Journal of Theoretical and Applied Information Technology
15th January 2022. Vol.100. No 1
© 2022 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

METHODS FOR THE SECURE USE OF EXTERNAL SERVERS TO SOLVE COMPUTATIONALLY-COMPLEX PROBLEMS WITH SECRET PARAMETERS

BANU B. YERGALIYEVA, YERZHAN N. SEITKULOV, DINA ZH. SATYBALDINA
Gumilyov Eurasian National University, Kazakhstan
E-mail: banu.yergaliyeva@gmail.com

ABSTRACT

In this work, we study methods for the secure use of external insecure computers (servers) when solving computationally-complex problems with secret parameters. This problem is one of the important scientific directions in the field of information security of cloud computing. The article presents methods for secure outsourcing of the problem of finding the extremum of a function, as well as one method for finding the value of an analytical (holomorphic) function on a secret argument. Note that our main goal is to demonstrate new methods of secure outsourcing of scientific computing, so we model classes of problems in such a way as to clearly show the essence of these methods.

Keywords: *Information Security, Big Data, Secure Outsourcing, Cloud Computing, Computationally-Complex Problem, Client-Server Interactions.*

1. INTRODUCTION

In this work, we study methods for the safe use of external insecure servers when solving computationally-complex problems with secret parameters. This problem is one of the important scientific directions in the field of information security in applied problems of cloud computing and security issues of client-server interactions. The paper presents methods of secure outsourcing for the following classes of computationally-complex tasks with secret parameters:

- Problems of finding the extremum of a function with secret parameters;

- Linear programming tasks with secret parameters;

- Problems of finding the value of an analytical function on a secret argument.

The theory of secure outsourcing of scientific computations is rapidly developing in various fields, since in modern conditions, the processing of big data is no longer possible to imagine without the use of powerful computing resources. That is why leading scientists in the field of information security offer a variety of methods for the secure outsourcing of scientific computing [1-50].

The main problem of secure outsourcing of scientific computations is that the client cannot transfer all the data of the original computationally-complex task to the server since the task contains confidential parameters. Therefore, the client first needs to convert the original task to another task, where already secret information cannot be detected. A new computationally-complex task is sent to the cloud to solve it. Then, the result of the cloud processing is transferred back to the client. From this intermediate result, the client must independently be able to calculate the result of the original computationally-complex task for a time that is acceptable for him.

This client-server interaction can be given the following protocol view [9]:

Protocol Z:

Suppose that a client needs to solve some computationally-complex task Z, depending on the secret parameter α: $Z(\alpha)$. Suppose that there is an algorithm A for solving the problem $Z(\alpha)$, which can be effectively implemented on the server, but not on the client's computer.

Step 1. The client decomposes algorithm A into two algorithms B and C so that the following conditions are satisfied:

- Implementation of algorithms B and C allows you to solve the problem Z;

- Algorithm B may depend on the secret parameter, and the C algorithm either does not depend on the secret parameter *α*, or the time it takes for the server to discover the secret from the *C* algorithm is unacceptable for it.

- The client can calculate *B* fairly quickly in a reasonable time.

Step 2. The client implements the *B* algorithm on its small computer, and sends the *C* algorithm to the server.

Step 3. The server implements the *C* algorithm, and returns the result of the calculation back to the client.

Step 4. The client, having received the result of calculating *C*, solves the problem *Z (α).*

Note that our main goal is to demonstrate new methods of secure outsourcing of scientific computations, so we model classes of problems in such a way as to clearly show the essence of these methods.

In Section 2 we will study several problems of finding the extremum of a function with secret parameters. At the same time, we offer several examples, generalizing them step by step, and moving on to non-trivial problem statements.

Section 3 deals with linear programming problems with secret parameters. Such problems arise, for example, in production planning, which, as well-know, can be mathematically represented as a problem of determining the maximum value of a linear function under certain constraints. Also presented is one mathematical programming problem where the objective function is not a linear function.

In Section 4 we present a solution to the problem of finding the value of an analytical function of several complex variables on a secret argument. For an approximate solution of this problem using a server, we used Osgood's lemma and the generalized Cauchy integral formula from the theory of functions of several complex variables.

2. PROBLEMS OF FINDING THE EXTREMUM OF A FUNCTION WITH SECRET PARAMETERS

In In this subsection, we will consider methods for secure finding the extremum of a function with secret parameters using external servers. At the same time, in order to show the essence of the methods, we will demonstrate them with specific examples.

First, consider a real function of one variable:

$$y = f(x) \qquad (1)$$

Suppose that this function is twice differentiable on some interval (a, b). Then the critical points $x_0 \in (a, b)$, are known to be determined from the condition

$$y' = f'(x_0) = 0$$

Therefore, the main difficulty in finding the extremum of a function is to find the critical points, after which it is easy to calculate the value of the extremum of the function by the formula (1).

Example No 1. Suppose that it is necessary to find the critical point for the function

$$f(x) = ax^2 + bx + c \qquad (2)$$

using an insecure server, while some of the parameters a, b, c are the client's secret numbers.

Option 1. Let the number c is a secret, and the rest of the parameters are not secrets. Then is sent to the insecure server the following equation: $2ax = -b$. If x_0 is the solution of this equation, then the client finds the extremum of the function (2) by calculating the value at that point.

Option 2. Let the number b is a secret, and the numbers a, c are not secrets. Then, to find the critical point on an insecure server, we can use the following protocol:

Step 1. The client chooses a random secret number β and calculates the numbers $d = 2a\beta$ and $p = -d - b$. And then sends to the server the following equation: $2az = p$

Step 2. Insecure server finds a solution z_0 of the equation $2az = p$, and returns this solution to the client.

Step 3. The client finds a solution to the equation $2ax = -b$ by the formula $x = \beta + z_0$.

We can, of course, give other options, for example, consider that the number *a* is a secret parameter or all parameters at once are a secret, but, in fact, they are solved in a similar way.

Example No 2. Suppose we need to find the critical point for a function of two variables.

$$f(x, y) = ax^2 + by^2 + cxy + dx + ey + f \qquad (3)$$

The critical point of this function is found by solving the system of equations

$$\begin{cases} 2ax + cy + d = 0 \\ 2by + cx + e = 0 \end{cases}$$

That is, it is necessary to solve the linear equation

$$\begin{pmatrix} 2a & c \\ c & 2b \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -d \\ -e \end{pmatrix} \qquad (4)$$

There can also be various options here, but we will consider only one option, when the numbers d, e are the secret parameters of the client. To solve equation (4) using a server, we can use the following protocol:

Step 1. The client chooses a secret vector at random

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}$$

and calculates the vector

$$w = \begin{pmatrix} 2a & c \\ c & 2b \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

Then the client calculates the vector

$$r = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + \begin{pmatrix} d \\ e \end{pmatrix};$$

and sends the equation to the server

$$\begin{pmatrix} 2a & c \\ c & 2b \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \qquad (5)$$

relatively unknown z_1, z_2.

Step 2. The server solves equation (5) and returns the solution to the client. We denote this solution z^0.

Step 3. The client finds a solution to equation (4) according to the following formula

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} - z^0.$$

Example No 3. Consider an arbitrary twice continuously differentiable function y=$f(x)$ on the segment $[a, b]$. It is necessary to find the extremum of this function, while we will assume that the function itself is a secret.

We make the following types of change of variables:

- shift along the abscissa: $x = t + c$, where t is a new explanatory variable, and c is some fixed secret number that provides a shift of the function graph along the abscissa axis;

- shift along ordinates: $y = w + d$, where w is new dependent variable, and d is a some fixed secret number that provides a shift of the function graph along the ordinate axis;

- compression of the function along the abscissa axis: $x = \alpha t$, where α, $0 < \alpha < 1$, is a compression ratio along the abscissa;

- compression of the function along the ordinate axis: $y = \beta w$, where β, $0 < \beta < 1$, is a compression ratio along the ordinate.

Now we will make sequentially all these types of change of variables:

$$\beta w + d = f(\alpha t + c),$$

$$w = \frac{f(\alpha t + c) - d}{\beta} \equiv g(t).$$

Suppose that at the interior point $x_0 \in (a, b)$ extreme value is reached $y_0 = f(x_0)$, then the function $w = g(t)$ reaches its extremum at the point

$$t_0 = \frac{x_0 - c}{\alpha} \in \left(\frac{a - c}{\alpha}, \frac{b - c}{\alpha}\right),$$

in this case, the extreme value will be equal to

$$w_0 = \frac{f(x_0) - d}{\beta}.$$

Now finding the extremum of a function can be described by the following protocol.

Step 1. The client transforms the original function $y = f(x)$, successively applying all of the above types of variable substitutions, and the numbers c, d, α, β keeps secret. Next, it sends to the server next function $w = g(t)$ and interval $\left(\frac{a-c}{\alpha}, \frac{b-c}{\alpha}\right)$.

Step 2. The server finds the breaking point t_0 and extremumw_0 for function$w = g(t)$ in the interval $\left(\frac{a-c}{\alpha}, \frac{b-c}{\alpha}\right)$, and passes a couple of numbers (t_0, w_0) to the client.

Step 3. The client finds the critical point and extremum of the original function using the following simple formulas

$$x_0 = \alpha t_0 + c, \qquad f(x_0) = \beta w_0 + d.$$

Example No 4. Let us now consider the general case. Note that instead of a simple shift along the abscissa axis, we can make the following change of variable:

$$x = x(t),$$

where t is a new explanatory variable, and function $x(t)$ is a any strictly increasing continuously differentiable function on some segment $[m,n]$, and let at all points $t \in [m,n]$ the strict inequality holds

$$x'_t(t) > 0.$$

Suppose that the conditions

$$x(m) = a, \qquad x(n) = b.$$

Further, since

$$y'_t(t) = f'_x(x(t))x'_t(t),$$

and $x'_t(t) > 0$, then the problem of finding the extremum of the function $f(x)$ on the segment $[a,b]$ is reduced to the problem of finding the extremum of the function

$$g(t) \equiv f\left(x(t)\right)$$

On the segment $[m,n]$.

So, we need the following types of variable changes:

- replacement of dependent variable: $x = x(t)$, where t is a new explanatory variable;

- shift along ordinates: $y = w + d$, where w is new dependent variable, and d is a some fixed number that provides a shift of the function graph along the ordinate axis;

- compression of the function along the ordinate axis: $y = \beta w$, where β, $0 < \beta < 1$, - compression ratio along the ordinate.

Now, making sequentially all these types of change of variables, we get as a result the function

$$w(t) = \frac{g(t) - d}{\beta}.$$

Suppose that at the interior point $x_0 \in (a,b)$ the extremal value $y_0 = f(x_0)$ is reached, then the function $w(t)$ reaches its extremum at the point t_0, such that $x(t_0) = x_0$. In this case, the extreme value will be equal to

$$w_0 = \frac{g(t_0) - d}{\beta}.$$

Now finding the extremum of a function can be described by the following protocol.

Step 1. The client transforms the original function $y = f(x)$, successively applying the described types of change of variables, while the function $x(t)$, as well as numbers d, β keeps secret. Next, it sends a function $w(t)$ and interval (m,n) to the server.

Step 2. The server finds the extreme point t_0 and extremum w_0 for function $w(t)$ in the interval (m,n), and passes a couple of numbers (t_0, w_0) to the client.

Step 3. The client finds the extreme point and extremum of the function $y = f(x)$ by the following simplest formulas

$$x_0 = x(t_0), \qquad f(x_0) = \beta w_0 + d.$$

Example No 5. In some cases, when modeling economic problems, we get a certain optimization problem, where it is required to find the extremum of a function under certain constraints. Consider the following problem of determining the extremum of a function

$$f(x_1, \dots, x_n) = \sum_{i,j=1}^{n} a_{ij} x_i x_j - \sum_{k=1}^{n} b_k x_k \quad (6)$$

and impose communication conditions

$$c_1 x_1 + c_2 x_2 + \cdots + c_n x_n = P, \quad (7)$$

where $a_{ij} = a_{ji}$.

The Lagrange function will have the form:

$$L = \sum_{i,j=1}^{n} a_{ij} x_i x_j - $$
$$- \sum_{k=1}^{n} b_k x_k + \lambda(\sum_{i=1}^{n} c_i x_i - P)$$

Differentiating the Lagrange function with respect to each variable and taking into account the constraint condition, we compose the following system of equations:

$$\begin{cases} L'_{x_1} = 2a_{11}x_1 + \cdots + 2a_{1n}x_n - b_1 + c_1\lambda = 0 \\ L'_{x_2} = 2a_{21}x_1 + \cdots + 2a_{2n}x_n - b_2 + c_2\lambda = 0 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ L'_{x_n} = 2a_{n1}x_1 + \cdots + 2a_{nn}x_n - b_n + c_n\lambda = 0 \\ c_1x_1 + c_2x_2 + \cdots + c_nx_n = P \end{cases}$$

Let us denote by A matrix

$$A = \begin{pmatrix} 2a_{11} & 2a_{12} & \dots & 2a_{1n} & c_1 \\ 2a_{12} & 2a_{22} & \dots & 2a_{2n} & c_2 \\ \dots & \dots & \dots & \dots & \dots \\ 2a_{1n} & 2a_{2n} & \dots & 2a_{nn} & c_n \\ c_1 & c_2 & \dots & c_n & 0 \end{pmatrix}$$

Then we obtain the system of linear algebraic equations

$$Aq = b, \tag{8}$$

where $b = (b_1, b_2, \dots, b_n, P)^T$ is right column, $q = (x_1, x_2, \dots, x_n, \lambda)^T$ is a unknown vector. As we see, the problem of finding the critical point is reduced to solving equation (8).

The client needs to find the extremum of the function (6) under the condition of communication (7) using the computational means of an unsafe server. In this case, the server does not need to know the following secret client parameters: $b_1, b_2, \dots, b_n, a_{ij}$, c_k and P. And the decision itself must also remain a secret.

To solve this equation, you can use the following protocol:

Step 1. The client takes the secret vector at random w and the secret invertible matrix D:

$$w = \begin{pmatrix} w_1 \\ w_2 \\ \dots \\ w_{n+1} \end{pmatrix},$$

$$D = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1,n+1} \\ d_{21} & d_{22} & \dots & d_{2,n+1} \\ & & \dots & \\ d_{n+1,1} & d_{n+1,2} & \dots & d_{n+1,n+1} \end{pmatrix}$$

and calculates a new matrix $G = AD$ and vector $b - Gw = f$, and sends to the server the equation

$$Gy = f \tag{9}$$

Step 2. The server solves the system of linear algebraic equations (9) and returns the solution to the client y_0.

Step 3. Then the client finds a solution to equation (8) by the formula

$$q = D(y_0 + w)$$

That is, it finds the critical point for function (6) under the condition of connection (7).

Note that in the case when for the client the numbers a_{ij} and c_k are not secret, then instead of the matrix D the identity matrix is taken.

3. LINEAR PROGRAMMING PROBLEMS WITH SECRET PARAMETERS

Now we will consider the general problem of production planning, which, as you know, can be mathematically represented as the problem of determining the maximum value of the function

$$L = c_1x_1 + c_2x_2 + \cdots + c_nx_n \to \max, \tag{10}$$

with restrictions (terms of communication)

$$\sum_{j=1}^{n} a_{ij}\, x_j \le b_i, \quad i = 1,2,\dots,m; \tag{11}$$

$$x_i \ge 0, \quad i = 1, 2, \dots, n. \tag{12}$$

Here a_{ij}, b_i, c_j are some constants.

Suppose that the coefficients c_i and a_{ij} are the client's secret numbers, and the numbers b_i are not a secret. Also suppose the solution must also be kept secret.

Let D is a positive diagonal matrix of the following form

$$D = \begin{pmatrix} d_1 & 0 & 0 & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & d_n \end{pmatrix} \equiv diag(d_1, d_2, \dots, d_n),$$

$$d_i \ge 0, \qquad i = 1, \dots, n.$$

Let's make the change of variables

$$x \equiv \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} =$$

$$= \begin{pmatrix} d_1 & 0 & 0 & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & d_n \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \equiv Dy$$

Then we get the following problem

$$M = r_1 y_1 + r y_2 + \dots + r y_n \to \max, \quad (13)$$

with restrictions (terms of communication)

$$\sum_{j=1}^{n} p_{ij}\, y_j \le b_i, \quad i = 1,2,\dots,m; \quad (14)$$

$$y_i \ge 0, \quad i = 1,2,\dots,n. \quad (15)$$

Here $p_{ij} = a_{ij} d_j$, $r_j = c_j d_j$.

Now we can imagine a protocol for solving the problem (10-12):

Step 1. The client takes a random diagonal matrix $D = diag(d_1, d_2, \dots, d_n), d_i \ge 0, i = 1, \dots, n.$

and makes the change of variables $x = Dy$. Now sends to the server a task (13-15).

Step 2. The server solves the problem (13-15) and sends the solution y^0 to the client.

Step 3. The client finds a solution x^0 of the task (10-12) by the formula

$$x^0 = D\,y^0.$$

Now we present a typical economic optimization problem in which the objective function is not linear, but which can only be solved using the computing power of the servers.

Let $C = (c_i^j)$ is a is a positive matrix, and the numbers $p_1, \dots, p_m$ are positive numbers.

It is required to find a vector x^s, maximizing function

$$p_1\, f_1\left(\sum_{i=1}^{n} c_i^1 x_i^1\right) + \dots +$$

$$+ p_m\, f_m\left(\sum_{i=1}^{n} c_i^m x_i^m\right) \to max \quad (16)$$

with restrictions

$$x_i^1 + \dots + x_i^m = h_i, i = 1, \dots, n \quad (17)$$

$$x_i^s \ge 0, \ \ i = 1, \dots, n, \ \ s = 1, \dots, m. \quad (18)$$

Where f_i are upward convex scalar functions of one variable (this is necessary to be able to use general convex optimization methods, for example, the natural logarithm is a convex upward function).

So, suppose the client needs to solve problem (16) - (18). The secret elements are the matrix *C*, numbers h_i and vector solution $x = (x_1, \dots, x_n)$. The vector $p = (p_1, \dots, p_m)$ is not secret.

Protocol

Step 1. The client randomly chooses m secret diagonal matrices

$$\begin{cases} D^1 = diag\left(d_1{}^1, \dots, d_n{}^1\right), \\ d_i{}^1 > 0, \quad i = 1, 2, \dots, n \\ D^2 = diag\left(d_1{}^2, \dots, d_n{}^2\right), \\ d_i{}^2 > 0, \quad i = 1, 2, \dots, n \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ D^m = diag\left(d_1{}^m, \dots, d_n{}^m\right), \\ d_i{}^m > 0, \quad i = 1, 2, \dots, n \end{cases}$$

and calculates $r_i^s = c_i^s d_i^s$. Further, at $s = 2, \dots, m$ calculates $k_i^s = \frac{d_i^s}{d_i^1}$, $b_i = \frac{h_i}{d_i^1}$ and sends to the server a task (19) − (21) :

$$p_1\, f_1\left(\sum_{i=1}^{n} r_i^1 z_i^1\right) + \dots +$$

$$+ p_m\, f_m\left(\sum_{i=1}^{n} r_i^m z_i^m\right) \to max \quad (19)$$

$$z_i^1 + k_i^1 z_i^1 + \dots + k_i^m z_i^m = b_i, \quad i = 1, \dots, n \quad (20)$$

$$z_i^s \ge 0, i = 1, \dots, n; \ s = 1, \dots, m \quad (21)$$

Step 2. The server solves problem (19) - (21), and an approximate solution z_i^s returns to the client.

Step 3. The client finds a solution to problem (16) - (18) by the formula

$$x_i^s = d_i^s z_i^s. \quad (22)$$

It is easy to see that this protocol solves the original problem. Let's make the change of variables

$$x_i^s = d_i^s z_i^s.$$

Then, considering that

$$r_i^s = c_i^s d_i^s, k_i^s = \frac{d_i^s}{d_i^1}, b_i = \frac{h_i}{d_i^1}$$

then (16) - (17) take the form (19) - (20), respectively. Further, since d_i^s are positive numbers, then condition (18) can be written in the form of condition (21).

Protocol security. Insecure server gets a system of equations

$$r_i^s = c_i^s d_i^s,\ k_i^s = \frac{d_i^s}{d_i^1},\ b_i = \frac{h_i}{d_i^1},$$
$$i = 1,\dots,n;\ \ s = 2,\dots,m. \qquad (23)$$

Since the system of equations (23) contains $2nm + n$ unknowns, and the number of equations themselves in this system is $2nm$, system (23) is ambiguously solvable, that is, the server will not be able to unambiguously determine the secret parameters of the client.

4. PROBLEMS OF FINDING THE VALUE OF AN ANALYTIC FUNCTION ON A SECRET ARGUMENT

Consider a computationally complex complex function defined in some open subset $D \subseteq C^n = C \times \dots \times C$, where C^n is cartesian product of n copies of the complex plane. We need to approximately calculate the value of the complex function $y = f(z)$ on a secret argument $z \in D$, using the computing resources of the server.

Suppose the function $f(z)$ is a continuous in an open set $D \subset C^n$ and is holomorphic with respect to each variable separately, then according to the well-known Osgood lemma, this function is holomorphic in the entire domain D. Let the closed polydisk

$$\Delta(w,r) = \{z \in C^n : |z_j - w_j| \le r_j, 1 \le j \le n\} \subset D.$$

Then, using repeatedly the Cauchy integral formula for a function of one variable, we obtain

$$f(z) = \left(\frac{1}{2\pi i}\right)^n \oint\limits_{|w_1-\theta_1|=r_1} \frac{d\theta_1}{\theta_1 - z_1} \times$$
$$\times \oint\limits_{|w_2-\theta_2|=r_2} \frac{d\theta_2}{\theta_2 - z_2} \cdots \oint\limits_{|w_n-\theta_n|=r_n} \frac{d\theta_n}{\theta_n - z_n} f(\theta)$$

valid for any interior point

$$z = (z_1, z_2, \dots, z_n) \in \Delta(w,r).$$

Without loss of generality, suppose that $n = 2$. Then the value of the complex function on the secret argument can be calculated approximately by the following integral sum:

$$f(z) \approx$$

$$\approx \left(\frac{1}{2\pi i}\right)^2 \sum_{i=1}^{p} \sum_{j=1}^{q} \frac{f(\theta_1^i, \theta_2^j)(\theta_2^{j+1} - \theta_2^j)(\theta_1^{i+1} - \theta_1^i)}{(\theta_2^j - z_2)(\theta_1^i - z_1)},$$

where the points θ_1^i, θ_2^j are uniformly selected at the boundaries of the disks $\partial\Delta_1, \partial\Delta_2$ respectively, where

$$\Delta_l = \{z \in C : |z - w_l| < r_l, 1 \le l \le 2\},$$
$$z_1 \in \Delta_1,\ z_2 \in \Delta_2.$$

Now the approximate calculation of the value of the analytic function on the secret argument can be represented by the following protocol:

Step 1. The client chooses large enough numbers p and q, as well as points θ_1^i, θ_2^j, uniformly selected along the boundaries $\partial\Delta_1, \partial\Delta_2$ accordingly, and sends the function to the server f.

Step 2. The server calculates the value of the function f at every point: $f(\theta_1^i, \theta_2^j)$, $i = 1,\dots,p; j = 1,\dots,q$. And also calculates numbers $h(\theta_1^i, \theta_2^j) \equiv f(\theta_1^i, \theta_2^j)(\theta_2^{j+1} - \theta_2^j)(\theta_1^{i+1} - \theta_1^i)$.

Calculated numbers $h(\theta_1^i, \theta_2^j)$ server sends to client.

Step 3. The client calculates the approximate value of the function on the secret argument by the formula

$$f(z) \approx \left(\frac{1}{2\pi i}\right)^2 \sum_{i=1}^{p} \sum_{j=1}^{q} \frac{h\left(\theta_1^i, \theta_2^j\right)}{\left(\theta_2^j - z_2\right)\left(\theta_1^i - z_1\right)},$$

Note that this method assumes that the server calculates the values quickly enough $f\left(\theta_1^i, \theta_2^j\right)$, at every $i = 1, \dots, p; j = 1, \dots, q$. Then the client performs fairly simple arithmetic operations.

5. CONCLUSION

In this work, new methods and algorithms for the safe use of external unsafe servers in solving computationally-complex problems with secret parameters are obtained. Namely, methods of secure outsourcing were presented for the following classes of computationally-complex tasks with secret parameters:

In Section 2, we presented new methods for finding the extremum of a function with secret parameters. The methods presented in examples 1-4 are fundamental and can be used for future theoretical research in the field of secure outsourcing of scientific computations. In the applied aspect, these methods can be used in computational mathematics with an approximate finding of the extremum of a function. For example, classical methods of finding the approximate value of the maximum or minimum of a function, such as the grid method, the uniform search method, the successive approximation method, the half-division method, the golden ratio, and others, require powerful computational tools. If the client does not have such computer facilities, he is forced to use external facilities (server, supercomputer, etc.) on a contractual basis. But in this case, as a rule, the client is faced with the problem of storing confidential data in a task that he would like to approximately solve on external insecure computing resources. The methods presented in Section 2 ensure the security of secret parameters when finding an approximate value of the extremum of a function using insecure external servers.

In Section 3, we presented a number of methods that are of practical importance. Optimization problems, that is, finding the maximum or minimum of a function under certain constraints on the connection conditions, often arise in the mathematical modeling of economic problems.

Section 4 presents a completely new approach to the theory of secure outsourcing of scientific computations. The theoretical results from the theory of complex functions of several variables are used. The result obtained in this section, namely the protocol for calculating the value of a holomorphic function on a secret argument, is of both fundamental and applied nature. We will present the applications of the results of this section in the following works.

The research results can serve as a source for further analysis and research in the development of methods for the safe outsourcing of computationally-complex tasks with secret parameters.

6. ACKNOWLEDGEMENTS

This research is funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan, grant No. AP09559249.

REFERENCES:

[1] Y.N. Seitkulov, S.N. Boranbayev, G.B. Ulyukova, B.B. Yergaliyeva, D. Satybaldina "Methods for secure cloud processing of big data", *Indonesian Journal of Electrical Engineering and Computer Science*, 22(3), pp. 1650–1658, 2021, doi: 10.11591/ijeecs.v22.i3.pp1650-1658.

[2] Ye. Seitkulov, "New methods of secure outsourcing of scientific computations", *The Journal of Supercomputing*, vol. 65, issue 1, pp. 469-482, 2013, doi: 10.1007/s11227-012-0809-3.

[3] Jianhua Yu, Xueli Wang, Wei Gao, "Improvement and applications of secure outsourcing of scientific computations", *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, pp.763–772, 2015, doi: 10.1007/s12652-015-0280-0.

[4] Xing Hu and Chunming Tang, "Secure outsourced computation of the characteristic polynomial and eigenvalues of matrix", *Journal of Cloud Computing,* 2015, URL: https://eprint.iacr.org/2014/442.pdf, doi: 10.1186/s13677-015-0033-9.

[5] C. Wang, K. Ren, J. Wang, "Secure Optimization Computation Outsourcing in Cloud Computing: A Case Study of Linear Programming", *IEEE Transactions on Computers*, vol. 65, issue 1, pp.216-229, 2016, doi: 10.1109/TC.2015.2417542.

[6] Ronak Vyas, Alok Singh, Jolly Singh, Gunjan Soni, B. R. Purushothama, "Design of an efficient verification scheme for correctness of

outsourced computations in cloud computing", *Security in Computing and Communications*, Springer, vol. 536, pp.66–77, 2015, doi: 10.1007/978-3-319-22915-7_7.

[7] M. Atallah M. and K. Frikken, "Securely outsourcing linear algebra computations", *ASIACCS '10: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pp.48-59, 2010, doi: 10.1145/1755688.1755695.

[8] D. Benjamin and M. Atallah, "Private and cheating-free outsourcing of algebraic computations", *Proceedings of 6th conference on privacy, security, and trust (PST*). -2008. -P.240-245, 2008, doi: 10.1109/PST.2008.12.

[9] Tsutomu Matsumoto, Koki Kato, Hideki Imai, "Speeding Up Secret Computations with Insecure Auxiliary Devices", *In: Goldwasser S. (eds) Advances in Cryptology — CRYPTO' 88. CRYPTO 1988. Lecture Notes in Computer Science*, vol. 403, pp.497-506, 1990, doi: 10.1007/0-387-34799-2_35.

[10] Thierry Mefenza, Damien Vergnaud, "Cryptanalysis of Server-Aided RSA Protocols with Private-Key Splitting", *The Computer Journal*, vol. 62, issue 8, August 2019, pp. 1194–1213, doi: 10.1093/comjnl/bxz040.

[11] Kai Zhou, M.Y. Afifi, Jian Ren, "ExpSOS: Secure and Verifiable Outsourcing of Exponentiation Operations for Mobile Cloud Computing", *IEEE Transactions on Information Forensics and Security*, vol.12, issue 11, pp. 2518-2531, doi:10.1109/TIFS.2017.2710941.

[12] S. Hohenberger and A. Lysyanskaya, "How to Securely Outsource Cryptographic Computations", *In: Kilian J. (eds) Theory of Cryptography. TCC* 2005. Lecture Notes in Computer Science, vol. 3378, pp.264–282, 2005, doi: 10.1007/978-3-540-30576-7_15.

[13] P. Béguin and J.-J. Quisquater, "Fast Server-Aided RSA Signatures Secure Against Active Attacks", *In: Coppersmith D. (eds) Advances in Cryptology — CRYPT0' 95. CRYPTO 1995. Lecture Notes in Computer Science*, vol. 963, pp. 57-69, 1995, doi: 10.1007/3-540-44750-4_5.

[14] C.H. Lim and P.J. Lee, "Security and Performance of Server-Aided RSA Computation Protocols", *In: Coppersmith D. (eds) Advances in Cryptology — CRYPT0' 95. CRYPTO 1995. Lecture Notes in Computer Science,* vol. 963, pp.70–83, 1995, doi: 10.1007/3-540-44750-4_6.

[15] C. Castelluccia, E. Mykletun, G. Tsudik, "Improving Secure Server Performance by Re-balancing SSL//TLS Handshakes", *ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security,* pp.26–34, 2006, doi: 10.1145/1128817.1128826.

[16] X. Chen, J. Li, J. Ma, Q. Tang, W. Lou, "New Algorithms for Secure Outsourcing of Modular Exponentiations", *In Foresti, S., Yung, M. and Martinelli, F. (eds.) ESORICS, Lecture Notes in Computer Science*, vol. 7459, pp. 541–556, 2012, doi: 10.1007/978-3-642-33167-1_31.

[17] Y. Wang and et al, "Securely Outsourcing Exponentiations with Single Untrusted Program for Cloud Storage", *In Kutylowski, M. and Vaidya, J. (eds.) ESORICS, Lecture Notes in Computer Science,* vol. 8712, pp.326–343, 2014, doi: 10.1007/978-3-319-11203-9_19.

[18] P.Q. Nguyen and I. Shparlinski, "On the Insecurity of a Server-Aided RSA Protocol", *In: Boyd C. (eds) Advances in Cryptology — ASIACRYPT 2001. ASIACRYPT 2001. Lecture Notes in Computer Science*, vol. 2248, pp. 21–35, 2001, doi: 10.1007/3-540-45682-1_2.

[19] J. Merkle, "Multi-round Passive Attacks on Server-Aided RSA Protocols", *CCS '00: Proceedings of the 7th ACM conference on Computer and Communications Security*, pp.102–107, 2000, doi: 10.1145/352600.352616.

[20] B. Pfitzmann and M. Waidner, "Attacks on Protocols for Server-Aided RSA Computation", *In: Rueppel R.A. (eds) Advances in Cryptology — EUROCRYPT' 92. EUROCRYPT 1992. Lecture Notes in Computer Science*, vol.658, pp. 153–162, 1993, doi: 10.1007/3-540-47555-9_13.

[21] M. Jakobsson and S. Wetzel, "Secure Server-Aided Signature Generation", *In: Kim K. (eds) Public Key Cryptography. PKC 2001. Lecture Notes in Computer Science,* vol. 1992, pp. 383–401, 2001, doi: 10.1007/3-540-44586-2_28.

[22] J. Merkle and R. Werchner, "On the Security of Server-Aided RSA Protocols", *In: Imai H., Zheng Y. (eds) Public Key Cryptography. PKC 1998. Lecture Notes in Computer Science*, vol. 1431, pp. 99–116, 1998, doi: 10.1007/BFb0054018.

[23] Y. Aono, "A New Lattice Construction for Partial Key Exposure Attack for RSA", *In Jarecki, S. and Tsudik, G. (eds.) PKC, Lecture Notes in Computer Science,* vol.5443, pp. 34–53, 2009, doi: 10.1007/978-3-642-00468-1_3.

[24] J. Blömer and A. May, “New Partial Key Exposure Attacks on RSA”, *In Boneh, D. (ed.) CRYPTO, Lecture Notes in Computer Science*, vol. 2729, pp. 27–43, 2003, doi: 10.1007/978-3-540-45146-4_2.

[25] M. Ernst, E. Jochemsz, A. May, B. de Weger, “Partial Key Exposure Attacks on RSA up to Full Size Exponents”, *In Cramer, R. (ed.) EUROCRYPT Lecture Notes in Computer Science,* vol. 3494, pp. 371–386, 2005, doi: 10.1007/11426639_22.

[26] R. M. Ospanov, Ye. N. Seitkulov, N. M. Sissenov, B. B. Yergaliyeva, “An example of an internal function for the SPONGE scheme”, *Vestnik S.-Petersburg Univ. Ser. 10. Prikl. Mat. Inform. Prots. Upr.*, **17**:3 (2021), pp. 287–293. doi: 10.21638/11701/spbu10.2021.306

[27] M.A. Seksembayeva, N.N. Tashatov, G. Ovechkin, D.Zh. Satybaldina, Y.N. Seitkulov, “Study of the Principles of Error Correcting Code in A Multipath Communication Channel with Intersymbol Interference”, *Journal of Theoretical and Applied Information Technology, 2021, 99(18), pp. 4387–4398.*

[28] A. Boranbayev, S. Boranbayev, Y. Seitkulov, A. Nurbekov, “Proposing Recommendations for Improving the Reliability and Security of Information Systems in Governmental Organizations in the Republic of Kazakhstan”, *Advances in Intelligent Systems and Computing, 2021, 1290, pp. 854–868. doi: 10.1007/978-3-030-63092-8_57*

[29] Y.N. Seitkulov, S.N. Boranbayev, N.N. Tashatov, H.V. Davydau, A.V. Patapovich, “Speech information security assessing in case of combined masking signals”, *Journal of Theoretical and Applied Information Technology, 2020, 98(16), pp. 3270–3281.*

[30] Y.N. Seitkulov, S.N. Boranbayev, B.B. Yergaliyeva, H.V. Davydau, A.V. Patapovich, “Method for speech intelligibility assessment with combined masking signals”, *Journal of Theoretical and Applied Information Technology, 2020, 98(8), pp. 1173–1186*

[31] Y.N. Seitkulov, S.N. Boranbayev, H.V. Davydau, A.V. Patapovich, “Speakers and auditors selection technique in assessing speech information security”, *Journal of Theoretical and Applied Information Technology, 2019, 97(12), pp. 3305–3316*

[32] Y.N. Seitkulov, S. Boranbayev, H.V. Davydau, A.V. Patapovich, “Algorithm of forming speech base units using the method of dynamic programming”, *Journal of Theoretical and Applied Information Technology, 2018, 96(23), pp. 7928–7941.*

[33] Y.N. Seitkulov, S.N. Boranbayev, B.B. Yergaliyeva, H.V. Davydau, A.V. Patapovich, “The base of speech structural units of Kazakh language for the synthesis of speech-like signals”, *12th IEEE International Conference on Application of Information and Communication Technologies, AICT 2018*, doi: 10.1109/ICAICT.2018.8747120

[34] M. Tatur, D. Adzines, Y. Seitkulov, M. Lukashevich, “Data mining processing based on problem-oriented machine architecture”, *International Conference on Information and Digital Technologies, IDT 2015, 2015, pp. 372–375, doi: 10.1109/DT.2015.7222999*

[35] N.L. Verenik, Y.N. Seitkulov, A.I. Girel, M.M. Tatur, “Some regularities and objective limitations of implementing semantic processing algorithms on computing systems with massive parallelism”, *Eurasian Journal of Mathematical and Computer Applications, 2014, 2(2), pp. 92–101*, doi: 10.32523/2306-3172-2014-2-2-92-101

[36] A.H. Ali, M.N. Abbod, M.K. Khaleel, M.A. Mohammed, T. Sutikno, “Large scale data analysis using MLlib”, *Telkomnika (Telecommunication Computing Electronics and Control),* 19(5), pp. 1735-1746. doi: 10.12928/TELKOMNIKA.v19i5.21059

[37] K. Zhou, J. Ren, “CASO: Cost-Aware Secure Outsourcing of General Computational Problems”, *IEEE Transactions on Services Computing,* 14(2),8316961, 2021, pp. 386-399. Doi 10.1109/TSC.2018.2814991

[38] D. Huang, L. Dai, L. Wei, Q. Wei, G. Wu, “A Secure Outsourced Fusion Denoising Scheme in Multiple Encrypted Remote Sensing Images”, *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, 54(10), 2017, pp. 2378-2389. Doi: 10.7544/issn1000-1239.2017.20170427

[39] C. Kuruba, K. Gilbert, P. Sidhaye, G. Pareek, P.B. Rangappa, “ Outsource-secured calculation of closest pair of points”, *Communications in Computer and Information Science*, 625, 2016 pp. 377-389. Doi: 10.1007/978-981-10-2738-3_33

[40] J. Ye, H. Zhang, C. Fu, “Verifiable delegation of polynomials”, *International Journal of Network Security*, 18(2), 2016, pp. 283-290

[41] X. Hu, C. Tang, “Secure outsourced computation of the characteristic polynomial and eigenvalues of matrix”, *Journal of Cloud*

Computing, 4(1), 2015, doi: 10.1186/s13677-015-0033-9

[42] Y. Seitkulov, A. Tokhtabayev, S. Atanov, N.L. Verenik, A.I. Girel, M.M. Tatur, “Methodology of building intelligent systems on parallel processor”, *8th IEEE International Conference on Application of Information and Communication Technologies, AICT 2014 - Conference Proceedings, doi: 10.1109/ICAICT.2014.7035973*

[43] M. Assefi, E. Behravesh, G. Liu, and A. P. Tafti, “Big data machine learning using apache spark MLlib”, *in 2017 IEEE International Conference on Circuits and Systems (ICCS) international conference on big data (big data),* 2017, pp. 3492–3498, doi: 10.1109/BigData.2017.8258338.

[44] M. A. Mohammed, Z. H. Salih, N. Ţăpuş, and R. A. K. Hasan, “Security and accountability for sharing the data stored in the cloud”, *in 2016 15th RoEduNet Conference: Networking in Education and Research,* 2016, pp. 1–5, doi: 10.1109/RoEduNet.2016.7753201.

[45] M. A. Mohammed and N. ŢĂPUŞ, “A novel approach of reducing energy consumption by utilizing enthalpy in mobile cloud computing”, *Stud. Informatics Control*, vol. 26, no. 4, pp. 425–434, 2017, doi: 10.24846/v26i4y201706.

[46] A. H. Ali and M. Z. Abdullah, “Recent trends in distributed online stream processing platform for big data: Survey”, *in 2018 1st Annual International Conference on Information and Sciences (AiCIS),* 2018, pp. 140–145, doi: 10.1109/AiCIS.2018.00036

[47] B.-E. B. Semlali, C. El Amrani, and S. Denys, “Development of a Java-based application for environmental remote sensing data processing”, *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, pp. 1978–1986, 2019, doi: 10.11591/ijece.v9i3.pp1978-1986.

[48] J. G. Shanahan and L. Dai, “Large scale distributed data science using apache spark”, *in Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining,* 2015, pp. 2323–2324.

[49] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst, “HaLoop: Efficient iterative data processing on large clusters”, *Proc. VLDB Endow*., vol. 3, no. 1–2, pp. 285–296, 2010.

[50] G. Malewicz et al., “Pregel: a system for large-scale graph processing”, *in Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 2010, pp. 135–146.