Math-Net.Ru

Общероссийский математический портал

Nguyen Bui Cuong, Nguyen Van Long, Hoang Dinh Linh, Анализ влияния линейной избыточности в S-блоках на аффинную эквивалентность в раундовых функциях XSL-схем, *Матем. вопр. криптогр.*, 2017, том 8, выпуск 1, 51–68

DOI: <https://doi.org/10.4213/mvk214>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 3.85.73.92

26 апреля 2023 г., 02:55:57

Analyzing the influence of linear redundancy in S-boxes on the affine equivalence within XSL-like round functions

Nguyen Bui Cuong, Nguyen Van Long, Hoang Dinh Linh

Institute of Cryptography Science and Technology, Government Information Security Committee,
Viet Nam

Получено 10.VI.2016

Abstract. We show that S-boxes based on finite field inversion always possess complete linear redundancy. Next, we consider the influence of linear redundancy of S-boxes on the affine equivalence of component functions within XSL-like round functions in the general case. Then, we propose an effective practical approach to test this. Finally, some experimental results on the round functions within the Kuznyechik and AES are presented.

Keywords: Boolean functions, S-boxes, round function, block cipher, affine equivalence, linear redundancy

Анализ влияния линейной избыточности в S-блоках на аффинную эквивалентность в раундовых функциях XSL-схем

Нгуен Буй Куонг, Нгуен Ван Лонг, Хоанг Динь Линь

Институт криптографических наук и технологий, Государственный комитет защиты информации, Вьетнам

Аннотация. Показано, что S-боксы, основанные на операции обращения в конечном поле, всегда обладают полной линейной избыточностью. В общем случае рассматривается влияние линейной избыточности S-боксов на аффинную эквивалентность компонентных функций в раундовых функциях XSL-схем. Предлагается эффективный практический метод проверки наличия этих свойств. Приводятся экспериментальные результаты для раундовых функций в Кузнечике и AES.

Ключевые слова: булевы функции, S-боксы, раундовая функция, блочный шифр, аффинная эквивалентность, линейная избыточность

Citation: *Mathematical Aspects of Cryptography*, 2017, v. 8, № 1, pp. 51–68 (Russian)

© Академия криптографии Российской Федерации, 2017 г.

1. Introduction

S-boxes play an important role in the round functions of any block cipher. The cryptographic properties of S-boxes are often meticulously examined including linear and differential approximations in order to ensure that the block cipher is secure against some particular attacks such as linear cryptanalysis, differential analysis etc. Normally, these properties may be provided by choosing optimized Boolean functions as the component functions of S-boxes. However, to find a compromise for optimizing Boolean functions in order to satisfy many security criteria simultaneously is truly a problem that need to be considered carefully. The linear redundancy, which was proposed in [1], and the nonlinearity is such a case. In [1] the authors had proved that the inversion mapping of the finite field $\mathbb{F}_{2^8}$ has high nonlinearity and complete linear redundancy. There are many S-boxes in real block ciphers (such as AES [2], Camellia [3], Shark [4], Square, Hierocrypt, SC2000, BMGL, MUGI etc.) based on the inversion mapping of the finite field. The designers of these algorithms have applied some affine transformations to input and output to cover the inversion mapping of the finite field and stated that this will prevent algebraic attacks in $\mathbb{F}_{2^8}$.

Related works. The concept of linear redundancy was proposed and considered for the first time in [1]. Recently, linear redundancy was considered as an important criterion for generating S-boxes with good cryptographic properties [5]. Moreover, a theoretical approach to examine the affine equivalence of round functions within AES was presented in [6]. However, these results were considered only in specific cases.

Our contributions. In this paper we clarify some results on the linear redundancy of S-boxes. In particular, we show that S-boxes generated by the algebraic method based on the inversion mapping often possess complete linear redundancy. Next, we study the influence of linear redundancy of S-boxes on the affine equivalence within XSL-like round functions, extending some theoretical results from [6]. Moreover, we propose a practical approach permitting to examine the affine equivalence of coordinate functions within XSL-like round functions in general case and discuss some experimental results on the round functions of Kuznyechik and AES.

Organization. This paper is organized as follows: section 2 introduces some basic definitions and general notations connected with linear redundancy of S-boxes. In this section we study the linear redundancy of AES-like S-boxes in detail. In section 3 we consider the affine equivalence of XSL-like round functions for S-boxes that are either completely linear redundant or incompletely linear redundant. Finally, some experimental results are presented.

2. Preliminaries

2.1. Notions

In this section we recall some necessary algebraic notions. An arbitrary element of a finite field $\mathbb{F}_{2^n}$ may be represented as follows:

$$b_{n-1}\alpha^{n-1} + b_{n-2}\alpha^{n-2} + \dots + b_0, \quad b_i \in \mathbb{F}_2,$$

where α is the primitive element of $\mathbb{F}_{2^n}$; the binary representation of this element is $b_{n-1}b_{n-2}\dots b_0$. For $\lambda \in \mathbb{F}_{2^n}$ the trace of λ with respect to the subfield $\mathbb{F}_2$ is defined as

$$\text{Tr}_{\mathbb{F}_2}^{\mathbb{F}_{2^n}}(\lambda) = \lambda + \lambda^2 + \lambda^{2^2} + \dots + \lambda^{2^{n-1}}.$$

For simplicity in the case of the field $\mathbb{F}_{2^n}$ and subfield $\mathbb{F}_2$ we will use the notation $\text{Tr}(\lambda)$ instead of $\text{Tr}_{\mathbb{F}_2}^{\mathbb{F}_{2^n}}(\lambda)$.

Let $\{\alpha_{n-1}, \dots, \alpha_0\}$ be a basis of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$ and $\{\beta_{n-1}, \dots, \beta_0\}$ be its dual basis.

Further, let $f(x_{n-1}, \dots, x_0) = (f_{n-1}(x), \dots, f_0(x))$ be a permutation on $\mathbb{F}_2^n$. Since $\mathbb{F}_{2^n}$ and $\mathbb{F}_2^n$ are isomorphic then f may be considered as a mapping from $\mathbb{F}_{2^n}$ to $\mathbb{F}_{2^n}$ defined as follows:

$$\begin{aligned} f: \mathbb{F}_{2^n} &\rightarrow \mathbb{F}_{2^n}, \\ x = \sum_{i=0}^{n-1} x_i \alpha_i &\mapsto (f_{n-1}(x) f_{n-2}(x) \dots f_0(x)) = \sum_{i=0}^{n-1} \alpha_i f_i(x), \end{aligned}$$

here f is also a bijection on $\mathbb{F}_{2^n}$ (see [7]). For this representation of f the output coordinate functions of $f(x)$ are defined as (see [7])

$$f_i(x) = \text{Tr}(f(x) \times \beta_i), \quad x = \sum_{i=0}^{n-1} x_i \alpha_i.$$

In the case of n -bit S-box $S: \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}$ we can consider S as a mapping from $\mathbb{F}_2^n$ to itself:

$$S: \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n, \quad x = (x_{n-1}, \dots, x_0) \mapsto (S_{n-1}(x), \dots, S_0(x)).$$

Thus, S-box S has n coordinate functions S_i which may be defined by the trace functions:

$$S_i(x) = \text{Tr}(S(x) \times \beta_i) \quad \text{for } 0 \leq i \leq n-1.$$

In addition, for a linear combination of coordinate functions (namely component function of S-box) we have:

$$\begin{aligned} S_\lambda(x) &= \sum_{i=0}^{n-1} \lambda_i S_i(x) = \sum_{i=0}^{n-1} \lambda_i \text{Tr}(\beta_i \times S(x)) = \\ &= \text{Tr} \left(\left(\sum_{i=0}^{n-1} \lambda_i \beta_i \right) \times S(x) \right) = \text{Tr}(\beta_\lambda \times S(x)), \end{aligned}$$

where $\lambda = (\lambda_{n-1}, \lambda_{n-2}, \dots, \lambda_0)$, $\beta_\lambda = \sum_{i=0}^{n-1} \lambda_i \beta_i$. Since $(\beta_{n-1}, \dots, \beta_0)$ is a basis of $\mathbb{F}_{2^n}$ then β_λ will have all values in $\mathbb{F}_{2^n}$, therefore all 2^n Boolean component functions of S may be represented by the trace function.

2.2. Linear redundancy in S-boxes

First, we recall the notion of affine equivalence between two Boolean functions. Two n -input Boolean functions f and g are *affine equivalent* if there exist a non-singular linear transformation $\mathcal{D}$ from $\mathbb{F}_2^n$ into itself, two n -element binary vectors a, b and a binary constant c such that

$$g(x) = f(\mathcal{D}x \oplus a^\top) \oplus bx^\top \oplus c,$$

where $bx^\top = b_{n-1}x_{n-1} \oplus b_{n-2}x_{n-2} \oplus \dots \oplus b_0x_0$.

In [1], Fuller and Millan have proposed a criterion of existence of S-box possessing good cryptographic properties, the notion of *possession of linear redundancy*, which is defined as follows.

Definition 1. A $n \times n$ S-box with the coordinate Boolean functions $(f_{n-1}, f_{n-2}, \dots, f_0)$ possesses linear redundancy if there exist at least two affine equivalent component Boolean functions

$$f_\lambda(x) = \lambda_{n-1}f_{n-1} \oplus \lambda_{n-2}f_{n-2} \oplus \dots \oplus \lambda_0f_0$$

and

$$f_\beta(x) = \beta_{n-1}f_{n-1} \oplus \beta_{n-2}f_{n-2} \oplus \dots \oplus \beta_0f_0,$$

where $\lambda = (\lambda_{n-1}, \dots, \lambda_0)$, $\beta = (\beta_{n-1}, \dots, \beta_0) \in \mathbb{F}_2^n \setminus \{0\}$ and $\lambda \neq \beta$. The number of distinct pairs (λ, β) such that component Boolean functions f_λ, f_β are affine equivalent is called *the linear redundancy* of S-box.

Denote linear redundancy of S-box S by $\mathcal{LR}(S)$. If all component Boolean functions of S (non-zero linear combination) are affine equivalent (i.e., $\mathcal{LR}(S) = (2^n - 1)(2^n - 2)/2$), then S possesses the complete linear redundancy.

On the other hand, if there does not exist any two affine equivalent non-zero linear combinations of coordinate functions (i. e., $\mathcal{LR}(S) = 0$), then S does not possess linear redundancy. Many small S-boxes possess linear redundancy because there are few affine equivalence classes when the number of variables of Boolean function are small. However the number of affine equivalence classes of S-box increases rapidly with the size of S-box. The presence of linear redundancy in the S-boxes is an indicator of the non-randomness and it may be exploited by the new cryptanalytic methods (see [1]). In [1] a challenge into the cryptographic community was launched: *prove that there does not exist an attack based on the linear redundancy*. If such a proof does not exist then there is still a certain doubt on the security of the block cipher that uses S-box possessing linear redundancy and, consequently, its non-random properties. Therefore, the designer constructing a secure block cipher should take into account this characteristic of the S-boxes to ensure the necessary preventive security of the cipher algorithm against the attacks that may arise.

2.3. The affine equivalence classes of Boolean functions and the algorithm testing the affine equivalence

The affine equivalence classes of Boolean functions are often used when considering some important cryptographic properties of not only Boolean functions but also of the S-boxes because affine transformations preserve such properties as algebraic degree and non-linearity or, more specifically, the absolute values of the Walsh transform and the auto-correlation function. However, the problem of detecting the affine equivalence of two Boolean functions and evaluation the number of affine equivalence classes of the set of n -input Boolean functions is a difficult problem. In fact, the number of affine equivalence classes increase rapidly when the number of variables n increases (see Table 1).

Table 1. The number of affine equivalence classes of n -input Boolean functions

n	1	2	3	4	5	6	7	8
Number	1	2	3	8	48	> 69000	$> 10^{19}$	$> 10^{53}$

Therefore, in the case of large n we cannot list all the classes, but we need an affine equivalence test algorithm with a practicable complexity permitting to calculate the linear redundancy of S-box. Before [1] there was not any effective method to determine the affine equivalence of two Boolean functions. The authors of [1] suggest a test algorithm based on the preservation of some cryptographic properties under the affine transformations. This algorithm provides a general procedure that may be applied to determine the affine equivalence relationship between the two functions or to prove such equivalence does not exist. It should be noted that the algorithm complexity depends on the actual function pair.

We have implemented and ran several experiments with some actual S-boxes. We have observed that if the S-box does not possess linear redundancy, then the experiments run fast; otherwise, it will take more time to test the property and to find the corresponding matrices, vectors and constant.

3. Linear redundancy in AES-like S-boxes

It is known (see [2]) that S-box of the AES algorithm has very good properties in both algebraic degree and non-linearity. However, 255 non-trivial component Boolean functions of this S-box have the same frequency distributions of the absolute values of the Walsh transform and auto-correlation. Since these properties are preserved under affine transformations, a natural question arises: whether the component functions are affine equivalent? In [1] the authors had proved that the S-box based on the inversion mapping in finite field will possess the complete linear redundancy. In what follows we recall and extend the results on the S-boxes that are affine equivalent to this S-box. For the output Boolean functions of the inversion mapping of the finite field the following statement is valid.

Theorem 1. *The output component functions of the finite field inversion are related by linear transformations.*

Proof. See [1]. □

Here we will prove an important result related to the AES-like S-boxes, which are the S-boxes based on the affine transformations of input and output of the finite field inversion. We will show that the linear redundancy existing in finite field inversion is preserved in the S-boxes generated by this method. First, we recall the concept of affine equivalence of two S-boxes.

Definition 2. Two $n \times n$ bit S-boxes S_1 and S_2 are affine equivalent (it is denoted by $S_1 \sim S_2$), if there exist invertible linear maps $\mathcal{A} \in GL(n, \mathbb{F}_2)$, $\mathcal{B} \in GL(n, \mathbb{F}_2)$ and constants $a \in \mathbb{F}_2^n$, $b \in \mathbb{F}_2^n$ such that

$$S_2(x) = \mathcal{B}(S_1(\mathcal{A}(x) \oplus a^\top)) \oplus b^\top$$

(specifically, if $a = 0_n, b = 0_n$ then we say that S_1 and S_2 are linear equivalent, where $0_n = (0, 0, \dots, 0) \in \mathbb{F}_2^n$).

Proposition 1. *Let S_1 and S_2 be two affine equivalent S-boxes S_1, S_2 of size $n \times n$ bits. If S_1 possesses linear redundancy then S_2 also possesses linear redundancy. Moreover, if S_1 possesses the complete linear redundancy then so does S_2 .*

Proof. Since S_1 and S_2 are affine equivalent S-boxes then there exist invertible maps $\mathcal{A} \in GL(n, \mathbb{F}_2)$, $\mathcal{B} \in GL(n, \mathbb{F}_2)$ represented by $n \times n$ matrices $A = (a_{ij})_{n \times n}$ and $B = (b_{ij})_{n \times n}$ over $\mathbb{F}_2$ respectively and constants $c \in \mathbb{F}_2^n$, $d \in \mathbb{F}_2^n$ such that $S_1(x) = \mathcal{B} S_2(\mathcal{A}x \oplus c^\top) \oplus d^\top$.

Assume that two S-boxes have representations in the form of coordinate Boolean functions as $(f_{n-1}, \dots, f_0)$, $(g_{n-1}, \dots, g_0)$ respectively, where $f_i, g_i: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$. Then we have the following equality implied by the affine equivalent relationship between two S-boxes:

$$f_i(x) = \sum_{j=0}^{n-1} b_{ij} g_j(Ax^\top \oplus c^\top) \oplus d_i^\top, \quad i = 0, \dots, n-1. \quad (1)$$

Assume that S_1 possesses linear redundancy, so there exist two component functions f_λ, f_β where $\lambda = (\lambda_{n-1}, \dots, \lambda_0)$, $\beta = (\beta_{n-1}, \dots, \beta_0)$, a non-trivial linear mapping $\mathcal{D} \in GL(n, \mathbb{F}_2)$ represented by the matrix D and n -dimension vectors $u, l \in \mathbb{F}_2^n$ element $m \in \mathbb{F}_2$ such that

$$\sum_{i=0}^{n-1} \lambda_i f_i(x) = \sum_{i=0}^{n-1} \beta_i f_i(Dx^\top \oplus u^\top) \oplus lx^\top \oplus m. \quad \forall x \in \mathbb{F}_2^n. \quad (2)$$

Substituting (1) into (2), we obtain:

$$\begin{aligned} & \sum_{i=0}^{n-1} \lambda_i \left(\sum_{j=0}^{n-1} b_{ij} g_j(Ax^\top \oplus c^\top) \oplus d_i^\top \right) = \\ & = \sum_{i=0}^{n-1} \beta_i \left(\sum_{j=0}^{n-1} b_{ij} g_j(A(Dx^\top \oplus u^\top) \oplus c^\top) \oplus d_i^\top \right) \oplus lx^\top \oplus m. \end{aligned}$$

It follows that

$$\begin{aligned} & \sum_{j=0}^{n-1} \sum_{i=0}^{n-1} \lambda_i b_{ij} g_j(Ax^\top \oplus c^\top) \oplus \sum_{i=0}^{n-1} \lambda_i d_i = \\ & = \sum_{j=0}^{n-1} \sum_{i=0}^{n-1} \beta_i b_{ij} g_j(A(Dx^\top \oplus u^\top) \oplus c^\top) \oplus \sum_{i=0}^{n-1} \beta_i d_i \oplus lx^\top \oplus m. \end{aligned}$$

Let

$$\begin{aligned} \lambda_j^* &= \sum_{i=0}^{n-1} \lambda_i b_{ij}, \\ \beta_j^* &= \sum_{i=0}^{n-1} \beta_i b_{ij}, \\ m^* &= m \oplus \sum_{i=0}^{n-1} \lambda_i d_i \oplus \sum_{i=0}^{n-1} \beta_i d_i. \end{aligned}$$

We have:

$$\sum_{j=0}^{n-1} \lambda_j^* g_j (A x^\top \oplus c^\top) = \sum_{j=0}^{n-1} \beta_j^* g_j \left(A (D x^\top \oplus u^\top) \oplus c^\top \right) \oplus l x^\top \oplus m^*.$$

Next,

$$\begin{aligned} \sum_{j=0}^{n-1} \lambda_j^* g_j (A x^\top \oplus c^\top) &= \\ &= \sum_{i=0}^{n-1} \beta_j^* g_j \left(A (D x^\top \oplus u^\top) \oplus c^\top \right) \oplus l x^\top \oplus m^* = \\ &= \sum_{i=0}^{n-1} \beta_j^* g_j \left(ADA^{-1} (A x^\top \oplus c^\top) \oplus ADA^{-1} c^\top \oplus A u^\top \oplus c^\top \right) \oplus l x^\top \oplus m^*. \end{aligned}$$

Let $D^* = ADA^{-1}$, $c^* = ADA^{-1}c \oplus Au^\top \oplus c$; then we get the following equality:

$$\begin{aligned} \sum_{j=0}^{n-1} \lambda_j^* g_j (A x^\top \oplus c^\top) &= \\ &= \sum_{j=0}^{n-1} \beta_j^* g_j \left(D^* (A x^\top \oplus c^\top) \oplus c^{*\top} \right) \oplus l x^\top \oplus m^*. \end{aligned}$$

Since $A \in GL(n, \mathbb{F}_2)$ then $Ax \oplus c$ gets in turn all values in $\mathbb{F}_2^n$ if x gets in turn all values in $\mathbb{F}_2^n$. Denoting $X = A^\top x \oplus c$, we have:

$$(\forall X \in \mathbb{F}_2^n) \quad \sum_{j=0}^{n-1} \lambda_j^* g_j (X) = \sum_{j=0}^{n-1} \beta_j^* g_j (D^* X^\top \oplus c^\top) \oplus l x^\top \oplus m^*.$$

It follows that S_2 has two non-trivial affine equivalent linear components, i.e. S_2 possesses linear redundancy. It is easy to see that the transformation and role of S_1 and S_2 in each equality are equivalent because A, B, D are invertible maps in $GL(n, \mathbb{F}_2)$. Therefore, for any two components of S_2 we always obtain two corresponding Boolean component combinations of S_1 , in other words, the numbers of pairs (λ, β) and (λ^*, β^*) are equal. Thus, $\mathcal{LR}(S_1) = \mathcal{LR}(S_2)$. Hence, if S_1 possesses the complete linear redundancy, then so does S_2 . $\square$

Applying this result to 8 bit S-boxes based on the finite field inversion (AES-like S-boxes), we get the following corollary.

Corollary 1. *All AES-like 8 bit S-boxes possess the complete linear redundancy.*

4. The affine equivalence of XSL-like round functions

In this section by means of results on linear redundancy of S-boxes used in XSL-like round functions we propose some analysis of the affine equivalence of Boolean functions that are linear combinations of output coordinate functions of XSL-like functions. The idea is similar to the method used in [6].

4.1. XSL-like round functions

Here we consider XSL-like round functions in a general form with m bit input and m bit output, in which the non-linear layer S uses k S-boxes of size n bit (so, $m = k \times n$) and the linear layer L is represented by a $k \times k$ matrix over $\mathbb{F}_2$. Since XOR operation with key do not affect affine equivalence of output Boolean function then we consider further only the nonlinear layer S and the linear layer L . Then a round function has the following form (Fig. 1):

$$f : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^m, (x_{m-1}, \dots, x_0) \mapsto (f_{m-1}, \dots, f_0).$$

Fig. 1. The XSL-like function

Let's denote

$$X_0 = x_{n-1} x_{n-2} \dots x_0 \in \mathbb{F}_{2^n},$$

$$X_i = x_{n-1+n \cdot i} x_{n-2+n \cdot i} \dots x_{0+n \cdot i} \in \mathbb{F}_{2^n}, \quad 1 \leq i \leq k-1.$$

Then the output of the round function may be represented as

$$L(S(X_{k-1}), \dots, S(X_0)),$$

where $S: \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}$ is S-box, $L: \mathbb{F}_{2^n}^k \rightarrow \mathbb{F}_{2^n}^k$ is a linear transformation defined by the equality

$$\begin{aligned} L(X_{k-1}, \dots, X_0) &= \\ &= (L_{k-1}(X_{k-1}, \dots, X_0), \dots, L_0(X_{k-1}, \dots, X_0)) = \\ &= (X_{k-1}, \dots, X_0) \begin{pmatrix} h_{k-1,k-1} & \cdots & h_{0,k-1} \\ \vdots & \ddots & \vdots \\ h_{k-1,0} & \cdots & h_{0,0} \end{pmatrix} = \\ &= (h_{k-1,k-1} \times X_{k-1} \oplus \dots \oplus h_{k-1,0} \times X_0, \dots, h_{0,k-1} \times X_{k-1} \oplus \dots \oplus h_{0,0} \times X_0); \end{aligned}$$

here L_i is defined as

$$L_i: \mathbb{F}_{2^n}^k \rightarrow \mathbb{F}_{2^n}, (X_{k-1}, \dots, X_0) \mapsto h_{i,k-1} \times X_{k-1} \oplus \dots \oplus h_{i,0} \times X_0.$$

Then, the n bit output of the i^{th} block ($0 \leq i \leq k-1$) may be defined as follows

$$(f_{n-1+n \cdot i}(x_{m-1}, \dots, x_0) \cdots f_{n \cdot i}(x_{m-1}, \dots, x_0)) = L_i(X_{k-1}, \dots, X_0).$$

Now, the coordinate Boolean functions of n bit output of the i^{th} block may be represented via the trace function:

$$f_{j+n \cdot i}(x_{m-1}, \dots, x_0) = \text{Tr}[\beta_j \times (L_i(X_{k-1}, \dots, X_0))], \quad 0 \leq j \leq n-1.$$

We consider the t^{th} output Boolean function of the round function, which belongs to the block number $\lfloor \frac{t}{n} \rfloor$ (having n -bit length) and number $t - \lfloor \frac{t}{n} \rfloor \cdot n$ in this n -bit block. We have:

$$\begin{aligned} f_t(x_{m-1}, \dots, x_0) &= \\ &= \text{Tr}[\beta_t \times h_{\lfloor \frac{t}{n} \rfloor, k-1} \times S(X_{k-1})] \oplus \dots \oplus \text{Tr}[\beta_t \times h_{\lfloor \frac{t}{n} \rfloor, 0} \times S(X_0)] \end{aligned}$$

for $0 \leq t \leq m-1$, $\beta_t \in \mathbb{F}_{2^n}$.

4.2. The case when S-box possesses complete linear redundancy

Here we extend the result of [6] to XSL-like round functions with S-boxes possessing complete linear redundancy.

Proposition 2. *All output Boolean functions of XSL-like round function with S-boxes possessing complete linear redundancy belong to the same affine equivalence class. Moreover, their linear combinations also belong to this affine equivalence class.*

Proof. Consider two output Boolean functions of XSL-like round function $f_t, f_{t'}$ for arbitrary $t, t' \in \{0, 1, \dots, m-1\}, t \neq t'$. We have

$$\begin{aligned} f_t(x_{m-1}, \dots, x_0) &= \\ &= \text{Tr}\left[\beta_t \times h_{\lfloor \frac{t}{n} \rfloor, k-1} \times S(X_{k-1})\right] \oplus \dots \oplus \text{Tr}\left[\beta_t \times h_{\lfloor \frac{t}{n} \rfloor, 0} \times S(X_0)\right] = \\ &= \text{Tr}[\tilde{\beta}_{t_{k-1}} \times S(X_{k-1})] \oplus \dots \oplus \text{Tr}[\tilde{\beta}_{t_0} \times S(X_0)] = \\ &= S_{\tilde{t}_{k-1}}(X_{k-1}) \oplus \dots \oplus S_{\tilde{t}_0}(X_0), \end{aligned} \quad (3)$$

$$\begin{aligned} f_{t'}(x_{m-1}, \dots, x_0) &= \\ &= \text{Tr}\left[\beta_{t'} \times h_{\lfloor \frac{t'}{n} \rfloor, k-1} \times S(X_{k-1})\right] \oplus \dots \oplus \text{Tr}\left[\beta_{t'} \times h_{\lfloor \frac{t'}{n} \rfloor, 0} \times S(X_0)\right] = \\ &= \text{Tr}[\tilde{\beta}'_{t'_{k-1}} \times S(X_{k-1})] \oplus \dots \oplus \text{Tr}[\tilde{\beta}'_{t'_0} \times S(X_0)] = \\ &= S_{\tilde{t}'_{k-1}}(X_{k-1}) \oplus \dots \oplus S_{\tilde{t}'_0}(X_0). \end{aligned} \quad (4)$$

Here $\tilde{\beta}_i, \tilde{\beta}'_i \in \mathbb{F}_{2^n}, \tilde{t}_i, \tilde{t}'_i \in \{0, \dots, 2^n-1\}$ for $i \in \{0, \dots, 2^n-1\}$, and $S_i: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ ($i \in \{0, \dots, 2^n-1\}$) are component Boolean functions of S . Since S possesses complete linear redundancy, then S_λ is linear equivalent to $S_{\lambda'}$ for arbitrary $\lambda, \lambda' \in \{1, \dots, 2^n-1\}, \lambda \neq \lambda'$. Therefore, there exists non-singular linear map $\mathcal{D}_i \in \text{GF}(n, \mathbb{F}_2)$ and $a_i, b_i \in \mathbb{F}_2^n, c_i \in \mathbb{F}_2$ such that

$$\begin{aligned} S_{\tilde{t}'_i}(x_{n-1+n \cdot i}, \dots, x_{n \cdot i}) &= \\ &= S_{\tilde{t}_i}[D_i(x_{n-1+n \cdot i}, \dots, x_{n \cdot i})^\top \oplus a_i^\top] \oplus b_i(x_{n-1+n \cdot i}, \dots, x_{n \cdot i})^\top \oplus c_i \end{aligned}$$

for all $(x_{n-1+n \cdot i}, \dots, x_{n \cdot i}) \in \mathbb{F}_2^n, i = 0, \dots, k-1$. Thus, let D be a $m \times m$ matrix over $\mathbb{F}_2$ corresponding to non-singular linear map $\mathcal{D}$ in $\text{GF}(m, \mathbb{F}_2)$, let $a, b \in \mathbb{F}_2^m$ and $c \in \mathbb{F}_2$. We have:

$$D = \begin{bmatrix} D_{k-1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & D_0 \end{bmatrix},$$

$$a = (a_{k-1}, \dots, a_0), b = (b_{k-1}, \dots, b_0), c = \bigoplus_{i=0}^{k-1} c_i.$$

Then

$$\begin{aligned}
 & f_t [D(x_{m-1}, \dots, x_0)^\top \oplus a^\top] \oplus b x^\top \oplus c = \\
 & = \bigoplus_{i=0}^{k-1} (S_{\tilde{t}_i} [D_i(x_{n-1+k \cdot i}, \dots, x_{k \cdot i})^\top \oplus a_i^\top] \oplus b_i(x_{n-1+k \cdot i}, \dots, x_{k \cdot i})^\top \oplus c_i) = \\
 & = \bigoplus_{i=0}^{k-1} (S_{\tilde{t}'_j}(x_{n-1+k \cdot j}, \dots, x_{k \cdot j})) = f_{t'}(x_{m-1}, \dots, x_0)
 \end{aligned}$$

for all $(x_{m-1}, \dots, x_0) \in \mathbb{F}_2^m$. Therefore, functions f_t and $f_{t'}$ belong to the same affine equivalence class. Consequently, all non-zero linear combinations of m output functions $(f_{m-1}, \dots, f_0)$ belong to the same affine equivalence class. Combinations of output functions of the round function have the form (4) for $\tilde{\lambda} = (\tilde{\lambda}_{k-1}, \dots, \tilde{\lambda}_0) \in \mathbb{F}_2^n$. Since S possesses complete linear redundancy, then we can easily prove by similar arguments that these combinations belong to the same affine equivalence class.

4.3. The case when S-box possesses incomplete linear redundancy

Here we consider S-box which possesses some affine equivalent component pairs of Boolean functions. Let

$$\mathcal{C}_S = \{(i, j) \in \mathbb{Z}^2 \mid 0 \leq i < j \leq 2^n - 1, S_i \sim S_j\}$$

and

$$\mathcal{C}_{S_i} = \{j \in \mathbb{Z} \mid 0 \leq j \leq 2^n - 1, i \neq j, S_i \sim S_j\}.$$

In this case we have

$$|\mathcal{C}_S| < \frac{(2^n - 1)(2^n - 2)}{2}, \quad |\mathcal{C}_{S_i}| < 2^n - 2 \quad \forall i \in \{0, \dots, 2^n - 1\}.$$

Let $f_t, f_{t'}$ ($t, t' \in \{0, 1, \dots, m-1\}, t \neq t'$) be two coordinate Boolean functions of a round function and their representations have the forms (3), (4). It is clear that if for each $i \in \{0, \dots, k-1\}$ two component functions $S_{\tilde{t}'_i}, S_{\tilde{t}_i}$ of S are affine equivalent, then $f_t, f_{t'}$ are also affine equivalent. However, it is important to note that $\tilde{t}, \tilde{t}'$ depend on the relation between given distinct coefficients $\beta_t, \beta_{t'}$ and $h_{\lfloor \frac{t}{n} \rfloor, i}, h_{\lfloor \frac{t'}{n} \rfloor, i}$ respectively, i.e. we need to consider each bit block independently since the elements of the matrix L are different. On the other hand, if $f_t, f_{t'}$ are affine equivalent then there exist the matrices $(D_{k-1}, \dots, D_0)$ from the representation of D and $a_i, b_i \in \mathbb{F}_2^n, c_i \in \mathbb{F}_2$ that realize the affine equivalence of component Boolean functions with respect to each different bit block. Therefore, if such pair does not exist, we can conclude that two considered coordinate functions are not affine equivalent.

Consequently, we can test affine equivalence of coordinate functions in the round functions instead of two round functions directly (in fact, it is impossible since the size of the round functions is very huge), i.e. we just need to consider the affine equivalence of component Boolean functions of n -bit S-box.

Let *checkaffine* (f, g) be the procedure for testing affine equivalence of any two Boolean functions from [1], then the pseudo-code of the test algorithm is as follows.

Algorithm 1. TESTING THE AFFINE EQUIVALENCE OF TWO COORDINATE FUNCTIONS OF XSL-LIKE ROUND FUNCTION

Inputs: INDEXES t, t' OF TWO COORDINATE FUNCTIONS FOR $t, t' = 0, \dots, m-1$.

Output: IF TWO FUNCTIONS ARE EQUIVALENT THEN RETURN 1 AND OUTPUT $D = \{D_{k-1}, \dots, D_0\}$, $a = \{a_{k-1}, \dots, a_0\}$, $b = \{b_{k-1}, \dots, b_0\}$, c ; OTHERWISE RETURN 0.

```

1.  $D \leftarrow \emptyset$ ,  $a \leftarrow \emptyset$ ,  $b \leftarrow \emptyset$ ,  $c = 0$ 
2. FOR  $i = 0$  TO  $k-1$ 
    IF (checkaffine ( $\text{Tr} [\beta_t \times h_{\lfloor \frac{t}{n} \rfloor, i} \times S(X)]$ ,  $\text{Tr} [\beta_{t'} \times h_{\lfloor \frac{t'}{n} \rfloor, i} \times S(X)]$ )
         $D_i \leftarrow D_{\text{temp}}$ ,
         $a_i \leftarrow a_{\text{temp}}$ ,
         $b_i \leftarrow b_{\text{temp}}$ ,
         $c_i \leftarrow c \oplus c_{\text{temp}}$ ;
        ( $D_{\text{temp}}, a_{\text{temp}}, b_{\text{temp}}, c_{\text{temp}}$  OUTPUTS OF THE ALGORITHM 1 IN [1])
    WHEN (checkaffine ( $\text{Tr} [\beta_t \times h_{\lfloor \frac{t}{n} \rfloor, i} \times S(X)]$ ,  $\text{Tr} [\beta_{t'} \times h_{\lfloor \frac{t'}{n} \rfloor, i} \times S(X)]$ ))
    ELSE RETURN 0
3. RETURN 1

```

The complexity of this algorithm is at most k times the complexity of the algorithm 1 from [1]. If two considered coordinate functions belong to the same bit block $\lfloor \frac{t}{n} \rfloor$ (i.e., they have the same coefficient $h_{\lfloor \frac{t}{n} \rfloor, i}$) of k bit blocks of the round function, then testing may be reduced as follows:

$$\begin{aligned}
 f_t(x_{m-1}, \dots, x_0) &= \\
 &= \text{Tr} \left[\beta_t \times h_{\lfloor \frac{t}{n} \rfloor, k-1} \times S(X_{k-1}) \right] \oplus \dots \oplus \text{Tr} \left[\beta_t \times h_{\lfloor \frac{t}{n} \rfloor, 0} \times S(X_0) \right].
 \end{aligned}$$

Let $S^\lambda(X) = \lambda \times S(X)$ ($X, \lambda \in \mathbb{F}_{2^n}$) for $S: \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}$.

Then we have $f_t(x_{m-1}, \dots, x_0) = \text{Tr}[\beta_t \times S^{h_{\lfloor \frac{t}{n} \rfloor, k-1}}(X_{k-1})] \oplus \dots \oplus \text{Tr}[\beta_t \times S^{h_{\lfloor \frac{t}{n} \rfloor, 0}}(X_0)]$ and $f_{t'}(x_{m-1}, \dots, x_0) = \text{Tr}[\beta_{t'} \times S^{h_{\lfloor \frac{t}{n} \rfloor, k-1}}(X_{k-1})] \oplus \dots \oplus \text{Tr}[\beta_{t'} \times S^{h_{\lfloor \frac{t}{n} \rfloor, 0}}(X_0)]$.

Thus, if two considered Boolean functions belong to the same n bit block then the indexes of coefficients h in the corresponding matrix representations are equal. However, since $\beta_t, \beta_{t'}$ in the equalities are equal, then in each i^{th} input block there exists a corresponding affine equivalent pair. Therefore, the condition $f_t \sim f_{t'}$ that depend on testing affine equivalence is:

$$\bigcap_{i=1}^{k-1} \mathcal{C}_{S_{h_{\lfloor \frac{t}{n} \rfloor, i}}} \neq \emptyset. \quad (5)$$

Indeed, assume that $\bigcap_{i=1}^{k-1} \mathcal{C}_{S_{h_{\lfloor \frac{t}{n} \rfloor, i}}} \neq \emptyset$, then there exists a pair such that $(u, v) \in \mathcal{C}_{S_{h_{\lfloor \frac{t}{n} \rfloor, i}}}$ for all i . It follows that $S_u^{h_{\lfloor \frac{t}{n} \rfloor, i}} \sim S_v^{h_{\lfloor \frac{t}{n} \rfloor, i}}$, or $\text{Tr}[u \times S^{h_{\lfloor \frac{t}{n} \rfloor, i}}] \sim \text{Tr}[v \times S^{h_{\lfloor \frac{t}{n} \rfloor, i}}]$ for all i . Hence, there exist $\beta_t, \beta_{t'}$ corresponding to (u, v) such that $(f_t, f_{t'})$ are affine equivalent.

4.4. Some experimental results

In this section we describe some experiments performed to analyze the affine equivalence in the round function of Kuznyechik block cipher described in [8] and published as GOST 34.12-2015 standard (see [9]). Kuznyechik algorithm uses XSL construction with parameters $m = 128, n = 8, k = 16$. We analyze the affine equivalence of coordinate functions in Kuznyechik block cipher.

The right multiply operation of a 16-byte row vector by the matrix L may be represented as the product of a 128 bit vector (16 bytes of the row vector) and a 128×128 bit binary matrix M (corresponding to the matrix L). The i^{th} row of the matrix M is represented in binary form as 16 bytes, each byte being the result of multiplying each byte in the row $\lfloor \frac{i}{8} \rfloor$ of matrix L by $2^{7-(i \bmod 8)}$ in $\mathbb{F}_{2^8}$. Let $(f_{127}^*, f_{126}^*, \dots, f_0^*)$ be 128 bit row vector, where each f_i^* is an output Boolean function of the S-box layer (Fig. 2).

Each Boolean function f_i is a XOR addition of functions $f_{127}^*, f_{126}^*, \dots, f_0^*$ multiplied by elements of the i^{th} row of matrix M , i. e.

$$f_i = \underbrace{\bigoplus_{k=120}^{127} m_{i,k} \cdot f_k^*}_{A_{15}} \oplus \underbrace{\bigoplus_{k=112}^{119} m_{i,k} \cdot f_k^*}_{A_{14}} \oplus \dots \oplus \underbrace{\bigoplus_{k=0}^7 m_{i,k} \cdot f_k^*}_{A_0},$$

where $i = 127, 126, \dots, 0$.

Fig. 2. The bit-form of transformations S and L in Kuznyechik

Each $A_l, l = 15, \dots, 0$, is a Boolean function equal to a linear combination of the corresponding output component functions of S-boxes S_l ; in fact, it is

$$\text{Tr} \left[\beta_l \times h_{\lfloor \frac{l}{n} \rfloor, l} \times S(X_l) \right].$$

Different Boolean functions $A_l, l = 15, \dots, 0$, have different sets of input variables. Therefore, in order for (f_i, f_j) to be equivalent the component Boolean functions $(A_{l,i}, A_{l,j})$ should be equivalent for all $l = 15, \dots, 0$ as was stated above.

To perform this analysis we construct for each given 8×8 bit S-box corresponding tuple consisting of 128 Boolean functions with 8-bit input. Depending on the elements of the 128×128 bit binary matrix the first tuple is linear combination of 8 component Boolean functions with respect to the 15th S-box, the second tuple is a linear combination of 8 component Boolean functions with respect to the 14th S-box, etc. In order to test the affine equivalence of output Boolean functions we search for a pair of equivalent Boolean functions in the set of 128 Boolean functions from the first tuple. The number of pairs that have to be tested equals $(128 \times 127) / 2$. For each equivalent pair (i, j) found in the first tuple we continue testing this pair in the next tuples. Therefore, we don't have to test all pairs of the next tuples but just test only the necessary pairs.

We have carried out experiments with various options as follows.

1. Check the affine equivalence of output functions of the Kuznyechik's round function. We've been running test for the entire 128 coordinate functions of the round function though Kuznyechik's S-box possesses linear redundancy ($\mathcal{LR}(S) = 19$), but there does not exist any pair of affinely equivalent functions. This occurs not only because S-box has very few affinely equivalent pairs but also because the linear layer has many different elements in the matrix representation. So it is very difficult to satisfy the affine equivalent condition for next tuples.

2. Check the affine equivalence of output functions of the round function that includes an nonlinear layer with 16 identical Kuznyechik's S-boxes and the AES's linear layer (including a Shiftrow and a MixColumns transformation). The experimental results indicate that there are no pairs affinely equivalent output Boolean functions of the above mentioned round functions.
3. Check the affine equivalence of output functions of the round function that include an S-box layer consisting of 16 identical S-boxes that possess large linear redundancy ($\mathcal{LR}(S) = 8449$, S-box table is given in Appendix 1) and the AES's linear transformations. The experimental results show that there are 28 output function pairs of the above round function which are affine equivalent to each other.

For example, the Boolean function pair (32, 33) is affinely equivalent. In this case the matrix D is of the form

$$D = \begin{pmatrix} D^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_{16} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{16} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & D^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{16} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & D^3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{16} \end{pmatrix},$$

where 0 is the 16×16 zero matrix, I_{16} is the 16×16 unit matrix and D^1, D^2, D^3, D^4 are the following 16×16 matrices:

$$D^1 = \begin{pmatrix} M^1 & 0 \\ 0 & I_8 \end{pmatrix}, \quad D^2 = \begin{pmatrix} I_8 & 0 \\ 0 & M^2 \end{pmatrix},$$

$$D^3 = \begin{pmatrix} M^3 & 0 \\ 0 & I_8 \end{pmatrix}, \quad D^4 = \begin{pmatrix} I_8 & 0 \\ 0 & M^3 \end{pmatrix},$$

where M^1, M^2, M^3 are 8×8 binary matrices defined by the following arrays:

$$M^1 = \{\text{fc}, 49, \text{fd}, 3\text{d}, 31, 77, 50, 05\},$$

$$M^2 = \{\text{a8}, 18, 54, 0\text{c}, 2\text{a}, 06, 15, 03\},$$

$$M^3 = \{\text{d1}, \text{dd}, 7\text{a}, 50, \text{e5}, \text{e3}, 3\text{d}, 28\}.$$

4. Check the affine equivalence of output functions of the round function including the S-box layer which has 16 identical S-box that possess a large linear redundancy and the Kuznyechik's linear transformation layer. We have conducted check on 5000 S-boxes possessing large linear redundancy like the S-box in Appendix 1 and could not find any pair of affinely equivalent Boolean functions of the linear layer.

Conclusion

In this paper, we show the influence of linear redundancy of S-boxes on the affine equivalence of output coordinate functions in XSL-like round function. Using S-boxes possessing complete linear redundancy implies that both output coordinate Boolean functions of round function and their linear combinations belong to the same affine equivalence class. In case that the used S-box possesses high linear redundancy, we can point out the existence and determine affinely equivalent output functions. Hence, to construct a secure cipher the designers should consider the linear redundancy of S-box as a criterion for generating S-boxes with good cryptographic properties.

Список литературы

- [1] Fuller J., Millan W., "Linear redundancy in S-boxes". In: *'Fast Software Encryption'*, Lect. Notes Comput. Sci., **6061**, 2010, 73–80
- [2] Daemen J., Rijmen V., *The Design of Rijndael: AES — The Advanced Encryption Standard*, Heidelberg etc.: Springer, 2002, xvii+238 ppp.
- [3] Aoki K., Ichikawa T., Kanda M., Matsui M., Moriai S., Nakajima J., Tokita T., "Camellia: A 128-bit block cipher suitable for multiple platforms—design and analysis". In: *"Selected Areas in Cryptography"*, Lect. Notes Comput. Sci., **2012**, 2001, 39–56.
- [4] Rijmen V., Daemen J., Preneel B., Bosselaers A., De Win E., "The cipher SHARK". In: *"Fast Software Encryption"*, Lect. Notes Comput. Sci., **1039**, 1996, 99–111.
- [5] Ivanov G., Nikolov N., Nikova S., "Reversed genetic algorithms for generation of bijective S-boxes with good cryptographic properties", Cryptology ePrint Archive, Report 2014/801 (2014), <http://eprint/iacr.org/2014/801.pdf>.
- [6] Youssef A. M., Tavares S. E., "Affine equivalence in the AES round function", *Discrete Appl. Math.*, **148**:2 (2005), 161–170.
- [7] Lidl R., Niederreiter H., *Finite Fields. 2nd ed.*, Cambridge: Cambridge Univ. Press, 1997, xiv+755 ppp.
- [8] Dygin D. M., Lavrikov I. V., Marshalko G. B., Rudskoy V. I., Trifonov D. I., Shishkin V. A., "On a new Russian Encryption Standard", *Mathematical Aspects of Cryptography*, **6**:2 (2015), 29–34.
- [9] *Information technology. Cryptographic Data Security. Block ciphers, GOST R 34.12-2015*, Federal Agency on Technical Regulating and Metrology, Moscow: Standardinform, 2015. (In Russian.)

Appendix. S-box possessing a large linear redundancy ($\mathcal{LR}(S) = 8449$)

```

S_box[] = {
7c, 63, 77, 7b, f2, 6b, 6f, c5, 30, 01, 67, 2b, fe, d7, ab, 76,
ca, 82, c9, 7d, fa, 59, 47, f0, ad, d4, a2, af, 9c, a4, 72, c0,
b7, fd, 93, 26, 36, 3f, f7, cc, 34, a5, e5, f1, 71, d8, 31, 15,
04, c7, 23, c3, 18, 96, 05, 9a, 07, 12, 80, e2, eb, 27, b2, 75,
09, 83, 2c, 1a, 1b, 6e, 5a, a0, 52, 3b, d6, b3, 29, e3, 2f, 84,
53, d1, 00, ed, 20, fc, b1, 5b, 6a, cb, be, 39, 4a, 4c, 58, cf,
d0, ef, aa, fb, 43, 4d, 33, 85, 45, f9, 02, 7f, 50, 3c, 9f, a8,
51, a3, 40, 8f, 92, 9d, 38, f5, bc, b6, da, 21, 10, ff, f3, d2,
cd, 0c, 13, ec, 5f, 97, 44, 17, c4, a7, 7e, 3d, 64, 5d, 19, 73,
60, 81, 4f, dc, 22, 2a, 90, 88, 46, ee, b8, 14, de, 5e, 0b, db,
e0, 32, 3a, 0a, 49, 06, 24, 5c, c2, d3, ac, 62, 91, 95, e4, 79,
e7, c8, 37, 6d, 8d, d5, 4e, a9, 6c, 56, f4, ea, 65, 7a, ae, 08,
ba, 78, 25, 2e, 1c, a6, b4, c6, e8, dd, 74, 1f, 4b, bd, 8b, 8a,
70, 3e, b5, 66, 48, 03, f6, 0e, 61, 35, 57, b9, 86, c1, 1d, 9e,
e1, f8, 98, 11, 69, d9, 8e, 94, 9b, 1e, 87, e9, ce, 55, 28, df,
8c, a1, 89, 0d, bf, e6, 42, 68, 41, 99, 2d, 0f, b0, 54, bb, 16
};

```