# Development of a Control System for Computations in BOINC with Homomorphic Encryption in Residue Number System

Mikhail Babenko, Nikolay Kucherov
North Caucasian Federal University
Stavropol, Russia, 355000
mgbabenko@ncfu.ru, nkucherov@ncfu.ru

Andrei Tchernykh*
CICESE Research Center
Ensenada, B.C., Mexico, 22860
chernykh@cicese.mx

Nikolay Chervyakov, Elena Nepretimova and Irina Vashchenko
North Caucasian Federal University
Stavropol, Russia, 355000
ncherviakov@ncfu.ru, nev1973@mail.ru, irishechka.26@mail.ru

## Abstract

In this paper, we propose approaches to constructing reliable schemes using the Residue Number System (RNS) for the BOINC volunteer computing systems. We show that application of RNS to homomorphic ciphers allows to build completely homomorphic information security system that not only ensure security but possibility to process encrypted data without its decryption. We present an algorithm for localizing and correcting errors for moduli of a special type.

## 1 Introduction

BOINC type systems allows obtaining significant functional and economic advantages [BCT+17, IG15, SJJT86]. On the other hand, volunteer computing systems require special attention to security, since, they lead to risks of confidentiality, integrity and correctness of the obtained results [ASS+14, TBC+17]. Homomorphic ciphers are used to ensure the security of information [CBT+17, Gen10]. Users can send information to BOINC servers that is not the result of the requested calculations [TSTB16]. For instance, they could send results with an error or a set of random bits [KPT+13].

Here, we consider two main problems: uncompleted tasks, and deliberately distorting results.

1. The problem of an uncompleted task. After the data for processing is sent to one of the participants, he could not be able to return results, due to software failures, participant's refusal, natural cataclysm, etc [TSAT15]. The BOINC systems try to solve this problem by setting a deadline for participant task execution.

2. The problem of deliberately distorting results by users can lead to the failure of all computations performed by other participants, and need for recalculations, which could requires significant additional time. The

*Corresponding author: chernykh@cicese.mx

In: E. Ivashko, A. Rumyantsev (eds.): Proceedings of the Third International Conference BOINC:FAST 2017, Petrozavodsk, Russia, August 28 - September 01, 2017, published at http://ceur-ws.org

BOINC system solves this problem by using at least 5 replicas (by default)[IG15, TPBS14]. In each project, this value can be either increased or decreased. The result, adopted by the quorum of client programs, is considered to be correct or incorrect. Incorrect answers are usually rejected [KPT+16].

To protect the system from the information distortion, we propose to apply RNS for detection and correction of errors. RNS improves performance, reliability, security, since, computations are not performed over original large numbers, but over small projections of large numbers [QPV02, SJJT86].

Operations on projections can be executed in parallel and independently. Encryption schemes constructed by using RNS allow building asymptotically optimal security systems, both from a practical and theoretical point of view [GTN11].

Homomorphic encryption can be naturally used in cloud computing. RNS creates several pieces of data, and operations over individual pieces are homomorphic with respect to: addition, subtraction and multiplication.

These properties of RNS can be used to develop a homomorphic encryption function. In homomorphic encryption schemes, we consider two types of security: data security and moduli security.

In the volunteer computing systems like BOINC, when we use redundant RNS to ensure the reliability of information, the probability of distortion of one or several projections of the result is high.

In order to provide verification of the result, we propose a model, in which the Boing server does not trust any user. We develop a simplified mechanism of result verification with a given probability of the correctness.

# 2 RNS homomorphic ciphers and their properties

## 2.1 Method of conversion from positional number system to RNS

There exist several methods for conversion between positional number system and RNS. Methods based on the principle of sequential summation of bit modular products are not efficient [Gen10].

To convert the numbers from positional number system to RNS, we propose to use the method of recursive doubling, the parallel summation of bit modular products described in the following way [CBD+16, CBKG15]:

$$\left|\left(a_i 2^i + a_{i+1} 2^{i+1}\right)\right|_{p_j} = |\cdot|^{+}_{p_j}\ ,\ 0 \le i \le k,\ i.e.\, \alpha_i \equiv \left|\sum_{i=0}^{k} \left|a_i 2^i\right|^{+}_{p_j}\right|^{+}_{p_j} \tag{1}$$

where $\alpha_i$ – is the least non negative residue $|p_j|$, $\alpha_i \in \{0,\ 1\}$ in case of binary number system.

For a further parallelization this method uses the associativity of addition. The details of this parallel method is described in [Gen10].

## 2.2 Method of conversion from RNS to positional number system

For an efficient implementation of decryption algorithms we use the approximate method from [CBD+16]. The idea of the approximate method of comparison of modular numbers is based on a quotient from division of the value of a number by the dynamic range of RNS, Chinese reminder theorem (CRT), which relates the positional number $X$ with its representation with residues $(x_1, x_2, \ldots, x_n)$, where $x_i$ – is the least non negative residue, from division by modules from RNS moduli set $(p_1, p_2, \ldots, p_n)$, with the following expression

$$X = \left|\sum_{i=1}^{n} \frac{P}{p_i} \left|P_i^{-1}\right| x_i\right|_P \tag{2}$$

where $P = \prod_{i=1}^{n} p_i$, $p_i$ – RNS moduli set, $\left|P_i^{-1}\right|$ – multiplicative inversion of $P_i$ with respect to $p_i$, and $P_i = \frac{P}{p_i}$.

If we divide (2) by the constant $P$, then we obtain an approximate value

$$\frac{X}{P} = \left|\sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i}}{p_i} x_i\right|_1 = \left|\sum_{i=1}^{n} k_i x_i\right|_1 \tag{3}$$

where $k_i = \frac{\left|P_i^{-1}\right|_{p_i}}{p_i}$ – for all $i$ from 1 to $n$ are the constants of the system, $x_i$ – digits in the RNS representation. The value of each sum is in the interval [0, 1). The final result of the sum is defined after the summation and is the fractional part of the sum. The fractional part also can be represented as $X$ mod 1, because $X = \lfloor X \rfloor + X$ mod 1. The number of digits in the fractional part is defined by the maximum possible difference between adjacent numbers [CBD+16].

Here, we briefly review the concept of diagonal function [DIP93]. First, for a given moduli set $\{p_1, p_2, \ldots, p_n\}$, where the moduli $m_i$ are mutually prime, we define a parameter "Sum of Quotients (SQ)", where $SQ = \sum_{i=1}^{n} P_i$, for all $P_i = P/p_i$. Then, we also define constants $\tilde{k}_i = \left|-\frac{1}{p_i}\right|_{SQ}$ for $i = 1, 2, \ldots, n$. The diagonal function corresponding to a given number $X$ with residues $(x_1, x_2, \ldots, x_n)$ is defined as: $D\left(X\right) = \left|\sum_{i=1}^{n} \tilde{k}_i x_i\right|_{SQ}$.

Note that $D\left(X\right)$ is a monotonic function. Two numbers $X$ and $Y$ can be compared based on the $D\left(X\right)$ and $D\left(Y\right)$ values. However, if they are equal, we need to compare the coordinates (residues corresponding to modulus) of $X$ with respect to $Y$ in order to determine whether $X > Y$ or $X = Y$ or $X < Y$. Pirlo and Impedovo [PI13] have observed that diagonal function does not support RNS to binary conversion. Mohan [Moh16] shows that it is possible to perform RNS to binary conversion using diagonal function by equations:

$$X = \frac{P \cdot D\left(X\right) + \sum_{i=1}^{n} x_i P_i}{SQ}. \tag{4}$$

It is worth to note that RNS moduli set has the form $p_1 = 2^a - c$ and $p_2 = 2^a + c$, where $c$ is impair, and $SQ = 2^{a+1}$. Considering that $SQ$ is a power of 2, to convert a number from RNS to binary number system it is not necessary to compute residues from division by large numbers used in methods such as CRT and nCRT.

Since $\lim\limits_{a \to \infty} \frac{2a+1}{a+1} = 2$ , the size of coefficients is asymptotically twice smaller than in methods that allow to compute $X$ with lower complexity than CRT, nCRT and aCRT.

## 3 Algorithm for error detection, localization and correction for moduli set of the form $\left\{2^l - 3, 2^l - 1, 2^l + 1, 2^l + 3\right\}$

Taking into account the works on constructing a reliable, secure and distributed storage system in the clouds with erasure codes, Byzantine protocol etc. with parameters (2, 4), we develop the error correction code with a moduli set of a special form that allows to detect and correct errors using error syndrome.

We can extend the applicability of the error detection, localization and correction with error syndrome using moduli set of the form $\left\{2^l - 3, 2^l - 1, 2^l + 1, \;\; 2^l + 3\right\}$ for cloud computing.

1. Calculate the value of $X$, using moduli set $\left\{2^l - 1, 2^l + 1\right\}$: $SQ_{23} = 2^l - 1 + 2^l + 1 = 2^{l+1}$. Constants of Diagonal function: $k_2 = \left|-\frac{1}{2^l-1}\right|_2 l + 1 = 2^l + 1$, $k_3 = \left|-\frac{1}{2^l+1}\right|_2 l + 1 = 2^l - 1$.

$$D_{23}\left(X\right) = \left|k_2 x_2 + k_3 x_3\right|_2 l + 1 = \left|2^l \left(x_2 + x_3\right) + x_2 - x_3\right|_2^{l+1}.$$

Using the equation from the paper [Moh16], we find the value $X$.

$$X = \frac{\left(2^{2l} - 1\right) D_{23}\left(X\right) + \left(2^l + 1\right) x_2 \left(2^l - 1\right) x_3}{2^{l+1}}. \tag{5}$$

2. Calculate the value of $X$, using moduli set $\{2^l - 3, 2^l + 3\}$: $SQ_{14} = 2^l - 3 + 2^l + 3 = 2^{L+1}$. Constants of Diagonal function: $k_1 = \left|-\frac{1}{2^l-3}\right|_2 l + 1 = \left|-\frac{2^l+1}{3}\right|_2 l + 1$, $k_4 = \left|-\frac{1}{2^l+1}\right|_2 l + 1 = \left|-\frac{2^l-1}{3}\right|_2 l + 1$. Then, the diagonal function value is:

$$D_{14}\left(X\right) = \left|k_1 x_1 + k_4 x_4\right|_2 l + 1 = \left|r_3 (2^l (x_1 + x_4) + x_1 - x_4\right|_2^{l+1}$$

where $r_3 = \left|\frac{1}{3}\right|_{2^{l+1}} = \begin{cases} \frac{2^{l+2}+1}{3}, & \text{if } n - \text{odd} \\ \frac{2^{l+1}+1}{3}, & \text{if } n - \text{even.} \end{cases}$

Using the equation from the work [8], we find the value of $X$.

$$X = \frac{\left(2^{2l} - 9\right) D_{14}\left(X\right) + \left(2^l + 3\right) x_1 \left(2^l - 3\right) x_4}{2^{l+1}}. \tag{6}$$

The use of error correction codes in the RNS with the given parameters can detect two or correct one error. We study the data obtained from the cloud using the Eqn. (5) and (6) we find the error syndrome.

1. If an error occurs in $x_1$, then

$$\left|X - x_1\right|_{2^l-3} \neq 0. \tag{7}$$

As the gcd $\left(2^l-3, 2^{l+1}\right) = 1$, then the condition Eqn. (7) is equivalent to:

$$\left|2^{l+1}X - 2^{l+1}x_1\right|_{2^l-3} \neq 0. \tag{8}$$

Substitute Eqn. (5) in Eqn. (8), then

$$\left|\left(2^{2l}-1\right) D_{23}\left(X\right) + \left(2^l+1\right) x_2 + \left(2^l-1\right) x_3 - 2^{l+1}x_1\right|_{2^l-3} \neq 0.$$

As the $\left|2^l\right|_{2^l-3} = 3$ Eqn. (8) can be written as

$$s_1 = \left|8D_{23}\left(X\right) + 4x_2 + 2x_3 - 6x_1\right|_{2^l-3} \neq 0. \tag{9}$$

2. If an error occurs in $x_2$, then

$$\left|X - x_2\right|_{2^l-1} \neq 0. \tag{10}$$

As the gcd $\left(2^l-1, 2^{l+1}\right) = 1$, Eqn. (10) can be written as

$$\left|2^{l+1}X - 2^{l+1}x_2\right|_{2^l-1} \neq 0.$$

Substitute Eqn. (6) in Eqn. (8), then

$$\left|\left(2^{2l}-9\right) D_{14}\left(X\right) + \left(2^l+3\right) x_1 + \left(2^l-3\right) x_4 - 2^{l+1}x_2\right|_{2^l-1} \neq 0. \tag{11}$$

As the $\left|2^l\right|_{2^l-1} = 1$ Eqn. (11) can be written as

$$s_2 = \left|8D_{14}\left(X\right) - 4x_1 + 2x_4 + 2x_2\right|_{2^l-1} \neq 0. \tag{12}$$

3. If an error occurs in $x_3$, then

$$\left|X - x_3\right|_{2^l+1} \neq 0. \tag{13}$$

As the gcd $\left(2^l+1, 2^{l+1}\right) = 1$, Eqn. (13) can be written as

$$\left|2^{l+1}X - 2^{l+1}x_3\right|_{2^l+1} \neq 0. \tag{14}$$

Substitute Eqn. (6) in Eqn. (14), then

$$\left|\left(2^{2l}-9\right) D_{14}\left(X\right) + \left(2^l+3\right) x_1 + \left(2^l-3\right) x_4 - 2^{l+1}x_3\right|_{2^l+1} \neq 0. \tag{15}$$

As the $\left|2^l\right|_{2^l+1} = -1$ Eqn. (15) can be written as

$$s_3 = \left|8D_{14}\left(X\right) - 2x_1 + 4x_4 - 2x_3\right|_{2^l+1} \neq 0. \tag{16}$$

4. If an error occurs in $x_4$, then

$$\left|X - x_4\right|_{2^l+3} \neq 0. \tag{17}$$

As the gcd $\left(2^l+3, 2^{l+1}\right) = 1$, Eqn. (17) can be written as

$$\left|2^{l+1}X - 2^{l+1}x_4\right|_{2^l+3} \neq 0. \tag{18}$$

Substitute Eqn. (5) in Eqn. (18), then

$$\left|\left(2^{2l}-1\right) D_{23}\left(X\right) + \left(2^l+1\right) x_2 + \left(2^l-1\right) x_3 - 2^{l+1}x_4\right|_{2^l+3} \neq 0. \tag{19}$$

Table 1: Determination of error from error syndrome value

| No | $s_1$ | $s_2$ | $s_3$ | $s_4$ | Error | Value X |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | None | Eqn. (5) or (6) |
| 2 | 0 | 1 | 1 | 1 | $x_4$ | Eqn. (5) |
| 3 | 1 | 1 | 1 | 0 | $x_1$ | Eqn. (5) |
| 4 | 1 | 1 | 0 | 1 | $x_2$ | Eqn. (6) |
| 5 | 1 | 0 | 1 | 1 | $x_3$ | Eqn. (6) |
| 6 | 1 | 1 | 1 | 1 | Fail | - |

As the $\left|2^l\right|_{2^l+3} = -3$ Eqn. (8) can be written as

$$s_4 = \left|8D_{23}\left(X\right) - 2x_2 + 4x_3 - 6x_4\right|_{2^l+3} \neq 0. \tag{20}$$

Using the Eqn. (9), (12), (16) and (20) calculate the value of the error syndromes $s_1, s_2, s_3, s_4$. If $s_i \neq 0$, we assume that $s_i = 1$, for all $i = \overline{(1,4)}$ . Using Table 1, you can calculate the position of error. After excluding error true value can be restored

Example 1. Let $X = 92 \rightarrow (1, 2, 7, 16)$, RNS is defined by moduli set
$\left\{2^4 - 3, 2^4 - 1, 2^4 + 1, 2^4 + 3\right\}$.

1. If error vector is $E \rightarrow (1, 0, 0, 0)$, then $X' \rightarrow (2, 2, 7, 16)$. If the error vector is given by: $E \rightarrow (1, 0, 0, 0)$, then $X' = X + E \rightarrow (2, 2, 7, 16)$.

Compute diagonal function values:

$$D_{23}\left(X'\right) = \left|16(2 + 7) + 2 - 7\right|_{32} = \left|16 - 5\right|_{32} = 11.$$

Because $n = 4$, then $r_3 = \frac{2^{4+1}+1}{3} = 3$, consequently:

$$D_{14}\left(X'\right) = \left|11(16(2 + 16) + 2 - 16)\right|_{32} = \left|11 \cdot 18\right|_{32} = 6.$$

Compute error syndrome value:
$s_1 = \left|8 \cdot 11 + 4 \cdot 2 + 2 \cdot 7 - 6 \cdot 2\right|_{13} = 7$, since $7 \neq 0$, it follows that $s_1 = 1$.
$s_2 = \left|8 \cdot 6 - 4 \cdot 2 + 2 \cdot 16 + 2 \cdot 2\right|_{15} = 1$.
$s_3 = \left|8 \cdot 6 - 2 \cdot 2 + 4 \cdot 16 - 2 \cdot 7\right|_{17} = 9$, since $9 \neq 0$, it follows that $s_3 = 1$.
$s_4 = \left|8 \cdot 11 - 2 \cdot 2 - 4 \cdot 7 + 6 \cdot 16\right|_{19} = 0$.

Since $s_1 = s_2 = s_3 = 1$ and $s_4 = 0$, then the case 3 of Table 1, and therefore an error occurred in the $x_1$, therefore, the true value of $X$ is reduced using the Eqn. (5).

$$X = \frac{(2^8 - 1) \cdot 11 + (2^4 + 1) \cdot 2 + (2^4 - 1) \cdot 7}{2^5} = 92.$$

True value: $X = 92$.

2. If error vector is $E \rightarrow (0, 1, 0, 0)$, then we obtain $X' = X + E \rightarrow (1, 3, 7, 16)$.

Compute diagonal function values:

$$D_{23}\left(X'\right) = \left|16(3 + 7) + 3 - 7\right|_{32} = \left|32 - 4\right|_{32} = 28.$$

Because $n = 4$, then $r_3 = \frac{2^{4+1}+1}{3} = 11$, consequently:

$$D_{14}\left(X'\right) = \left|11(16(1 + 16) + 1 - 16)\right|_{32} = 11.$$

Compute error syndrome value:
$s_1 = \left|8 \cdot 28 + 4 \cdot 3 + 2 \cdot 7 - 6 \cdot 1\right|_{13} = 10$, since $10 \neq 0$, it follows that $s_1 = 1$.
$s_2 = \left|8 \cdot 11 - 4 \cdot 1 + 2 \cdot 16 + 2 \cdot 3\right|_{15} = 1$ since $2 \neq 0$, it follows that $s_2 = 1$.
$s_3 = \left|8 \cdot 11 - 2 \cdot 1 + 4 \cdot 16 - 2 \cdot 7\right|_{17} = 0$.
$s_4 = \left|8 \cdot 28 - 2 \cdot 3 - 4 \cdot 7 + 6 \cdot 16\right|_{19} = 1$.

Since $s_1 = s_2 = s_4 = 1$ and $s_3 = 0$, then the case 4 of Table 1, and therefore an error occurred in the $x_2$, therefore, the true value of $X$ is reduced using the Eqn. (6).

$$X = \frac{(2^8 - 9) \cdot 11 + (2^4 + 3) \cdot 1 + (2^4 - 3) \cdot 16}{2^5} = 92.$$

True value: $X = 92$.
3. If error vector is $E \to (0,0,1,0)$, then we obtain $X' = X + E \to (1,2,8,16)$.
Compute diagonal function values:

$$D_{23}(X') = |16(2+8) + 2 - 8|_{32} = |32 - 6|_{32} = 26.$$

Because $n = 4$, then $r_3 = \frac{2^{4+1}+1}{3} = 11$, consequently:

$$D_{14}(X') = |11(16(1+16) + 1 - 16)|_{32} = 11.$$

Compute error syndrome value:
$s_1 = |8 \cdot 26 + 4 \cdot 2 + 2 \cdot 8 - 6 \cdot 1|_{13} = 5$, since $5 \neq 0$, it follows that $s_1 = 1$.
$s_2 = |8 \cdot 11 - 4 \cdot 1 + 2 \cdot 16 + 2 \cdot 2|_{15} = 0$.
$s_3 = |8 \cdot 11 - 2 \cdot 1 + 4 \cdot 16 - 2 \cdot 8|_{17} = 15$ since $15 \neq 0$, it follows that $s_3 = 1$.
$s_4 = |8 \cdot 26 - 2 \cdot 2 - 4 \cdot 8 + 6 \cdot 16|_{19} = 8$ since $8 \neq 0$, it follows that $s_4 = 1$.
Since $s_1 = s_3 = s_4 = 1$ and $s_2 = 0$, then the case 5 of Table 1, and therefore an error occurred in the $x_3$, therefore, the true value of $X$ is reduced using the Eqn. (6).

$$X = \frac{(2^8 - 9) \cdot 11 + (2^4 + 3) \cdot 1 + (2^4 - 3) \cdot 16}{2^5} = 92.$$

True value: $X = 92$.
4. If error vector is $E \to (0,0,0,1)$, then we obtain $X' = X + E \to (1,2,7,17)$.
Compute diagonal function values:

$$D_{23}(X') = |16(2+7) + 2 - 7|_{32} = |16 - 5|_{32} = 11.$$

Because $n = 4$, then $r_3 = \frac{2^{4+1}+1}{3} = 11$, consequently:

$$D_{14}(X') = |11(16(1+17) + 1 - 17)|_{32} = 16.$$

Compute error syndrome value:
$s_1 = |8 \cdot 11 + 4 \cdot 2 + 2 \cdot 7 - 6 \cdot 1|_{13} = 0$.
$s_2 = |8 \cdot 16 - 4 \cdot 1 + 2 \cdot 17 + 2 \cdot 2|_{15} = 12$ since $12 \neq 0$, it follows that $s_2 = 1$.
$s_3 = |8 \cdot 16 - 2 \cdot 1 + 4 \cdot 17 - 2 \cdot 7|_{17} = 10$ since $10 \neq 0$, it follows that $s_3 = 1$.
$s_4 = |8 \cdot 11 - 2 \cdot 2 - 4 \cdot 7 + 6 \cdot 17|_{19} = 6$ since $6 \neq 0$, it follows that $s_4 = 1$.
Since $s_2 = s_3 = s_4 = 1$ and $s_1 = 0$, then the case 2 of Table 1, and therefore an error occurred in the $x_4$, therefore, the true value of $X$ is reduced using the Eqn. (5).

$$X = \frac{(2^8 - 1) \cdot 11 + (2^4 + 1) \cdot 2 + (2^4 - 1) \cdot 7}{2^5} = 92.$$

True value: $X = 92$.
Analyzing the values of error syndromes from the Table 1 we can note that in order to determine which parts should be used to obtain a true value it is sufficient to know either the syndromes $\{s_1, s_4\}$ or $\{s_2, s_3\}$ . In terms of computational complexity of computing $\{s_1, s_4\}$ are better. Thus using $\{s_1, s_4\}$, we determine which parts should be taken to obtain the true value. The results are shown in the Table 2.

Table 2: Determination of error from error syndrome value

| No | $s_1$ | $s_4$ | Correct chuncks | Value X |
|---|---|---|---|---|
| 1 | 0 | 0 | $x_2$, $x_3$ | Eqn. (5) |
| 2 | 0 | 1 | $x_2$, $x_3$ | Eqn. (5) |
| 3 | 1 | 0 | $x_2$, $x_3$ | Eqn. (5) |
| 4 | 1 | 1 | $x_1$, $x_4$ | Eqn. (6) |

Its worth noting that in Table 2 in case 4 to control the result of computations it is required to use the value of hash function MD5, which ensure the correctness of the data with probability $2^{-32}$.

## 4 Conclusion

In this paper, we propose approaches to construct reliable schemes for BOINC type volunteer computing systems using Residue Number System, and a scheme for controlling computations. We show how an application of RNS allows to build completely homomorphic information security systems. We present a new algorithm for localizing and correcting errors for moduli of a special type.

*Acknowledgment.* The work is partially supported by CONACYT (Consejo Nacional de Ciencia y Tecnología, México), grant No. 178415. Part of the work was supported State task No. 2.6035.2017 and Russian Federation President Grant SP-1215.2016.5. We gratefully acknowledge Evgeny Ivashko for valuable discussions and comments.

## References

[ASS$^{+}$14] Sameer Hasan Albakri, Bharanidharan Shanmugam, Ganthan Narayana Samy, Norbik Bashah Idris, and Azuan Ahmed. Security risk assessment framework for cloud computing environments. *Security and Communication Networks*, 7(11):2114–2124, 2014.

[BCT$^{+}$17] Mikhail Babenko, Nikolay Chervyakov, Andrei Tchernykh, Nikolay Kucherov, Maria Shabalina, Irina Vashchenko, Gleb Radchenko, and Daniil Murga. Unfairness correction in P2P grids based on residue number system of a special form. *1st International Workshop on Uncertainty in Cloud Computing, in conjunction with 28th International Conference on Database and Expert Systems Applications (DEXA17)*, pages 147–151, 2017.

[CBD$^{+}$16] Nikolay Ivanovich Chervyakov, Mikhail Grigor'evich Babenko, Maxim Anatolievich Deryabin, Anton Sergeevich Nazarov, and Maria Nikolaevna Shabalina. Computation of positional characteristics of numbers in RNS based on approximate method. In *NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW), 2016 IEEE*, pages 177–179. IEEE, 2016.

[CBKG15] Nikolay Ivanovich Chervyakov, Mikhail Grigorevich Babenko, Nikolay Nikolaevich Kucherov, and Anastasiia Igorevna Garianina. The effective neural network implementation of the secret sharing scheme with the use of matrix projections on FPGA. In *International Conference in Swarm Intelligence*, pages 3–10. Springer, 2015.

[CBT$^{+}$17] Nikolay Chervyakov, Mikhail Babenko, Andrei Tchernykh, Nikolay Kucherov, Vanessa Miranda-López, and Jorge M. Cortés-Mendoza. AR-RRNS: Configurable reliable distributed data storage systems for internet of things to ensure security. *Future Generation Computer Systems*, 2017.

[DIP93] Giovanni Dimauro, Sebastiano Impedovo, and Giuseppe Pirlo. A new technique for fast number comparison in the residue number system. *IEEE transactions on computers*, 42(5):608–612, 1993.

[Gen10] Craig Gentry. Computing arbitrary functions of encrypted data. *Communications of the ACM*, 53(3):97–105, 2010.

[GTN11] Mahadevan Gomathisankaran, Akhilesh Tyagi, and Kamesh Namuduri. HORNS: A homomorphic encryption scheme for cloud computing using residue number system. In *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, pages 1–5. IEEE, 2011.

[IG15] Evgeny Ivashko and Alexander Golovin. Partition algorithm for association rules mining in BOINC–based enterprise desktop grid. In *International Conference on Parallel Computing Technologies*, pages 268–272. Springer, 2015.

[KPT$^{+}$13] Dzmitry Kliazovich, Johnatan E Pecero, Andrei Tchernykh, Pascal Bouvry, Samee U Khan, and Albert Y Zomaya. CA-DAG: communication-aware directed acyclic graphs for modeling cloud computing applications. In *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on*, pages 277–284. IEEE, 2013.

[KPT$^{+}$16] Dzmitry Kliazovich, Johnatan E Pecero, Andrei Tchernykh, Pascal Bouvry, Samee U Khan, and Albert Y Zomaya. CA-DAG: modeling communication-aware applications for scheduling in cloud computing. *Journal of Grid Computing*, 14(1):23–39, 2016.

[Moh16] PV Ananda Mohan. RNS to binary conversion using diagonal function and pirlo and impedovo monotonic function. *Circuits, Systems, and Signal Processing*, 35(3):1063–1076, 2016.

[PI13] Giuseppe Pirlo and Donato Impedovo. Verification of static signatures by optical flow analysis. *IEEE Transactions on Human-Machine Systems*, 43(5):499–505, 2013.

[QPV02] Michaël Quisquater, Bart Preneel, and Joos Vandewalle. On the security of the threshold scheme based on the Chinese remainder theorem. In *International Workshop on Public Key Cryptography*, pages 199–210. Springer, 2002.

[SJJT86] Michael A Soderstrand, W Kenneth Jenkins, Graham A Jullien, and Fred J Taylor. *Residue number system arithmetic: modern applications in digital signal processing.* IEEE press, 1986.

[TBC$^+$17] Andrei Tchernykh, Mikhail Babenko, Nikolay Chervyakov, Jorge M. Cortés-Mendoza, Nikolay Kucherov, Vanessa Miranda-López, Maxim Deryabin, Inna Dvoryaninova, and Gleb Radchenko. Towards mitigating uncertainty of data security breaches and collusion in cloud computing. *1st International Workshop on Uncertainty in Cloud Computing, in conjunction with 28th International Conference on Database and Expert Systems Applications (DEXA17)*, pages 137–141, 2017.

[TPBS14] Andrei Tchernykh, Johnatan E Pecero, Aritz Barrondo, and Elisa Schaeffer. Adaptive energy efficient scheduling in Peer-to-Peer desktop grids. *Future Generation Computer Systems*, 36:209–220, 2014.

[TSAT15] Andrei Tchernykh, Uwe Schwiegelsohn, Vassil Alexandrov, and El-ghazali Talbi. Towards understanding uncertainty in cloud computing resource provisioning. *Procedia Computer Science*, 51:1772–1781, 2015.

[TSTB16] Andrei Tchernykh, Uwe Schwiegelsohn, El-ghazali Talbi, and Mikhail Babenko. Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability. *Journal of Computational Science*, 2016.