Computer Science Journal of Moldova, vol.**18**, no.3(54), 2010

Information encryption systems based on Boolean functions

Aureliu Zgureanu

Abstract

An information encryption system based on Boolean functions is proposed. Information processing is done using multidimensional matrices, performing logical operations with these matrices. At the basis of ensuring high level security of the system the complexity of solving the problem of building systems of Boolean functions that depend on many variables (tens and hundreds) is set. Such systems represent the private key. It varies both during the encryption and decryption of information, and during the transition from one message to another.

Keywords: Boolean functions, multidimensional matrices, private keys, security of the system, the complexity of the problem.

1 Introduction

The most popular information encryption systems (IES), based on prime numbers, are shown in [1, 2]. In [3], using ideas from [1], there is proposed a new encryption algorithm, which considerably increases resistance to breakage, keeping the speed encryption and decryption. In [4] there has been proposed another encryption system with a cryptographic power not smaller than those two shown in [1,3], but, at the same time, with an encrypting-decrypting time much smaller. Together with the improving of computing means, the requirements towards IES also increase. Public keys and those private become bigger and bigger, and arithmetic operations with very big numbers become more difficult. As a result, the productivity of the systems decreases considerably. The

situation may be changed if we replace these arithmetic operations with logical operations on systems of Boolean functions, represented by multidimensional matrices [4]. Such a solution to the problem is proposed in this paper.

2 Sets of relations and multidimensional matrixes

In accordance with [11], a system A of $N_A = n_1 n_2 n_3 \cdot ... \cdot n_p$ elements $a_{i_1 i_2 i_3 ... i_p}$ $(i_\alpha = 1, 2, 3, ..., n_\alpha;\ \alpha = 1, 2, 3, ..., p)$ that belong to the set Ω and are placed in the points of p-dimensional space of coordinates $i_1, i_2, ..., i_p$ is called a *multidimensional matrix* over the set Ω. The number p is called the *size of the matrix* and shows the number of indexes in the notation of the matrix elements. Size N_A shows the total number of elements in this matrix. Size n_α of the index i_α shows how many values (from 1 to n_α) this index runs. So in this paper, a multidimensional matrix is a direct generalization of the usual two-dimensional matrix.

Consider a family of sets $X = \{X_1, X_2, ..., X_n\}$, where $X_i = \{x_{i1}, x_{i2}, ..., x_{i\lambda_i}\}$, $i = \overline{1, n}$ and the set $\Omega = \{\omega_1, ..., \omega_r\}$ with arbitrary elements (in our case – integer numbers). There are k relations $R_j = R_{X_{j_1} ... X_{j_{d_j}}}$ $(2 \le d_j \le n, j = \overline{1, k},\ j_1, j_2, ..., j_{d_j} \in \{1, 2, ..., n\})$ defined on this family as subsets of Cartesian products $X_{j_1} \times X_{j_2} \times ... \times X_{j_{d_j}}$. The matrixes of these relations are d_j-dimensional with elements from Ω. Let's mark by $\overrightarrow{R}$ the vector with components R_j, that is $\overrightarrow{R} = (R_1, ..., R_j, ..., R_k)$. Let's correlate the following n-dimensional matrix to this vector:

$$A_R = \Phi(\vec{R}). \tag{1}$$

The elements of this matrix are denoted by $a_{s_1 ... s_\tau ... s_n}$. Let's explain how these elements are obtained.

We build the Cartesian product $X_1 \times X_2 \times ... \times X_n = \{x_{11}, ..., x_{1\lambda_1}\} \times ... \times \{x_{n1}, ..., x_{n\lambda_n}\}$, which obviously contains $u = \lambda_1 \cdot ... \cdot \lambda_n$ elements.

With these elements compose a two-dimensional matrix with u rows and n columns (Figure 1, left side).

$$A_R: \quad \begin{array}{c} \\ 1 \\ \vdots \\ i \\ \vdots \\ u \end{array} \begin{array}{c} \begin{array}{cccccc} X_1 & \cdots & X_\tau & \cdots & X_n & \end{array} \\ \begin{pmatrix} x_{11} & \cdots & x_{\tau 1} & \cdots & r_{n1} \\ & & \ddots & & \\ x_{1s_1} & \cdots & x_{\tau s_\tau} & \cdots & x_{ns_n} \\ & & \ddots & & \\ x_{1\lambda_1} & \cdots & x_{\tau\lambda_\tau} & \cdots & x_{n\lambda_n} \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{cccccc} R_1 & \cdots & R_j & \cdots & R_k \end{array} \\ \begin{pmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1k} \\ & & \ddots & & \\ r_{i1} & \cdots & r_{ij} & \cdots & r_{ik} \\ & & \ddots & & \\ r_{u1} & \cdots & r_{uj} & \cdots & r_{uk} \end{pmatrix} \end{array}$$

Figure 1.

Compose another two-dimensional matrix $||r_{ij}||$ with u rows and k columns (Figure 1, right side), where $r_{ij} = r_{s_{j_1}\ldots s_{j_d}}$ with elements $s_{j_1}, ..., s_{j_d}$ selected from line i at the places $j_1, ..., j_d$, that indicate the sets $X_{j_1}, ..., X_{j_d}$ where relation R_j is defined.

For simplicity replace the element $x_{\tau s_\tau}$ with its second index as it is shown in Figure 2. The lines of the matrix in the left side of Figure 2 represent indices of the matrix A_R elements. The lines of the matrix in the right side of Figure 2 form the elements of matrix A_R:

$$a_{s_1\ldots s_\tau\ldots s_n} = (r_{i1}, ..., r_{ij}, ..., r_{ik}), \tag{2}$$

To the vector (2) there is associated a number c_i in the base y which satisfies the condition $y > \max \omega_h$, $h = \overline{1, r}$:

$$c_i = r_{i1}y^{k-1} + ... + r_{ij}y^{k-j} + ... + r_{ik} = \sum_{j=1}^{k} r_{ij}y^{k-j}, \ i = \overline{1, u}. \tag{3}$$

So, we obtain the vector

$$\vec{c} = (c_1, ..., c_i, ..., c_u). \tag{4}$$

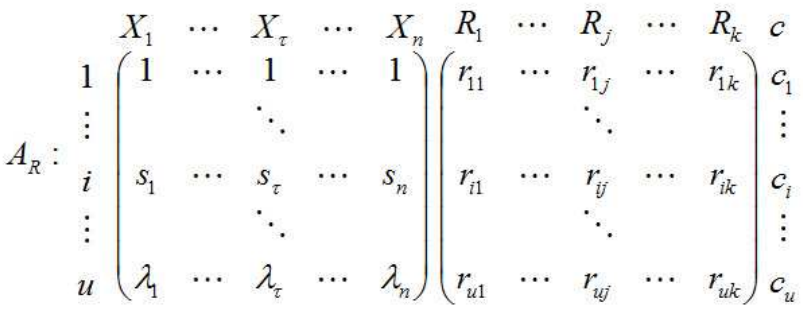

Figure 2.

Thus, using the transformation (1), the vector $\vec{c}$ (3), (4) is put into correspondence to the vector $\vec{R}$. The reverse transformation

$$\vec{R} = \Phi^{-1}(\vec{c}), \tag{5}$$

generally is much more complicated [5], [7], [8].

In some particular cases we can find vector $\vec{R}$ coordinates by vector $\vec{c}$ coordinates. This was achieved when investigating of the distribution of prime numbers in the range of integer numbers. As the result an algorithm for prime numbers generating has been elaborated [9], [10].

If the transformation (5) is difficult we can use this when elaborating the IES.

3 Information encryption systems

We consider a particular case of the exposed above, i.e. $X_1 = X_2 = ... = X_n = \Omega = \{0, 1\}$. We denote the relations defined on these sets by $M_j = M_{X_{j_1}...X_{j_{d_j}}}$ $(2 \leq d_j \leq n,\ j = \overline{1,\ k},\ j_1, j_2, ..., j_{d_j} \in \{1, 2, ..., n\})$, thus obtaining the vector $\vec{M} = (M_1, ..., M_j, ..., M_k)$. Let's correlate an n-dimensional matrix $A_M = \Phi(\vec{M})$, $i = \overline{0, u}$, $j = \overline{1, k}$ [5], presented at Figure 3, to this vector. In this matrix $M_j = M_{X_\tau...X_n}$ and

$m_{ij} = m_{\sigma_\tau \dots \sigma_n} \in \{0, 1\}$. Therefore, this matrix represents a system of k Boolean functions with variables $x_1, ..., x_n$. We correlate the following vector to this matrix:

$$\vec{m} = (m_0, ..., m_i, ..., m_t), t \leq u, \text{ where } m_i = \sum_{j=1}^{k} m_{ij} \cdot 2^{k-j}, i = \overline{0, t}, \quad (6)$$

$$n = \lceil \log_2 t \rceil, k = \lceil \log_2 \max m_i \rceil \quad (7)$$

$$A_R : \begin{array}{c} \\ 0 \\ \vdots \\ i \\ \vdots \\ u \end{array} \begin{array}{c} \begin{array}{cccccc} x_1 & \cdots & x_\tau & \cdots & x_n & \end{array} \\ \begin{pmatrix} 0 & \cdots & 0 & \cdots & 0 \\ & & \ddots & & \\ \sigma_1 & \cdots & \sigma_\tau & \cdots & \sigma_n \\ & & \ddots & & \\ 1_1 & \cdots & 1 & \cdots & 1 \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{ccccc} M_1 & \cdots & M_j & \cdots & M_k \end{array} \\ \begin{pmatrix} m_{01} & \cdots & m_{0j} & \cdots & m_{0k} \\ & & \ddots & & \\ m_{i1} & \cdots & m_{ij} & \cdots & m_{ik} \\ & & \ddots & & \\ m_{u1} & \cdots & m_{uj} & \cdots & m_{uk} \end{pmatrix} \end{array} \begin{array}{c} m \\ m_0 \\ \vdots \\ m_i \\ \vdots \\ m_u \end{array}$$

Figure 3.

By analogy (Figure 4) we create another matrix A_D to which we correlate a vector

$$\vec{d} = (d_0, ..., d_i, ..., d_t), t \leq u, \text{ where } d_i = \sum_{j=1}^{k} d_{ij} \cdot 2^{k-j}, i = \overline{0, t}. \quad (8)$$

We may perform logical operations with these matrixes: $A_M \wedge A_D$, $A_M \vee A_D$, $A_M \oplus A_D$ and other, as the result we obtain other matrixes. Let's analyze the operation $\oplus$ (sum modulo 2). Suppose that $A_M \oplus A_D = A_C$. In this case $c_{ij} = m_{ij} \oplus d_{ij}$. Taking into account properties of this operation, we obtain:

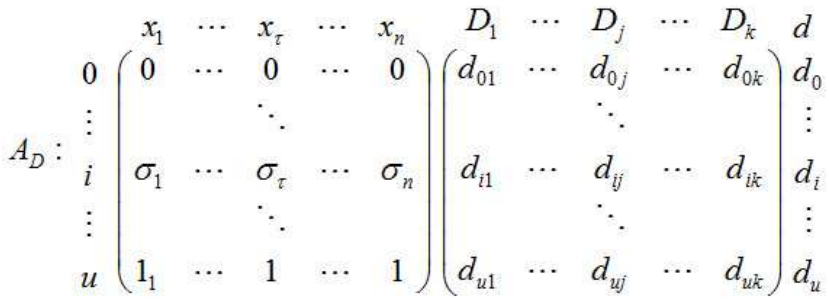

Figure 4.

$$(A_M \oplus A_D) \oplus A_D = A_M \oplus (A_D \oplus A_D) = A_M.$$

Thus

$$A_M \oplus A_D = A_C,\ A_C \oplus A_D = A_M. \tag{9}$$

From (9) it results that the matrix A_D may serve as private key for encryption and decryption of vector $\vec{m}$ which is the ASCII encoding (or any other encoding) of the plaintext M through vector $\overrightarrow{c}$ (ciphertext)

$$\overrightarrow{c} = (c_0, ..., c_i..., c_t),\ t \leq u,$$
$$\text{where } c_i = \sum_{j=1}^{k} c_{ij} \cdot 2^{k-j},\ i = \overline{0,t},\ c_{ij} = m_{ij} \oplus d_{ij}. \tag{10}$$

Let's see how we may create the private key. Suppose that the function is defined by veracity table (see Table 1), where $\varepsilon_0, ..., \varepsilon_u \in \{0, 1\}$.

Let's create the partition $\{\tilde{X}_1, \tilde{X}_2\} = \{\{x_1, ..., x_\tau\}, \{x_{\tau+1}, ..., x_n\}\}$ on set $x = \{x_1, ..., x_n\}$. We create two sets:

- $Y = \{y_0, y_1, ..., y_p, ..., y_{2^\tau - 1}\}$ (formed of binary states that correspond to variables from $\tilde{X}_1$);

Table 1.

	x_1	$\cdots$	x_τ	$\cdots$	x_n	$F(x_1,...,x_n)$
0	0	$\cdots$	0	$\cdots$	0	ε_0
$\vdots$			$\ddots$			$\vdots$
i	σ_1	$\cdots$	σ_τ	$\cdots$	σ_n	ε_i
$\vdots$			$\ddots$			$\vdots$
u	1	$\cdots$	1	$\cdots$	1	ε_u

- and $Z = \{z_0, ..., z_q, ..., z_{2^{n-\tau}-1}\}$ (formed of binary states that correspond to variables from $\tilde{X}_2$).

Then, the Boolean function $F(x_1, ..., x_n)$ may be considered as a binary relation R_{YZ} between the sets Y and Z with the matrix

$$R_{YZ} = \begin{array}{c} \\ y_0 \\ \vdots \\ y_i \\ \vdots \\ y_h \end{array} \begin{array}{c} \begin{array}{ccccc} z_0 & \cdots & z_j & \cdots & z_s \end{array} \\ \left[\begin{array}{ccccc} a_{00} & \cdots & a_{0j} & \cdots & a_{0s} \\ & & \ddots & & \\ a_{i0} & \cdots & a_{ij} & \cdots & a_{is} \\ & & \ddots & & \\ a_{h0} & \cdots & a_{hj} & \cdots & a_{hs} \end{array}\right] \end{array}, h = 2^\tau - 1, s = 2^{n-\tau} - 1,$$

$$\forall i, j a_{ij} = \begin{cases} 1, \text{ if } F(y_i, z_j) = 1, \\ 0, \text{ if } F(y_i, z_j) = 0. \end{cases}$$

According to [6], the subset $S^{z_j}_{F^\varepsilon}$ of the set Y is called ***subset of column*** of the function $F(x_1, ..., x_n)$ for the column z_j and is composed of the elements y_i for which $a_{ij} = \varepsilon$, $\varepsilon \in \{0, 1\}$.

The Boolean function may be defined by the *table of subsets of column* (see Table 2):

It is obvious that $S^{z_j}_{F^0} = Y \backslash S^{z_j}_{F^1}$. Because of this, the subsets $S^{z_j}_{F^0}$ are not indicated in the Table 2. We create partitions $\pi_{x_1}, ..., \pi_{x_\tau}$ on the set Y [6].

Table 2.

	z_0	...	z_j	...	z_s
F^1	$S_{F^1}^{z_0}$	...	$S_{F^1}^{z_j}$	...	$S_{F^1}^{z_s}$

Let's consider a specific case: n=5, $\tau = 3$ (see Table 3). In this case $\tilde{X}_1 = \{x_1, x_2, x_3\}$, $\tilde{X}_2 = \{x_4, x_5\}$ and $Y = \{y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7\} = \{000, 001, 010, 011, 100, 101, 110, 111\}$, $Z = \{z_0, z_1, z_2, z_3\} = \{00, 01, 10, 11\}$. We create the Table 3 ($\varepsilon_0, ..., \varepsilon_{31} \in \{0, 1\}$) and the partitions $\pi_{x_i} = \{\bar{m}_i^0; \bar{m}_i^1\}$, $i = \overline{1,3}$ according to the following conditions:

$$y_j \in \bar{m}_i^{\sigma_i}, \text{ if } x_i = \sigma_i \tag{11}$$

$$\pi_{x_1} = \{\overline{y_0, y_1, y_2, y_3}\,{}_1^0; \overline{y_4, y_5, y_6, y_7}\,{}_1^1\}, \pi_{x_2} = \{\overline{y_0, y_1, y_4, y_5}\,{}_2^0; \overline{y_2, y_3, y_6, y_7}\,{}_2^1\},$$

$$\pi_{x_3} = \{\overline{y_0, y_2, y_4, y_6}\,{}_3^0; \overline{y_1, y_3, y_5, y_7}\,{}_3^1\}.$$

Table 3.

		x_4x_5			
		z_0	z_1	z_2	z_3
	$x_1\,x_2\,x_3$	00	01	10	11
y_0	0 0 0	ε_0	ε_1	ε_2	ε_3
y_1	0 0 1	ε_4	ε_5	ε_6	ε_7
y_2	0 1 0	ε_8	ε_9	ε_{10}	ε_{11}
y_3	0 1 1	ε_{12}	ε_{13}	ε_{14}	ε_{15}
y_4	1 0 0	ε_{16}	ε_{17}	ε_{18}	ε_{19}
y_5	1 0 1	ε_{20}	ε_{21}	ε_{22}	ε_{23}
y_6	1 1 0	ε_{24}	ε_{25}	ε_{26}	ε_{27}
y_7	1 1 1	ε_{28}	ε_{29}	ε_{30}	ε_{31}

To simplify this, we replace elements y_i by their indexes i. Thus, such partitions are obtained:

$$\pi_{x_1} = \{\overline{0,1,2,3}\,{}^{0}_{1}; \overline{4,5,6,7}\,{}^{1}_{1}\}, \pi_{x_2} = \{\overline{0,1,4,5}\,{}^{0}_{2}; \overline{2,3,6,7}\,{}^{1}_{2}\},$$

$$\pi_{x_3} = \{\overline{0,2,4,6}\,{}^{0}_{3}; \overline{1,3,5,7}\,{}^{1}_{3}\}.$$

Let's mark by $\bar{m}^{\sigma_i, ..., \sigma_j, ..., \sigma_p}_{i, ..., j, ..., p}$ the bloc of product of partitions $\pi_{x_i}, ..., \pi_{x_j}, ..., \pi_{x_p}$, where $\sigma_j = 0\,(1)$ if the elements of this bloc belong to the bloc $\bar{m}^0_j$ ($\bar{m}^1_j$) for $j = \overline{i,p}$. We also mark the indicated partitions product by $\pi_{x_i,..., x_j,..., x_p}$. For partitions above we get the following products:

$$\pi_{x_1,x_2} = \{\overline{0,1}^{0,0}_{1,2}; \overline{2,3}^{0,1}_{1,2}; \overline{4,5}^{1,0}_{1,2}; \overline{6,7}^{1,1}_{1,2}\},$$

$$\pi_{x_1,x_3} = \{\overline{0,2}^{0,0}_{1,3}; \overline{1,3}^{0,1}_{1,3}; \overline{4,6}^{1,0}_{1,3}; \overline{5,7}^{1,1}_{1,3}\},$$

$$\pi_{x_2,x_3} = \{\overline{0,4}^{0,0}_{2,3}; \overline{1,5}^{0,1}_{2,3}; \overline{2,6}^{1,0}_{2,3}; \overline{3,7}^{1,1}_{2,3}\},$$

$$\pi_{x_1,x_2,x_3} = \{\overline{0}^{0,0,0}_{1,2,3}; \overline{1}^{0,0,1}_{1,2,3}; \overline{2}^{0,1,0}_{1,2,3}; \overline{3}^{0,1,1}_{1,2,3}; \overline{4}^{1,0,0}_{1,2,3}; \overline{5}^{1,0,1}_{1,2,3}; \overline{6}^{1,1,0}_{1,2,3}; \overline{7}^{1,1,1}_{1,2,3}\}.$$

The Table 2 is obtained when the function is given by veracity table. This table may be also obtained in the case when the function is given in analytical form, for instance in disjunctive normal form:

$$F(x_1, ..., x_n) = u_1 \vee ... \vee u_i \vee ... \vee u_e,$$

where $u_i = x_{i_1}^{\sigma_{i_1}} \wedge ... \wedge x_{i_a}^{\sigma_{i_a}}$, $i_1, i_2, ..., i_a \in \{1, ..., n\}$, $\sigma_{i_1}, ..., \sigma_{i_a} \in \{0, 1\}$, $i = \overline{1,e}$.

There may be distinguished the following 3 cases:

a) $x_{i_1}, ..., x_{i_a} \in \tilde{X}_1$

In this case u_i doesn't depend on variables $x_{\tau+1}, ..., x_n$ and, therefore, the subsets of column are equal and are formed of the elements of the bloc $\bar{m}^{\sigma_{i_1} \dots \sigma_{i_\alpha}}_{i_1 \dots i_\alpha}$ [6]:

$$S^{z_0}_{u_i^1} = ... = S^{z_s}_{u_i^1} = \bar{m}^{\sigma_{i_1} \dots \sigma_{i_\alpha}}_{i_1 \dots i_\alpha}$$

b) $x_{i_1}, ..., x_{i_a} \in \tilde{X}_2$

Taking into account the property

$$u_i = \begin{cases} 1, \text{ if } \forall x_{i_t} \in \{x_{i_1}, ..., x_{i_a}\},\ x_{i_t} = \sigma_{i_t}, \\ 0, \text{ if } \exists\, x_{i_t} \in \{x_{i_1}, ..., x_{i_a}\},\ x_{i_t} \neq \sigma_{i_t} \end{cases}$$

and the definition of the subset of column we get:

$$S^{z_j}_{u_i^1} = \begin{cases} Y, \text{ if for } \forall x_{i_t} \in \{x_{i_1}, ..., x_{i_a}\},\ x_{i_t} = \sigma_{i_t}, \\ \emptyset, \text{ if } \exists\, x_{i_t} \in \{x_{i_1}, ..., x_{i_a}\},\ x_{i_t} \neq \sigma_{i_t}; \end{cases}$$

c) $x_{i_1}, ..., x_{i_b} \in \tilde{X}_1,\ x_{i_{b+1}}, ..., x_{i_a} \in \tilde{X}_2$

In this case

$$S^{z_j}_{u_i^1} = \begin{cases} \bar{m}^{\sigma_{i_1} \dots \sigma_{i_s}}_{i_1 \dots i_s} \text{ if for } \forall x_{i_t} \in \{x_{i_{s+1}}, ..., x_{i_b}\},\ x_{i_t} = \sigma_{i_t}, \\ \emptyset \text{ if } \exists\, x_{i_t} \in \{x_{i_{s+1}}, ..., x_{i_b}\}, \text{ for which } x_{i_t} \neq \sigma_{i_t} \text{ holds.} \end{cases}$$

Considering every conjunction as a Boolean function, we get their subsets of column according to the cases mentioned above. These subsets are given in Table 4. The subsets of column of the given function are obtained in last line. They represent the union of the subsets from every column.

As any analytical form of Boolean function may be reduced to the form (11), then any function given in analytical form may be represented by the table of subsets of column.

The representation of Boolean function by subsets of column gives us the possibility to create the private key in a compact form. Suppose that functions $F_1, ..., F_j, ..., F_k$ with values from the respective columns from the Figure 4 correspond to relations $D_1, ..., D_j, ..., D_k$.

Table 4.

	z_0	...	z_j	...	z_s
u_1^1	$S_{u_1^1}^{z_0}$	...	$S_{u_1^1}^{z_j}$	...	$S_{u_1^1}^{z_s}$
u_2^1	$S_{u_2^1}^{z_0}$	...	$S_{u_2^1}^{z_j}$	...	$S_{u_2^1}^{z_s}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
u_e^1	$S_{u_e^1}^{z_0}$	...	$S_{u_e^1}^{z_j}$	...	$S_{u_e^1}^{z_s}$
	$S_{F^1}^{z_0}=\bigcup_{i=1}^{e} S_{u_i^1}^{z_0}$	...	$S_{F^1}^{z_j}=\bigcup_{i=1}^{e} S_{u_i^1}^{z_j}$	...	$S_{F^1}^{z_s}=\bigcup_{i=1}^{e} S_{u_i^1}^{z_s}$

Consider functions F_j for which the following conditions are achieved:

$$S_{F_j^1}^{z_0} = S_{F_j^1}^{z_1} = ... = S_{F_j^1}^{z_s} = S_j,\ j = \overline{1,k}.$$

For all the values of j we'll get the vector $\vec{S} = (S_1, ..., S_j, ..., S_k)$ – private key. Suppose that the values of the first τ variables in the index i of d_i form the binary state $\sigma_1 ... \sigma_\tau = y_q$ (Figure 4). Thereby, according to the definition of the subset of column, the values of d_{ij} are obtained from the relation

$$d_{ij} = \begin{cases} 1, \text{if } y_q \in S_j \\ 0, \text{if } y_q \notin S_j \end{cases} \tag{12}$$

Thus, the vector $\vec{S}$ determines univocally the matrix A_D. The subsets S_j are chosen on condition that

$$\bigcup_{j=1}^{k} S_j = Y \tag{13}$$

This condition assures changing the components of the vector $\vec{m}$ through vector $\vec{d}$. Relation (12) assures a rapid calculation of function value on binary state $i = \sigma_1 ... \sigma_\tau ... \sigma_n$.

As $|Y| = 2^\tau$, then for a single function we may create 2^{2^τ} subsets of column, and for k functions we have

$$\lambda = 2^{k \cdot 2^\tau}$$

different keys. As a result, the security of the private key may be chosen by parameter τ and subsets S_j.

According to those mentioned, in the computational software program *Mathematica 6*, there has been elaborated an IES composed of:

- ***key generator,*** which generates vector $\vec{S} = (S_1, ..., S_j, ..., S_k)$. The components S_j are selected randomly as subsets of the set Y and on condition (13). Using (12) and (8) we create vector $\vec{d} = (d_0, ..., d_t)$;

- ***codifier,*** which creates vector $\vec{c} = (c_0, ..., c_t)$ (on the basis of vectors $\vec{m}$ and $\vec{d}$ using (10)), codifies vector $\vec{S} = (S_1, ..., S_j, ..., S_k)$ with the help of the system from [4] or other secure system, and, concatenating it with vector $\vec{c}$, creates vector $\vec{g}$ [4];

- ***decoder,*** which restores the vector $\vec{S} = (S_1, ..., S_j, ..., S_k)$ from the vector $\vec{g}$, creates vector $\vec{d}$ using (8) and (12), creates vector $\vec{m}$ on the basis of the vectors $\vec{c} = (c_0, ..., c_t)$ and $\vec{d} = (d_0, ..., d_t)$ using (9). The initial text is printed on the basis of the vector $\vec{m}$.

Some data concerning functioning of this system (Cripto 3) in comparison with the system RSA are brought in the Table 5. We notice that for RSA both the encrypting and decrypting time grow almost linearly. Beginning with $t = 100000$ the system already meets some difficulties in creating vector $\vec{c}$ because of its too big components. This fact is marked in the Table 5 by symbol ∞. The data correspond to the public key of 2057 bits. The time grows much slowly for the system Cripto 3 and as a result it manages handling messages that contain millions of symbols, and, in the same time, has a very high security.

Table 5.

Encrypting systems	Number of symbols	Encrypting time (sec.)	Decrypting time (sec.)
RSA	100	0.34	5.34
Cripto3		0.31	0.07
RSA	500	0.48	26.90.
Cripto3		0.36	0.08
RSA	1000	0.93.	53.66
Cripto3		0.46	0.10
RSA	10000	9.60	533.40
Cripto3		1.01	0.51
RSA	100000	∞	∞
Cripto3		14.67	5.46
RSA	500000	∞	∞
Cripto3		76.23	34.37
RSA	1000000	∞	∞
Cripto3		82.62	67.20
RSA	2000000	∞	∞
Cripto3		192.06	193.23
RSA	4000000	∞	∞
Cripto3		582.75	431.90

For instance, if $t = 1000000$, then $k = 14$. Consider $\tau = 4$ and, therefore, $\lambda = 2^{224}$. This number is bigger than the number of atoms in the galaxy.

More than that, the key is the variable one. It changes both from one message to another and during the information encrypting. It changed 334 times in the case mentioned above. The data from Table 5 were got using *Athlon (tm) Processor3500*.

This system may be generalized for the case when the functions $F_1, ..., F_j, ..., F_k$ are from q-valent logics. In such a case, both variables $x_1, ..., x_n$ and functions F_j admit values from the set $\Omega = \{0, 1, ..., q-1\}$.

In the Table 1 we have $u = q^n - 1$ for these functions and the last state has the form $q-1...q-1...q-1$. In Figure 3 $m_{ij} \in \Omega$ and in Figure 4 $d_{ij} \in \Omega$. These matrices represent systems of q-valent functions. The formulas (14), (15) and (16) correspond respectively to the formulas (6), (7) and (8):

$$\vec{m} = (m_0, ..., m_i, ..., m_t),\ t \leq u,$$
$$\text{where } \ m_i = \sum_{j=1}^{k} m_{ij} \cdot q^{k-j}, i = \overline{0,t}, \tag{14}$$

$$n = \left\lceil \log_q t \right\rceil,\ k = \left\lceil \log_q \max m_i \right\rceil, \tag{15}$$

$$\vec{d} = (d_0, ..., d_i, ..., d_t),\ t \leq u,\ \text{where } \ d_i = \sum_{j=1}^{k} d_{ij} \cdot q^{k-j}, i = \overline{0,t}. \tag{16}$$

Let's create a new matrix $A_C = A_M + A_D(\text{mod } q)$, where $c_{ij} = m_{ij} + d_{ij}(\text{mod } q)$. Since for q matrices A_D the following relation holds:

$$\overbrace{A_D + A_D + ... + A_D}^{q\ times}(\text{mod } q) = 0\,(\text{zero matrix}),$$

then the equalities (17) and (18) correspond respectively to equalities (9) and (10):

$$A_M + A_D(\text{mod } q) = A_C,\ A_C + (q-1)A_D(\text{mod } q) = A_M, \tag{17}$$

$$\vec{c} = (c_0, ..., c_i..., c_t),\ t \leq u,$$
$$\text{where } c_i = \sum_{j=1}^{k} c_{ij} \cdot q^{k-j},\ \ i = \overline{0,t}\,, c_{ij} = m_{ij} + d_{ij}(\text{mod } q). \tag{18}$$

From (18) it results that if for encrypting the vector $\vec{m}$ we apply the matrix A_D, then for decrypting this vector we apply the matrix $(q-1)A_D$.

From (16) it results that components d_h of the vector $\vec{d}$ belong to the set $\{1, ..., q^k-1\}$ (0 is not included in this set because the state 0....0 doesn't change the components of the vector $\vec{m}$). In order to create this vector we take the last τ variables from the set $\{x_1, ..., x_{n-\tau}, ..., x_n\}$, choose randomly q^τ numbers from the set $\{1, ..., q^k-1\}$ and create the following vector with these numbers:

$$\vec{d} = (d_0, ..., d_h, ..., d_{q^\tau-1}), \quad q^\tau - 1 \leq t,$$

which represents the private key. Components d_h may be repeated an arbitrary number of times. Thereby, the number of different private keys is

$$\lambda = (q^k - 1)^{q^\tau}.$$

Using (14) and (18) we create the vector $\vec{c}$. It results from (17) that

$$m_i = \sum_{j=1}^{k} (c_{ij} + (q-1)d_{ij}) (\text{mod } q) q^{k-j}, i = \overline{0,t}.$$

For the examined case, in the computation software *Mathematica* 6, there was also elaborated an encryption system with a higher speed, depending on q and τ values. For example, the encrypting and decrypting time for $t = 2000000$, $q = 3$, $\tau = 4$ is equal to 130.62 *sec* and 127.45 *sec* respectively in comparison with 192.06 and 193.23 (see Table 5). Generally, a deeper investigation is needed to determine the optimal values for parameters q, τ and t.

For $q > 2$ the private key may be also represented by subsets of column. For this case, in the vector $\vec{S} = (S_1, ..., S_j, ..., S_k)$, every component S_j represents sets of form $\{\{S_j^1\}, \{S_j^2\}, ..., \{S_j^{q-1}\}\}$, where S_j^ε is a subset of the set $\{0, 1, ..., q^\tau - 1\}$, for $\forall \varepsilon \in \{1, ..., q-1\}$, and $S_j^k \bigcap S_j^s = \emptyset$ occurs for $\forall k, s \in \{1, ..., q-1\}$. But, together with the growth of q, there appear difficulties concerning the representation and transmitting of private key. Additional investigations are needed here.

4 Conclusions

1. The elaborated system has information processing speed much higher and also a capacity of solving the problems of much bigger dimensions in comparison with existent encryption systems. The priorities of the system have been highlighted during its testing with vectors that contain hundreds, thousands and millions of components.

2. Due to the fact that the system can operate with small numbers, it may be easily created using different programming languages.

3. The system may be improved using functions with q-valent logics. Deeper investigations are needed in order to achieve this.

References

[1] R. L. Rivest, A. Shamir, and L. Adleman. *A method for obtaining digital signatures and public-key cryptosystems.* CACM, 21(2), February 1978, pp. 120–126.

[2] El Gamal, *A public key cryptosystem and a signature scheme based on discrete logarithms.* IEEE TRANS. On inform. Theory, vol. IT-31. pp. 469–472, July 1985.

[3] Bulat M., Zgureanu A., Ciobanu I., Bivol L, *Encryption systems with vector keys.* International scientific conference "Mathematical modeling, optimization and information technologies", Chişinău, 9-21 martie, 2008. ATIC. pp. 281–285.

[4] Bulat M.S., Zgureanu A.F., Chobanu Ya.I., Bivol L.G., *Encryption systems based on n-ary relations.* Systems of management, control and measuring (UKI-08), Russian Conference with international participation, Moscow IPU RAN, 2008. pp. 66–67.

[5] M. Bulat, *Some applications of multidimensional matrices.* Annals of ATIC-2002, v.I (II), pp. 75–82.

[6] M. Bulat, *About one method of Boolean functions differentiation.* Annals of ATIC-2001, v.I (I), pp. 40–47.

[7] M. Bulat M, *Isomorfismo de grandes sistemas.* Acta Academia 2001, Evrica, Chiinu, pp. 161–170.

[8] M. Bulat, A. Zgureanu, I. Ciobanu, L. Bivol, *The inverse transformations of multidimensional matrices.* ASADE Moldova, August 21, 2007, p. 34.

[9] M. Bulat, D. Leon, A. Zgureanu, I. Ciobanu, L. Bivol, *Generadores de numeros primos y factorizadores de numeros compuestos.* Revista de Matematica: Teoria y Aplicaciones, 2006, 13(1) CIMPA-UCR-CCSS: pp.1–15.

[10] M. Bulat, A. Zgureanu, I. Ciobanu, L. Bivol, *Generating of prime numbers based on the multidimensional matrices.* Intern. Algebraic Conf. dedic. to the 100th an-ry of D. K. Fadeev. St P-rg, Russia, 2007, pp. 98–99.

[11] N. P. Sokolov, *Spatial matrices and their application.* Moscow, 1960.

Aureliu Zgureanu, Received November 10, 2010

Aureliu Zgureanu
The Academy of Transport, Computer Science and Communication
Munceşti, 121-a, Chişinău MD-2002 Moldova
E–mail: *aurelzgureanu@gmail.com*
Phone: +373 79 234829, +373 22 473056