
User	Server
Convert the original problem	
$f(\boldsymbol{x}),  \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},  \boldsymbol{x} \ge 0, \text{ into the}$	
new version. $f(\mathbf{z}), \mathbf{A}'\mathbf{z} = \mathbf{b}',$	$f(\mathbf{z}) \mathbf{A}' \mathbf{b}' \mathbf{z}$
$oldsymbol{z} \geq oldsymbol{r}.$	$\xrightarrow{f(z), A, b, r}$ [open channel] Use the reduced gradient method to obtain the solution
Derive the true solution by $x^* = z^* - r$ .	$\underbrace{z^*}_{[\text{open channel}]} z^*, \text{ and return it to the user.}$

Notice that the original target vector  $\boldsymbol{x}$  is converted into

$$\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{r} \tag{1}$$

where r is a random vector to mask x. The encryption involves only the simple translation transformation. In the later outsourcing process, the user needs to transfer the data

 $\{f(z), A', b', r\}, \text{ not } \{f(z), A', b'\},\$ 

via an open channel.

Without the vector  $\mathbf{r}$ , the server cannot recover the inequality constraints  $\mathbf{z} \geq \mathbf{r}$ , instead only the nonnegative constraints  $\mathbf{z} \geq \mathbf{0}$ . But one cannot derive the solutions to  $f(\mathbf{z})$ , s.t.,  $\mathbf{A}'\mathbf{z} = \mathbf{b}', \mathbf{z} \geq \mathbf{r}$ , from that of  $f(\mathbf{z})$ , s.t.,  $\mathbf{A}'\mathbf{z} = \mathbf{b}', \mathbf{z} \geq \mathbf{0}$ . The original presentation has confused the inequality constraints  $\mathbf{z} \geq \mathbf{r}$  with the nonnegative constraints  $\mathbf{z} \geq \mathbf{0}$ . It neglected the necessity to transfer the vector  $\mathbf{r}$  to the server.

The server returns the solution  $z^*$  to the user via the open channel. Clearly, the server who knows the vector r, can retrieve the target solution  $x^*$  by computing  $x^* = z^* - r$ . Besides, any adversary who has captured the vectors  $r, z^*$  via open channels can also retrieve the true solution  $x^*$ . Therefore, the Algorithm I cannot keep the output privacy, not as claimed. Du et al. have neglected the fact that the masking vector r must be sent to the server.

To fix the flaw, one needs to use a hybrid transformation which combines the usual translation transformation and resizing transformation. For example, the target vector can be converted into

$$\boldsymbol{z} = \lambda \boldsymbol{x} + \boldsymbol{r} \tag{2}$$

where  $\lambda$  is a random positive number, and r is a random vector. In this case, only the user who knows the random number  $\lambda$  can retrieve the final solution by computing

$$oldsymbol{x}^* = (oldsymbol{z}^* - oldsymbol{r})/\lambda$$
 (3)

By the way, in the Algorithm II, the masking vector r is not exposed due to that

$$\hat{a}_k \le z_k \le \hat{u}_k, k = 1, \cdots, n,$$
$$\hat{a}_k = a_k + r_k, \ \hat{u}_k = u_k + r_k$$

One cannot retrieve the masking vector  $\boldsymbol{r}$  from the outsourced data  $\{(\hat{a}_k, \hat{u}_k)\}_{k=1,\dots,n}$ .

#### 5 Further discussions

 $\diamond$  Feasible region similarity. To keep the similarity between the feasible region of new outsourced problem and the original, one needs to use the general translation transformation and resizing transformation. If the target vector  $\boldsymbol{x}$  is converted into

$$\boldsymbol{z} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{r} \tag{2'}$$

where D is a random nonsingular positive diagonal matrix, and r is a random vector, the feasible region of the outsourced problem could be incompatible with the original. In this case, the recovered solution

$$x^* = D^{-1}(z^* - r)$$
 (3')

makes no sense.

 $\diamondsuit$  Representations for nonlinear constraints. The Algorithm II is designed for the case of nonlinear constraints

$$g_i(\mathbf{x}) = 0, i = 1, \cdots, m, \ h_j(\mathbf{x}) \le b_j, j = 1, \cdots, l.$$

In general, the nonlinear constraints cannot be represented as

$$Gx = 0, Hx \leq b$$

where  $\boldsymbol{G} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^{l \times 1}$ , and  $\boldsymbol{H} \in \mathbb{R}^{l \times n}$ . Naturally, the above representation is only suitable for linear constraints, not nonlinear constraints. These nonlinear constraints include  $x_j^2$  or  $x_i x_j (i \neq j)$ terms at least [2]. Du et al. have confused the representations for linear constraints and that for nonlinear constraints. Eventually, the Algorithm II is just a generalized version of Algorithm I, which is not suitable for the case of nonlinear constraints.

#### 6 Conclusion

We show that the Du et al.'s outsourcing algorithms are flawed due to the simple translation transformation, and clarify that the algorithm for the case of nonlinear constraints is unsuitable. We also suggest a method to fix one algorithm. The findings in this note could be helpful for the future works on this topic.

# References

- W. Du, A. Li, Q. Li, P. Zhou: Privacy-Preserving and Secure Cloud Computing: A Case of Large-Scale Nonlinear Programming. *IEEE Trans. Cloud Comput.*, 2023, 11(1), 484-498.
- [2] F. Hillier and G. Lieberman: Introduction to Operations Research, 11th Edition, McGraw Hill, 2021.