UDC 003.26:512.5

Recovering S-boxes from the Differential Distribution Table and Affine Equivalence Classes of S-boxes with Respect to Modular Addition

Stepan Yershov[1,a], Serhii Yakovliev[1,b]

[1]*National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Institute of Physics and Technology*

Abstract

This paper considers the problem of S-box recovery from its differential distribution table (DDT) with respect to modular addition. We describe the structure of DDT for affine S-boxes and affine transformations of S-boxes. We found some unexpected internal symmetry in DDT w.r.t. modular addition, which holds for other algebraic operations, but not for bitwise addition (XOR). We describe two classes of affine transformations (affine shifts) which preserve the structure of DDT. For a recovery of S-box from its DDT we propose a backtracking-based algorithm, which is moderately effective for medium-size S-boxes. We apply our algorithm for three-bit S-boxes and describe the structure of their DDT equivalence classes; among other things, it was shown that affine shifts do not cover all DDT equivalence class members.

Keywords: symmetric cryptography, S-box, differential cryptanalysis, difference distribution table, DDT, S-box recovery, affine equivalency, backtracking

Introduction

Nowadays, one of the most significant primitives of symmetric cryptology are block ciphers. The modern block ciphers are developed based on methodologies that guarantee security against all known cryptographic attacks. Their security, in most cases, is based on the properties of specific transformations — *substitution boxes* (further referred to as *S-boxes*). The primary purpose of S-boxes is to perform a reversible, nonlinear transformation on the input bits. As a result, they provide provable security against differential and linear cryptanalysis attacks ([1, 2, 3]).

There are numerous methods and tools for studying the security characteristics of S-boxes and ciphers based on differential cryptanalysis. An example could be the work of Nyberg *et al.* [4], where the scheme of DES-like ciphers is described in terms of probabilities of differentials. Another important tool of differential cryptanalysis is so-called *Difference Distribution Table*, or *DDT*, which contains a meaningful description of the differential probabilities of a given S-box.

One of the popular techniques of S-box creation is to generate an S-box randomly or by specific features, and then study its properties using known methods, one of which is to analyze its DDT. From the other side, obtaining an S-box from a given DDT is an important practical task for several reasons. First, such an algorithm allows us to obtain an S-box with predefined properties, which is a much better method for selecting an S-box. Thus, having an algorithm for recovering an S-box from a DDT, we can first generate a DDT that describes an S-box that satisfies our needs, and then obtain an S-box based on the given DDT.

However, alternatively, S-box generation could be done in another way: it's known in cryptology world cases when S-boxes were obtained with some sophisticated design principles and kept secret during publishing, like it was in the case of DES [5]. Additionally, Perrin *et al.* designed methodologies for the decomposition of S-box secret structure, illustrated in work [6] for cipher Skipjack [7] or cipher Kuznyechik, described in series of works (e.g. [8, 9, 10]).

[a]yershovstepan@gmail.com
[b]yasv@rl.kiev.ua

Secondly, there are attacks in which an attacker can obtain a DDT, and, having an algorithm for recovering the S-box from the DDT, they can get a scheme for generating the S-box, simplifying the process of attacking the cipher. Such cryptoattacks are often the case for cryptoprimitives with secret S-boxes, such as Blowfish block cipher [11]. Thus, Bar-On *et al.* showed in their work on effective slide attacks, particularly on GOST [12], that it's possible to obtain DDT and thus recover secret S-boxes, which is helpful for cryptoanalysis.

Additionally, the considered problem might have rather theoretical interest from the research perspective about the properties of DDTs. For instance, Boura *et al.* [13] investigated whether two S-boxes that share the same DDT are necessarily coupled with some linear transformation (with respect to bitwise addition).

However, despite the practical value, the problem of S-box recovery from DDT has not yet been fully solved. The main obstacle to S-box recovery is the *DDT equivalence classes* since there are cases when different S-boxes have the same DDT. Hence, there is no bijective mapping from the set of S-boxes to the set of DDTs, which makes solving this problem much harder.

These issues have been studied in the case of bitwise addition (XOR). This article will focus on the modular arithmetic modulo power of two. The usage of modular addition is also a rather crucial practical case, since there is a separate trend in differential cryptanalysis where differentials constructed with respect to the modulo power of two are used. As an example, probabilities for differentials with respect to the modular addition are studied in [14, 15].

It should be noted that there are several algorithms for obtaining an S-box with a specific DDT. Some are efficient, while others are no better than a simple brute-force search. Dunkelman and Huang presented a new algorithm for solving the problem of reconstructing an S-box from a given DDT [16], which is based on the known relationship between DDT and Linear Approximation Table (LAT), studied in [17, 18, 19]. The invented algorithm results better than the already known guess-and-determine algorithm presented in [13]. Note that the mentioned works concentrate on the bitwise addition, rather than a modulo power of two.

In this paper, we present theoretical results describing the influence of S-box properties on the structure of the corresponding Difference Distribution Table (DDT) with respect to addition modulo 2^n. In particular, we study affine S-boxes, S-boxes related through affine transformations, and their interconnections in terms of the corresponding DDTs and DDT equivalence classes. Furthermore, we provide computational results that characterize the structure and size of DDT equivalence classes for S-boxes of small dimensions. Specifically, we present explicit classifications and numerical data for three-bit S-boxes, demonstrating the actual powers of their DDT equivalence classes and illustrating how these values relate to the underlying affine transformations. Finally, we propose a backtracking-based algorithm for recovering an S-box from its DDT, accompanied by an analysis of its time efficiency.

The results of this work were partially presented at the XVII Scientific and Practical Conference «Theoretical and Applied Problems of Physics, Mathematics, and Informatics» (April 26-27, 2019, Kyiv, Ukraine).

The rest of the paper is organized as follows. The first section introduces the basic definitions and preliminary concepts used throughout the paper. In the second section, we examine the dependence of the differential in the DDT constructed with respect to addition modulo a power of two on affine transformations, provide analytical estimates of the sizes of DDT equivalence classes defined by affine transformations, and discuss practical properties of DDTs and DDT equivalence classes under modulo a power of two operations. The third section presents the design of a backtracking-based algorithm for recovering an S-box from its DDT, while the fourth section provides an analysis of the algorithm's time efficiency. Finally, the fifth section presents computational results for small-size S-boxes, including a detailed description of three-bit S-box DDT equivalence classes and their actual powers.

1. Terms and Notation

In this work, V_n denotes the space of n-bit vectors: $V_n = \{0,1\}^n$. Vectors of V_n are naturally interpreted as numbers modulo 2^n. In this sense we associate V_n with $\mathbb{Z}_{2^n}$.

n-bit S-box is a Boolean function of form $S\colon V_n \to V_n$. In this work, we consider only bijective S-boxes with equal length of input and output vectors.

The differential of S-box S is an arbitrary pair (a, b) of vectors $a, b \in V_n$

The difference distribution table (DDT) for an S-box is a two-dimensional table where each cell contains a counter indicating how many times a pair of input differences equals a and a pair of output differences equals b.

Thus, for an input difference $a \in V_n$ and an output difference $b \in V_m$, the cell $\delta_S(a, b)$ of the DDT is defined as:

$$\delta_S(a, b) = |\{z \in V_n \mid S(z \oplus a) \oplus S(z) = b\}|,$$

where $\oplus$ denotes bitwise addition. We will address to DDTs constructed in this manner as $\mathrm{DDT}_{\oplus}$.

Another variant of the DDT is constructed based on addition modulo 2^n. In this case, the DDT cell looks like this:

$$\delta_S(a, b) = |\{z \in V_n \mid S(z + a) = b + S(z)\}|,$$

where $+$ denotes addition modulo 2^n. This variant of the DDT will be referred to as DDT_+. Further in this paper, if not specified otherwise, by DDT we mean DDT_+.

Even though the $\mathrm{DDT}_{\oplus}$ and DDT_+ have similar appearance, properties of these tables are different.

S-boxes $S_1(x)$ and $S_2(x)$ are called *DDT equivalent* if they have the same DDT.

Two n-bit S-boxes $S_1(x)$ and $S_2(x)$ are *affine equivalent* if there exist affine bijective mappings $L_1, L_2\colon V_n \to V_n$ such that

$$S_1(x) = L_1(S_2(L_2(x))).$$

Note that affine equivalencies with respect to XOR and with respect to modular addition are different relations.

2. DDT Equivalence of S-Boxes

As mentioned before, one of the main difficulties that arise when recovering an S-box from its DDT_+ is the fact that there are cases where different S-boxes have the same DDT_+ or are pretty similar based on some criterion, which significantly complicates the task at hand. Therefore, in practice, *S-boxes equivalence classes* are introduced.

Thus in this section, we will consider the S-box transformations that, after applying, produce DDT_+-equivalent S-box, e.g., do not change the table's structure.

2.1. Affine S-Boxes

At first we describe the structure of DDT_+ equivalence classes by considering the class of the simplest transformations, namely *affine S-boxes* with respect to modular addition:

$$S(x) = ux + w, \text{ where } u \in \mathbb{Z}_{2^n}^*, w \in \mathbb{Z}_{2^n}.$$

Note that we impose the condition $u \in \mathbb{Z}_{2^n}^*$ to satisfy the condition of S-box bijectivity. Thus, the number of affine n-bit S-boxes (w.r.t. $+$) is $2^n\varphi(2^n) = 2^{2n-1}$, where $\varphi(x)$ is the Euler totient function.

Consider in detail the DDT_+ element $\delta_S(a, b)$ of the affine S-box. We have

$$S(x + a) = S(x) + b$$
$$ux + au + w = ux + w + b$$
$$au = b,$$

which is simultaneously true or false for every $x \in \mathbb{Z}_{2^n}$. Therefore,

$$\delta_S(a, b) = 2^n \cdot [au = b],$$

where $[.]$ is the Iverson's brackets (indicator function). Thereby, there is only one nonzero element in the DDT_+ row a, which is located at the position au and is equal to 2^n, and all the others are 0. Note that this position also has some boundaries.

Consider odd rows, i.e., $a = 2k + 1, k \in \mathbb{N}$. Based on the condition $u \in \mathbb{Z}_{2^n}^*$, we have that $u = 2t + 1$. As a result, we can see that:

$$b = au = (2k + 1)(2t + 1) = 2(2kt + k + t) + 1,$$

i.e., a nonzero element 2^n can only be at odd positions in odd rows.

In the case of even rows ($a = 2k, k \in \mathbb{N}$), note that, based on

$$b = au = 2k(2t + 1),$$

element 2^n can be placed only in even positions.

Since the DDT_+ of an affine S-box $S(x) = (ux + w) \bmod 2^n$ is affected only by the coefficient u, for a fixed value of u, all S-boxes have equivalent DDTs. This is a special case of a more global relation that will be given further.

The number of unique DDT_+ equivalence classes of a given affine n-bit S-box is equal to $\varphi(2^n) = 2^{n-1}$; therefore, the cardinality of each DDT_+ equivalence class of the affine S-box is 2^n.

2.2. Affine Transformations of S-boxes

In this section we describe connections between DDT_+ of S-boxes obtained by affine transformations of inputs and/or outputs. We formalize these results in the following statements.

Claim 1. *Let $S_1(x)$ and $S_2(x)$ be n-bit S-boxes, $u \in \mathbb{Z}_{2^n}^*$, $w \in \mathbb{Z}_{2^n}$*

1) If $S_1(x) = S_2(ux + w)$, then

$$\forall\, a, b \in V_n \colon \delta_{S_1}(a, b) = \delta_{S_2}(ua, b)$$

2) If $S_1(x) = uS_2(x) + w$, then

$$\forall\, a, b \in V_n \colon \delta_{S_1}(a, b) = \delta_{S_2}(a, u^{-1}b)$$

Proof. The proofs of given statements are similar.

1) Consider the case: $S_1(x) = S_2(ux + w)$.

From the equation $S_1(x + a) = S_1(x) + b$ we can obtain

$$S_2(ux + ua + w) = S_2(ux + w) + b,$$

and, with substitution $y = ux + w$,

$$S_2(y + ua) = S_2(y) + b.$$

Therefore,

$$\delta_{S_1}(a, b) = \delta_{S_2}(ua, b).$$

Thus, in the case of affine transformation of the arguments, the DDT_+ of S-boxes are equivalent up to the permutation of rows.

2) Consider the case $S_1(x) = uS_2(x) + w$.

From the equation $S_1(x + a) = S_1(x) + b$ we similarly obtain

$$uS_2(x + a) + w = uS_2(x) + b + w,$$
$$uS_2(x + a) = uS_2(x) + b,$$

and, since u is invertible,

$$S_2(x + a) = S_2(x) + u^{-1}b.$$

Therefore,

$$\delta_{S_1}(a, b) = \delta_{S_2}(a, u^{-1}b).$$

Thus, when considering affine transformations of S-box outputs, we obtain equal DDT_+ up to column permutations. □

2.3. DDT Equivalence Classes from Affine Transformations

It follows from Claim 1 that affine transformations with $u = 1$, applied to inputs or outputs, do not change DDT_+ of corresponding S-box. In other words, any class of affine equivalency, described with transformations of form

$$S(x) \to S(x + a) + b,$$

where $a, b \in V_n$, are nested into some DDT equivalence class. We will refer to such transformations as forming an *affine shift class* and call the transformation itself an *affine shift*.

Surprisingly, there is another class of affine equivalency which doesn't change DDT_+. We describe it in next two statements.

Claim 2. *$S_1(x)$ and $S_2(x)$ are n-bit S-boxes connected with an affine transformation of the form*

$$S_1(x) = -S_2(-x + u) + w, \quad u, w \in V_n.$$

Then

$$\forall\, a, b \in V_n \colon \delta s_1(a, b) = \delta s_2(-a, -b),$$

where the negative numbers are naturally interpreted as $(-a) = 2^n - a$.

Proof. Since $(-1)^{-1} \equiv (-1) \pmod{2^n}$, this statement follows directly from Claim 1. □

Claim 3. *For every n-bit S-box S and every differential (a, b), $a, b \in \mathbb{Z}_{2^n}$ holds*

$$\delta_S(a, b) = \delta_S(-a, -b).$$

Proof. Similarly to proof of Claim 1, we have

$$S(x + a) = S(x) + b;$$
$$S(x) = S(x + a) - b;$$

introducing the substitution $y = x - a$, we obtain

$$S(y - a) = S(y) - b.$$

The first and the last equations are equal, so the numbers of their solutions, namely $\delta_S(a, b)$ and $\delta_S(-a, -b)$, are equal too. □

We will address the family of affine transformations considered in Claim 2 as *inverse affine shifts*, and the class generated by combinations of inverse and non-inverse affine shifts as the *class of equivalence affine shifts*.

From Claims 2 and 3 it follows that inverse affine shifts also doesn't change DDT_+ of corresponding S-box. Moreover, Claim 3 states that

DDT_+ of arbitrary S-box has a specific structure, namely it is symmetric with respect to the central point $a = b = 2^{n-1}$.

This specific structure has no analogues in $\mathrm{DDT}_\oplus$, because every $a \in V_n$ is an element of order 2 w.r.t. XOR, so $\ominus a \equiv a$; thus, $\mathrm{DDT}_\oplus$ doesn't have any predefined internal symmetries in general case. But for every other algebraic operation with elements of order higher than 2 such symmetry will appear. There is only one vector of order 2 w.r.t. addition modulo 2^n, namely $a = 2^{n-1}$, so differential $(2^{n-1}, 2^{n-1})$ is "self-symmetric" (and it is the central point of DDT_+).

Note that statement of Claim 3 holds for differentials with respect to any appropriate algebraic operation.

Further experiments (listed below) show that there are more complex transformations rather than affine that preserve DDT of S-boxes. These dependencies can be traced in S-boxes of high bit depth. Future research may focus on identifying and describing such transformations.

At last, we consider simple special case of DDT equivalence class representative.

Claim 4. *Let $S_1(x)$ be an arbitrary n-bit S-box belonging to the DDT equivalence class $\mathbb{G}$. Then $\exists\, S_2(x) \in \mathbb{G}$ such that $S_2(0) = 0$.*

Proof. Suppose $S_1(0) = a$ for some $a \in V_n$. Consider the S-box $S_2(x)$, which is connected to $S_1(x)$ by the following affine shift:

$$S_2(x) = S_1(x) - a.$$

It is obvious from the form $S_2(x)$ that $S_2(0) = 0$. It was also previously established that affine shifts do not change the DDT of an S-box. This means that $S_1(x), S_2(x) \in \mathbb{G}$, where $\mathbb{G}$ is the DDT equivalence class of S-boxes.

So, $S_2(x)$ is the desired S-box. □

Thus, it was established that at least one representative exists in each DDT equivalence class such that $S(0) = 0$. Further we will use this fact as a grounding for S-box recovering from DDT.

Finally, it should be noted that results of Claims 1-4 will hold for generalized S-box of form $S\colon \mathbb{Z}_p \to \mathbb{Z}_p$ and the addition modulo p. Such S-boxed are widely used in our days in cryptographic mappings for smart contracts and blockchain protocols.

3. Algorithm for Recovering an S-box from Its DDT_+

This section will describe an algorithm for recovering an S-box from its DDT_+. Note that we're considering recovery problem in case when there is no additional information about the recoverable S-box other than the DDT, and thus, we can recover the DDT equivalence class only. Note that along with the S-box all its affine and inverse affine shift are also recovered.

In the previous section, we studied the structure and properties of DDT equivalence classes. Thus, taking into account the above results, to solve the problem of recovering a DDT equivalence class, we can find only one representative of the class with a given DDT, which will be called the *generating element*, and on its basis, generate a class of affine equivalence shifts, which for S-boxes with bit size three or less coincides with the whole DDT equivalence class.

For the purpose of further description, let's establish a few notations. Let us denote the set of a-th row's positions of non-zero DDT elements as $\mathbb{A}_a$. Let's denote the set of checked assumptions of $S(a)$ for a of a given row as $\mathbb{T}_a$.

The main idea of this algorithm is to make step-by-step assumptions about the value of an S-box at a certain point and check the correctness of the assumption. Based on the check results, one can decide whether to guess the value of the S-box at the next point or to step back by recalculating previously made assumptions. With the described approach, it is possible to obtain much better results than with a naive search of all possible options, which is commonly called *full search* (or *brute force*) because we can discard entire sets of S-boxes that do not have a given DDT. This approach is not innovative and is known as *backtracking*.

To implement a backtracking-based algorithm, it is necessary to choose a rule based on which unsatisfactory sets of S-boxes can be identified and rejected.

For convenience and small optimization, due to the Claim 4, we set $S[0] = 0$. Then, we can make assumptions about the S-box $S(x)$ elements, starting with $a = 1$.

The pseudocode of the algorithm initialization for the case of an n-bit S-box described in a Fig. 1.

Input: DDT ddt corresponding to an unknown S-box
Output: The generating S-box $S\colon V_n \to V_n$

for $i \leftarrow 0$ **to** $n-1$ **do**
  $S(i) \leftarrow 0$
end
foreach *entry* $\delta_{i,j}$ *in* ddt **do**
  if $\delta_{i,j} \neq 0$ **then**
    $\mathbb{A}_i = \mathbb{A}_i \cup \{j\}$;
  end
end
for $i \leftarrow 0$ **to** $n-1$ **do**
  $\mathbb{T}_i \leftarrow \emptyset$
end
$\texttt{guess}(1, \{\mathbb{A}_a\}, \{\mathbb{T}_a\}, ddt)$;

Figure 1: Recovering the n-bit S-box from a DDT

After describing the initialization for the algorithm, we need to define the guessing procedure. We will address to it as `guess` procedure.

Let's consider the value of the derivative of the S-box $S(x)$ in the direction a, which is $S(x+a) - S(x)$. Further, we can consider the value of the derivative at point 0; relying on the strategy for S-box selection, we can use the fact that $S(0) = 0$, and thus it takes the following form: $S(a)$. In other words, it is the location of the point 0 of the S-box on any DDT row a. Then we can make an assumption of the $S(a)$ value based on the structure of DDT a-th row structure. Knowing the DDT of the S-box $S(x)$, we can conclude that the possible values of the derivative $S(a)$ at the a-th row are limited by the indices of the non-zero elements of the a-th row of the DDT $\mathbb{A}_a$.

We can organize the search in our backtracking algorithm as follows: we assume the value of the expression $S(a)$ and then check the validity of this assumption. To do this, after fixing 0 on the row a ($a \geq 1$), we need to check the following statement for all $k \in \overline{1, a-1}$:

$$\delta(a-k, S(a) - S(k)) \neq 0$$

Note that the possible values of k are obtained from the following considerations. Checking the value 0 is irrelevant since, assuming that $S(0) = 0$, in this case, $\delta(a, S(a)) \neq 0$ is always true given the strategy of choosing the element $S(a)$. If, for all k, the given DDT element is possible, meaning it has a non-zero value, we consider the assumption of the location 0 to be possible and move on to the next DDT row $a+1$. If for at least one k there is a DDT element $\delta(a-k, S(a)-S(k))$ equal to zero, then we consider this branch unsatisfying and re-guess the location of point 0. Then, in the case $\mathbb{A}_a = \mathbb{T}_a$, we should go back one step above and zero the set of tested variants.

The pseudocode of a recursive procedure `guess` that guesses the element of the S-box and returns a generative element of the DDT equivalence class is listed in the Fig. 2

Input: An S-box position a to guess;
$\{\mathbb{A}_a : 0 \leq a < n\}$; $\{\mathbb{T}_a : 0 \leq a < n\}$;
DDT ddt.
Output: The generating S-box $S\colon V_n \to V_n$ with DDT ddt.

if $\mathbb{A}_a = \mathbb{T}_a$ **then**
  $\mathbb{T}_a \leftarrow \emptyset$;
  $S(a) \leftarrow 0$;
  $\texttt{guess}(a-1, \{\mathbb{A}_a\}, \{\mathbb{T}_a\}, ddt)$;
end
foreach $x' \in \mathbb{A}_a$ **do**
  $\mathbb{T}_a = \mathbb{T}_a \cup \{x'\}$;
  if x' *is not in* $S(x)$ **then**
    $S(a) \leftarrow x'$;
    Break the loop;
  end
end
if $S(a) = 0$ **then**
  $\mathbb{T}_a \leftarrow \emptyset$;
  $\texttt{guess}(a-1, \{\mathbb{A}_a\}, \{\mathbb{T}_a\}, ddt)$;
end
if $a = n-1$ **then**
  if $DDT(S(x)) = ddt$ **then**
    return $S(x)$
  else
    $\mathbb{T}_a \leftarrow \emptyset$;
    $S(a) \leftarrow 0$;
    $\texttt{guess}(a-1, \{\mathbb{A}_a\}, \{\mathbb{T}_a\}, ddt)$;
  end
end
for $x \leftarrow 1$ **to** $a-1$ **do**
  if $\delta\big(a-x, S(a) - S(x)\big) = 0$ **then**
    $\texttt{guess}(a, \{\mathbb{A}_a\}, \{\mathbb{T}_a\}, ddt)$;
  end
end
$\texttt{guess}(a+1, \{\mathbb{A}_a\}, \{\mathbb{T}_a\}, ddt)$;

Figure 2: `guess` procedure

4. Complexity Estimations of Proposed Algorithm

In this section, we present some estimates of the time complexity of proposed algorithm. It should be noted that the upper asymptotic bound of the algorithm's complexity remains the same as in the case of full search, i.e., $O(n!)$. However, the proposed algorithm contains some optimizations that make the algorithm much faster. If we take into account the fact that the first element of the S-box is fixed to zero, the complexity is decreased to $(n-1)!$. If at the a-th step, the rule says that further searching does not make sense, then in this case, it is necessary to search for $(n-a)!$ fewer options.

Note that in the case of affine S-boxes, the algorithm completes almost instantly due to the specific structure of the tables of distribution of differentials of linear S-boxes since there is only one possible option for each element of S-box $S(a)$.

Since asymptotic estimates for the proposed algorithm are not descriptive enough, we present actual measurements on the personal computer described below as a function of the average recovery rate versus the bit-size of S-boxes in the Table 1, where the algorithm from this section is called backtracking.

This recovery algorithm is implemented in the Java programming language. The presented measurements were done on a personal computer (hereinafter referred to as PC) with the following computing resources: CPU 2.3 GHz, RAM 8 GB.

Note that it is not recommended to implement the described recovery algorithm recursively because this option is not an optimal implementation in terms of memory since, for S-boxes of large bit sizes, it leads to an overflow of the function call stack, as a result of which the algorithm terminates prematurely and does not produce any results. Thus, when recovering S-boxes of large bit sizes (more than 6) on the presented PC, the problem of function call stack overflow arises. Since any recursive algorithm can be implemented iteratively, this algorithm was not implemented recursively for the above optimizations.

Table 1
Algorithm running time in seconds depending on the bit size of the recovery S-box; BF – brute force, B – backtracking, * – means theoretical approximation

	2-bit	3-bit	4-bit	5-bit
BF	0.17	135.9	$1.7 \times 10^{11}*$	$1.8 \times 10^{34}*$
B	0.17	0.15	105	128

5. Properties of Three-bit S-box DDT Equivalence Classes

Due to limited computational resources, most of the experimental studies were conducted on S-boxes of small bit sizes. Most of the results were obtained when studying 3-bit S-boxes, which were the best candidate for these purposes because their number does not exceed the computing resources of a regular personal computer. At the same time, they have enough statistics to trace specific dependencies for further generalization.

Thus, it was found that not all DDT equivalence classes have the same cardinality. For example, all DDT equivalence classes of 3-bit S-boxes have only five possible cardinalities: 8, 16, 32, 64, and 128. We will name DDT equivalence class cardinality as *metaclass*.

This difference in cardinalities occurs because, within these classes, S-boxes have the same or similar structure. In some DDT equivalence classes, due to the structure of S-boxes, not all affine shifts and inverse shifts form unique S-boxes. For example, all linear S-boxes are always concentrated in the metaclass with the lowest cardinality value. In the case of 3-bit S-boxes, all linear S-boxes form their own metaclass with cardinality 4, and all DDT equivalence classes that lies outside this metaclass have cardinality 8.

Let's consider an arbitrary 3-bit linear S-box $S(x)$, which has the following form $S(x) = ax + b$, where the coefficient $a \in \mathbb{Z}_8^*$ and the coefficient $b \in \mathbb{Z}_8$. Thus, based on the Claims 1 and 2, as a result of all possible combinations of affine shifts of inputs and outputs and all possible such combinations in the case of inverse affine shifts, the power of the resulting class is 8 S-boxes. The power of this metaclass is explained by the properties of affine shifts and the power of the multiplicative group $\mathbb{Z}_8^*$.

Note that for 3-bit S-boxes, the affine shift equivalence class is ultimately the same as the DDT equivalence class.

However, this is not true for all bit sizes of S-boxes. For example, for 4-bit S-boxes, there are cases where two S-boxes belong to the same DDT equivalence class but are not affine equivalent. An example is the following 4-bit S-boxes:

$$S_1(x) = (0, 5, 3, 14, 6, 4, 12, 2, 8, 13, 10, 1, 9, 7, 15, 11),$$
$$S_2(x) = (8, 15, 4, 9, 7, 13, 5, 3, 11, 6, 12, 1, 14, 10, 2, 0).$$

The S-boxes $S_1(x)$ and $S_2(x)$ are the generating elements for two classes of non-overlapping affine equivalence shifts, each of 512 S-boxes.

We calculated the number of such S-boxes in the DDT equivalence class depending on the power of the metaclass. The results are shown in Table 2, where $|M|$ is the value of the metaclass and $\#S(x)$ is the number of S-boxes:

Table 2
Dependence of the number of S-boxes in the DDT equivalence class that $S(0) = 0$ on the value of the metaclass $|M|$

$\|M\|$	8	16	32	64	128
$\#S(x)$	1	2	4	8	16

Among other things, it has been noticed that the larger the power of the metaclass that includes the DDT equivalence class, the fewer zero elements in the DDT, and, as a result, the greater the frequency of occurrence of small elements among the values of DDT cells.

Conclusions

This paper focuses on the problem of recovering an S-box based on its differential distribution table (DDT_+) with respect to addition modulo 2^n. This problem can be solved only within the so-called DDT equivalence class, since the correspondence between S-boxes and DDTs is not bijective. The structure of DDT_+ for affine S-boxes has been described. Affine transformations of S-boxes were considered; their effect on the structure of DDT_+ was formulated. It was discovered that every DDT_+ has a specific internal symmetry with respect to a central point, which is not characteristic for $DDT_\oplus$. Based on this fact, two subsets of affine transformations that preserve structure of DDT, namely affine shifts and inverse affine shifts, were found.

A backtracking-based algorithm for recovering S-box from DDT_+ was designed based on considered properties. The time efficiency of the algorithm is analyzed. Although the algorithm is asymptotically equivalent to the brute force approach, it shows much better results in practice, as proven empirically and compared with the brute-force-based analog. It should be noted that it is impossible to adapt the known algorithm for S-box recovery from $DDT_\oplus$, proposed by Dunkelman and Huang, since this algorithm is based on connections between XOR-differentials and linear approximations with respect to XOR. Differentials with respect to modular addition have no connection with XOR LAT. Thus, it may be interesting to investigate the possible connections between differentials and generalized linear approximations from the point of view of S-box recovery.

Finally, the paper presents empirical observations for the structure of DDT_+-equivalence classes of three-bit S-boxes, describing how DDT_+-equivalence classes are distributed in dependence of underlying S-boxes. It was also discovered that (inverse) affine shifts cannot form an entire DDT_+ equivalence class, even for some four-bit S-boxes. Future research can concentrate on searching and formalizing other transformations that preserve the structure of DDT_+, which allows one to describe DDT equivalence classes more efficiently.

References

[1] E. Biham and A. Shamir, “Differential Cryptanalysis of DES-Like Cryptosystems,” Journal of Cryptology, no. 4, 1991. DOI: 10.1007/BF00630563.

[2] M. Matsui, “Linear Cryptanalysis Method for DES Cipher,” in EUROCRYPT 1993 (T. Helleseth, ed.), vol. 765 of Lecture Notes in Computer Science, pp. 386–397, Heidelberg: Springer, 1994. DOI: 10.1007/3-540-48285-7_33.

[3] K. Nyberg, “Linear Approximation of Block Ciphers,” in EUROCRYPT 1994 (A. De Santis, ed.), vol. 950 of Lecture Notes in Computer Science, pp. 439–

444, Heidelberg: Springer, 1995. DOI: 10.1007/BFb0053460.

[4] K. Nyberg and L. R. Knudsen, "Provable Security Against a Differential Attack," Journal of Cryptology, vol. 8, pp. 27–37, Dec. 1995. DOI: 10.1007/BF00204800.

[5] Data Encryption Standard Committee, "Data Encryption Standard," Tech. Rep. 112, Federal Information Processing Standards Publication, 1999. URL: https://csrc.nist.gov/pubs/fips/46-3/final.

[6] A. Biryukov and L. Perrin, "On Reverse-Engineering S-Boxes With Hidden Design Criteria or Structure," in Advances in Cryptology – CRYPTO 2015, 2015. DOI: 10.1007/978-3-662-47989-6_6.

[7] National Security Agency, "SKIPJACK and KEA Algorithm Specifications," tech. rep., National Security Agency, 1998. URL: https://csrc.nist.rip/CSRC/media/Projects/Cryptographic-Algorithm-Validation-Program/documents/skipjack/skipjack.pdf.

[8] L. Perrin, "Partitions in the S-Box of Streebog and Kuznyechik," IACR Transactions on Symmetric Cryptology, pp. 302–329, 2019. DOI: 10.13154/tosc.v2019.i1.302-329.

[9] L. Perrin, "Streebog and Kuznyechik: Inconsistencies in the Claims of Their Designers," in Proceedings of the 105th IETF Meeting, 2019. URL: https://datatracker.ietf.org/meeting/105/materials/slides-105-cfrg-streebog-and-kuznyechik-00.

[10] L. Perrin and X. Bonnetain, "Russian Style (Lack of) Randomness." Preprint on HAL, 2019. URL: https://hal.archives-ouvertes.fr/hal-02396792.

[11] B. Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)," in Fast Software Encryption (R. Anderson, ed.), (Berlin, Heidelberg), pp. 191–204, Springer Berlin Heidelberg, 1994. DOI: 10.1007/3-540-58108-1_24.

[12] A. Bar-On, E. Biham, O. Dunkelman, and N. Keller, "Efficient Slide Attacks," Journal of Cryptology, vol. 31, no. 3, pp. 641–670, 2018. DOI: s00145-017-9266-8.

[13] C. Boura, A. Canteaut, J. Jean, and V. Suder, "Two Notions of Differential Equivalence on S-Boxes," Des. Codes Cryptography, vol. 87, pp. 185–202, Mar. 2019. DOI: 10.1007/s10623-018-0496-z.

[14] P. Hawkes and L. O'Connor, "XOR and Non-XOR Differential Probabilities," in Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, vol. 1592 of Lecture Notes in Computer Science, pp. 272–285, Springer, 1999. DOI: 10.1007/3-540-48910-X_19.

[15] S. Yakovliev and V. Bakhtigozin, "Asymptotic Distributions for S-Box Heterogeneous Differential Probabilities," Theoretical and Applied Cybersecurity, vol. 1, no. 1, pp. 37–41, 2019. DOI: 10.20535/tacs.2664-29132019.1.169029.

[16] O. Dunkelman and S. Huang, "Reconstructing an S-Box From Its Difference Distribution Table," IACR Transactions on Symmetric Cryptology, pp. 193–217, Jun. 2019. DOI: 10.13154/tosc.v2019.i2.193-217.

[17] C. Blondeau, G. Leander, and K. Nyberg, "Differential-Linear Cryptanalysis Revisited," Journal of Cryptology, vol. 30, pp. 859–888, Jul 2017. DOI: s00145-016-9237-5.

[18] F. Chabaud and S. Vaudenay, "Links Between Differential and Linear Cryptanalysis," Advances in Cryptology — EUROCRYPT'94, pp. 356–365, 1995. DOI: 10.1007/BFb0053450.

[19] C. Blondeau and K. Nyberg, "New Links Between Differential and Linear Cryptanalysis," Advances in Cryptology – EUROCRYPT 2013, pp. 388–404, 2013. DOI: 10.1007/978-3-642-38348-9_24.