PRACE NAUKOWE Akademii im. Jana Długosza w Częstochowie
Technika, Informatyka, Inżynieria Bezpieczeństwa 2016, t. IV, s. 415–426

http://dx.doi.org/10.16926/tiib.2016.04.34

Mikhail Selianinau
Wydział Matematyczno-Przyrodniczy
Akademia im. Jana Długosza w Częstochowie
al. Armii Krajowej 13/15, 42-200 Częstochowa
e-mail: m.selianinov@ajd.czest.pl

APPLICATION OF MODULAR COMPUTING TECHNOLOGY FOR CREATING A CRYPTOGRAPHIC INFORMATION SECURITY SYSTEM

Abstract. In the present paper, we deal with the application of the parallel modular computing structures for creating a cryptographic information security system. The proposed computer-arithmetical base of the modular computing technology within the minimal redundant modular coding allows us to attain an essential increase of performance, degree of internal parallelism and data encryption rate of cryptographic algorithms. The designed parallel encrypting algorithm is characterized by a maximum level of unloading of the real time computing process from the labor-consuming calculations, which can be realized by means of the look-up tables formed at a stage of preliminary calculations.
Keywords: information security, cryptosystem, cryptogram, encryption, modular arithmetic, modular number system.

ZASTOSOWANIE MODULARNEJ TECHNOLOGII OBLICZENIOWEJ DO TWORZENIA KRYPTOGRAFICZNYCH SYSTEMÓW OCHRONY INFORMACJI

Streszczenie. W niniejszej pracy mamy do czynienia z zastosowaniem równoległych modularnych struktur obliczeniowych przy tworzeniu kryptograficznych systemów ochrony informacji. Zaproponowano komputerowo-arytmetyczną bazę modularnej technologii obliczeniowej, która, w ramach minimalnie nadmiernego kodowania modularnego, pozwala znacznie poprawić skuteczność oraz zwiększyć stopień równoległości i szybkość realizacji przekształceń kryptograficznych. Opracowany algorytm szyfrowania charakteryzuje się maksymalnym zmniejszeniem rzędu złożoności obliczeniowej

przez zrealizowanie w czasie rzeczywistym skomplikowanych i żmudnych obliczeń za pomocą wstępnie wygenerowanych tabel.
Słowa kluczowe: bezpieczeństwo informacji, kryptosystem, kryptogram, szyfrowanie, modularna arytmetyka, modularne systemy liczbowe.

Introduction

The development of computer technology, emergence of new multimedia means and high-bandwidth networks has led to the need to apply the new technologies providing the real time processing and transmission of large volumes of data in the modern information systems. Therefore, the problem of ensuring confidentiality and integrity of information requires the high performance of used data security systems.

When the cryptographic information security systems (CISS) is created it is necessary increasingly more often to solve the problems requiring fast carrying out of large volumes of high-precision calculations. The conventional methods for implementation of such tasks based on positional arithmetic are not adequately efficient and in many cases are simply unacceptable owing to internal structure of sequential computer algorithms. The actuality of creation and implementation in practice of essentially new computing technologies of fast parallel high-precision calculations in CISS, first of all, a modular technology of information processing, is also defined by this circumstance [2, 5, 7, 9, 10].

In this paper, we consider the possibility of using modular computing structures (MCS), in possession of the maximal level of internal parallelism, in constructing of CSII. The computer-arithmetical base of modular computing technology (MCT) which within the minimum excess modular coding allows us significantly to increase performance, degree of parallelism and execution speed of cryptographic algorithms is offered.

Principles of block chipper in CISS

The mathematical data encryption algorithms that prevent confidential information leakage are the basis behind the methods used in CISS. The plaintext message is a linear sequence of symbols from some alphabet. As examples of the alphabets used in modern information systems we can cite the following alphabets such as the common alphabet of Latin letters, the Cyrillic alphabet, the ASCII alphabet, etc. [1, 3, 4, 8, 17].

Let us denote the plaintext message by T, and corresponding to it ciphertext (cryptogram) by C. Then, the encryption can be represented as a function E which converts a plaintext T into a ciphertext C:

$$C = E(T, K_E) \tag{1}$$

Similarly, the decryption can be represented as a function D which converts a ciphertext C into a plaintext T:

$$T = D(C, K_D). \tag{2}$$

In formulas (1) and (2) K_E and K_D designate an encryption and decryption keys respectively. In the case when $K_E = K_D$ we have a symmetric cryptographic system, otherwise when $K_E \neq K_D$ a cryptographic system is asymmetric [1, 3, 17].
The development of data encryption algorithms based on the rational choice of functions which transform the plaintext message to the cryptogram. The idea of direct application of such a function to the whole message is implemented very rarely. In practice, all applicable cryptographic transformations are associated with the partition of a message into a lot of fixed-length blocks, each one of them is encrypted separately. This approach simplifies essentially the encryption problem and allows an unlimited length data burst enciphering.

Let us consider a method of block encryption of a data stream. The plaintext T is partitioned into blocks $T_1, T_2, \dots, T_s$, where T_j $(j = 1, 2, \dots, s)$ represents a finite sequence of characters of some alphabet having a specified length. At the same time the formation of the corresponding sequence of cryptograms $C_1, C_2, \dots, C_s$ requires s encryption operations. Therefore, the transformation (1) can be rewritten in the following form:

$$\begin{cases} C_1 = E(T_1, K_{E,1}), \\ C_2 = E(T_2, K_{E,2}), \\ \quad \dots \ \dots \ \dots \\ C_s = E(T_s, K_{E,s}). \end{cases} \tag{3}$$

It should be noted that the plaintext blocks $T_1, T_2, \dots, T_s$ can be formed out of messages belonging to the same user or out of independent messages belonging to s different users. Both a one key $K_E = K_{E,1} = K_{E,2} = \dots = K_{E,s}$ and a set $\{K_{E,1}, K_{E,2}, \dots, K_{E,s}\}$ of s keys corresponding to the number of plaintext blocks can be used for encryption. At the same time, it does not matter whether in series or in parallel each of enciphering processes is occurred.

The process of block decryption of a data stream may be also described similarly. The transformation (2) can be represented as follows

$$\begin{cases} T_1 = D(C_1, K_{D,1}), \\ T_2 = D(C_2, K_{D,2}), \\ \quad \dots \ \dots \ \dots \\ T_s = D(C_s, K_{D,s}). \end{cases} \tag{4}$$

At the same time, the sequential or parallel implementation of (4) is possible too.

Block ciphers are the basis on which realized almost all modern cryptosystems and their characteristic feature consists in the fast processing speed. In the block cryptographic algorithms all operations performed on data are based on the fact that the transformable block can be represented as a non-negative integer number which belongs to the range corresponding to its digit capacity.

The plaintext block coding

Let us an alphabet $A = \{a_1, a_2, \dots, a_k\}$ containing k unique characters be generally set. Let us consider the plaintext block T_j $(j = 1, 2, \dots, s)$ representing a character string of length l over the alphabet A. It is possible to use the following method of encoding of a given string. For this purpose, we will set a numbering on the alphabet, i.e. we will assign a number from 0 to $k - 1$ to each character of the alphabet A: $a_1 \to 0, a_2 \to 1, \dots, a_k \to k - 1$. Now the string represents a sequence of numbers of the $\lceil \log_2 k \rceil$-bit length from the set $\{0, 1, \dots, k - 1\}$ ($\lceil x \rceil$ designates the nearest to x integer at the right). In this case, the number of bits for representation of the same plaintext block is equal to $l \cdot \lceil \log_2 k \rceil$.

Another method of encoding consists in representation of the considered string T_j as some natural number X_j in the number system with a base p $(j = 1, 2, \dots, s)$. At the same time, we have:

$$X_j = \sum_{i=0}^{l-1} T_j[i] \cdot p^i, \qquad (5)$$

where $0 \le X < p^l, T_j[i]$ is a number corresponding to the ith character of string T_j, $T_j[i] \in \boldsymbol{Z}_p = \{0, 1, \dots, p - 1\}$. Thus, some integer X_j from the range $[0, p^l)$ corresponds to the specified string T_j of length l over the alphabet A. It is clear, that to the different blocks there will correspond the different integer numbers and vice versa. Therefore, now it is possible to operate with these numbers instead of character strings. For example, in order to transfer a block T_j over a communication channel it is enough to transmit the corresponding number X_j which requires $L = \lceil \log_2 p^l \rceil = \lceil l \cdot \log_2 p \rceil$ bits for binary representation. The defined encoding version is called a polynomial encoding because it is necessary to calculate a polynomial for receiving the values of X_j (see. (5)).

Thus, in conventional CISS operating at positional number system the plaintext T is interpreted as a set of blocks T_j $(j = 1, 2, \dots, s)$ representing a binary code of corresponding numbers X_j:

$$\begin{cases} X_1 = \left(x_{L-1}^{(1)},\ x_{L-2}^{(1)}, \dots, x_0^{(1)}\right)_2, \\ X_2 = \left(x_{L-1}^{(2)},\ x_{L-2}^{(2)}, \dots, x_0^{(2)}\right)_2, \\ \dots\ \dots\ \dots \\ X_s = \left(x_{L-1}^{(s)},\ x_{L-2}^{(s)}, \dots, x_0^{(s)}\right)_2. \end{cases} \quad (6)$$

The binary number $Y_j = \left(y_{L-1}^{(j)},\ y_{L-2}^{(j)}, \dots, y_0^{(j)}\right)_2$ belonging to the range $[0, 2^L)$ similarly corresponds to the cryptogram C_j $(j = 1, 2, \dots, s)$.

The level of computational complexity of the used encryption algorithms (3) and (4) is a main factor that does exert a significant influence on qualitative characteristics of CISS. At the present time there exists a situation when the use of conventional representation of information and positional arithmetic in many cases cease to meet the increased requirements for the CISS performance. One of the ways to improve a CISS is the transition to the unconventional computing arithmetic, i.e. by performing all the computations using a modular number system (MNS), that allows us to parallelize the encrypting and decrypting processes and to attain the essential increase in data processing rate [2, 5, 7, 9, 10].

The computer-arithmetical basis of modular computing technology for creating a CISS

At the present time a MCT is widely used in parallel processing systems to solve problems demanding fast and exact calculations. The spectrum of such tasks covers, in particular, procedures of digital signal processing, carrying out calculations in the ranges of large numbers, creation of high-speed computer aids for performing of multiplication and exponentiation operations over big modules, mainly multiplicative procedures on the basis of Barret's and Montgomery's reduction schemes and so on [2, 5, 7, 9, 10-12, 15, 16].

A classic MNS on the set of integers $\boldsymbol{Z}$ is determined by means of pairwise relatively prime modules $m_1, m_2, \dots, m_k$ $(k \geq 2)$. In the given MNS the number $X \in \boldsymbol{Z}$ is represented as $X = (\chi_1, \chi_2, \dots, \chi_k)$, where $\chi_i = |X|_{m_i}$; we shall designate through $|x|_m$ the element of the set $\boldsymbol{Z}_m = \{0, 1, \dots, m-1\}$ that is congruent to x modulo m. In the nonredundant MNS with the bases $m_1, m_2, \dots, m_k$ it is possible to code at most $M_k = \prod_{i=1}^{k} m_i$ integers. At the same time the sets $\boldsymbol{Z}_{M_k} = \{0, 1, \dots, M_k - 1\}$ and $\boldsymbol{Z}_{\boldsymbol{M_k}}^{-} = \{-\lfloor M_k/2 \rfloor, -\lfloor M_k/2 \rfloor + 1, \dots, \lceil M_k/2 - 1 \rceil\}$

are usually used as a range of MNS (the designation $\lfloor x \rceil$ is used for the nearest to x integer at the left) [2, 5, 13, 14].

In the MNS with the bases $m_1, m_2, \ldots, m_k$ the modular operations (addition, subtraction and multiplication without overflow check) on any two integers A and B, represented by means of modular codes (MC): $A = (\alpha_1, \alpha_2, \ldots, \alpha_k)$, $B = (\beta_1, \beta_2, \ldots, \beta_k)$ ($\alpha_i = |A|_{m_i}$, $\beta_i = |B|_{m_i}, i = 1, 2, \ldots, k$), are carried out independently for each base, i.e. by the rule

$$A \circ B = (\alpha_1, \alpha_2, \ldots, \alpha_k) \circ (\beta_1, \beta_2, \ldots, \beta_k) =$$
$$= \left(|\alpha_1 \circ \beta_1|_{m_1}, |\alpha_2 \circ \beta_2|_{m_2}, \ldots, |\alpha_k \circ \beta_k|_{m_k}\right) (\circ \in \{+, -, \times\}). \quad (7)$$

The natural internal parallelism of MNS caused by the lack of interdigit carry propagation during performance of modular operations (7) holds a central position in all the advantages of modular arithmetic (MA). Since the components of the MC have a small code length and the ring operations are performed in the MNS independently for each module then the MA gives in essence new possibilities to increase the computation speed. The mentioned property is of particular importance especially for applications of MCT in cryptography.

The positional value of the number X can be obtained by its MC. The decoding mapping $\boldsymbol{\Phi}_{MNS}: \boldsymbol{Z}_{m_1} \times \boldsymbol{Z}_{m_2} \times \ldots \times \boldsymbol{Z}_{m_k} \to \boldsymbol{D}$ for the MNS with a range $\boldsymbol{D} = \boldsymbol{Z}_{M_k}$ which associates the MC $(\chi_1, \chi_2, \ldots, \chi_k)$ with an element $X \in \boldsymbol{D}$ can be realized according to the Chinese Remainder Theorem by means of the relations

$$X = \sum_{i=1}^{k} M_{i,k} \left| M_{i,k}^{-1} \chi_i \right|_{m_i} + \rho_k(X) M_k, \quad (8)$$

$$X = \sum_{i=1}^{k-1} M_{i,k-1} \left| M_{i,k-1}^{-1} \chi_i \right|_{m_i} + I(X) M_{k-1}, \quad (9)$$

where $M_{i,l} = M_l / m_i$, $M_l = \prod_{i=1}^{l} m_i$ $(l = k-1, k)$; $\rho_k(X)$ and $I(X)$ are an integral characteristics of MC called a rank and an interval index (II) of a number X with respect to the modules $m_1, m_2, \ldots, m_k$, respectively; $|c^{-1}|_m$ designates the multiplicative inversion of an integer c modulo m which is defined as an element d of a ring $\boldsymbol{Z}_m$ such that $|cd|_m = 1$. For any c relatively prime to m the value $d = |c^{-1}|_m$ always exists and is unique. The expressions (8) and (9) are called a rank form and an interval-modular form (IMF) of an integer X, respectively [2, 5].

The main component of optimization process of the MCS applied to creation a CISS consists in simplification of the basic non-modular procedures, first of all, the operations of transformation and expansion of MC realized on the basis of

relations (8) and (9). In the classical MNS the calculation of a rank characteristic $\rho_k(X)$ and an interval index $I(X)$ needs an application of the general algorithm for generating the integral characteristics of MC which is quite difficult and labor-consuming [2, 6].

As is generally known, it is possible to improve significantly the arithmetic properties of MNS and to optimize the algorithms of MA by introducing the so-called minimal additional redundancy which is carried out by some reduction of the effective range of MNS. The minimal redundant modular coding $\boldsymbol{\Phi}_{MRMNS}$: $\boldsymbol{Z_{m_1}} \times \boldsymbol{Z_{m_2}} \times ... \times \boldsymbol{Z_{m_k}} \to \boldsymbol{D}$ provides the use of the range $\boldsymbol{D}$ with a cardinal number $|\boldsymbol{D}| < M_k$. The resulting redundant MNS is naturally a restriction of the original non-redundant MNS and possesses all its advantages. In the case of redundant MNS a set $\boldsymbol{Z_{2M}^{-}} = \{-M, -M+1, ..., M-1\}$ is usually applied as a range $\boldsymbol{D}$, where $M = m_0 M_{k-1}, m_0$ is a fixed natural number.

The essence of the principle of minimal redundant modular coding is disclosed in [2, 5, 13, 14]. In this case, the calculation of the II $I(X)$ of a number $X \in \boldsymbol{D}$ becomes extremely simple since its value is completely defined by the so-called computer II $\hat{I}_k(X) = |I(X)|_{m_k}$ and is reduced to summation of a set of k residues modulo m_k. The required configuration of minimum redundant MNS (MRMNS) is achieved by the choice of the kth module m_k satisfying a condition $m_k \geq 2m_0 + \rho$, where

$$\rho = \left\lfloor \sum_{i=1}^{k-1} \frac{m_i - 1}{m_i} \right\rfloor = k - 1 - \left\lceil \sum_{i=1}^{k-1} \frac{1}{m_i} \right\rceil \leq k - 2 \qquad (10)$$

represents the maximum value of the rank characteristic $\rho_{k-1}(X)$ determined by the equality

$$|X|_{M_{k-1}} = \sum_{i=1}^{k-1} M_{i,k-1} \left| M_{i,k-1}^{-1} \chi_i \right|_{m_i} + \rho_{k-1}(X) M_{k-1}$$

(the designation $\lceil x \rceil$ is used for the nearest to x integers at the right). The minimal redundancy is attained in the case when the equality $m_k - 2m_0 - \rho = |m_k - \rho|_2$ holds.

At the same time, the following relation is true:

$$I(X) = \begin{cases} \hat{I}_k(X), & \text{if } \hat{I}_k(X) < m_0, \\ \hat{I}_k(X) - m_k, & \text{if } \hat{I}_k(X) \geq m_k - m_0 - \rho, \end{cases} \qquad (11)$$

where the residue $\hat{I}_k(X)$ is determined according to the calculation relations

$$\hat{I}_k(X) = \left| \sum_{i=1}^{k} R_{i,k}(\chi_i) \right|_{m_k}; \qquad (12)$$

$$R_{i,k}(\chi_i) = \left| - \frac{\left| M_{i,k-1}^{-1} \chi_i \right|_{m_i}}{m_i} \right|_{m_k} \quad (i \neq k), \;\; R_{k,k}(\chi_k) = \left| \frac{\chi_k}{M_{k-1}} \right|_{m_k}. \quad (13)$$

For many computer applications including also most of modern CISS it is enough to use as basic the version of minimal redundant MA (MRMA) oriented on operating only with non-negative integers. In this case, a set $\boldsymbol{Z}_M = \{0, 1, \dots, M-1\}$ is usually applied as a range $\boldsymbol{D}$. Then the configuration of used MRMNS is achieved by the choice of the kth module m_k satisfying a condition $m_k \geq m_0 + \rho$. The relation for $I(X)$ looks like

$$I(X) = \begin{cases} \hat{I}_k(X), & \text{if } \hat{I}_k(X) < m_0, \\ \hat{I}_k(X) - m_k, & \text{if } \hat{I}_k(X) \geq m_k - \rho. \end{cases} \quad (14)$$

In spite of the fact that the input additional redundancy is very small, just it allows us to simplify significantly the algorithms of performance of non-modular operations. It is seen from the relations (11) – (14) that in comparison with conventional (non-redundant) configurations of the MA a minimal redundant modular coding allows us to attain an essentially new level of optimization of nonmodular procedures on such qualitative characteristics as performance and computational burden. This is caused by the fact that the nonmodular procedures synthesized on the basis of IMF (9) use an interval index which is calculated by means of the simple relations and is formed precisely, without an error inherent in the calculation of the rank characteristic [2, 5]. The main advantages of applied modular computing technology for the construction of CISS are determined by the reason mentioned above, and a MRMA represents an effective computer-arithmetical basis for the realization of various cryptographic tasks.

As regards the commonly used and promising methods of cryptographic information protection, for most of them the modular multiplication and exponentiation consist a most time-consuming part of the basic encryption equations. In other words, these equations to a great extent possess a modular computer structure. This reason naturally led to the idea of application of the adequate methodological, algorithmic and hardware aids generated by modular coding in the CISS.

The parallel encryption system in the MNS

Let us consider the realization of parallel encryption of the plaintext block T_j $(j = 1, 2, \dots, s)$ using the minimal redundant modular coding. This

block of length L bits represents some non-negative integer number $X_j = \left(x_{L-1}^{(j)}, x_{L-2}^{(j)}, \dots, x_0^{(j)}\right)_2$ from the range $[0, 2^L)$.

Let us set the basic MRMNS with the bases $m_1, m_2, \dots, m_k$ and the range $\boldsymbol{Z}_M$. At the same time the modules m_i $(i = 1, 2, \dots, k)$ are chosen so as to satisfy the relation $M > 2^L$. The combination of the chosen modules of MRMNS represents the confidential information in the CISS.

The number $X_j \in \boldsymbol{Z}_{2^L}$ corresponding to the information block is uniquely coded in the MRMNS by the set of residues $\chi_i^{(j)} = \left|X_j\right|_{m_i}$ modulo m_i $(i = 1, 2, \dots, k)$, i.e. $X_j = (\chi_1^{(j)}, \chi_2^{(j)}, \dots, \chi_k^{(j)})$. The transformation of the positional binary code $\left(x_{L-1}^{(j)}, x_{L-2}^{(j)}, \dots, x_0^{(j)}\right)_2$ of an integer X_j to the minimal redundant MC (MRMC) is carried out within the parallel and pipelined MCS of look-up table type [2, 5].

Further a procedure for the encryption is applied to the block represented in MRMC. For this purpose at first the key sequence $K_{E,j}$ of length L bits (see (3)) has to be generated by means of pseudorandom sequence generator. This key sequence in the MRMNS is represented by means of the set of residues $(\kappa_1^{(E,j)}, \kappa_2^{(E,j)}, \dots, \kappa_k^{(E,j)})$, $\kappa_i^{(E,j)} = \left|K_{E,j}\right|_{m_i}$, $K_{E,j} \in \boldsymbol{Z}_M; j = 1, 2, \dots, s;\ i = 1, 2, \dots, k$.

The process of encrypting represents the imposition of key sequence over the information block in an MRMNS. This procedure can be considered as realization of some transformation $Y_j = \left|F\left(X_j, K_{E,j}\right)\right|_{M_k}$ which is carried out in parallel over the modules $m_1, m_2, \dots, m_k$. An encryption procedure adapted to MRMC has the form

$$\begin{cases} \gamma_1^{(j)} = \left|E\left(\chi_1^{(j)}, \kappa_1^{(E,j)}\right)\right|_{m_1}, \\ \gamma_2^{(j)} = \left|E\left(\chi_2^{(j)}, \kappa_2^{(E,j)}\right)\right|_{m_2}, \\ \dots\ \dots\ \dots \\ \gamma_k^{(j)} = \left|E\left(\chi_k^{(j)}, \kappa_k^{(E,j)}\right)\right|_{m_k}, \end{cases}$$

where $\gamma_i^{(j)} = \left|Y_j\right|_{m_i}$, $Y_j \in \boldsymbol{Z}_M; j = 1, 2, \dots, s;\ i = 1, 2, \dots, k$.

In the MRMNS the various types of linear and nonlinear cryptographic functions and their combinations can be realized, for example

$$\gamma_i^{(j)} = \left|\chi_i^{(j)} + \kappa_i^{(E,j)}\right|_{m_i}, \gamma_i^{(j)} = \left|\chi_i^{(j)} \cdot \kappa_i^{(E,j)}\right|_{m_i}, \gamma_i^{(j)} = \left|\chi_i^{(j)\,\kappa_i^{(E,j)}}\right|_{m_i}$$

$(i = 1, 2, \dots, k)$ and their combinations. The considered operations are performed in parallel over all modules of the MRMNS. The resulting MRMC of the number $Y_j = (\gamma_1^{(j)}, \gamma_2^{(j)}, \dots, \gamma_k^{(j)})$ corresponding to the cryptogram C_j enters to a communication channel.

Similar to parallel encrypting operation the decrypting operation is reduced to calculation of an MC of the number X_j corresponding to information block T_j $(j = 1, 2, \dots, s)$. This process can be represented as follows

$$\begin{cases} \chi_1^{(j)} = \left| D\left(\gamma_1^{(j)}, \kappa_1^{(D,j)}\right)\right|_{m_1}, \\ \chi_2^{(j)} = \left| D\left(\gamma_2^{(j)}, \kappa_2^{(D,j)}\right)\right|_{m_2}, \\ \quad \dots \quad \dots \quad \dots \\ \chi_k^{(j)} = \left| D\left(\gamma_k^{(j)}, \kappa_k^{(D,j)}\right)\right|_{m_k}, \end{cases}$$

where $\kappa_i^{(D,j)}$ represents the ith residue of the MRMC $(\kappa_1^{(D,j)}, \kappa_2^{(D,j)}, \dots, \kappa_k^{(D,j)})$ of key sequence $K_{D,j} \in \boldsymbol{Z}_M$; $j = 1, 2, \dots, s$; $i = 1, 2, \dots, k$. Further the generated MRMC $\left(\chi_1^{(j)}, \chi_2^{(j)}, \dots, \chi_k^{(j)}\right)$ of the number X_j in accordance with an IMF (9) converts to the positional code within the framework of the parallel and pipelined MCS of table type [2, 5]. As a result, we receive the plaintext block T_j.

Final remarks

The usefulness of application of the MA in CISS is imposed first of all by internal parallelism of MCS that provides them a number of significant advantages over position structures at an implementation of cryptographic algorithms. These advantages include:

- any operation in MNS is always reduced to sequence of single-cycle operations over the low-bit residues;
- the independence of the execution times of modular operations (addition, subtraction and multiplication without overflow check) of the number of modules, and thus of the length of MNS code;
- the perfect suitability to application of a tabular method of information processing at boss the hardware and software levels;
- the extreme simplicity of the pipelining the calculation at the level of low-bit tabular operations;
- the high regularity, uniformity and technological effectiveness of basic modular architecture.

Thus, the use of the MCS for the CISS design allows us to attain essential increase of performance due to the data representation in the MRMC and, accordingly, the parallel implementation of cryptographic transformations. It should be noted that the designed parallel encrypting algorithm is characterized by a maximum level of unloading of the real time computing process from the labor-consuming calculations, which can be realized by means of the look-up tables formed at a stage of preliminary calculations. It gives the capability to use an extremely simple table-summation configuration of the CISS, which is only implementing extraction of residues from tabular memory and their summation over the bases of the MRMNS.

References

[1] Buchmann J., *Introduction to Cryptography,* Springer, New York, 2004, DOI: http://dx.doi.org/10.1007/978-1-4419-9003-7.

[2] Chernyavsky A.F. (red.), *High-speed Methods and Systems of Digital Information Processing*, Belarusian State University Press, Minsk, 1996 (in Russian).

[3] Katz J., Lindell Y., *Introduction to Modern Cryptography*, Chapman & Hall/CRC Press, Boca Raton, 2008.

[4] Koblitz N., *Algebraic Aspects of Cryptography*, Springer, New York, 2004.

[5] Kolyada A.A., Pak I.T., *Modular Structures of Pipeline Digital Information Processing*, University Press, Minsk, 1992 (in Russian).

[6] Kolyada A.A., Selyaninov M.Y., *On the formation of the integral characteristics of the codes of residue number systems with the symmetrical range,* Cybernetics, 4, (1986), 20–24 (in Russian).

[7] Kornerup P., Matula D.W., *Finite Precision Number Systems and Arithmetic*, Cambridge University Press, Cambridge, 2010.

[8] Mao W., *Modern Cryptography*. Theory and Practice, Prentice Hall PTR, Upper Saddle River, NJ, 2003.

[9] Mohan P.V. Ananda, *Residue Number Systems: Algorithms and Architectures*, Kluwer Academic Publishers, 2002.

[10] Omondi A., Premkumar B., Residue Number Systems. *Theory and Implementation*, Imperial College Press, London, 2007.

[11] Selianinau M., *High– speed modular structures for parallel computing in the space of orthogonal projections*, Scientific Issues, Jan Długosz University of Częstochowa, Ser. Technical and IT Education, V, (2010), 87–96.

[12] Selianinau M., *Modular principles of high-speed adaptive filtration of discrete signals*, Scientific Issues, Jan Długosz University of Częstochowa, Ser. Technical and IT Education, VI, (2011), 75–84.
[13] Selyaninov M., *Modular technique of parallel information processing*, Scientific Issues of Jan Długosz University of Czestochowa, Mathematics XIII, (2008), 43–52.
[14] Selyaninov M., *Construction of modular number system with arbitrary finite ranges,* Scientific Issues of Jan Długosz University of Czestochowa, Mathematics XIV (2009), 105–115.
[15] Selyaninov M., *Modular technique of high-speed parallel computing on the sets of polynomials*, Scientific Issues of Jan Długosz University of Czestochowa, Mathematics XVII (2012), 69–76.
[16] Selyaninov M., *Application of modular computing technique for high-speed implementation of cyclic convolution*, Scientific Issues of Jan Długosz University of Czestochowa, Mathematics XIX (2014), 213–222.
[17] Stinson D.R., Cryptography. Theory and Practice, Chapman & Hall/CRC Press, Boca Raton, 2006.