Mathematical Problems of Computer Science 45, 111—121, 2016.

Linear Cryptanalysis of Modified SAFER + Algorithm

Melsik K Kyureghyan, Knarik M. Kyuregyan

Institute for Informatics and Automation Problems of NAS RA
e-mail: melsik@ipia.sci.am, knarikyuregyan@gmail.com

Abstract

In paper [1] a modified SAFER+ [2] algorithm was presented. It was shown that those modifications made it possible to speed up a modified algorithm implementation about 1,7 times and obtain the same security level as compared with SAFER+ in terms of differential analysis. In this paper a linear cryptanalysis of modified SAFER+ algorithm is presented and it is shown that it has the same resistance against linear cryptanalysis as compared with an original SAFER+.

Keywords: Block cipher**,** Linear cryptanalysis, Balanced function.

1. Introduction

Linear cryptanalysis is one of the powerful cryptanalyses against iterated block ciphers. It was introduced by Matsui as a theoretical attack on DES and it is in fact a known-plaintext attack: that is a cryptanalyst has an access to the set of some plaintexts and their corresponding ciphertexts. Matsui exploits a cipher’s weakness that he quantifies in terms of “unbalanced linear expression” that involves plaintext bits, ciphertext bits (actually bits from the second last round output) and subkey bits. A linear expression is unbalanced if the equation is satisfied with the probability different from $1/2$ when the plaintext and the keys are uniformly random and independent. The cryptanalyst applies the last round attack using this linear expression for the entire cipher excluding the last round. He or she guesses wrong keys and by processing a large number of plaintext/ciphertext pairs and tries to find the last round key. The success of linear cryptanalysis depends on the original linear expression being more imbalanced than the linear expression obtained with wrong keys. In this paper we will consider a linear cryptanalysis of modified SAFER+ [1], whose round function is depicted in Fig. 1.

Let $\mathrm{X} = \mathrm{X1X2} \ldots \mathrm{X16}$ be the 16 bytes input of the i-th round. The round function consist of the following four layers:

1. XOR/ADD, first round key is “added” to the round input either modulo 2 (XOR) or modulo 256 (ADD): $\mathrm{U} = \mathrm{XOR/ADD}(\mathrm{X}, \mathrm{K}_{2\mathrm{i}-1})$.
2. Non-Linear (NL), where each byte is subjected to either the non-linear function EXP: $\mathrm{x} \rightarrow 45^{\mathrm{x}}\mathrm{modulo}257$ (except that 45^{128} is taken as 0 rather than $-1 = 257$) or its inverse function LOG: $\mathrm{V} = \mathrm{NL}(\mathrm{U})$.
3. ADD/XOR, by this layer the second round key is inserted: $\mathrm{W} = \mathrm{ADD/XOR}(\mathrm{V}, \mathrm{K}_{2\mathrm{i}})$.

Invertible Linear Transform or Pseudo-Hadamard Transform (PHT) consisting of four times applied *Armenian shuffle* and four times applied eight 2-PHT boxes: Y = PHT(W), is equivalent to

$$\begin{pmatrix} Y1 \\ Y2 \\ Y3 \\ Y4 \\ Y5 \\ Y6 \\ Y7 \\ Y8 \\ Y9 \\ Y10 \\ Y11 \\ Y12 \\ Y13 \\ Y14 \\ Y15 \\ Y16 \end{pmatrix} = \begin{pmatrix} 16 & 1 & 4 & 2 & 2 & 2 & 4 & 1 & 2 & 1 & 4 & 4 & 1 & 2 & 1 & 8 \\ 8 & 1 & 2 & 2 & 2 & 1 & 2 & 1 & 1 & 1 & 4 & 2 & 1 & 1 & 1 & 4 \\ 1 & 2 & 1 & 2 & 4 & 1 & 4 & 4 & 16 & 1 & 4 & 2 & 2 & 8 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 4 & 2 & 8 & 1 & 2 & 2 & 2 & 4 & 1 & 1 \\ 4 & 2 & 2 & 2 & 16 & 1 & 1 & 8 & 1 & 4 & 2 & 1 & 4 & 1 & 4 & 2 \\ 4 & 1 & 2 & 1 & 8 & 1 & 1 & 4 & 1 & 2 & 1 & 1 & 2 & 1 & 2 & 2 \\ 4 & 1 & 2 & 1 & 1 & 2 & 16 & 1 & 4 & 2 & 2 & 8 & 4 & 2 & 1 & 4 \\ 2 & 1 & 1 & 1 & 1 & 1 & 8 & 1 & 4 & 1 & 2 & 4 & 2 & 2 & 1 & 2 \\ 2 & 4 & 4 & 8 & 4 & 2 & 1 & 2 & 1 & 2 & 16 & 1 & 2 & 1 & 4 & 1 \\ 2 & 2 & 4 & 4 & 2 & 2 & 1 & 1 & 1 & 1 & 8 & 1 & 1 & 1 & 2 & 1 \\ 4 & 1 & 16 & 1 & 4 & 8 & 2 & 2 & 4 & 1 & 1 & 2 & 1 & 4 & 2 & 2 \\ 2 & 1 & 8 & 1 & 4 & 4 & 1 & 2 & 2 & 1 & 1 & 1 & 1 & 2 & 2 & 1 \\ 2 & 2 & 4 & 1 & 1 & 4 & 4 & 1 & 2 & 8 & 1 & 2 & 16 & 1 & 4 & 2 \\ 1 & 2 & 2 & 1 & 1 & 2 & 2 & 1 & 2 & 4 & 1 & 1 & 8 & 1 & 4 & 1 \\ 1 & 8 & 1 & 4 & 2 & 1 & 2 & 2 & 4 & 2 & 4 & 1 & 4 & 2 & 16 & 1 \\ 1 & 4 & 1 & 2 & 1 & 1 & 2 & 1 & 2 & 2 & 2 & 1 & 4 & 1 & 8 & 1 \end{pmatrix} \cdot \begin{pmatrix} W1 \\ W2 \\ W3 \\ W4 \\ W5 \\ W6 \\ W7 \\ W8 \\ W9 \\ W10 \\ W11 \\ W12 \\ W13 \\ W14 \\ W15 \\ W16 \end{pmatrix} (\mathrm{mod}\ 256).$$

2. Preliminaries

In this paper we follow the terminology and notation in [3].

A binary-valued function is said to be *balanced* if it takes on the value 0 for exactly half of its possible arguments and the value 1, otherwise. An *I/O sum* for the i^{th} round, denoted by $S^{(i)}$, is a modulo-two sum of balanced binary-valued function f_i of the round input $X^{(i)} = Y^{(i-1)}$ and a balanced binary-valued function g_i of the round output $Y^{(i)}$, that is

$$S^{(i)} := f_i\left(Y^{(i-1)}\right) \oplus g_i\left(Y^{(i)}\right),$$

where $\oplus$ denotes modulo two addition, i.e., the XOR operation.

The functions f_i and g_i are called the *input function* and the *output function*, respectively, of the I/O sum $S^{(i)}$. I/O sums for successive rounds are said to be *linked* if the output function of each of these I/O sums, except the last, coincides with the input function of the following I/O sum (i.e., $g_i = f_{i+1}$). When $S^{(1)}, S^{(2)}, \ldots, S^{(\rho)}$are linked, then their modulo-two sum is also I/O sum, namely

$$S^{(1\ldots\rho)} := \bigoplus_{i=1}^{\rho} S^{(i)} = f_1\left(Y^{(1)}\right) \oplus g_\rho\left(Y^{(\rho)}\right),$$

where $Y^{(1)} = X^{(1)} = X$ is the input to the first round, and is called a *ρ-round I/O sum.*

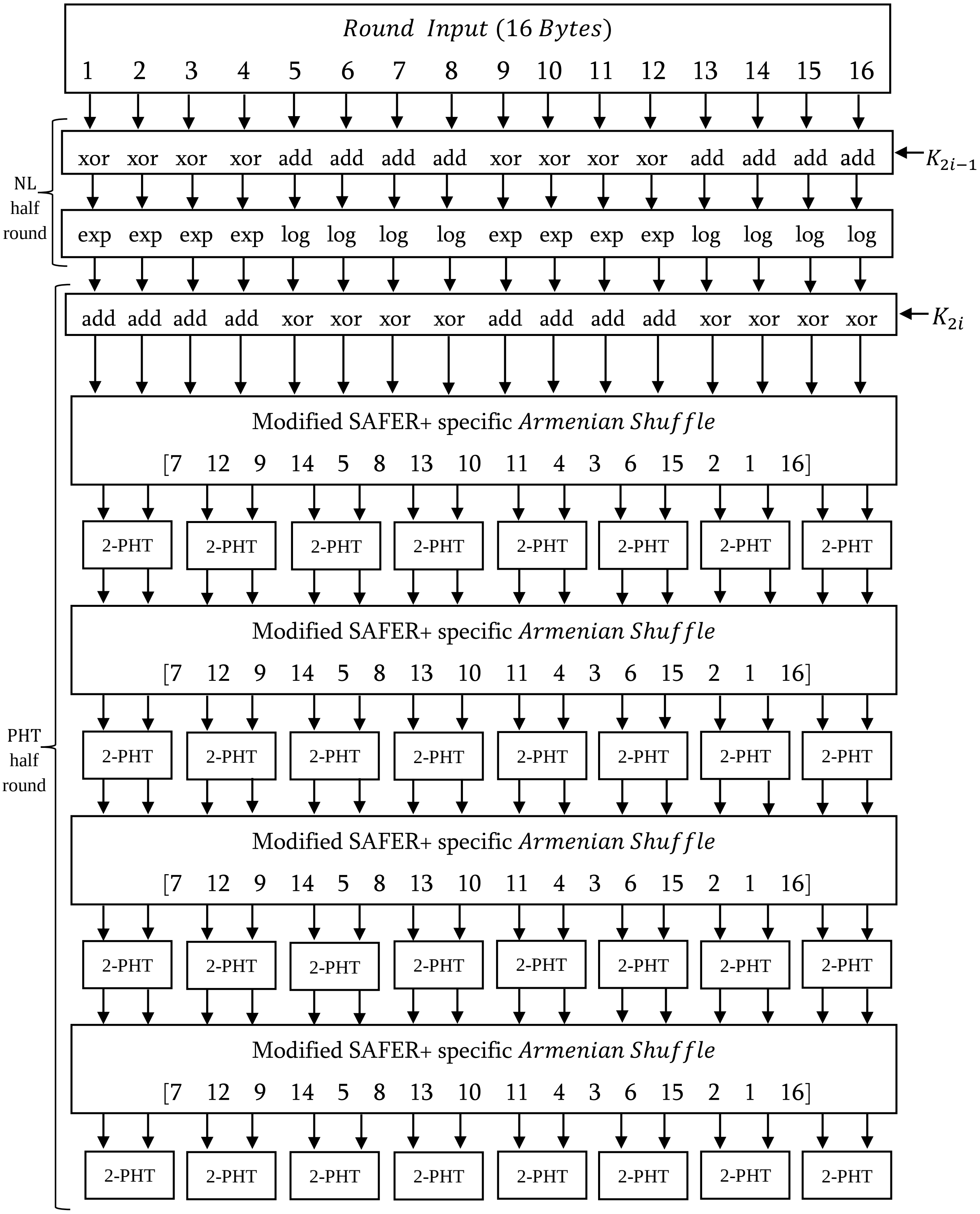
$Round\ \ Input$ (16 $Bytes$)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
xor xor xor xor add add add add xor xor xor xor add add add add
K_{2i-1}
exp exp exp exp log log log log exp exp exp exp log log log log
NL half round
add add add add xor xor xor xor add add add add xor xor xor xor
K_{2i}
Modified SAFER+ specific $Armenian\ Shuffle$
[7 12 9 14 5 8 13 10 11 4 3 6 15 2 1 16]
2-PHT
PHT half round

Fig. 1. Design of the i-th round of modified SAFER+.

Linear cryptanalysis depends critically on the notation of "$imbalance$" for binary-valued random variables. The $imbalance$ $I(V)$ of a binary-valued random variable V is the real number $|2P[V = 0] - 1|$, where $P[V = 0]$ is the probability that V takes on the value 0. Note that

$$0 \leq I(V) \leq 1$$

with equality on the left if and only if V is constant random variable and with equality on the right if and only if $2P[V=0]=1/2$.

In linear cryptanalysis we always assume that the plaintext and all round keys are independent and uniformly random over the appropriate sets. In practice, however, the full key is usually produced from a user selected secret key by key scheduling algorithm. Sometimes we explicitly fix keys by specifying e. g., $K^{(1\dots\rho)}=k^{(1\dots\rho)}$, which implies that the statistics of the random experiment are the conditional statistics given that $K^{(1\dots\rho)}=k^{(1\dots\rho)}$.

A round iterated block cipher of block-size n encrypts a plaintext X by ρ successive application of a keyed round function with different key in each round. The full key will be denoted by $K^{(1\dots\rho)}=\left(K^{(1)},K^{(2)},\dots,K^{(\rho)}\right)$, where $K^{(i)}$ is the round key applied in the i-th round for $i=1,2,\dots,\rho$. Note that, in the ciphers of SAFER family, $K^{(i)}=(K_{2i-1},K_{2i})$.

The $key\text{-}dependent\ imbalance$ $I\left(S^{(1\dots\rho)}|K^{(1\dots\rho)}\right)$ of the I/O sum $S^{(1\dots\rho)}$ is the random variable whose value when $K^{(1\dots\rho)}=k^{(1\dots\rho)}$ is the key-dependent imbalance of $S^{(1\dots\rho)}$, i.e., $I\left(S^{(1\dots\rho)}|k^{(1\dots\rho)}\right)$. The $average\text{-}key\ imbalance$ $\bar{I}\left(S^{(1\dots\rho)}\right)$ of the I/O sum $S^{(1\dots\rho)}$ is the expectation of this key-dependent imbalance computed under the assumption that the round keys are chosen independently and uniformly at random, i.e.,

$$\bar{I}\left(S^{(1\dots\rho)}\right):=E\left[I\left(S^{(1\dots\rho)}|K^{(1\dots\rho)}\right)\right]=\frac{1}{|\mathcal{K}^{\rho}|}\sum_{k^{(1\dots\rho)}\in\mathcal{K}^{\rho}}I\left(S^{(1\dots\rho)}|k^{(1\dots\rho)}\right),$$

where $|\mathcal{K}^{\rho}|$ denotes the cardinality of the set of full keys.

An I/O sum is said to be $effective$ if it has a “large” average-key imbalance, and is said to be $guaranteed$ if its average-key imbalance is 1, the maximum possible.

A $threefold\ sum$ $T^{(i)}$ for the i-th round is a modulo-two sum of three terms: the first, a balanced binary-valued function f_i of the round input $X^{(i)}=Y^{(i-1)}$; the second, a balanced binary-valued function g_i of the round output $Y^{(i)}$; the third, some binary-valued function h_i of the round key $K^{(i)}$, i.e.,

$$T^{(i)}=f_i\left(Y^{(i-1)}\right)\oplus g_i\left(Y^{(i)}\right)\oplus h_i\left(K^{(i)}\right).$$

The function h_i is the $key\ function$ of the threefold sum $T^{(i)}$ and $S^{(i)}=f_i\left(Y^{(i-1)}\right)\oplus g_i\left(Y^{(i)}\right)$ is the $parent\ I/O\ sum$ for $T^{(i)}$.

Theorem 1.1: (Threefold sum imbalance and I/O sum average-key imbalance) *Let* $T^{(1\dots\rho)}=S^{(1\dots\rho)}\oplus h_i\left(K^{(1\dots\rho)}\right)$ *be a threefold sum. Then the average-key imbalance of the parent I/O sum* $S^{(1\dots\rho)}$ *is lower bounded by the threefold sum imbalance, i.e.,*

$$\bar{I}\left(S^{(1\dots\rho)}\right)\geq I\left(T^{(1\dots\rho)}\right).$$

Moreover, equality holds if and only if h is equal to

$$\begin{cases} 0 & if \quad P[S=0/K=0]>1/2 \\ 1 & if \quad P[S=0/K=0]<1/2 \\ arbitrary & if \quad P[S=0/K=0]=1/2 \end{cases}$$

(in this case the function h is called a maximizing key function for $T^{(1\dots\rho)}$*) or is the complement of such a function.*

A threefold sum is said to be $guaranteed$ if it has imbalance 1. It follows from Theorem 1.1 that $T^{(1\dots\rho)}=S^{(1\dots\rho)}\oplus h\left(K^{(1\dots\rho)}\right)$ is guaranteed if only if the parent I/O sum $S^{(1\dots\rho)}$ is guaranteed and h is a maximizing key function.

Lemma 1.1: (Matsui’s Piling-up Lemma) *The imbalance of a modulo-two sum of independent binary-valued variables* $V^{(1)},V^{(2)},\dots,V^{(\rho)}$ *is the product of their imbalance, i.e.,*

$$I\left(\bigoplus_{i=1}^{\rho} V^{(i)}\right) = \prod_{i=1}^{\rho}\left(I(V)^{(i)}\right).$$

Remark 1.1: I*f* $T^{(1)}, T^{(2)}, \dots, T^{(\rho)}$ *are independent threefold sums, then it follows from lemma 1 that*

$$I\left(\bigoplus_{i=1}^{\rho} T^{(i)}\right) = \prod_{i=1}^{\rho}\left(I(T)^{(i)}\right),$$

where $\oplus_{i=1}^{\rho} T^{(i)}$ *is* ρ*-round threefold sum provided that* $T^{(1)}, T^{(2)}, \dots, T^{(\rho)}$ *are linked.*
The round functions of our cipher can be written as

$$Y^{(i)} = R_{K^{(i)}}^{(i)}\left(Y^{(i-1)}\right) = \phi(Y^{(i-1)} \otimes_i K_{2i-1}, K_{2i}),$$

where $\otimes_i$ is the group operation by which one of the round keys is inserted in the i-th round (Fig. 1). We will show that this structure guarantees the independence of any “homomorphic” one-round threefold sum and the input to that round.

Definition 1.1: *Consider an iterated block cipher whose* i*-th round function inserts the key* K_{2i-1} *with a group operation* $\otimes_i$ *at the input to each round (Fig. 1). Let* B^n *denote the set of binary* n*-tuples, very often considered as* $B^n = \{0, 1, 2, \dots, 2^n - 1\}$*. A homomorphism from a group* $(B^n, \otimes)$ *onto the group* $(B, \oplus)$ *is called* $binary\text{-}valued\ homomorphism\ for$ $\otimes$[1]*. An I/O sum for the rounds i to j* $(i \leq j)$ *is homomorphic if the input function is a binary-valued homomorphism for* $\otimes_i$ *and the output function is a binary-valued homomorphism for* $\otimes_{j+1}$*. A threefold sum is* $homomorphic$ *if the parent I/O sum is homomorphic.*

If for all rounds $i,$ the operation $\otimes_i$ is the bitwise XOR operation in B^n, then the only binary-valued homomorphisms are the linear function $l_a(x) = a \circ x$ for x in B^n, where a is a non zero n-tuple and $a \circ x$ denotes the scalar product with operations of $GF(2)$. An I/O sum (or a threefold sum) whose input and output functions are l_a and l_b, respectively, will be called $linear$ with $linear\ mask$ (a, b).

Theorem 1.2: *Consider the cascade of* ρ *rounds with round functions* $R^{(1)}, R^{(2)}, \dots, R^{(\rho)}$ *for which*

$$Y^{(i)} = R_{K^{(i)}}^{(i)}\left(Y^{(i-1)}\right) = \phi_i\left(Y^{(i-1)} \otimes_i K_{2i-1}, K_{2i}\right),$$

where $\otimes_i$ *denotes a group operation in* B^n *and where* $\phi_i(., k_{2i})$ *is a bijection on* B^n *for all* k_{2i} *(Fig. 1). Let*

$$T^{(i)} = f_i\left(Y^{(i-1)}\right) \oplus f_{i+1}\left(Y^{(i)}\right) \oplus \left(f_i(K_{2i-1}) \oplus h_i(K_{2i})\right) \tag{1}$$

be a homomorphic threefold sum for the i*-th round, so that* $T^{(1)}, T^{(2)}, \dots, T^{(\rho)}$ *are linked. Then, the average-key imbalance for the parent I/O sum* $S^{(1\dots\rho)}$ *of the threefold sum* $T^{(1\dots\rho)} := \oplus_{i=1}^{\rho} T^{(i)}$ *is lower bounded by the product of the one-round threefold sum imbalance, i.e.,*

$$\bar{I}\left(S^{(1\dots\rho)}\right) \geq \prod_{i=1}^{\rho} I\left(T^{(i)}\right). \tag{2}$$

Note that, as $\phi_i(., k_{2i})$ is a bijection and X is uniformly distributed, $Y^{(0)}, Y^{(1)}, \dots, Y^{(\rho-1)}$ are uniformly distributed. Thus, $I(T^{(i)})$ for $i = 2, 3, \dots, \rho$ can be computed quite easily.

For a given $S^{(1\dots\rho)}$ we can evaluate the right side of (2) for many choices of linked homomorphic one-round threefold sums whose sum has $S^{(1\dots\rho)}$ as its parent I/O sum. We can then use the maximum such threefold-sum imbalance as an approximation $\bar{I}\left(S^{(1\dots\rho)}\right)$, i.e.,

[1] f is such homomorphism if, for all $U, V \in B^n$ $f(U \otimes V) = f(U) \oplus f(V)$ and if f is not identically zero.

$$\bar{I}\big(S^{(1\ldots\rho)}\big) \approx \max_{\substack{f_2,\ldots,f_\rho \\ h_1,\ldots,h_\rho}} \prod_{i=1}^{\rho} I(T^{(i)}), \tag{3}$$

where $T^{(i)}$ is given by (1). The following is an effective procedure for finding this maximum.

Procedure for finding an effective ρ-round homomorphic I/O sum for a cipher whose round functions insert a key by using a group operation at the input.

1. For $i = 1,2,\ldots,\rho+1$ find the set $\mathcal{H}_i$ of all binary-valued functions on B^n that are binary-valued homomorphisms for $\otimes_i$.
2. For $i = 1,2,\ldots,\rho$ find the imbalance of all i-th round homomorphic threefold sums with input function f_i in $\mathcal{H}_i$, output function f_{i+1} in $\mathcal{H}_{i+1}$, and maximizing key function. Discard the threefold sums with small imbalance.
3. Consider each possible choice of ρ linked threefold sums containing one of the threefold sums found in step 2 in each round.
   Use the right side of (3) as an estimate of the imbalance of the ρ-round I/O sums.
   Output the ρ-round I/O sums having the largest estimated imbalance.

3. Linear Cryptanalysis of Modified SAFER+

Now we will find effective homomorphic I/O sums to a cascade of half-rounds of modified SAFER+ algorithm. We first find all binary-valued homomorphisms for ADD/XOR and for XOR/ADD. There are $2^8 - 1$ binary-valued homomorphisms for 8 bit XOR, namely the functions defined as $l_{a2}(V2) := a2 \circ V2$, where $a2$ is a non-zero binary 8-tuple " $\circ$ " operation denotes the modulo two “dot product”. There is only one binary-valued homomorphism for modulo 256 addition, namely the function l_{a1} where $a1 = 0000\ 0001 = 01$ (hex notation of byte).

Thus, there exist $2^{72} - 1$ balanced homomorphisms for ADD/XOR, namely the functions l_a defined as $l_a(V) = a \circ V$ where a lies in the set of 128-tuples.

$$\mathcal{A} = \{a\colon a \in \{0,1\}^{128} \backslash \{00\};\, a1, a2, a3, a4, a9, a10, a11, a12\} \in \{00, 01\}\}.$$

Similarly, there are $2^{72} - 1$ balanced homomorphism for XOR/ADD, namely the functions $l_b(V) = b * V$, where b lies in the set

$$\mathcal{B} = \{b\colon b \in \{0,1\}^{128} \backslash \{00\};\, b5, b6, b7, b8, b13, b14, b15, b16\} \in \{00, 01\}\}.$$

The set of all homomorphic functions for XOR/ADD and the set of all homomorphic functions for ADD/XOR are subsets of the set of all linear binary-valued functions.

We now consider the part of round (half-round) containing the PHT function. The following lemma specifies all homomorphic I/O sums that have non-zero imbalance. The input function must be balanced and homomorphic for ADD/XOR; the output function must be balanced and homomorphic for XOR/ ADD. There are $(2^{72} - 1)^2$ such I/O sums, namely

$$\mathrm{S}_{a,b}^{\mathrm{PHT-hr}} := l_a(\mathrm{V}) \oplus l_b(\mathrm{Y}) \quad a \in \mathcal{A}\,, b\ \in \mathcal{B}.$$

Lemma 2.1: *For the PHT-half-round, the only homomorphic I/O sums that have non-zero imbalance are the $2^{16} - 1$ guaranteed I/O sums obtained by XOR-ing together any positive number of the 16 guaranteed I/O sums listed in Table 1.*

Table 1. Effective I/O sums for the PHT-function.

(a,b)	$l_b(Y)$	$l_a(W)$	$I\left(S_{a,b}^{\mathrm{PHT}}\right)$
$(0100000101001010, 1000000000000000)$	$Y1_0$	$W2_0 \oplus W8_0 \oplus W10_0 \oplus W13_0 \oplus W15_0$	1
$(0100010111001110, 0100000000000000)$	$Y2_0$	$W2_0 \oplus W6_0 \oplus W8_0 \oplus W9_0 \oplus W10_0 \oplus W11_0 \oplus$ $W13_0 \oplus W14_0 \oplus W15_0$	1
$(1010010001000001, 0010000000000000)$	$Y3_0$	$W1_0 \oplus W3_0 \oplus W6_0 \oplus W10_0 \oplus W16_0$	1
$(1111010001000011, 0001000000000000)$	$Y4_0$	$W1_0 \oplus W2_0 \oplus W3_0 \oplus W4_0 \oplus W6_0 \oplus W10_0 \oplus$ $W15_0 \oplus W16_0$	1
$(0000011010010100, 0000100000000000)$	$Y5_0$	$W6_0 \oplus W7_0 \oplus W9_0 \oplus W12_0 \oplus W14_0$	1
$(0101011010110100, 0000010000000000)$	$Y6_0$	$W2_0 \oplus W4_0 \oplus W6_0 \oplus W7_0 \oplus W9_0 \oplus W11_0 \oplus$ $W12_0 \oplus W14_0$	1
$(0101100100000010, 0000001000000000)$	$Y7_0$	$W2_0 \oplus W4_0 \oplus W5_0 \oplus W8_0 \oplus W15_0$	1
$(0111110101000010, 0000000100000000)$	$Y8_0$	$W2_0 \oplus W3_0 \oplus W4_0 \oplus W5_0 \oplus W6_0 \oplus W8_0 \oplus$ $W10_0 \oplus W15_0$	1
$(0000001010010101, 0000000010000000)$	$Y9_0$	$W7_0 \oplus W9_0 \oplus W12_0 \oplus W14_0 \oplus W16_0$	1
$(0000001111011101, 0000000001000000)$	$Y10_0$	$W7_0 \oplus W8_0 \oplus W9_0 \oplus W10_0 \oplus W12_0 \oplus$ $W13_0 \oplus W14_0 \oplus W16_0$	1
$(0101000001101000, 0000000000100000)$	$Y11_0$	$W2_0 \oplus W4_0 \oplus W10_0 \oplus W11_0 \oplus W13_0$	1
$(0101001001111001, 0000000000010000)$	$Y12_0$	$W2_0 \oplus W4_0 \oplus W7_0 \oplus W10_0 \oplus W11_0 \oplus W12_0 \oplus$ $W13_0 \oplus W16_0$	1
$(0001100100100100, 0000000000001000)$	$Y13_0$	$W4_0 \oplus W5_0 \oplus W8_0 \oplus W11_0 \oplus W14_0$	1
$(1001100100110101, 0000000000000100)$	$Y14_0$	$W1_0 \oplus W4_0 \oplus W5_0 \oplus W8_0 \oplus W11_0 \oplus W12_0 \oplus$ $W14_0 \oplus W16_0$	1
$(1010010000010001, 0000000000000010)$	$Y15_0$	$W1_0 \oplus W3_0 \oplus W6_0 \oplus W6_0 \oplus W12_0 \oplus W16_0$	1
$(1010110100010101, 0000000000000001)$	$Y16_0$	$W1_0 \oplus W3_0 \oplus W5_0 \oplus W6_0 \oplus W8_0 \oplus W12_0 \oplus$ $W14_0 \oplus W16_0$	1

Since $I\left(S_{a,b}^{\mathrm{PHT-hr}} | k_{2i}\right) = I\left(S_{a,b}^{\mathrm{PHT}}\right)$ for any $k_{2i} \in \{0,1\}^{128}$, where $S_{a,b}^{\mathrm{PHT}} := l_a(W) \oplus l_b(Y)$, ($a \in \mathcal{A}$ and $b \in \mathcal{B}$) is an I/O sum for the PHT function alone, we will look for I/O sums $S_{a,b}^{\mathrm{PHT}}$ with non-zero imbalance instead of looking for $S_{a,b}^{\mathrm{PHT-hr}}$ with non-zero imbalance. So our main purpose is to find $a \in \mathcal{A}$ and $b \in \mathcal{B}$ such that $S_{a,b}^{\mathrm{PHT}}$ sum has the form $W_\alpha \oplus \phi(W_{127}, \ldots, W_{\alpha-1}, W_{\alpha+1}, \ldots, W_0)$ for some input bit W_α, since this implies that the I/O sum imbalance is 0.

First we consider PHT function. Table 2 shows three kind of dependences for some input and output bits. For example, for $Y10_2$ we conclude that

$$Y10_2 = W1_1 \oplus W2_2 \oplus W9_2 \oplus W10_2 \oplus W11_1 \oplus W12_1 \oplus \phi(W2_1, W9_1, W10_1; W1_0, W2_0, \\ W3_0, W4_0, W9_0, W10_0, W11_0, W12_0; W5, W6, W7, W8, W13, W14, W15, W16)$$

for some function ϕ.

Table 2. Dependencies for certain bits of the PHT output Y on certain bits of the PHT input W

"0"-no dependence, "1"-binary linear dependence (i.e., complementing this input bit complements the corresponding output bit), "n"- non-linear binary dependence.

Input bit Output bit		$W1$	$W2$	$W3$	$W4$	$W9$	$W10$	$W11$	$W12$
		7654321	7654321	7654321	7654321	7654321	7654321	7654321	7654321
$Y1$	7	00001nn	1nnnnnn	001nnnn	01nnnnn	01nnnnn	1nnnnnn	001nnnn	001nnnn
	6	000001n	01nnnnn	0001nnn	001nnnn	001nnnn	01nnnnn	0001nnn	0001nnn
	5	0000001	001nnnn	00001nn	0001nnn	0001nnn	001nnnn	00001nn	00001nn
	4	0000000	0001nnn	000001n	00001nn	00001nn	0001nnn	000001n	000001n
	3	0000000	00001nn	0000001	000001n	000001n	00001nn	0000001	0000001
	2	0000000	000001n	0000000	0000001	0000001	000001n	0000000	0000000
	1	0000000	0000001	0000000	0000000	0000000	0000001	0000000	0000000
	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y2$	7	0001nnn	1nnnnnn	01nnnnn	01nnnnn	1nnnnnn	1nnnnnn	001nnnn	01nnnnn
	6	00001nn	01nnnnn	001nnnn	001nnnn	01nnnnn	01nnnnn	0001nnn	001nnnn
	5	000001n	001nnnn	0001nnn	0001nnn	001nnnn	001nnnn	00001nn	0001nnn
	4	0000001	0001nnn	00001nn	00001nn	0001nnn	0001nnn	000001n	00001nn
	3	0000000	00001nn	000001n	000001n	00001nn	00001nn	0000001	000001n
	2	0000000	000001n	0000001	0000001	000001n	000001n	0000000	0000001
	1	0000000	0000001	0000000	0000000	0000001	0000001	0000000	0000000
	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y3$	7	1nnnnnn	01nnnnn	1nnnnnn	01nnnnn	00001nn	1nnnnnn	001nnnn	01nnnnn
	6	01nnnnn	001nnnn	01nnnnn	001nnnn	000001n	01nnnnn	0001nnn	001nnnn
	5	001nnnn	0001nnn	001nnnn	0001nnn	0000001	001nnnn	00001nn	0001nnn
	4	0001nnn	00001nn	0001nnn	00001nn	0000000	0001nnn	000001n	00001nn
	3	00001nn	000001n	00001nn	000001n	0000000	00001nn	0000001	000001n
	2	000001n	0000001	000001n	0000001	0000000	000001n	0000000	0000001
	1	0000001	0000000	0000001	0000000	0000000	0000001	0000000	0000000
	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y4$	7	1nnnnnn	1nnnnnn	1nnnnnn	1nnnnnn	0001nnn	1nnnnnn	01nnnnn	01nnnnn
	6	01nnnnn	01nnnnn	01nnnnn	01nnnnn	00001nn	01nnnnn	001nnnn	001nnnn
	5	001nnnn	001nnnn	001nnnn	001nnnn	000001n	001nnnn	0001nnn	0001nnn
	4	0001nnn	0001nnn	0001nnn	0001nnn	0000001	0001nnn	00001nn	00001nn
	3	00001nn	00001nn	00001nn	00001nn	0000000	00001nn	000001n	000001n
	2	000001n	000001n	000001n	000001n	0000000	000001n	0000001	0000001
	1	0000001	0000001	0000001	0000001	0000000	0000001	0000000	0000000
	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y5$	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y6$	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y7$	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y8$	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y9$	7	01nnnnn	001nnnn	001nnnn	0001nnn	1nnnnnn	01nnnnn	00001nn	1nnnnnn
	6	001nnnn	0001nnn	0001nnn	00001nn	01nnnnn	001nnnn	000001n	01nnnnn
	5	0001nnn	00001nn	00001nn	000001n	001nnnn	0001nnn	0000001	001nnnn
	4	00001nn	000001n	000001n	0000001	0001nnn	00001nn	0000000	0001nnn
	3	000001n	0000001	0000001	0000000	00001nn	000001n	0000000	00001nn
	2	0000001	0000000	0000000	0000000	000001n	0000001	0000000	000001n
	1	0000000	0000000	0000000	0000000	0000001	0000000	0000000	0000001
	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y10$	7	01nnnnn	01nnnnn	001nnnn	001nnnn	1nnnnnn	1nnnnnn	0001nnn	1nnnnnn
	6	001nnnn	001nnnn	0001nnn	0001nnn	01nnnnn	01nnnnn	00001nn	01nnnnn
	5	0001nnn	0001nnn	00001nn	00001nn	001nnnn	001nnnn	000001n	001nnnn
	4	00001nn	00001nn	000001n	000001n	0001nnn	0001nnn	0000001	0001nnn
	3	000001n	000001n	0000001	0000001	00001nn	00001nn	0000000	00001nn
	2	0000001	0000001	0000000	0000000	000001n	000001n	0000000	000001n
	1	0000000	0000000	0000000	0000000	0000001	0000001	0000000	0000001
	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y11$	7	001nnnn	1nnnnnn	00001nn	1nnnnnn	001nnnn	1nnnnnn	1nnnnnn	01nnnnn
	6	0001nnn	01nnnnn	000001n	01nnnnn	0001nnn	01nnnnn	01nnnnn	001nnnn
	5	00001nn	001nnnn	0000001	001nnnn	00001nn	001nnnn	001nnnn	0001nnn
	4	000001n	0001nnn	0000000	0001nnn	000001n	0001nnn	0001nnn	00001nn
	3	0000001	00001nn	0000000	00001nn	0000001	00001nn	00001nn	000001n
	2	0000000	000001n	0000000	000001n	0000000	000001n	000001n	0000001
	1	0000000	0000001	0000000	0000001	0000000	0000001	0000001	0000000
	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y12$	7	01nnnnn	1nnnnnn	0001nnn	1nnnnnn	01nnnnn	1nnnnnn	1nnnnnn	1nnnnnn
	6	001nnnn	01nnnnn	00001nn	01nnnnn	001nnnn	01nnnnn	01nnnnn	01nnnnn
	5	0001nnn	001nnnn	000001n	001nnnn	0001nnn	001nnnn	001nnnn	001nnnn
	4	00001nn	0001nnn	0000001	0001nnn	00001nn	0001nnn	0001nnn	0001nnn
	3	000001n	00001nn	0000000	00001nn	000001n	00001nn	00001nn	00001nn
	2	0000001	000001n	0000000	000001n	0000001	000001n	000001n	000001n
	1	0000000	0000001	0000000	0000001	0000000	0000001	0000001	0000001
	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y13$	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y14$	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y15$	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000
$Y16$	0	0000000	0000000	0000000	0000000	0000000	0000000	0000000	0000000

Secondly, $S_{a,b}^{\mathrm{PHT}}$ depends linearly on some W_α if $l_b(Y)$ depends linearly on W_α and $a_\alpha = 0$. Since $a \in \mathcal{A}$, then the rows of Table 2 contain input bits that cannot appear in $l_a(W)$. Whenever $l_b(Y)$ contains an output bit that depends linearly on such a W_α and contain no other output bit that depends on W_α $I(S_{a,b}^{\mathrm{PHT}}) = 0$. By using Table 2 we can iteratively show that I/O sums $S_{a,b}^{\mathrm{PHT}}$ with none-zero imbalance cannot contain any of the output bit Yi_j, where $i = 1,2,3,4,9,10,11,12$ and $j = 1,2,3,4,5,6,7$.

Finally, we consider the $2^{16} - 1$ balanced output functions obtained as linear combinations of the remaining 16 output bits that might occur if $b \in \mathcal{B}$. For each of these output functions, we found all input functions such that $S_{a,b}^{\mathrm{PHT}}$ doesn't depend linearly on any input bit. It is easy to show that $\left((a,b); a,b \in \{0,1\}^{128}, I(\mathrm{S}_{a,b}^{\mathrm{PHT}}) = 1\right)$ is a subgroup of $(\mathcal{A} \times \mathcal{B}, \oplus_{256\mathrm{bits}})$. (In fact if $I(\mathrm{S}_{a,b}^{\mathrm{PHT}}) = 1$ and $I(\mathrm{S}_{a',b'}^{\mathrm{PHT}}) = 1$ for some 128 tuples a' and b', then $I(\mathrm{S}_{a\oplus a',b\oplus b'}^{\mathrm{PHT}}) = I(\mathrm{S}_{a,b}^{\mathrm{PHT}} \oplus \mathrm{S}_{a',b'}^{\mathrm{PHT}}) = I(\mathrm{S}_{a,b}^{\mathrm{PHT}}) \cdot I(\mathrm{S}_{a',b'}^{\mathrm{PHT}}) = 1$ as well, whch shows that $\mathcal{A} \times \mathcal{B}$ is closed under "$\oplus$".) Thus, we obtain all I/O sums with non-zero imbalance, as is done in the statement of the lemma.

We next consider the half round containing the nonlinear function NL and find homomorphic I/O sums for NL that have non-zero imbalance. Here the input function is homomorphic for XOR/ADD and the output function is homomorphic for ADD/XOR. Such I/O sums can be obtained by summing I/O sums for its EXP and LOG blocks. For the function EXP with the input byte U1 and output byte V1, the only homomorphic I/O sums are

$$S_{a1,b1}^{\mathrm{EXP}} = l_{a1}(U1) \oplus l_{b1}(V1), \text{ for } a1 \in B^8 \backslash \{00\};\ b1 = 01.$$

The most effective ones are obtained when $(a1, b1)$ is equal to $(cd, 01)$ or $(ff, 01)$ (the imbalance being $\frac{28}{128}$) or to $(86{,}01), (bf, 01), (c0, 01)$ or $(f7{,}01)$ (the imbalance being $\frac{24}{128}$). Computing all these I/O sums for EXP , establishes the following.

Remark 2.1: $I(S_{01,01}^{\mathrm{EXP}}) = I(S_{02,01}^{\mathrm{EXP}}) = I(S_{03,01}^{\mathrm{EXP}}) = 0$. *Furthermore, for all* $a1$ *and* $b1$ *in* B^8, *if* $a1_7 = 0$, *then* $I(S_{a1,b1}^{EXP}) = 0$.

For the function LOG with the input U2 and output V2, the only homomorphic I/O sums are

$$S_{a2,b2}^{\mathrm{LOG}} = l_{a2}(\mathrm{U2}) \oplus l_{b2}(\mathrm{V2}), \text{ for } a2 = 01;\ b2 \in B^8 \backslash \{00\}.$$

Their imbalance is easily deduced since $I(S_{a1,b1}^{\mathrm{EXP}}) = I(S_{b1,a1}^{\mathrm{LOG}})$.

Remark 2.2: *For all* $a1$ *and* $b1$ *in* B^8, *if* $b1_7 = 0$, *then* $I(S_{a1,b1}^{\mathrm{LOG}}) = 0$.

Finally we have link I/O sums for successive half rounds.

Theorem 2.1: *The procedure for finding effective homomorphic I/O sums doesn't find an I/O sum with non-zero imbalance for cascade of half rounds taken in the same order as they are used in modified SAFER+ and containing at least two PHT-layers.*

Proof: Let $T_{a,b}^{\mathrm{PHT-hr}}$, $T_{b,c}^{\mathrm{NL}}$ and $T_{c,d}^{\mathrm{PHT-hr}}$ be linked homomorphic half-round threefold sums with maximizing key function. If $T_{a,b}^{\mathrm{PHT-hr}}$ and $T_{c,d}^{\mathrm{PHT-hr}}$ have none zero imbalance, the 128 bit none zero masks a, b, c, d, can have a 1 only in the two least significant bits of each byte (bits of byte are numbered from 7 for the most significant bit to 0 for the least significant bit) (Lemma 2.1). Then $I(T_{b,c}^{\mathrm{NL}}) = 0$ since the I/O sum average key imbalance is also 0 (Remark 2.1) Therefore, the sum of the three half-round threefold sums has imbalance 0.

One of the most effective I/O sums is $S_{a,b}^{\mathrm{NL-PHT-NL}}$, where $a2, a4, a11$ and $b5, b6$ are either cd or ff and other bytes of a and b are zero. Their imbalance is $\left(\frac{28}{128}\right)^5$, because

$$I(S_{ff,01}^{\mathrm{EXP}}) = I(S_{cd,01}^{\mathrm{EXP}}) = I(S_{01,ff}^{\mathrm{LOG}}) = I(S_{01,cd}^{\mathrm{LOG}}) = \frac{28}{128}$$

and because

$$I(S_{a,b}^{\mathrm{PHT}}) = I(S_{a1\oplus a2,b1\oplus b2}^{\mathrm{PHT}}) = I(S_{a1,b1}^{\mathrm{PHT}}) \cdot I(S_{a2,b2}^{\mathrm{PHT}}) = 1,$$

if

$$\begin{aligned} a1\oplus a2 = {} & (00\ 00\ 00\ 00\ 00\ 01\ 01\ 00\ 01\ 00\ 00\ 01\ 00\ 01\ 00\ 00)\oplus \\ & (00\ 01\ 00\ 01\ 00\ 01\ 01\ 00\ 01\ 00\ 01\ 01\ 00\ 01\ 00\ 00) = \\ & (00\ 01\ 00\ 01\ 00\ 00\ 00\ 00\ 00\ 00\ 01\ 00\ 00\ 00\ 00\ 00) = a \text{ (hex notation)}, \end{aligned}$$

$$\begin{aligned} b1\oplus b2 = {} & (\ 00\ 00\ 00\ 00\ 01\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00)\oplus \\ & (00\ 00\ 00\ 00\ 00\ 01\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00) = \\ & (00\ 00\ 00\ 00\ 01\ 01\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00) = b \text{ (hex notation)}. \end{aligned}$$

4. Conclusion

We have proved that the procedure for finding effective homomorphic I/O sums, cannot find an I/O sum with none-zero imbalance for two rounds of modified SAFER+. Thus, as in the case with regular SAFER+, a modified SAFER+ is also secure against linear cryptanalysis after only three of its suggested six rounds.

Acknowledgement

The authors are thankful to Prof. Gurgen Khachatrian for very useful discussions and comments.

References

[1] K. Kyuregyan, “Some Modifications of SAFER+”, *In Reports of NAS RA,* vol. 115, no 1, pp. 33--39, Yerevan, Armenia, 2015.

[2] J. L. Massey, G. Khachatrian and M Kyuregian, “Nomination of SAFER+ as candidate algorithm for the advanced encryption standard”, (AES). *NIST AES Proposal,* 1998.

[3] Carlo H.: Cryptanalysis of iterated Block Ciphers, ETH Series In Information Processing (Ed Massey), v. 7, Konstanz: Hartung-Gorre Verlag, 1996.

Submitted 05.11.2016, accepted 03.02.2016

Ձևափոխած SAFER+ ալգորիթմի գծային վերլուծությունը

Մ Կյուրեղյան, Ք.Կյուրեղյան

Ամփոփում

[2] հոդվածում ներկայացված են SAFER+ համակարգի ձևափոխությունները, որոնք ալգորիթմի արագագործությունը բարելավում են մոտավորապես 1,7 անգամ, դիֆերենցիալ վերլուծության նկատմամբ ապահովելով նույն կրիպտոկայունությունը, ինչ՝ SAFER+ համակարգի ալգորիթմը: Այս հոդվածում ներկայացված է ձևափոխված SAFER+ համակարգի կրիպտոկայունությունը բլոկային ծածկագրական համակարգերի մյուս ամենաարդյունավետ կրիպտովերլուծության, գծային վերլուծության նկատմամբ:

Ցույց է տրվել, որ SAFER+ համակարգը ձևափոխություններից հետո (ձևափոխած SAFER+) գծային վերլուծության նկատմամբ ևս ունի միննույն կրիպտոկայունությունը, ինչ՝ SAFER+ համակարգը:

Линейнный криптоанализ модифицированного алгоритма SAFER+

М. Кюрегян, К. Кюрегян

Аннотация

В статье [2] представлены некоторые модификации системы SAFER+, которые примерно в 1,7 раза увеличивают скорость алгоритма, обеспечивая такую же криптостойкость по отношению к дифференциальному анализу, как алгоритм системы SAFER+. В этой статье представлена криптостойкость модифицированной системы SAFER+ по отношению к линейному анализу, который являетса одним из эффективных аттак против итеративных блочных шифров.

Было доказано, что после модификации система SAFER+ (модифицированный SAFER+) имеет ту же криптостойкость по отношению к линейному анализу, что и система SAFER+.