Tatra Mt. Math. Publ. **37** (2007), 113–122

TATRA MOUNTAINS
Mathematical Publications

A LINEAR EQUIVALENCE ALGORITHM

Marcin Serweciński

ABSTRACT. J. Fuller and W. Millan in: [*Linear redundancy in S-boxes.* In: FSE '03 (T. Johansson, ed.), Lect. Notes Comput. Sci., Vol. 2887, Springer--Verlag, Berlin, 2003, pp. 74–86] proved that coordinate Boolean functions of the AES s-box are linearly equivalent. Although to this time no one described the attack based on this observation, we believe that it affects the complexity of the cryptographic attack on AES. Therefore, a fast algorithm for testing if two Boolean functions are linearly equivalent can be very useful when designing and analyzing S-boxes. In this contribution we present a new algorithm for solving this task. Our algorithm not only tests if two Boolean functions are linearly equivalent but also creates, if it is possible, a matrix which establishes a linear equivalence for considered functions.

1. Preliminaries

In this paragraph we introduce a few basic definitions and notations which will be used in a sequel of this paper.

Let $\mathbb{F}_2$ be the two-element finite field. By $\mathbb{F}_2^n$ we denote an n-dimensional vector space of n-tuples of elements from $\mathbb{F}_2$. By e_i we denote the i-th vector of the standard basis of the vector space $\mathbb{F}_2^n$.

Let $v \in \mathbb{F}_2^n$. The number of ones in the vector v is called *the Hamming weight of the vector* v and will be denoted $\mathrm{hwt}(v)$. Let $v, w \in \mathbb{F}_2^n$. *The Hamming distance of* v *and* w, denoted as $d(v, w)$, is the number of coordinates on which the vectors v and w differs.

A Boolean function is a mapping from $\mathbb{F}_2^n$ into $\mathbb{F}_2$. Observe, that each Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}$ can be represented by a vector, called *the truth table*, defined in the following way

$$\big(f(\alpha_0), f(\alpha_1), \dots, f(\alpha_{2^n-1})\big) \in \mathbb{F}_2^{2^n},$$

2000 Mathematics Subject Classification: 94A60.
Keywords: Boolean functions, linear equivalence, cryptology.

where α_i denotes the vector in $\mathbb{F}_2^n$ whose integer representation is i, $0 \leq i \leq 2^n - 1$. Using these notations we can define in the natural way *the Hamming weight of the function* f as the Hamming weight of its truth table. Similarly, the Hamming distance of two Boolean functions is the Hamming distance of their truth tables.

It is easy to see, that each Boolean function f can be uniquely represented by a polynomial of the form

$$F(X_1, \ldots, X_n) = \bigoplus_{i=(i_1,\ldots,i_n)\in\mathbb{F}_2^n} a_i X_1^{i_1} X_2^{i_2} \ldots X_n^{i_n}, \qquad a_i \in \mathbb{F}_2.$$

The polynomial F is called *the algebraical normal form of the Boolean function* f. *The algebraic degree of the Boolean function* is defined as the algebraic degree of its algebraical normal form.

One of the most important families of the Boolean functions is the family of the affine Boolean functions. We say that a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is an affine function, if it can be described by the following formula

$$f(x) = \bigoplus_{i=1}^{n} x_i a_i \oplus b,$$

where $x = (x_1, \ldots, x_n), a = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$ and $b \in \mathbb{F}_2$. It is easy to see, that degree of the affine function is at most one. In the situation when $b = 0$, the function f will be called *the linear Boolean function*. The set of all affine Boolean functions over $\mathbb{F}_2^n$ will be denoted by $\mathrm{Aff}(n)$.

By *the nonlinearity of the Boolean function f*, denoted by N_f, we mean the distance between function f and the set $\mathrm{Aff}(n)$. It can be expressed as follows

$$N_f = \min\big\{d(f, g) : g \in \mathrm{Aff}(n)\big\}.$$

Let $\mathfrak{B}_n$ be a set of all Boolean functions over $\mathbb{F}_2^n$. We can consider the group $GL(n, 2)$ action on the set $\mathfrak{B}_n$ defined as follows

$$(A \cdot f)(x) = f(A^{-1}x), \text{ for all } x \in \mathbb{F}_2^n,$$

where $f \in \mathfrak{B}_n$ and $A \in GL(n, 2)$.

We say that the Boolean functions f and g are linearly equivalent if and only if they belong to the same orbit of the group $GL(n, 2)$ action on the set $\mathfrak{B}_n$. In other words, there must exist a matrix $A \in GL(n, 2)$, such that

$$\underset{x\in\mathbb{F}_2^n}{\forall} f(Ax) = g(x).$$

It is known that the number of equivalence classes of Boolean functions increases exponentially with n. H a r r i s o n [1] showed how to compute this number using cycle index. M a i o r a n a [3] showed how to classify Boolean functions for $n = 6$, but there is still no good criterion for testing if two Boolean functions belongs to the same equivalence class.

It is known, that some properties of Boolean functions, like the Hamming weight, the algebraic degree and the nonlinearity are invariant under action on the group $GL(n, 2)$. Let the symbol Φ denote the set of all known invariants of Boolean functions with respect to the linear equivalence. Let $\phi \in \Phi$. By $\phi(f) = \phi(g)$, we denote the fact, that the Boolean functions f and g have the same value of invariant ϕ.

So, when we want to test, if the Boolean functions f and g are linearly equivalent, we firstly test if

$$\underset{\phi \in \Phi}{\forall} \phi(f) = \phi(g).$$

If considered functions differ in any of invariants, then we know that they are not linearly equivalent. If not, we must test in another way if they are linearly equivalent or not.

2. The main idea of the algorithm

In this paragraph we show how to test if two Boolean functions are linearly equivalent. We show that for particular Boolean functions f and g, only some of the vectors of $\mathbb{F}_2^n$ can play a role of the rows of the matrix which establishes linear equivalence between functions f and g.

Let $a = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$. By l_a we denote the linear Boolean function defined in the following way

$$l_a(x) = \bigoplus_{i=1}^{n} a_i x_i, \quad \text{for all } x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n.$$

Let $A \in GL(n, 2)$. By $l_A : \mathbb{F}_2^n \to \mathbb{F}_2^n$ we denote the linear mapping defined in the following way

$$l_A(x) := A \cdot x, \qquad x \in \mathbb{F}_2^n.$$

Let $A_{[i]}$ denote the i-th row of the matrix A.

Theorem 2.1. *Let $f, g \in \mathfrak{B}_n$ be linearly equivalent Boolean functions and let $A \in GL(n, 2)$ be a matrix such that*

$$f(x) = g(Ax), \qquad x \in \mathbb{F}_2^n.$$

Then for all $i \in \{1, 2, \ldots, n\}$ the functions $f \cdot l_{e_i}$ and $l_{A_{[i]}}$ are linearly equivalent.

Proof. Let $i \in \{1, \ldots, n\}$. Let us consider a superposition of the Boolean function $f \cdot l_{e_i}$ and the linear mapping l_A. It is easy to see that

$$(f \cdot l_{e_i} \circ l_A)(x) = f(Ax) \cdot l_{e_i}(Ax) = g(x) \cdot l_{e_i}(Ax).$$

The vector e_i can be treated as a matrix with one column and n rows. Then

$$l_{e_i}(Ax) = e_i^T \cdot A \cdot x = A_{[i]}^T \cdot x = l_{A_{[i]}}(x).$$

Therefore, the functions $f \cdot l_{e_i}$ and $g \cdot l_{A_{[i]}}$ are linearly equivalent. □

Theorem 2.1 can be easily extended:

Theorem 2.2. *Let $f, g \in \mathfrak{B}_n$ be linearly equivalent Boolean functions and $A \in GL(n, 2)$ be a matrix, such that $f(Ax) = g(x)$. Let $a_i \in \mathbb{F}_2^n$ be the i-th row of the matrix A. Then, for $1 \leq k_1 < k_2 < \ldots < k_s \leq n$ the functions $f \cdot l_{e_{k_1}} \cdot \ldots \cdot l_{e_{k_s}}$ and $g \cdot l_{A_{[k_1]}} \cdot \ldots \cdot l_{A_{[k_s]}}$ are linearly equivalent.*

Let f and g be linearly equivalent Boolean functions. Presented above theorem shows us, how to obtain a matrix $A \in GL(n, 2)$ such that $f \circ l_A = g$. Let us consider the Boolean function $f \cdot l_{e_1}$. We know, that this function must be linearly equivalent with the function $g \cdot l_{A_{[1]}}$, where $A_{[1]}$ is the first row of the matrix A.

So, $A_{[1]} \in \mathbb{F}_2^n$ (the first row of the matrix A) must belong to the following set

$$L_1 = \big\{a \in \mathbb{F}_2^n : \underset{\phi \in \Phi}{\forall} \phi(f \cdot l_{e_1}) = \phi(g \cdot l_a)\big\}.$$

In a similar way we can compute sets: L_2, L_3, …, L_n containing possible values of the second row of the matrix A, the third row of the matrix A, ..., the n-th row of the matrix A.

We can interpret each tuple $(a_1, a_2, \ldots, a_n) \in L_1 \times L_2 \times \ldots \times L_n$ as a matrix, such that the first row is equal to a_1, the second to a_2, …, the n-th row to a_n. Therefore, by testing all matrices chosen in that way, we can test if the Boolean functions f and g are linearly equivalent or not.

Definition 2.3. Let $f, g \in \mathfrak{B}_n$ and $S \subset \Phi$. Then

$$L_{(S,f)}(g) := \big\{a \in \mathbb{F}_2^n : \underset{\phi \in S}{\forall} \phi(f) = \phi(g \cdot l_a)\big\}.$$

In the situation when $S = \{\phi\}$ instead of $L_{(S,f)}(g)$ we will write $L_{(\phi,f)}(g)$ or even $L_{\left(\phi,\phi(f)\right)}(g)$.

Let $f, g \in \mathfrak{B}_n$ and $1 \le i \le n$. Observe, that $L_i = L_{(\Phi, f\cdot l_{e_i})}(g)$

Lemma 2.4. *Let $f, \xi_1, \xi_2 \in \mathfrak{B}_n$ and $\phi \in \Phi$ be such that, $\phi(\xi_1) \neq \phi(\xi_2)$, then*

$$L_{(\phi,\xi_1)}(f) \cap L_{(\phi,\xi_2)}(f) = \emptyset.$$

Lemma 2.5. *Let $f, g \in \mathfrak{B}_n$ be linearly equivalent functions. Then*

$$\underset{\substack{\phi\in\Phi\\ \xi\in\mathfrak{B}_n}}{\forall} \quad \#L_{(\phi,\xi)}(f) = \#L_{(\phi,\xi)}(g).$$

P r o o f. Let $A \in GL(n, 2)$ be a matrix, such that

$$(f \circ l_A)(x) = g(x), \quad \text{for all } x \in \mathbb{F}_2^n.$$

Observe, that

$$\begin{aligned}
\#L_{(\phi,\xi)}(f) &= \#\{a \in \mathbb{F}_2^n : \phi(f \cdot l_a) = \phi(\xi)\} \\
&= \#\left\{a \in \mathbb{F}_2^n : \phi\big((f \cdot l_a) \circ l_A\big) = \phi(\xi)\right\} \\
&= \#\left\{a \in \mathbb{F}_2^n : \phi\big((f \circ l_A) \cdot (l_a \circ l_A)\big) = \phi(\xi)\right\} \\
&= \#\left\{a \in \mathbb{F}_2^n : \phi\big(g \cdot (l_a \circ l_A)\big) = \phi(\xi)\right\}.
\end{aligned}$$

It is obvious that the function $(l_a \circ l_A)(x)$ is a linear mapping. So, there must exist $a^{'} \in \mathbb{F}_2^n$ such that $(l_a \circ l_A)(x) = l_{a'}(x)$, for all $x \in \mathbb{F}_2^n$. Therefore

$$\begin{aligned}
\#\left\{a \in \mathbb{F}_2^n : \phi\big(g \cdot (l_a \circ l_A)\big) = \phi(\xi)\right\} &= \#\left\{a \in \mathbb{F}_2^n : \phi(g \cdot l_a) = \phi(\xi)\right\} \\
&= \#L_{(\phi,\xi)}(g).
\end{aligned}$$

□

When we are testing the linear equivalence of the Boolean functions $f, g \in \mathfrak{B}_n$, we often replace the function f by some Boolean function $\hat{f} \in \mathfrak{B}_n$, linearly equivalent with the function f, which is more convenient for testing the linear equivalence with g. The next theorem shows how to do it properly.

Theorem 2.6. *Let $f \in \mathfrak{B}_n$ and $S \subset \Phi$. There exists the Boolean functions $\hat{f}, \xi_1, \ldots, \xi_m \in \mathfrak{B}_n$ and $1 \le r \le m$, such that f and $\hat{f}$ are linearly equivalent,*

$$\#L_{(S,\xi_1)}(\hat{f}) \le \#L_{(S,\xi_2)}(\hat{f}) \le \ldots \le \#L_{(S,\xi_m)}(\hat{f}),$$

$$\mathit{span}\Big(L_{(S,\xi_1)}(\hat{f}) \cup L_{(S,\xi_2)}(\hat{f}) \cup \ldots \cup L_{(S,\xi_m)}(\hat{f})\Big) = \mathbb{F}_2^n$$

and

$$e_1, e_2, \ldots, e_{\dim\left(L_{(S,\xi_1)}(\hat{f})\right)} \in L_{(S,\xi_1)}(\hat{f}),$$
$$e_{\dim\left(L_{(S,\xi_1)}(\hat{f})\right)+1}, \ldots, e_{\dim\left(L_{(S,\xi_1)}(\hat{f}) \cup L_{(S,\xi_2)}(\hat{f})\right)} \in L_{(S,\xi_2)}(\hat{f}),$$
$$\vdots$$
$$e_{\dim\left(L_{(S,\xi_1)}(\hat{f}) \cup L_{(S,\xi_2)}(\hat{f}) \cup \ldots \cup L_{(S,\xi_{r-1})}(\hat{f})\right)+1}, \ldots, e_n \in L_{(S,\xi_r)}(\hat{f}).$$

P r o o f. Let $f \in \mathfrak{B}_n$. If we take into account Lemma 2.4, we get immediately, that there must exist $\xi_1, \xi_2, \ldots, \xi_m \in \mathbb{N}_0$, such that

$$\#L_{(S,\xi_1)}(f) \leq \#L_{(S,\xi_2)}(f) \leq \ldots \leq \#L_{(S,\xi_m)}(f),$$

$$\operatorname{span}\left(L_{(S,\xi_1)}(f) \cup \#L_{(S,\xi_2)}(f) \cup \ldots \cup L_{(S,\xi_m)}(f)\right) = \mathbb{F}_2^n.$$

It is easy to see, that we can choose a basis $(\alpha_1, \ldots, \alpha_n)$ of the vector space $\mathbb{F}_2^n$ as follows

$$\alpha_1, \alpha_2, \ldots, \alpha_{\dim\left(L_{(S,\xi_1)}(f)\right)} \in L_{(S,\xi_1)}(f),$$

$$\alpha_{\dim\left(L_{(S,\xi_1)}(f)\right)+1}, \ldots, \alpha_{\dim\left(L_{(S,\xi_1)}(f) \cup L_{(S,\xi_2)}(f)\right)} \in L_{(S,\xi_2)}(f),$$
$$\vdots$$
$$\alpha_{\dim\left(L_{(S,\xi_1)}(f) \cup \ldots \cup L_{(S,\xi_{r-1})}(f)+1\right)}, \ldots \alpha_n \in L_{(S,\xi_r)}(\hat{f}).$$

Let us consider an invertible linear mapping l_A defined on the basis $(\alpha_1, \ldots, \alpha_n)$ in the following way

$$l_A(\alpha_i) = e_i, \quad \text{for } i = 1, \ldots, n.$$

Let $A \in GL(n, 2)$ be a matrix of l_A and let $1 \leq i \leq n$. Observe, that

$$(f \cdot l_{\alpha_i}) \circ l_{A_T} = (f \circ l_{A^T}) \cdot (l_{\alpha_i} \circ l_{A^T})$$

Note also, that

$$(l_{\alpha_i} \circ l_{A^T})x = \alpha_t^T \cdot A^T \cdot x = (A \cdot \alpha_i)^T \cdot x = e_i^T \cdot x = l_{e_i}(x)$$

Therefore, if $\hat{f} := f \circ l_{A^T}$

$$\underset{S \in S}{\forall} S(f \cdot l_{\alpha_i}) = S(\hat{f} \cdot l_{e_i}).$$

$\square$

Let $f, g \in \mathfrak{B}_n$ and let $A \in GL(n,2)$ be a matrix such that $f \circ l_A = g$. The method of obtaining the matrix A, which has been discussed above can be summarized in the following algorithm:

Algorithm LIN_EQUIV
Input: $f, g \in \mathfrak{B}_n$

if $\underset{\phi\in\Phi}{\exists}\ \phi(f) \neq \phi(g)$ **then**
  return "f and g are not linearly equivalent"
end if

$(\hat{f}, A) \leftarrow$ Optimization(f) {see Theorem 2.6}

for $i \leftarrow 1$ **to** n **do**
  $L_i \leftarrow \left\{ a \in \mathbb{F}_2^n : \underset{\phi\in\Phi}{\forall}\ \phi(\hat{f} \cdot l_{e_i}) = \phi(g \cdot l_a) \right\}$ {see Theorem 2.1}
  if $L_i = \emptyset$ **then**
    return "f and g are not linearly equivalent"
  end if
end for

$m \leftarrow \lfloor \frac{n}{2} \rfloor$

for $i \leftarrow 1$ **to** m **do**
  $L_i^2 \leftarrow \left\{ (a,b) \in L_{2i-1} \times L_{2i} : \underset{\phi\in\Phi}{\forall}\ \phi(\hat{f} \cdot l_{e_{2i-1}} \cdot l_{e_{2i}}) = \phi(g \cdot l_a \cdot l_b) \right\}$
  {Theorem 2.2}

  if $L_i^2 = \emptyset$ **then**
    return "f and g are not linearly equivalent"
  end if
end for

if $n = 2m$ **then**
  $G \leftarrow L_1^2 \times L_2^2 \times \ldots \times L_m^2$
else
  $G \leftarrow L_1^2 \times L_2^2 \times \ldots \times L_m^2 \times L_n$
end if

```
for X ∈ G do
  if rank(X) ≠ n then
    continue the loop with the next matrix
  else
    if f ∘ l_X = g then
      return X · A^T
    end if
  end if
end for

return "f and g are not linearly equivalent"
```

This is the simplest version of our algorithm. It can be easily extended by using the divide and conquer method. In the next section we will show that even this simple version is very effective on the AES coordinate Boolean functions.

3. Remarks

The tests we have made, showed that our algorithm is very efficient for some functions and not efficient for others. For example our algorithm works very good for AES coordinate Boolean functions and works very unefficient for bent functions.
The main reason of this situation is that there is no good set of invariants, which could be used to distinguish nonequivalent Boolean functions. In our tests to construct the set L_i we use only the Hamming weight. In this situation the set L_i can be defined in more convenient way using the Walsh-Hadamard transform.

Definition 3.1. Let $f \in \mathfrak{B}_n$. The Walsh-Hadamard transform of f is a function $f^w : \mathbb{F}_2^n \to \mathbb{Z}$ defined in the following way

$$f^w(a) := \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+a\cdot x}, \qquad a \in \mathbb{F}_2^n.$$

Note that $\mathrm{hwt}(f \cdot l_a) = \frac{f^w(a)+2\mathrm{hwt}(f)}{4}$. So the set $L_{(\mathrm{hwt},\xi)}$ can be defined as follows

$$L_{(\mathrm{hwt},\xi)}(f) = \{a \in \mathbb{F}_2^n : f^w(a) = 4\mathrm{hwt}(\xi) - 2\mathrm{hwt}(f)\}.$$

It is known, that the Walsh-Hadamard transform of the Boolean function of the degree greater than 1 has at least two different values for non-zero arguments. So, after the optimization, the size of the set $L_{(\mathrm{hwt},\xi)}$ can be bounded by 2^{n-1}. It is known, that for bent functions this bound is strict. On the other hand for many functions the sizes of the $L_{(\mathrm{hwt},\xi)}$ are much smaller than 2^{n-1}. Next example shows how our algorithm works on the coordinate Boolean functions of the AES s-boxes.

To simplify the notation we will write vectors of the vector space $\mathbb{F}_2^n$ as integer numbers obtained in the following way

$$v = [v_1, \dots, v_n] \in \mathbb{F}_2^n \qquad \longmapsto \qquad \sum_{i=1}^{n} 2^{n-i} v_i.$$

Let aes_1 and aes_2 be the coordinate Boolean functions of two last significant bits of the output of AES s-box. In the first step we derive the sets

$$L_{(\mathrm{hwt},i)}(\mathrm{aes}_1) = \{a \in \mathbb{F}_2^n : \mathrm{hwt}(\mathrm{aes}_1 \cdot l_a) = i\}, \qquad i \in \{1, \dots 256\}.$$

Only 16 of them are nonempty. Moreover, the sets $L_{(\mathrm{hwt},56)}(\mathrm{aes}_1)$, $L_{(\mathrm{hwt},57)}(\mathrm{aes}_1)$ have the smallest size

$$\begin{aligned} L_{(\mathrm{hwt},56)}(\mathrm{aes}_1) &= \{45, 103, 142, 163, 196\}, \\ L_{(\mathrm{hwt},57)}(\mathrm{aes}_1) &= \{13, 25, 59, 112, 117, 134, 135, 136\}. \end{aligned}$$

Due to Theorem 2.6 we can replace the function aes_1 by the function $\overline{\mathrm{aes}_1}$ (more convenient for our purposes) defined as follows

$$\overline{\mathrm{aes}_1}(x) := \mathrm{aes}_1(A^T x),$$

where A is a matrix of the linear mapping l_A defined on the basis of $\mathbb{F}_2^8$ in the following way

$$\begin{array}{llll} l_A(45) = 1, & l_A(103) = 2, & l_A(142) = 4, & l_A(13) = 8, \\ l_A(25) = 16, & l_A(59) = 32, & l_A(112) = 64, & l_A(117) = 128. \end{array}$$

Note, that

$$1, 2, 4, 8, 16 \in L_{(\mathrm{hwt},56)}(\overline{\mathrm{aes}_1}) \quad \text{and} \quad 32, 64, 128 \in L_{(\mathrm{hwt},57)}(\overline{\mathrm{aes}_1}).$$

The functions aes_1 and $\overline{\mathrm{aes}_1}$ are linearly equivalent. We show that the functions $\overline{\mathrm{aes}_1}$ and aes_2 are linearly equivalent too. Let $B \in GL(8, 2)$ be the matrix of the linear mapping such that

$$\overline{\mathrm{aes}_1}(Bx) = \mathrm{aes}_2(x), \qquad x \in \mathbb{F}_2^n.$$

In the next step we derive the sets L_1, L_2, …, L_8.

$$\begin{aligned} L_1 = L_2 = L_3 &= \{77, 106, 151, 176, 253\}, \\ L_4 = L_5 = L_6 = L_7 = L_8 &= \{73, 95, 97, 139, 220, 223, 236, 238\}. \end{aligned}$$

Using L_1, L_2, ..., L_8 we obtain L_1^2, L_2^2, L_3^2, L_4^2.

$$\begin{aligned}
L_1^2 &= \{(77,106),(77,151),(106,77),(106,176),\\
&\quad (151,77),(151,176),(176,106),(176,151)\},\\
L_2^2 &= \{(77,73),(77,97),(77,139),(77,238),\\
&\quad (106,95),(106,97),(106,139),(106,223),\\
&\quad (151,73),(151,220),(151,236),(151,238),\\
&\quad (176,95),(176,220),(176,223),(176,236)\},\\
L_3^2 = L_4^2 &= \{(73,139),(73,220),(95,139),(95,236),\\
&\quad (97,223),(97,238),(139,73),(139,95),\\
&\quad (220,73),(220,223),(223,97),(223,220),\\
&\quad (236,95),(236,238),(238,97),(238,236)\}.
\end{aligned}$$

In the last step of the algorithm we consider all 32768 matrices which belong to the set $G = L_1^2 \times L_2^2 \times L_3^2 \times L_4^2$. Note that only 3072 of them are invertible.

There exists exactly eight matrices $B \in GL(8,2)$ such that

$$\overline{\mathrm{aes}_1}(Bx) = \mathrm{aes}_2(x), \qquad x \in \mathbb{F}_2^n.$$

By multiplying matrix A^T by each of this matrices we get the matrix of the linear mapping which establishes a linear equivalence between the functions aes_1 and aes_2. So we reduced the number of possible matrices to at most 2^{12}.

REFERENCES

1. HARRISON, M. A.: *On classification of Boolean functions by the general linear and affine groups*, J. Soc. Indust. Appl. Math. **12**(1964), 285–299.
2. FULLER, J.—MILLAN, W.: *Linear Redundancy in S-Boxes.* In: FSE '03 (T. Johansson, ed.), Lect. Notes Comput. Sci., Vol. 2887, Springer-Verlag, Berlin, 2003, pp. 74–86.
3. MAIORANA, J. A.: *A Classification of the Cosets of the Reed-Muller code r(1,6)*, Math. Comput., **57**(1991), 403–414.

Received November 10, 2005

University of Silesia
ul. Bankowa 14
PL–40-007 Katowice
POLAND
E-mail: ptasiek@ux2.math.us.edu.pl