УДК 621.3.06

L. Budaghyan[1], O. Kazymyrov[2]

[1] *University of Bergen, Norway*
[2] *Kharkov National University of Radioelectronics, Ukraine*

VERIFICATION OF RESTRICTED EA-EQUIVALENCE FOR VECTORIAL BOOLEAN FUNCTIONS

We present algorithms for solving the restricted extended affine equivalence (REA-equivalence) problem for any m-dimensional vectorial Boolean functions in n variables. The best of them has complexity $O(2^{2n+1})$ *for REA-equivalence* $F(x) = M_1 \cdot G(x \oplus V_2) \oplus M_3 \cdot x \oplus V_1$. *The algorithms are compared with previous effective algorithms for solving the linear and the affine equivalence problem for permutations by Biryukov et. al.*

Keywords: *EA-equivalence, matrix representation, S-box, vectorial Boolean function.*

Introduction

Vectorial Boolean functions play very important role in ensuring high-level security for modern ciphers. They are used in cryptography as nonlinear combining or filtering functions in the pseudo-random generators (stream ciphers) and as substitution boxes (S-boxes) providing confusion in block ciphers. Up to date an important question of generation of vectorial Boolean functions with optimal characteristics to prevent all known types of attacks remains open. Sometimes equivalence (i.e. EA or CCZ) is used for achieving necessary properties without losing other ones (e.g. δ-uniformity, nonlinearity) [1, 2].

However, very often, inverse problem occurs: it is needed to check several functions for equivalence. For instance, when finding a new vectorial Boolean function it is necessary to verify whether it is equivalent to already known ones as it happens in some of block ciphers, where several substitutions are used [3 – 5].

The complexity of exhaustive search for checking extended affine (EA) equivalence of functions from $\mathcal{F}_2^n = GF(2^n)$ to itself equals $O(n^{3n^2+2n})$. When $n = 6$ the complexity is already 2^{120} that makes it impossible to perform exhaustive computing.

In the paper [1] Alex Biryukov et al. have shown that in case when given functions are permutations of $\mathcal{F}_2^n$, the complexity of determining restricted extended affine equivalence (REA) equivalence equals $O(n^2 \cdot 2^n)$ for the case of linear equivalence and $O(n \cdot 2^{2n})$ for affine equivalence. In this paper we consider more general cases of REA-equivalence for functions from $\mathcal{F}_2^n$ to $\mathcal{F}_2^m$ and specify results, when time complexity can be reduced to polynomial. The complexities of our algorithms and the best previous known ones are given in Table 1.

Table 1

Complexities for solving REA-equivalence problem

Restricted EA-equivalence	Complexity	m	G(x)	Source
$F(x) = M_1 \cdot G(M_2 \cdot x)$	$O(n^2 \cdot 2^n)$	$m = n$	Permutation	[1]
$F(x) = M_1 \cdot G(M_2 \cdot x \oplus V_2) \oplus V_1$	$O(n \cdot 2^{2n})$	$m = n$	Permutation	[1]
$F(x) = M_1 \cdot G(x \oplus V_2) \oplus V_1$	$O(2^{2n+1})$	$m \geq 1$	†	Sec. 3
$F(x) = M_1 \cdot G(x \oplus V_2) \oplus V_1$	$O(m \cdot 2^{3n})$	$m \geq 1$	Arbitrary	Sec. 3
$F(x) = G(M_2 \cdot x \oplus V_2) \oplus V_1$	$O(n^2 \cdot 2^m)$	$m \geq 1$	Permutation	Sec. 3
$F(x) = G(x \oplus V_2) \oplus M_3 \cdot x \oplus V_1$	$O(n \cdot 2^n)$	$m \geq 1$	Arbitrary	Sec. 3
$F(x) = M_1 \cdot G(x \oplus V_2) \oplus M_3 \cdot x \oplus V_1$	$O(2^{2n+1})$	$m \geq 1$	‡	Sec. 3
$F(x) = M_1 \cdot G(x \oplus V_2) \oplus M_3 \cdot x \oplus V_1$	$O(m \cdot 2^{3n})$	$m \geq 1$	Arbitrary	Sec. 3

† – G is under condition $\{2^i \mid 0 \leq i \leq m-1\} \subset \mathrm{img}(G')$ where $G'(x) = G(x) + G(0)$.

‡ – G is under condition $\{2^i \mid 0 \leq i \leq m-1\} \subset \mathrm{img}(G')$ where $G'(x)$ is defined as (5).

Preliminaries

For any positive integers n and m, a function F from $\mathcal{F}_2^n$ to $\mathcal{F}_2^m$ is called differentially δ -uniform if for every $a \in \mathcal{F}_2^n \setminus \{0\}$ and every $b \in \mathcal{F}_2^m$, the equation $F(x)+F(x+a)=b$ admits at most δ solutions [6]. Vectorial Boolean functions used as S-boxes in block ciphers must have low differential uniformity to allow high resistance to differential cryptanalysis [7]. In the important case when $m=n$, differentially 2-uniform functions, called almost perfect nonlinear (APN), are optimal (since for any function $\delta \geq 2$).

The notion of APN function is closely connected to the notion of almost bent (AB) function [8], which can be described in terms of the Walsh transform of a function $F: \mathcal{F}_2^n \mapsto \mathcal{F}_2^m$:

$$\lambda(u,v) = \sum_{x \in \mathcal{F}_2^n} (-1)^{v \cdot F(x)+u \cdot x},$$

where " $\cdot$ " denotes inner products in $\mathcal{F}_2^n$ and $\mathcal{F}_2^m$, respectively. The set $\left\{\lambda(u,v) \,\middle|\, (u,v) \in \mathcal{F}_2^n \times \mathcal{F}_2^m, v \neq 0\right\}$ is called the Walsh spectrum of F and the set $\left\{\left|\lambda(u,v)\right| \,\middle|\, (u,v) \in \mathcal{F}_2^n \times \mathcal{F}_2^m, v \neq 0\right\}$ the extended Walsh spectrum of F. If $m=n$ and the Walsh spectrum of F equals $\left\{0, \pm 2^{\frac{n+1}{2}}\right\}$ then the function F is called AB [8]. AB functions exist for n odd only and oppose an optimum resistance to linear cryptanalysis [9]. Every AB function is APN but the converse is not true in general (see [10] for comprehensive survey on APN and AB functions).

The natural way of representing F as a function from $\mathcal{F}_2^n$ to $\mathcal{F}_2^m$ is by its algebraic normal form (ANF):

$$\sum_{I \subseteq \{1,\ldots,n\}} a_I \left(\prod_{i \in I} x_i\right), \quad a_I \in \mathcal{F}_2^m,$$

the sum being calculated in $\mathcal{F}_2^m$. The algebraic degree $\deg(F)$ of F is the degree of its ANF. F is called affine if it has algebraic degree at most 1 and it is called linear if it is affine and $F(0)=0$.

Any affine function $A: \mathcal{F}_2^n \mapsto \mathcal{F}_2^m$ can be represented in matrix form

$$A(x) = M \cdot x \oplus C \qquad (1)$$

where M is an $n \times m$ matrix and $C \in \mathcal{F}_2^m$. All operations are performed in $\mathcal{F}_2$, thus (1) can be rewritten as

$$\begin{pmatrix} a_0 \\ a_1 \\ \cdots \\ a_{m-1} \end{pmatrix}_x = \begin{pmatrix} k_{0,0} & \cdots & k_{0,n-1} \\ k_{1,0} & \cdots & k_{1,n-1} \\ \vdots & \ddots & \vdots \\ k_{m-1,0} & \cdots & k_{m-1,n-1} \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ \cdots \\ x_{n-1} \end{pmatrix} \oplus \begin{pmatrix} c_0 \\ c_1 \\ \cdots \\ c_{m-1} \end{pmatrix}$$

with $a_i, x_i, c_i, k_{j,s} \in \mathcal{F}_2^m$.

Two functions $F, G: \mathcal{F}_2^n \mapsto \mathcal{F}_2^m$ are called extended affine equivalent (EA-equivalent) if there exist an affine permutation A_1 of $\mathcal{F}_2^m$, an affine permutation A_2 of $\mathcal{F}_2^n$ and a linear function L_3 from $\mathcal{F}_2^n$ to $\mathcal{F}_2^m$ such that

$$F(x) = A_1 \circ G \circ A_2(x) + L_3(x).$$

Clearly A_1 and A_2 can be presented as $A_1(x) = L_1(x) + c_1$ and $A_2(x) = L_2(x) + c_2$ for some linear permutations L_1 and L_2 and some $c_1 \in \mathcal{F}_2^m$, $c_2 \in \mathcal{F}_2^n$.

Definition 1. Functions F and G are called restricted EA-equivalent (REA-equivalent) if some elements of the set $\{L_1(x), L_2(x), L_3(x), c_1, c_2\}$ are in $\{0, x\}$.

There are two special cases

– linear equivalence when $\{L_3(x), c_1, c_2\} = \{0,0,0\}$;

– affine equivalence when $L_3(x) = 0$.

In matrix form EA-equivalence is represented as follows

$$F(x) = M_1 \cdot G\left(M_2 \cdot x \oplus V_2\right) \oplus M_3 \cdot x \oplus V_1,$$

where elements of $\{M_1, M_2, M_3, V_1, V_2\}$ have dimensions $\{m \times m, n \times n, m \times n, m, n\}$.

We say that functions F and F' from $\mathcal{F}_2^n$ to $\mathcal{F}_2^m$ are Carlet Charpin Zinoviev (CCZ) equivalent if there exists an affine permutation $\mathcal{L}$ of $\mathcal{F}_2^n \times \mathcal{F}_2^m$ such that $G_F = \mathcal{L}(G_{F'})$, where $G_H = \left\{(x, H(x)) \,|\, x \in \mathcal{F}_2^n\right\}$, $H \in \{F, F'\}$. CCZ-equivalence is the most general known equivalence of functions for which differential uniformity and extended Walsh spectrum are invariants. In particular every function CCZ-equivalent to an APN (respectively, AB) function is also APN (respectively, AB). EA-equivalence is a specific case of CCZ-equivalence [11]. The algebraic degree of a function is invariant under EA-equivalence but, in general, it is not preserved by CCZ-equivalence.

Verification of Restricted EA-equivalence

Special types of REA-equivalence, which are considered in this paper, are shown in Table 2.

Hereinafter assume that obtaining the value $F(x)$ for any x takes one step. Pre-computed values of function $F(x), F^{-1}(x)$ and corresponding substitutions are used as input for the algorithms.

Table 2

Special types of REA-equivalence

REA-equivalence	Type
$F(x) = M_1 \cdot G(x) \oplus V_1$	I
$F(x) = G(M_2 \cdot x \oplus V_2)$	II
$F(x) = G(x) \oplus M_3 \cdot x \oplus V_1$	III
$F(x) = M_1 \cdot G(x) \oplus M_3 \cdot x \oplus V_1$	IV

Thereafter, complexity of representing functions in needed form is not taken into account, as well as memory needed for data storage. These assumptions are introduced to be able to compare complexities of algorithms of the present paper with those of [1] where the same assumptions were made.

There are $2^{n \cdot m}$ choices of linear mappings. The complexity of obtaining the $m \times n$ matrix M satisfying the equation

$$F(x) = M \cdot G(x)$$

using exhaustive search method is $O(2^n \cdot 2^{m \cdot n})$, where $O(2^{m \cdot n})$ and $O(2^n)$ are the complexities of checking all matrices for all possible $x \in \mathcal{F}_2^n$. Another natural method is based on system of equations. The complexity in this case depends only on the largest of the parameters n and m. Indeed, for square matrices we can benefit from the asymptotically faster Williams method based on system of equations with complexity $O(n^{2.3727})$ [12]. Besides, for $n \le 64$ we can use 64-bit processor instructions to bring the complexity to $O(n^2)$ because of two rows (columns) can be added in 1 step [13]. Since any system of m equations with n variables can be considered as a system of k equations with k variables where $k = \max\{n, m\}$ then the complexity of solving such system is

$$\mu = O(k^2), \qquad (2)$$

which gives the complexity of finding M by this method.

Proposition 1. Any linear function $L : \mathcal{F}_2^n \mapsto \mathcal{F}_2^m$ can be converted to a matrix with the complexity $O(n)$.

Proof. We need to find an $m \times n$ matrix M satisfying $L(x) = M \cdot x$. Suppose

$$\text{rows}_M(i) = (m_{ij}), \forall j \in \{0,1,\ldots,n-1\},$$

$$\text{cols}_M(j) = (m_{ij}), \forall i \in \{0,1,\ldots,m-1\}$$

are the i^{th} row and the j^{th} column of matrix M, respectively. Each value of $x \in \left\{2^i \,|\, 0 \le i \le n-1\right\}$ is equivalent to a vector with 1 at the i^{th} row

$$2^0 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad 2^1 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad \cdots \quad 2^{n-1} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}.$$

Clearly, every column, except the i^{th}, becomes zero when multiplying the matrix M by x. Hence, each column of the matrix M can be obtained using equation

$$\text{cols}_M(i) = L(2^i), \quad i \in \{0,1,\ldots,n-1\}.$$

For obtaining all columns of M it is necessary to compute n values of $L(2^i)$, $0 \le i \le n-1$. Consequently the complexity of transformation is $O(n)$. □

Proposition 2. Any $n \times n$ matrix M can be converted to a linear function $L : \mathcal{F}_2^n \mapsto \mathcal{F}_2^n$ with the complexity $O(n^3)$ filed operations.

Proof. Any linear transformation has the form

$$M \cdot x = L(x) = \sum_{i=0}^{n-1} \delta_i x^{2^i}, \qquad (3)$$

where □ in $\mathcal{F}_{2^n}$. Then using (3) for every $x = 2^i$, $0 \le i \le n-1$, the equation could be rewritten as

$$\begin{pmatrix} \text{cols}_M(0) \\ \text{cols}_M(1) \\ \vdots \\ \text{cols}_M(n-1) \end{pmatrix} =$$

$$= \begin{pmatrix} (2^0)^{2^0} & (2^0)^{2^1} & \cdots & (2^0)^{2^{n-1}} \\ (2^1)^{2^0} & (2^1)^{2^1} & \cdots & (2^1)^{2^{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ (2^{n-1})^{2^0} & (2^{n-1})^{2^1} & \cdots & (2^{n-1})^{2^{n-1}} \end{pmatrix} \times \begin{pmatrix} \delta_0 \\ \delta_1 \\ \vdots \\ \delta_{n-1} \end{pmatrix}.$$

The complexity of acquiring vector $\{\delta_0, \delta_1, \ldots, \delta_{n-1}\}$ is corresponding to obtaining the inverse matrix and equals $O(n^3)$ field operations. □

In practice, the Lagrange interpolation method [14] works faster for $n \le 6$ and much slower otherwise.

Proposition 3. Let $F, G : \mathcal{F}_2^n \mapsto \mathcal{F}_2^m$ and $G'(x) = G(x) \oplus G(0)$. Then the complexity of checking F and G for REA-equivalence of type I equals

– $O(2^{n+1})$ in case when for any $i \in \{0,\ldots,m-1\}$ there exists $x \in \mathcal{F}_2^n$ such that $G'(x) = 2^i$;

– $O(m \cdot 2^{2n})$ in case G is arbitrary.

Proof. Let $F'(x) = F(x) \oplus F(0)$. Then REA-equivalent of type I

$$F'(x) \oplus F(0) = M_1 \cdot G'(x) \oplus M_1 \cdot G(0) \oplus V_1$$

can be rewritten in the following form

$$\begin{cases} F(0) = M_1 \cdot G(0) \oplus V_1; \\ F'(x) = M_1 \cdot G'(x). \end{cases} \qquad (4)$$

In case of $G(0) = 0$, V_1 equals $F(0)$, but in general it's necessary first to find M_1 from equation $F'(x) = M_1 \cdot G'(x)$. If the set $\left\{2^i \mid 0 \le i \le m-1\right\}$ is the

subset of the image set of G', then the problem of finding $m\times m$ matrix M_1 is equivalent to the problem of converting linear function to matrix form with additional testing for all x in $\mathcal{F}_2^n$. It is possible to find M_1 with the complexity O(m) as was shown in Proposition 1. The complexity of finding the pre-images of G' of elements 2^i, $\forall i \in \{0,\dots,m-1\}$ equals $O(2^n)$ as well as the complexity of checking $F'(x) = M_1 \cdot G'(x)$ for given M_1. In cryptography, in most cases $2^n >> m$, so the complexity O(m) can be neglected. Therefore, the total complexity of verification for equivalence of F and G equals $O(2^n + 2^n + m) \approx O(2^{n+1})$.

Let now G be arbitrary and $F'(x)_i$ be the i^{th} bit of $F'(x)$. Denote img(G') the image set of G' and $u_{G'} = |\text{img}(G')|$ the number of elements in img(G'). Let also $N_{G'}$ be any subset of $\mathcal{F}_2^n$ such that $|N_{G'}| = u_{G'}$ and $\left|\{G'(a) \mid a \in N_{G'}\}\right| = u_{G'}$.

Then to find M_1 it is necessary to solve a system below for all $i \in \{0,\dots,m-1\}$

$$F'(x_j)_i = \text{rows}_{M_1}(i)\cdot G'(x_j), \forall x_j \in N_{G'}, 0 \le j \le u_{G'}-1 \Leftrightarrow$$

$$\Leftrightarrow \begin{cases} F'(x_0)_i = \text{rows}_{M_1}(i)\cdot G'(x_0); \\ F'(x_1)_i = \text{rows}_{M_1}(i)\cdot G'(x_1); \\ \dots \\ F'(x_{u_{G'}-1})_i = \text{rows}_{M_1}(i)\cdot G'(x_{u_{G'}-1}). \end{cases}$$

For every i in $\{0,\dots,m-1\}$, the complexity of solving the system highly depends on $u_{G'}$ and m and equals $O(\max\{u_{G'}, m\}^2)$ according to (2). Then the total complexity of obtaining M_1 for all m bits is $O(m\cdot\max\{u_{G'}, m\}^2)$. If value $u_{G'} \approx 2^n$, then $O(m\cdot 2^{2n})$. □

Remark 1. If it is known in advance that functions F and G in Proposition 3 are REA-equivalent of type I, then the complexity of verification $F'(x) = M_1 \cdot G'(x)$ can be ignored and the total complexity for the case $\left\{2^i \mid 0 \le i \le m-1\right\} \subset \text{img}(G')$ becomes $O(2^n)$.

Proposition 4. Let G be a permutation and $F, G : \mathcal{F}_2^n \mapsto \mathcal{F}_2^n$. Then the complexity of checking F and G for REA-equivalence of type II is $O(n^2)$.

Proof. Denote $H(x) = G^{-1}(F(x))$. Then the equation $F(x) = G(M_2 \cdot x \oplus V_2)$ takes the form

$$H(x) = M_2 \cdot x \oplus V_2.$$

Obviously for satisfying REA-equivalence H must be linear and invertible. Arbitrary linear function from $\mathcal{F}_2^n$ to $\mathcal{F}_2^m$ has the form (3) thence consists at most of n monomials. Suppose $\psi(n)$ the complexity of obtaining binary Hamming weight of the monomials exponents in H. Consequently the complexity for checking linearity of H with equals $O(n\cdot\psi(n))$. For the most of modern processors including Intel the value $\psi(n)$ is equal to O(1) [15]. Taking $x = 0$ we obtain $V_2 = H(0)$ and the equivalence can be represented as $H'(x) = M_2 \cdot x$, where $H'(x) = H(x) \oplus H(0)$. For satisfying REA-equivalence conditions the matrix M_2 must be nonsingular. Therefore, the total complexity is equal to the sum of checking linearity of H ($O(n)$), obtaining matrix M_2 ($O(n)$) and checking it on invertibility ($O(n^2)$) and approximately equals $O(n^2)$. □

Proposition 5. Let $F, G : \mathcal{F}_2^n \mapsto \mathcal{F}_2^m$. Then the complexity of checking F and G for REA-equivalence of type III equals $O(n)$.

Proof. Denote $H(x) = F(x) \oplus G(x)$, then REA-equivalence

$$F(x) = G(x) \oplus M_3 \cdot x \oplus V_1$$

takes the form

$$H(x) = M_3 \cdot x \oplus V_1$$

The situation is the same as in Proposition 4, but with arbitrary $m\times n$ matrix. Thus the complexity of obtaining M_3 and V_1 equals $O(n)$.

Every vectorial Boolean function admits the form

$$H(x) = H'(x) \oplus L_H(x) \oplus H(0), \qquad (5)$$

where L_H is a linear function and H' has terms of algebraic degree at least 2.

Proposition 6. Let G' be defined by (5) and $F, G : \mathcal{F}_2^n \mapsto \mathcal{F}_2^m$. Then the complexity of checking F and G for REA-equivalence of type IV equals

– $O(2^{n+1})$ in case $\left\{2^i \mid 0 \le i \le m-1\right\} \subset \text{img}(G')$,

– $O(m\cdot 2^{2n})$ in case G is arbitrary.

Proof. Using (5) REA-equivalence of type IV can be rewritten as

$$F'(x) \oplus L_F(x) \oplus F(0) = M_1 \cdot G'(x) \oplus M_1 \cdot L_G(x) \oplus$$
$$\oplus M_3 \cdot x \oplus M_1 \cdot G(0) \oplus V_1$$

and gives the system of equations

$$\begin{cases} F'(x) = M_1 \cdot G'(x) \\ L_F(x) = M_1 \cdot L_G(x) \oplus M_3 \cdot x \\ F(0) = M_1 \cdot G(0) \oplus V_1 \end{cases}$$

It's easy to see that for a given M_1 one can easily compute M_3 and V_1 from the second and the third equations of the system. The first equation of the system leads to the two different cases for the function G' considered in Proposition 3. Hence, according to Proposition 2, the total complexity for finding G'

equals $O(2^{n+1})$ and $O(m \cdot 2^{2n})$, respectively. It should be noted that the complexity of finding the matrix M_3 is not taken into account since $2^{n+1} >> n$.

If we add one of V_1, V_2 values to REA-equivalence, then the complexity will increase in 2^m or 2^n times respectively. REA-equivalance with V_1, V_2 and corresponding complexities are shown in Table 1. It should be mentioned that types I and III of REA-equivalence are particular cases of type IV. But taking into account different restrictions for the function G it is necessary to check all these types of EA-equivalence.

An algorithm for the type IV of REA-equivalence in case $\left\{2^i \,|\, 0 \le i \le m-1\right\} \subset \mathrm{img}(G')$ is given below.

Algorithm 1 Checking Functions for REA-equivalence of Type IV

Input: $F'(x), L_F(x), F(0), G'(x), L_G(x), G(0)$
Output: True if F is EA-equivalent to G
for $V_2 = 0$ **to** 2^n **do**
  $H'(x) \leftarrow G'(x \oplus V_2)$;
  $L_H(x) \leftarrow L_G(x \oplus V_2)$;
  $H(0) \leftarrow G(V_2)$;
  for $i = 0$ **to** $m-1$ **do**
    $x \leftarrow 2^i$;
    $find(2^i == G(y))$;
    SetColumn(M_1,i,$H(y)$);
  end for
  $V_1 \leftarrow M_1 \cdot H(0) \oplus F(0)$;
  for $i = 0$ **to** $n-1$ **do**
    $x \leftarrow 2^i$;
    SetColumn(M_3,i,$L_F(x) \oplus M_1 \cdot L_H(x)$);
  end for
  for $i = 0$ **to** $2^n - 1$ **do**
    if $F(x)$!= $M_1 \cdot H(x \oplus V_2) \oplus M_3 \cdot x \oplus V_1$ **then**
      goto next V_2;
    end if
  end for
  return True
end for
return False

Conclusions

The present paper studies complexities of checking functions for special cases of EA-equivalence and it is shown that for some of this cases the complexity of checking takes polynomial time. Obtained results give a practical method for checking functions on equivalence. The best result is with the complexity $O(2^{2n+1})$ for checking REA-equivalence of the form $F(x) = M_1 \cdot G(x \oplus V_2) \oplus M_3 \cdot x \oplus V_1$ under some condition on G.

References

1. Alex Biryukov, Christophe De Canniere, An Braeken, and Bart Preneel. A toolbox for cryptanalysis: Linear and affine equivalence algorithms. In Advances in Cryptology - EUROCRYPT 2003, Lecture Notes in Computer Science, pages 33–50. Eli Biham, editor, Springer, 2003.

2. Daemen, J., Rijmen, V. The Design of Rijndael, AES – The Advanced Encryption Standard. Springer, Heidelberg. 2002. ISBN: 978-3-540-42580-9.

3. Daesung Kwon et al., New Block Cipher: ARIA. In Jong In Lim and Dong Hoon Lee, editors, ICISC, volume 2971 of Lecture Notes in Computer Science, pages 432-445. Springer, 2003.

4. R. Oliynykov, I. Gorbenko, V. Dolgov, V. Ruzhentsev, Symmetric block cipher "Kalyna", Applied Radio Electronics 6 (2007), 46-63. In Ukrainian.

5. R. Oliynykov, I. Gorbenko, V. Dolgov, V. Ruzhentsev, Results of Ukrainian National Public Cryptographic Competition, Tatra Mt. Math. Publ. 47 2010, 99-113.

6. K. Nyberg. Differentially uniform mappings for cryptography, Advances in Cryptography, EUROCRYPT'93, LNCS, Springer-Verlag, New York, 765, pp. 55-64, 1994.

7. E. Biham and A. Shamir. Differential Cryptanalysis of DES-like Cryptosystems. Journal of Cryptology, vol. 4, No.1, pp. 3-72, 1991.

8. F. Chabaud and S. Vaudenay. Links between differential and linear cryptanalysis, Advances in Cryptology - EUROCRYPT'94, LNCS, Springer-Verlag, New York, 950, pp. 356-365, 1995.

9. M. Matsui. Linear cryptanalysis method for DES cipher. Advances in Cryptology - EUROCRYPT'93, LNCS, Springer-Verlag, pp. 386-397, 1994.

10. C. Carlet. Vectorial Boolean Functions for Cryptography. Chapter of the monograph "Boolean Models and Methods in Mathematics", Computer Science, and Engineering, Y. Crama and P. Hammer eds, Cambridge University Press, pp. 398-469, 2010.

11. C. Carlet, P. Charpin, and V. Zinoviev. Codes, bent functions and permutations suitable for DES-like cryptosystems. Designs, Codes and Cryptography, 15(2), pp. 125-156, 1998.

12. Virginia Vassilevska Williams, Breaking the Coppersmith-Winograd barrier, November 2011. [Electronic resource] / Mode of access: WWW/URL: http://www.cs.berkeley.edu/~virgi/matrixmult.pdf – Last access: 2013..

13. Sara Robinson, Toward an Optimal Algorithm for Matrix Multiplication. From SIAM News, Volume 38, Number 9, November 2005.

14. R. Lidl and H. Niederreiter. Finite Fields. Volume 20 of Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge (1997).

15. Intel SSE4 Programming Reference, April 2007. [Electronic resource] / Mode of access: WWW/URL: http://software.intel.com/sites/default/files/m/9/4/2/d/5/17971-intel_20sse4_20programming_20reference.pdf – Last access: 2013. – Title from the screen.

Надійшла до редколегії 22.11.2012

Рецензент: д-р техн. наук, проф. І.Д. Горбенко, Харківський національний університет радіоелектроніки, Харків.

ПЕРЕВІРКА ВЕКТОРНИХ БУЛЕВИХ ФУНКЦІЙ НА ОБМЕЖЕНУ РОЗШИРЕНО АФІННУ ЕКВІВАЛЕНТНІСТЬ

Л. Будагян, О. Казимиров

У статті представлені алгоритми для вирішення проблеми обмеженої расширено афінної еквівалентності (OPA-еквівалентності) у разі довільної m-мірної векторної булевої функції від n змінних. Кращий з отриманих алгоритмів має складність $O(2^{2n+1})$ *для OPA-еквівалентності* $F(x) = M_1 \cdot G(x \oplus V_2) \oplus M_3 \cdot x \oplus V_1$. *Складності розроблених методів порівнюються з запропонованими раніше Бірюковим та ін. алгоритмами для вирішення проблем лінійних та афінних еквівалентностей.*

Ключові слова: *PA-еквівалентність, матричне уявлення, S-блок, векторні булеві функції.*

ПРОВЕРКА ВЕКТОРНЫХ БУЛЕВЫХ ФУНКЦИЙ НА ОГРАНИЧЕННУЮ РАСШИРЕННО АФИННУЮ ЭКВИВАЛЕНТНОСТЬ

Л. Будагян, А. Казимиров

В статье представлены алгоритмы для решения проблемы ограниченной расширенно аффинной эквивалентности (ОРА-эквивалентности) в случае произвольной m-мерной векторной булевой функции от n переменных. Лучший из предложенных алгоритмов имеет сложность $O(2^{2n+1})$ *для ОРА-эквивалентности* $F(x) = M_1 \cdot G(x \oplus V_2) \oplus M_3 \cdot x \oplus V_1$. *Сложности разработанных методов сравниваются с предложенными ранее Бирюковым и др. алгоритмами для решения проблем линейных и аффинных эквивалентностей.*

Ключевые слова: *РА-эквивалентность, матричное представление, S-блок, векторные булевы функции.*