The Implementation of Information and Communication Technologies with the Use of Modular Codes

Dmitriy Yurdanov[1] Maksim Kalmykov[1] Dmitriy Gostev[1] Igor Kalmykov[1]

[1] North-Caucasian
Federal University
Stavropol
Russian Federation

Abstract

The purpose of the study is to improve the speed and accuracy of the implementation of information technology through the use of modular code. The paper presents the developed *Orthogonal Frequency Division Multiplexing (OFDM)* algorithm in the codes of *residue number system (RNS)*. The studies have shown that the performance of *OFDM* based on number-theoretic transforming in RNS code allows you to perform orthogonal transformational changes of signals without calculating the real and imaginary parts of the spectrum. In addition, the transition to integer calculations eliminates round-off errors that were caused by the use of irrational numbers when submitting twiddle *OFDM* coefficients. It is shown that the use of new modular technologies in protocols used in electronic payment systems allows for calculations in real-time by parallelizing the operations at the level of processing short numbers.

1 Introduction

Expanding the scope of information technologies and systems is largely determined by progress in the field of computer technology, as well as the acceleration of the process of informatization of modern society. Increasing requirements for technical and economic characteristics of modern communication systems has led to the need for parallel computing. To provide the data processing and transmission time in a real scale, the process of parallelizing can be performed at mixed levels. The most effective results can be obtained by using modular codes that provide parallelization at the level of arithmetic operations. Therefore, the algorithm elaboration for improving the efficiency of information and communication systems through the use of modular codes is an urgent task.

2 The Purpose of the Study

The most prominent feature of recent years is the expanding of spheres for modular arithmetic application. Currently, we can distinguish two large groups in position-independent modular codes. The basis of the first group is position-independent codes of the residue number system (RNS). In producing of such codes of residue

In: S. Hölldobler, A. Malikov, C. Wernhard (eds.): *YSIP2 – Proceedings of the Second Young Scientist's International Workshop on Trends in Information Processing, Dombai, Russian Federation, May 16–20, 2017*, published at http://ceur-ws.org.

classes, mutually prime integers are applied in the function of basis [Moh02]. Due to this fact, any code can be represented as a set of residues obtained by dividing this number by the based number

$$A = (\alpha_1,\, \alpha_2, ..., \alpha_n), \tag{1}$$

where $\alpha_i \equiv A \bmod p_i$; $i = 1, ..., n$.

The basis of the second group of position-independent codes comprises some modular polynomial codes such as codes of polynomial system of residue classes (PSRC) [Kal14]. In producing of such codes of residue classes, prime polynomials are applied in the function of basis. Because of this, any positional code at the beginning appears in polynomial form. Then the obtained polynom is put in correspondence with the set of residues obtained by dividing this number by the based number

$$A(z) = (\alpha_1(z),\, \alpha_2(z), ..., \alpha_n(z)), \tag{2}$$

where $\alpha_i(z) \equiv A(z) \bmod p_i(z)$; $i = 1, ..., n$.

Despite their differences, these modular codes have much in common. These codes, due to the parallel and independent processing of residues, can increase the speed of the following modular operations

$$|A \otimes B|^{+}_{p_i} = |\alpha_i \otimes \beta_i|^{+}_{p_i}\,, \tag{3}$$

where $A = (\alpha_1, \alpha_2, ..., \alpha_n) B = (\beta_1,\ \beta_2, ...,\ \beta_n)$ modular code in the residue ring; $\alpha_i \equiv A \bmod p_i$; $\otimes$ operations of addition, subtraction, multiplication with the module of RNS base code p_i; $i = 1, ..., n$.

At that, the data processing provides the minimum error, as it occurs with integers along with the module of RNS base code.

Thus, it is clear that the use of modular code improves the speed and accuracy of the data processing in those algorithms of information and communication systems that use only addition, subtraction and multiplication operations.

Therefore, the aim of this work is to improve the speed and accuracy of algorithms implemented in information and communication technologies through the use of modular position-independent codes.

3 Data and Methods of the Study

It is known that modular arithmetic codes are codes used to perform calculations. Low digit capacity of the processed residues allows for calculations in parallel and independently in computing channels that are defined by the code base in real time. These features of modular codes have predetermined the areas in which they get limiting specifications of information and communication systems. The studies allow to identify the most promising areas in which the modular codes have their most evident advantages.

The basis of the first direction is the classic techniques and digital signal processing algorithms (DSP) using some orthogonal transformational changes of signals in the field of complex numbers [Omo07,Bri02,Fri05]. In this case, the turning coefficients are represented by integers which are then converted into a RNS code. Using the modular code allows for high speed signal processing in a digital signal processing system (DSP). There is an example in the work [Kat13] about application of the RNS modular code in the system of secondary processing of navigation data. Using the RNS code has allowed to increase the computing speed and reduce errors in determining the space-time coordinates of the consumer.

The second area of application of the modular codes is associated with producing of fault-tolerant computing systems [Kal14,Ber04,Ste16]. The introduction of additional surplus bases in the modular code allows you to search for and correct errors that may arise due to the occurrence of faults and failures during operation of computer systems. As a result, such devices have the function of stability to failure. In the work [Ste16] the use of PSRC codes for error correction is shown. They arise when attacks such as failures with AES encryption algorithm occur.

As the base of the third direction we can put algorithms and methods of using modular codes in conducting a large-scale analysis of signals. So, the works [Han05,Kal15] show the feasibility of using modular arithmetic in the implementation of discrete wavelet decomposition (WPT). The increased interest in WPT implemented in the residue ring is due to the fact that such orthogonal transformation signals allow us to calculate the time-frequency characteristics of the signals with fewer errors.

Let us consider the possibility of using modular arithmetic in information and communication systems which use the method of *Orthogonal Frequency Division Multiplexing* (OFDM). OFDM based on Fast Fourier Transform

is the most popular installation (IFFT-Inverse Fast Fourier Transform) [Tsa05, Tar03]. So, FFT implementation is determined by

$$X(k) = \sum_{n=0}^{N/2-1} x_{v-1,0}(n) W_N^{2nk} + \sum_{n=0}^{N/2-1} x_{v-1,1}(n) W_N^{2(n+1)k}, \tag{4}$$

where $W_N^2 = e^{-\frac{2\pi}{N/2}}$;$x_{v-1,0}(n) = x(2nT),\ ; x_{v-1,1}(n) = x((2n+1)T)$ sequence with the even and odd numbers accordingly.

However, implementation of FFT is characterized by two computing lanes that affect the cost and reliability of the circuitry special processor (SP) of OFDM. Furthermore, irrational numbers are used in the FFT as twiddle factors; that reduces the accuracy of calculations. To eliminate these deficiencies is possible by using the algebraic system which has the function of a ring or a field in the implementation of OFDM.

Let us assume that $GF(M)$ is a finite Galois field, G_N cyclic group of order N, $\varepsilon = \sqrt[N]{1} \in GF(M)$. Then OFDM conversion can be performed using a number-theoretic transformation (NTT). In this case, we obtain

$$X(k) = \left(\sum_{n=0}^{N-1} x(n) \times \varepsilon^{-kn} \right) \bmod \mathrm{M}, \tag{5}$$

where $[x(0),\ x(1),\ ...,\ x(N-1)]$ - the input vector of X signal; $x(n) \in G_N$.

The reverse number-theoretic transformation has the following form

$$x(n) = \left(N^{-1} \sum_{k=0}^{N-1} X(k) \times \varepsilon^{kn} \right) \bmod M. \tag{6}$$

Properties of NTT are isomorphic to DFT properties. Particularly NTT can be calculated by fast algorithms the same algorithms that were used in the computation of the Fourier transform [Nus78]. Moreover, NTT by its structure is implemented by using of digital hardware components. For example, if we take εas a power of two, the multiplication by (5) in the degree $\varepsilon$when calculating NTT is replaced by shifts of code words and their further actuation in the module of M number.

Increase the speed of number-theoretic transformation is possible due to the use of modular application developed algorithm codes. If M number is a compound for which the numbers of Mersenne are widely used, then the expression (5) can be reduced to a multidimensional parallel processing. In this case, transformational changes of the signal in the ringZ_M is isomorphic to a transformation in the amount of rings$Z_{p_1} \dot{+} Z_{p_2} + ... \dot{+} Z_{p_L}$, where$M = \prod_{i=1}^{L} p_i$. Then it is fair enough for RNS code

$$\begin{aligned} X_1(k) &= \left(\sum_{n=0}^{N-1} x_1(n) \times \varepsilon_1^{-kn} \right) \bmod p_1 \\ &\vdots \\ X_L(k) &= \left(\sum_{n=0}^{N-1} x_L(n) \times \varepsilon_L^{-kn} \right) \bmod p_L \end{aligned}, \tag{7}$$

where $x_i(n) \equiv x(n) \bmod p_i;\ \varepsilon_i^{-kn} \equiv \varepsilon^{-kn} \bmod p_i;\ X_i(n) \equiv X(n) \bmod p_i;$.

The inverse transformation is given by

$$\begin{aligned} x_1(n) &= \left(N_1 \sum_{k=0}^{N-1} X_1(k) \times \varepsilon_1^{kn} \right) \bmod p_1 \\ &\vdots \\ x_L(n) &= \left(N_L \sum_{k=0}^{N-1} X_L(k) \times \varepsilon_L^{kn} \right) \bmod p_L \end{aligned}, \tag{8}$$

where $N_i(n) \equiv (N^{-1}) \bmod p_i;\ \varepsilon_i^{kn} \equiv \varepsilon^{kn} \bmod p_i;\ i = 0, ...,\ L.$

For moving from the modular code in the position code, you can use the Chinese remainder theorem

$$x(n) = \sum_{i=1}^{L} x_i(n) B_i \bmod M, \tag{9}$$

where B_i- orthogonal basis of the 1st base code; $B_i \equiv 1 \bmod p_i;\ i = 0, ...,\ L$.

Another area where modular codes can be effectively applied is electronic payment system (EPS). The electronic payment system is a set of methods, algorithms and protocols which allows to perform payment transactions between counterparties using e-money [Wan11]. Let us consider the possibility of using modular codes in the development of EPS protocols.

In the work [Sar14] the protocol of "withdrawals" is presented which uses the proof with zero knowledge proofs. This protocol is used when receiving an electronic purse in the bank. In order to increase its effectiveness the protocol with modular codes was developed.

For obtaining of e-purse, an owner of electronic money chooses a K secret key. Then it calculates the value of a public key which is transmitted to the bank

$$K_U = g^K \bmod q \tag{10}$$

where $q-$ prime number; $g-$ primitive element that generates q multiplicative group.

In the first step the user selects a base protocol $p_1, ..., p_L$, so that the P range of RNS code satisfies the condition

$$P = \prod_{i=1}^{L} p_i > q \tag{11}$$

where p_i- are primes in which g is the primitive element.

Then the user calculates the value of the parameter which is called "delivery" *(Pedersen)* to the residue number system

$$\begin{array}{l} C_1 = g^{K_1} g^{S_1} g^{T_1} \bmod p_1 \\ \vdots \\ C_L = g^{K_L} g^{S_L} g^{T_L} \bmod p_L \end{array} \tag{12}$$

where $C_i \equiv C \bmod p_i; K_i \equiv K \bmod p_i; S_i \equiv S \bmod p_i; T_i \equiv C \bmod p_i$; C − delivery; S − parameter that is used for calculation the number of the electronic coin; T − parameter that is used for detection of double payment of the coin.

This "delivery" in the form of RNS code $(C_1, C_2, \ldots, C_L)$ is sent to the bank. The user does not reveal its sensitive data to the bank.

Then, the user carries out "noise masking" of their sensitive data, i.e. he/she changes the value of K secret key, S and T numbers. It uses random values $\Delta K_i,\ \Delta S_i,\ \Delta T_i$.

$$\begin{array}{l} K_i^* = (K_i + \Delta K_i) \bmod p_i \\ S_i^* = (S_i + \Delta S_i) \bmod p_i \\ T_i^* = (T_i + \Delta T_i) \bmod p_i \end{array} . \tag{13}$$

The result is values $K^* \neq K$, $S^* \neq S$, $T^* \neq T$. After that the user calculates a new "noisy delivery" according to

$$\begin{array}{l} C_1^* = g^{K_1^*} g^{S_1^*} g^{T_1^*} \bmod p_1 \\ \vdots \\ C_L^* = g^{K_L^*} g^{S_L^*} g^{T_L^*} \bmod p_L \end{array} \tag{14}$$

where $C_i^* \equiv C^* \bmod p_i$.

In the next stage, the bank sends the user the number$d \in Z_q$. This number serves as a question that the user must answer. If he knows the secret value of the K key, S and T numbers, he will be able to answer the "question".

The user starts the calculation of the response to d question.

$$\begin{array}{l} r_i(1) = (K_i^* - dK_i) \bmod \varphi(p_i), \\ r_i(2) = (S_i^* - dK_i) \bmod \varphi(p_i), \\ r_i(3) = (T_i^* - dT_i) \bmod \varphi(p_i). \end{array} \tag{15}$$

The responses to this d question are transferred to the bank. The bank then proceeds to verification of evidence of true of the user.

$$\begin{array}{l} A_1 = (C_1^d g^{r_1(1)} g^{r_1(2)} g^{r_1(3)}) \bmod p_1 \\ \vdots \\ A_L = (C_k^d g^{r_L(1)} g^{r_L(2)} g^{r_L(3)}) \bmod p_L \end{array} \tag{16}$$

If the user is the actual owner of the secret parameters of K key, S and T numbers, the equality is fair enough

$$A_i = C_i^* \bmod p_i. \tag{17}$$

4 Results and Discussion

Let us consider the performance of OFDM based on single-module number-theoretic transformation using modular codes. We choose Mersenne number as a module $M = 255 = 3 \cdot 7 \cdot 17$. For this module M there is a 16-point NTT. Since $2^{16} \bmod 255 = 1$, then select $\varepsilon = 2$ root of unity of order N = 16. Let us assume an input vector submitted in $Z_M = Z_{255}$, $\overline{X} = \{x(0), x(1), x(2), \cdots, x(15)\} = \{0, 1, 2, 3, \cdots, 15\}$.

We carry out NTT according to the formula

$$X(k) = \left(\sum_{n=0}^{N-1} x(n) \cdot \varepsilon^{-kn} \right) \bmod 255. \tag{18}$$

As a result of calculations according to the formula (18) we obtain NTT range:

$\{X(0), \cdots, X(15)\} = \{120, 223, 177, 91, 60, 148, 147, 16, 120, 223, 177, 91, 60, 148, 147, 16\}$.

We implement NTT in the ring$Z_3 \dot{+} Z_5 \dot{+} Z_{17}$, that is in RNS code for modules $p_1 = 3, p_2 = 7, p_3 = 17$. Then the input signal appears in the code.

$\overline{X} \bmod 3 = \{0, 1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 2, 0\}$,

$\overline{X} \bmod 5 = \{0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0\}$,

$\overline{X} \bmod 17 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$.

We use the expression (7) and shall make the calculation of NTT in the residual classes system. As a result, we obtain

$\{X_1(0), X_1(1), \cdots, \mathrm{X}_1(15)\} = \{0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1\}$,

$\{X_2(0), X_2(1), \cdots, \mathrm{X}_2(15)\} = \{0, 3, 2, 1, 0, 3, 2, 1, 0, 3, 2, 1, 0, 3, 2, 1\}$,

$\{X_3(0), X_3(1), \cdots, \mathrm{X}_3(15)\} = \{1, 2, 7, 6, 9, 12, 11, 16, 1, 2, 7, 6, 9, 12, 11, 16\}$.

Let us give the NTT report in RNS code. We get X(0) = 120 = (0. 0. 1). Thus it is clear that by using of RNS the values were obtained identical with the results of NTT in M module = 255.

Let us consider the work of the developed protocol using RNS code. Let the bases were chosen $p_1 = 11, p_2 = 13, p_3 = 19$. Then the range of RNS will equal P = 2717. Let us take g = 2 as a primitive element of the group. Let the secret key value is K = 3. Let the values S = 5 and T = 5. We submit these parameters in RNS: K = (3, 3, 3), S = (5, 5, 5), T = (5, 5, 5). We use (3) to calculate the submission

$C_1 = g^{K_1} g^{S_1} g^{T_1} \bmod p_1 = (2^3 * 2^5 * 2^5) \bmod 11 = 2^3 \bmod 11 = 8$

$C_2 = g^{K_2} g^{S_2} g^{T_2} \bmod p_2 = (2^3 * 2^5 * 2^5) \bmod 13 = 2^1 \bmod 13 = 2$

$C_3 = g^{K_3} g^{S_3} g^{T_3} \bmod p_3 = (2^3 * 2^5 * 2^5) \bmod 19 = 2^{13} \bmod 19 = 3$

The result of C = (8, 2, 3) is sent to the bank.

Then, the user carries out "noisy making" of their sensitive data, i.e. changes the value of K key = (3, 3, 3), the numbers S = (5, 5, 5) and T = (5, 5, 5). In this case, the values are used $\Delta K = 2,\ \Delta S = 2,\ \Delta T = 2$. Then, according to (13) we get noise values of K^* = (5, 5, 5), S = (7, 7, 7) and T = (7, 7, 7).

After that the user calculates the "noisy delivery" in accordance with (14)

$C_1^* = g^{K_1^*} g^{S_1^*} g^{T_1^*} \bmod p_1 = (2^5 * 2^7 * 2^7) \bmod 11 = 2^9 \bmod 11 = 5$

$C_2^* = g^{K_2^*} g^{S_2^*} g^{T_2^*} \bmod p_2 = (2^5 * 2^7 * 2^7) \bmod 13 = 2^7 \bmod 13 = 11$

$C_3^* = g^{K_3^*} g^{S_3^*} g^{T_3^*} \bmod p_3 = (2^5 * 2^7 * 2^7) \bmod 19 = 2^1 \bmod 19 = 2$

The resulting noisy presentation of C^* = (5, 11, 2) is sent to the bank.

In the next stage, the bank sends the user the number d = 10.

The user starts the calculation of the response to the question of d = 10. The first answer in the RNS code equals

$r_1(1) = (K_1^* - dK_1) \bmod \phi(11) = (5 - 10 \cdot 3) \bmod 10 = 5$

$r_2(1) = (K_2^* - dK_2) \bmod \phi(13) = (5 - 10 \cdot 3) \bmod 12 = 11$

$r_3(1) = (K_3^* - dK_3) \bmod \phi(19) = (5 - 10 \cdot 3) \bmod 18 = 11$

The second answer in the RNS code equals

$r_1(2) = (S_1^* - dS_1) \bmod \phi(11) = (7 - 10 \cdot 5) \bmod 10 = 7$

$r_2(2) = (S_2^* - dS_2) \bmod \phi(13) = (7 - 10 \cdot 5) \bmod 12 = 5$

$r_3(2) = (S_3^* - dS_3) \bmod \phi(19) = (7 - 10 \cdot 5) \bmod 18 = 11$

The third answer in the RNS code equals

$r_1(3) = (T_1^* - dT_1) \bmod \phi(11) = (7 - 10 \cdot 5) \bmod 10 = 7$
$r_2(3) = (T_2^* - dT_2) \bmod \phi(13) = (7 - 10 \cdot 5) \bmod 12 = 5$
$r_3(3) = (T_3^* - dT_3) \bmod \phi(19) = (7 - 10 \cdot 5) \bmod 18 = 11$

The responses to this question (5,11,11), (7,5,11), (7,5,11) are transmitted to the bank. The bank then proceeds to verification of evidence of true of the user. To do this we need to calculate

$A_1 = C_1^d g^{r(1)_1} g^{r(2)_1} g^{r(3)_1} \bmod p_1 = (8^{10} * 2^5 * 2^7 * 2^7) \bmod 11 = 2^9 \bmod 11 = 5$
$A_2 = C_2^d g^{r(1)_2} g^{r(2)_2} g^{r(3)_2} \bmod p_2 = (2^{10} * 2^{11} * 2^5 * 2^5) \bmod 13 = 2^7 \bmod 13 = 11$
$A_3 = C_3^d g^{r(1)_3} g^{r(2)_3} g^{r(3)_3} \bmod p_3 = (3^{10} * 2^{11} * 2^{11} * 2^{11}) \bmod 19 = 2^1 \bmod 19 = 2$

Since the user is the actual owner of the secret parameters of K key, S and T numbers, the equality is fair enough

$$A_1 = C_1^* \bmod p_1 = 5,$$
$$A_2 = C_2^* \bmod p_2 = 11,$$
$$A_3 = C_3^* \bmod p_3 = 2.$$

After checking the bank gives the owner an e-purse.

5 Discussion of Results

The studies have shown that the use of RNS code can increase the speed of implementation of the orthogonal transformation of OFDM signals and the protocol "withdrawals" by parallel computing on the basis of RNS. It is known that the speed of the operation of multiplication according to the module is proportional to the digit capacity of operands. When using a single-module protocol of "withdrawals" as given in the work, the digit capacity of operands is equals to $L_1 = \lceil \log_2 q \rceil$ bit. The maximum digit capacity of operands in the developed protocol will be determined by a senior basis of RNS and it will be $L_2 = \lceil \log_2 p_k \rceil$. It is obvious that$L_1 > L_2$. As a result, when using q $L_1 = 64$ with digit capacity we can apply the RNS code (389,419,421,442,461,467,491,509). At that, this capacity of every basis of RNS equals to $L_2 = 9$ bits. Even with the additional time spent on the implementation of the transformation of the positional code numbers R, S, T in the modular code, the developed protocol will require less time for implementation.

In addition, the introduction of additional control bases in the modular code will perform an operation of control of the reliability of the obtained results. That is, using modular code in OFDM systems and electronic payment systems, we can improve the reliability of the data being processed.

6 Conclusion

The paper deals with information technology in which the modular codes are used effectively. An algorithm for performing orthogonal transformation of OFDM signals is presented here, as well as the protocol of "withdrawals" of electronic money which use RNS codes. The studies have shown that the use of the modular code allows you to increase the speed of implementation of information technology at the expense of parallel computing according to the bases of RNS. Thus arithmetic operations are performed on residues which have much smaller capacity than the original data. Furthermore, the use of integers allows to eliminate some rounding errors when performing OFDM.

References

[Moh02] P.V. Mohan. *Residue Number Systems. Algorithms and Architectures.* Springer, 2002.

[Omo07] A. Omondi, B. Premkumar *Residue Number Systems: Theory and Implementation.* Imperial College Press. UK, 2007

[Moh16] P.V. Mohan. *Residue Number Systems. Theory and Applications.* Springer, 2016.

[Kal14] I. Kalmykov, K. Katkov, D. Naumenko, A. Sarkisov, A. Makarova *Parallel Modular Technologies in Digital Signal Processing.* Life Science Journal, 2014.

[Ber04] V. Berezhnoy, N. Chervyakov, Yu. Shchelkunova, A. Shilov *Neural Network Realization in the Polynomial Residue Number System of the Digital Signal Processing Operations with Increased Number of Digits.* Neurocomputers: Development, Application, 2004.

[Bri02] E. Brigham *The Fast Fourier Transform.* New York: Prentice-Hall, 2002.

[Chu09] J. Chu, M. Benaissa *Polynomial Residue Number System GF(2m) Multiplier Using Trinomials.* In 17th European Signal Processing Conference, 2009.

[Fri05] M. Frigo, S. Johnson *The Design and Implementation of FFTW3.* Proceedings of the IEEE, 2005.

[Kat13] K. Katkov, I. Kalmykov *Application of Parallel Technologies in Navigation Management under the Conditions of Artificial Ionospheric Disturbances.* World Applied Sciences Journal, 2013.

[Ste16] E. Stepanova, I. Kalmykov, E. Toporkova, M. Kalmykov, R. Katkov, D, Rezenkov *Application of the codes of a polynomial residue number system, aimed at reducing the effects of failures in the AES cipher.* Journal of Digital Information Management, 2016.

[Sta05] H. Stark *Wavelets and signal processing.* Springer International Publishing Switzerland, 2005.

[Kal15] I. Kalmykov, K. Katkov, L. Timoshenko, A. Dunin, T. Gish *Application of Modular Technologies in the Large-Scale Analysis of Signals.* Journal of Theoretical and Applied Information Technology, 2015.

[Tsa05] C. Tsai, B. Huang *Concatenated codes design for OFDM based wireless local area networks.* Third international working conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs), 2005.

[Tar03] B. Tarokh, H.R. Sadjadpour *Construction of OFDM M-QAM sequences with low peak-to-average power ratio.* IEEE Trans. on Communications, 2003.

[Nus78] H.J. Nussbaumer *Fast multipliers for number theoretic transform.* IEEE Trans. Comput., 1978.

[Frn05] T. Frnqvist *Number theory meets cache locality - efficient implementation of a small prime FFT for the GNU Multiple Precision Arithmetic Library.* Masters thesis, Stockholm University, 2005.

[Alf96] L. Alfredson *VLSI architectures and arithmetic operations with application to the Fermat number transform.* Linkooping Studies in Sci. and Technology, Dissertation No.425, 1996.

[Arn11] J. Arndt *Matters Computational.* Springer, 2011.

[Wan11] Q. Wang *Compact k-spendable E-cash with Anonymity Control Based Offline TTP.* International Journal of Innovative Computing, Information and Control, 2011.

[Sar14] A. Sarkisov, A. Makarova *Extension of the Methods of Protection of the E-commerce Systems Based on the Modular Algebraic Schemes.* Proceedings of the Southern Federal University. Technical sciences, 2014.