Cryptography and Communications (2022) 14:849–874
https://doi.org/10.1007/s12095-021-00547-2

Differential uniformity and linearity of S-boxes by multiplicative complexity

Yongjin Jeon[1] · Seungjun Baek[1] · Hangi Kim[1] · Giyoon Kim[1] · Jongsung Kim[2]

Received: 28 July 2021 / Accepted: 25 October 2021 /Published online: 13 January 2022

Abstract

A cryptographic primitive with low multiplicative complexity (MC) makes various applications efficient, but it may lead to cryptographic vulnerabilities. To find a trade-off between cryptographic resistance and MC, we propose a new tool called *A-box*, which is constructed using AND gates. In this paper, we prove several important properties of A-boxes, which provide the theoretical lower bounds of differential uniformity and linearity of corresponding S-boxes by MC. Specifically, we show that the differential uniformity (resp. linearity) of an (n, m)-bit S-box is at least 2^{n-l}, where its MC is $\lfloor \frac{n-1}{2} \rfloor + l$ (resp. $m - 1 + l$). Furthermore, we develop an algorithm to find S-boxes with differential uniformity equal to the bounds with respect to their MC. We improve the algorithm previously proposed by Zajac and Jókay (Cryptogr. Commun. **6**(3), 255–277, 2014), which is applicable only to S-boxes of size lower than 5 bits, whereas ours can run on larger-sized S-boxes. We found a bijective (8, 8)-bit S-box with differential uniformity 16, linearity 128, and 8 nonlinear gates: this has better cryptographic security than the SKINNY S-box with differential uniformity 64, linearity 128, and 8 nonlinear gates. We believe that our results provide a better understanding of the relationship between cryptographic resistance and MC of S-boxes.

Keywords A-box · S-box · Multiplicative complexity · Differential uniformity · Linearity

✉ Jongsung Kim
jskim@kookmin.ac.kr

Yongjin Jeon
idealtop18@kookmin.ac.kr

Seungjun Baek
hellosj3@kookmin.ac.kr

Hangi Kim
tiontta@kookmin.ac.kr

Giyoon Kim
gi0412@kookmin.ac.kr

[1] Department of Financial Information Security, Kookmin University, Seoul, Korea

[2] Department of Information Security, Cryptology and Mathematics / Department of Financial Information Security, Kookmin University, Seoul, Korea

Mathematics Subject Classification (2010) 06E30 · 94A60 · 94C11

1 Introduction

Reducing the number of nonlinear gates when implementing cryptographic primitives, especially using block ciphers, is an important concern in various fields [2, 34] such as post quantum zero-knowledge (PQZK) proofs, multi-party computation (MPC) protocols [23], fully homomorphic encryption (FHE), and side-channel attacks (SCAs) [29]. In PQZK proofs based on "MPC-in-the-head," the signature size increases proportionally to the number of nonlinear gates used in the underlying block cipher [18]. The computational complexity in the MPC protocol based on Yao's garbled circuit [32, 35] using the free-XOR technique depends on the number of nonlinear gates [30]. For FHE, an AND gate in the underlying block cipher is more expensive than an XOR gate and may generate noise during calculations [26]. Furthermore, the smaller the number of nonlinear gates required to implement the block cipher, the more efficient the SCA countermeasure technique that can be implemented. This is because the cost of Boolean masking increases sharply when it is applied to nonlinear gates rather than linear ones. Therefore, cryptographic primitives that can be implemented using a small number of nonlinear gates have many advantages in various applications.

The multiplicative complexity (MC) of a vectorial Boolean function is the minimal number of AND gates needed to implement it over the basis {AND, XOR, NOT} (called an *XOR-AND circuit* [13]). As stated above, obtaining a primitive with low MC is important for performance; however, such a primitive may have potential cryptographic vulnerabilities. For example, block ciphers based on an S-box with low MC may be vulnerable to cryptanalysis [16].

In this paper, we discuss security against differential and linear cryptanalyses (DC and LC, respectively) in an S-box based on its MC [6, 31]. As DC and LC are the most influential techniques among block cipher cryptanalyses [27], the *differential uniformity* and *linearity* of an S-box are considered the most important cryptographic properties. For an efficient and secure cryptographic primitive design, it is essential to clarify the trade-off between the MC and differential uniformity/linearity of an S-box.

The lower bounds of differential uniformity and linearity independent of MC have been revealed. While the differential uniformity has an obvious lower bound [5], there are several bounds of linearity: the covering radius bound, Sidelnikov–Chabaud–Vaudenay's bound [17], and three types of linearity bounds in [16]. Zajac and Jókay investigated the MC of all the affine classes of bijective $(4, 4)$-bit S-boxes [37]. Their investigation used an *expansion-compression* method to accurately calculate the MC of S-boxes. This leads to the following two facts about bijective $(4, 4)$-bit S-boxes.

– A bijective $(4, 4)$-bit S-box with optimal differential uniformity 4 has MC at least 4.
– A bijective $(4, 4)$-bit S-box with optimal linearity 8 has MC at least 4.

Božilov et al. investigated the MC of all the affine classes of quadratic $(5, 5)$-bit S-boxes [14]. There are many open issues for the existence of S-boxes, such as $(5, 5)$-bit S-boxes with MC 6 and differential uniformity 2 (this study shows that there is no such $(5, 5)$-bit S-box, which will be explained below). In [10], Boyar and Find found that the number of AND gates, linearity, and the length of the shortest linear code are related. The bounds they found were proved in the $\Sigma\Pi\Sigma$ circuit, which consists of sequential XOR, AND, and XOR layers.

Contributions In this paper, we present a new cryptographic tool called A-box, motivated by the work of Zajac and Jókay [37]. While their *expansion function* consists of AND and identity parts, we restrict it to only the AND part to define the A-box. Specifically, an S-box can be divided into an A-box (as a nonlinear function) and linear functions. We used several properties of A-boxes to prove the theoretical lower bounds of differential uniformity and linearity of the corresponding S-boxes by MC.

We show that the differential uniformity (resp. linearity) of an (n, m)-bit S-box is at least 2^{n-l}, where its MC is $\lfloor \frac{n-1}{2} \rfloor + l$ (resp. $m - 1 + l$). Furthermore, we develop an algorithm to search for A-boxes with differential uniformity equal to our bounds with respect to their MC. Those A-boxes lead to S-boxes with the same differential uniformity as theirs. Table 1 presents the lowest differential uniformity we found, within (n, n)-bit S-boxes with MC k where $3 \leq n \leq 8$ and $1 \leq k \leq 7$. Our investigation reveals the following properties:

- A (5, 5)-bit S-box with differential uniformity 2 has MC at least 7.
- A (6, 6)-bit S-box with differential uniformity 4 has MC at least 7.
- An (8, 8)-bit S-box with differential uniformity 32 has MC at least 7.

Compared to differential uniformity, a relatively large MC is required to reduce linearity. The new S-boxes constructed by our method are compared with existing ones in Table 2. Either they have better differential uniformity or linearity than those of existing S-boxes with respect to the same nonlinear gates or less, or they achieve our lower bounds of both differential uniformity and linearity with respect to MC which is the same as implemented nonlinear gates. The detailed implementation codes are provided in Appendix A.

Organization The remainder of this paper is organized as follows. In Section 2, we introduce the expansion-compression method of [37], based on which we define the A-box, and we observe several properties of A-boxes. In Section 3, we use them to prove the theoretical lower bounds of the differential uniformity and linearity of S-boxes by MC. In Section 4, we present an S-box search process in terms of MC and low differential uniformity. We conclude this paper in Section 5.

Preliminary The following notations and definitions are used throughout this paper.

Table 1 Differential uniformity of (n, n)-bit S-boxes found by our experiments

n	Multiplicative Complexity						
	1	2	3	4	5	6	7
3	**8**	**4**	**2**	**2**	**2**	**2**	**2**
4	**16**	**8**	**4**	**2**	**2**	**2**	**2**
5	**32**	**32**	**16**	**8**	**4**	4	**2**
6	**64**	**64**	**32**	**16**	**8**	8	4
7	**128**	**128**	**128**	**64**	**32**	**16**	16
8	**256**	**256**	**256**	**128**	**64**	64	32

*The numbers in bold are equal to our lower bounds of differential uniformity.

Table 2 Comparision of (n, n)-bit S-boxes*

S-box	n	Differential uniformity	Linearity	Lower bound of MC	#NL gates	Bijectivity	Reference
S-box of [8]	3	4 (LB)	8 (LB)	2	2	YES	[8, 37]
S-box of CTC	3	2 (LB)	4 (LB)	3	3	YES	[19, 20]
S-box of [8]	4	8	16 (LB)	3	3	YES	[8, 37]
S-box of [1]	4	4 (LB)	16 (LB)	3	3	NO	[1]
S-box of PRESENT	4	4	8 (LB)	4	4	YES	[9, 21]
S-box of [1]	4	2 (LB)	8 (LB)	4	4	NO	[1]
S-box of [14]	5	16	32 (LB)	4	4	YES	[14]
Our S-box	5	8 (LB)	32 (LB)	4	4	YES	Listing 1
S-box of [14]	5	4 (LB)	16 (LB)	5	5	YES	[14]
S-box of PRIMATEs	5	2 (LB)	8 (LB)	7	7	YES	[3, 14, 33]
Our S-box	6	8 (LB)	32 (LB)	6	6	YES	Listing 2
Our S-box	6	4 (LB)	16 (LB)	7	7	NO	Listing 3
S-box of [28]	6	4	16	7	9	YES	[28]
S-box of [7]	6	2	16	8	9	NO	[7]
Our S-box	7	32 (LB)	128 (LB)	5	5	YES	Listing 4
Our S-box	7	4	32	9	10	NO	Listing 5
S-box of [28]	7	8	32	8	11	YES	[28]
S-box of SKINNY	8	64	128 (LB)	8	8	YES	[25]
Our S-box	8	16 (LB)	128 (LB)	8	8	YES	Listing 6
Our S-box	8	8	64	9	10	NO	Listing 7
S-box of PIPO	8	16	64	9	11	YES	[28]
S-box of Fantomas	8	16	64	9	11	YES	[4]
S-box of LILLIPUT	8	8	64	9	12	YES	[1]
S-box of AES	8	4	32	10	32	YES	[11, 22]

*We use ‘LB’ to denote that an S-box achieves our lower bound of differential uniformity or linearity by MC (same as implemented nonlinear(NL) gates)

(n, m)-bit S-box
A vectorial Boolean function with $\mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$.

Multiplicative Complexity (MC)
The MC of S-box S is the minimum number of AND gates necessary to implement the S-box as an XOR-AND circuit. The notation is $c_\wedge(S)$.

DDT
$\delta_S(\Delta a, \Delta b) = \#\{\mathbf{x} \in \mathbb{F}_2^n | S(\mathbf{x}) \oplus S(\mathbf{x} \oplus \Delta a) = \Delta b\}$, for (n, m)-bit S-box S.

Differential uniformity
$\delta(S) = \max\limits_{\Delta a \neq 0, \Delta b} \#\delta_S(\Delta a, \Delta b)$.

LAT
$\mathcal{L}_S(\Lambda a, \Lambda b) = \sum\limits_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{\Lambda b \cdot S(\mathbf{x}) \oplus \Lambda a \cdot \mathbf{x}}$, for (n, m)-bit S-box S.

Linearity
$\mathcal{L}(S) = \max\limits_{\Lambda a, \Lambda b \neq 0} |\mathcal{L}_S(\Lambda a, \Lambda b)|$.

For the convenience of notation, an n-dimensional vector is considered to be a column-matrix when performing matrix multiplication. Therefore, it is defined as follows.

$$\mathbb{x} = (x_{n-1}, \cdots, x_0) = \begin{pmatrix} x_{n-1} \\ \vdots \\ x_0 \end{pmatrix} \in \mathbb{F}_2^n$$

Any linear function can be expressed as the multiplication of a matrix. We indicate the matrix related to the linear function in the subscript of $\mathcal{T}$. For example, the matrix expression of the linear function $\mathcal{T}_M : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is an $m \times n$ matrix M and we denote $\mathcal{T}_M(\mathbb{x}) = M\mathbb{x}$ for $\mathbb{x} \in \mathbb{F}_2^n$. The dot product of $\mathbb{x}$ and $\mathbb{y}$ is denoted by $\mathbb{x} \cdot \mathbb{y}$, and its matrix expression is $\mathbb{x}^T\mathbb{y}$. I_n is the $n \times n$ identity matrix. The k LSBs and k MSBs of $\mathbb{x} = (x_{n-1}, \cdots, x_0)$ are expressed as $\mathbb{x}|_k = (x_{k-1}, \cdots, x_0)$ and $\mathbb{x}|^k = (x_{n-1}, \cdots, x_{n-k})$, respectively. $0^{(n)}$ is the zero vector of $\mathbb{F}_2^n$, and $0^{(n\times m)}$ is the zero matrix of size $n \times m$.

2 A-box and its equivalence classes

2.1 Definition of A-box

Consider an XOR-AND circuit of an S-box. The AND gates on the circuit are made as per the following rule. The inputs of the i^{th} AND gate are calculated as the linear combinations of three types of bits: the inputs of the S-box, the outputs of the previous $i - 1$ AND gates, and the constant 1. In addition, the outputs of the circuit are calculated as the linear combinations of three types of bits: the inputs of the circuit, the outputs of the sufficient number of AND gates, and the constant 1. In particular, an S-box with $0 \mapsto 0$ can be constructed with a circuit without using NOT gates. Zajac and Jókay constructed an XOR-AND circuit of such an S-box as the expansion–compression method for expansion function and compression function [37]. As the expansion function is helpful for understanding the concept of the A-box, we explain it first.

For two vectors $\mathbb{b}_0, \mathbb{b}_1 \in \mathbb{F}_2^i$ and an input $\mathbb{u} \in \mathbb{F}_2^i$, the expansion function E_i is defined as follows.

$$\begin{aligned} E_i &: \mathbb{F}_2^i \to \mathbb{F}_2^{i+1}, \\ E_i(\mathbb{u}) &= ((\mathbb{b}_0 \cdot \mathbb{u})(\mathbb{b}_1 \cdot \mathbb{u}))||\mathbb{u}. \end{aligned}$$

That is, E_i is a function that concatenates the output of one AND gate to the MSB of the input $\mathbb{u}$.. The $\mathbb{b}_j \cdot \mathbb{u}$ for $j \in \{0, 1\}$ means a linear combination of input bits. Zajac and Jókay constructed the following function with $\mathbb{F}_2^n \to \mathbb{F}_2^{n+k}$ by composing k expansion functions from E_n to E_{n+k-1}.

$$E_{n+k-1} \circ E_{n+k-2} \circ \cdots \circ E_n. \tag{1}$$

This function (1) describes a circuit using k AND gates. By setting $\mathbb{b}_0$ and $\mathbb{b}_1$ of each E_{n+i}, the linear combination of the inputs of each i^{th} AND gate will be determined. We define $\mathbb{b}_j$ as a *partner vector* and the $2k$-tuple that consists of all $\mathbb{b}_j$ in order as a *partner tuple*. By applying a linear function $\mathcal{T}_C : \mathbb{F}_2^{n+k} \to \mathbb{F}_2^m$ (which is called the compression function in [37]) to function (1), we obtain the following function with $\mathbb{F}_2^n \to \mathbb{F}_2^m$,

$$\mathcal{T}_C \circ E_{n+k-1} \circ E_{n+k-2} \circ \cdots \circ E_n. \tag{2}$$

By Lemma 6 of [37], function (2) expresses any (n, m)-bit S-box with $0 \mapsto 0$. In this method, a circuit with $k = C_\wedge(S)$ always exists. By XORing function (2) with a constant $\mathbb{v} \in \mathbb{F}_2^m$, we obtain an (n, m)-bit S-box expression with no necessary conditions.

$$\mathcal{T}_C \circ E_{n+k-1} \circ E_{n+k-2} \circ \cdots \circ E_n \oplus \mathbb{v}. \tag{3}$$

The output of function (1) consists of a k-bit (called the AND part) and n-bit (called the identity part); the latter n-bit represents the original input. The AND part becomes a type of (n, k)-bit S-box, which we define as an *A-box* (refer to Fig. 2). Naturally, the MC of an (n, k)-bit A-box is less than or equal to k. To avoid any confusing context, we denote an S-box as S and the corresponding A-box as S_A. The mathematical definition of the A-box is given in Definition 1.

Definition 1 Let $\mathbb{x} = (x_{n-1}, \cdots, x_0) \in \mathbb{F}_2^n$ and $\mathbb{y} = (y_{k-1}, \cdots, y_0) \in \mathbb{F}_2^k$ be the input and output, respectively, of an (n, k)-bit S-box S_A. For $2k$ vectors $\mathbb{b}_0, \cdots, \mathbb{b}_{2k-1}$ that satisfy the following inductive properties, S_A is called an **(n,k)**-bit A-box.

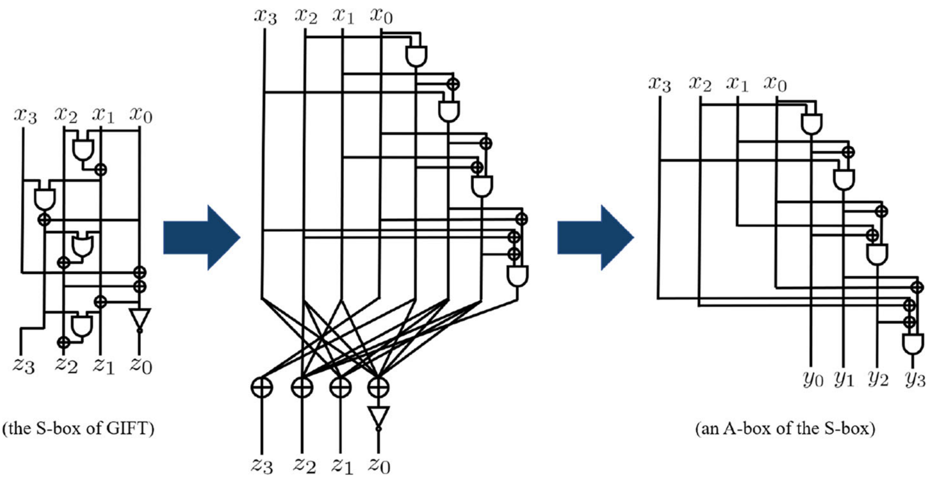
- $y_0 = (\mathbb{b}_0 \cdot \mathbb{x})(\mathbb{b}_1 \cdot \mathbb{x})$.
- For $1 \leq i < k$, $y_i = (\mathbb{b}_{2i} \cdot (y_{i-1}|| \cdots ||y_0||\mathbb{x}))(\mathbb{b}_{2i+1} \cdot (y_{i-1}|| \cdots ||y_0||\mathbb{x}))$.

For an (n, k)-bit A-box S_A, $\mathbb{b}_{2i}$ and $\mathbb{b}_{2i+1}$ are called i^{th} **partner vectors** for all i, and $(\mathbb{b}_{2k-1}, \cdots, \mathbb{b}_0)$ is called the **partner tuple** of S_A.

A-boxes can be taken from existing S-boxes. For example, Fig. 1 shows an A-box construction taken from the GIFT S-box.

We often write the ANF of an A-box S_A as $(f_{k-1}, \cdots, f_0)$. The ANF function f_i satisfies $c_\wedge(f_i) \leq i + 1$. We also denote the components of the i^{th} partner vectors as follows.

$$\begin{aligned} \mathbb{b}_{2i} &= (b_{2i}^{n+i-1}, \cdots, b_{2i}^0) \in \mathbb{F}_2^{n+i}, \\ \mathbb{b}_{2i+1} &= (b_{2i+1}^{n+i-1}, \cdots, b_{2i+1}^0) \in \mathbb{F}_2^{n+i}. \end{aligned}$$



(the S-box of GIFT)

(an A-box of the S-box)

Fig. 1 Process of obtaining an A-box from the implementation of GIFT S-box

Note that an A-box can have multiple partner tuples. For example, the output of the i^{th} AND gate of the A-box is invariant even though $\mathbb{b}_{2i}$ and $\mathbb{b}_{2i+1}$ are swapped.

2.2 Equivalence classes

For an S-box S and three affine functions A, B, and C, $(B \circ S \circ A) \oplus C$ is called an *extended affine transformation (EA transformation)*. Let $S' = (B \circ S \circ A) \oplus C$; then, S is the *extended affine equivalent (EA equivalent) to S'*. Furthermore, the set of all S-boxes that are EA equivalent to S is called the *extended affine equivalence class (EA class) of S* [15]. Similarly, linear equivalence is defined by $S' = B \circ S \circ A$ for two linear functions A and B. It is known that MC, differential uniformity, and linearity are invariant under EA transformation [15, 36].

Consider an (n, m)-bit S-box S. By using the form (3), S is given as

$$S(\mathbb{x}) = \mathcal{T}_C(S_A(\mathbb{x})||\mathbb{x}) \oplus \mathbb{v} \tag{4}$$

for an (n, k)-bit A-box S_A. As $\mathcal{T}_C$ is a linear function, we can decompose $\mathcal{T}_C(\mathbb{x}) = \mathcal{T}_{N'}(\mathbb{x}|^k) \oplus \mathcal{T}_N(\mathbb{x}|_n)$ into two linear functions, i.e., $\mathcal{T}_{N'} : \mathbb{F}_2^k \rightarrow \mathbb{F}_2^m$ and $\mathcal{T}_N : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$. Accordingly, function (4) is transformed as follows.

$$S(\mathbb{x}) = \mathcal{T}_C(S_A(\mathbb{x})||\mathbb{x}) \oplus \mathbb{v} = \mathcal{T}_{N'}(S_A(\mathbb{x})) \oplus \mathcal{T}_N(\mathbb{x}) \oplus \mathbb{v} = (\mathcal{T}_{N'} \circ S_A)(\mathbb{x}) \oplus \mathcal{T}_N(\mathbb{x}) \oplus \mathbb{v}.$$

The matrix N' can be decomposed by matrix multiplication of the $m \times k$ matrix M in reduced row echelon form (RREF) and the invertible $m \times m$ matrix D. Now, we obtain

$$\begin{aligned} S(\mathbb{x}) &= (\mathcal{T}_{N'} \circ S_A)(\mathbb{x}) \oplus \mathcal{T}_N(\mathbb{x}) \oplus \mathbb{v} \\ &= (\mathcal{T}_D \circ \mathcal{T}_M \circ S_A)(\mathbb{x}) \oplus \mathcal{T}_N(\mathbb{x}) \oplus \mathbb{v}. \end{aligned} \tag{5}$$

This is depicted in Fig. 2. Note that in this figure each rectangle box located to the right of the partner vectors is a linear operator that generates two inputs of the AND gate. Specifically, for the i^{th} AND gate, the corresponding rectangle box computes the two inputs $\mathbb{b}_{2i} \cdot (y_{i-1}|| \cdots ||y_0||\mathbb{x})$ and $\mathbb{b}_{2i+1} \cdot (y_{i-1}|| \cdots ||y_0||\mathbb{x})$.

As $\mathcal{T}_D(\mathbb{x})$ and $\mathcal{T}_N(\mathbb{x}) \oplus \mathbb{v}$ are affine functions, S and $\mathcal{T}_M \circ S_A$ are EA equivalent. Therefore, the following theorem holds.

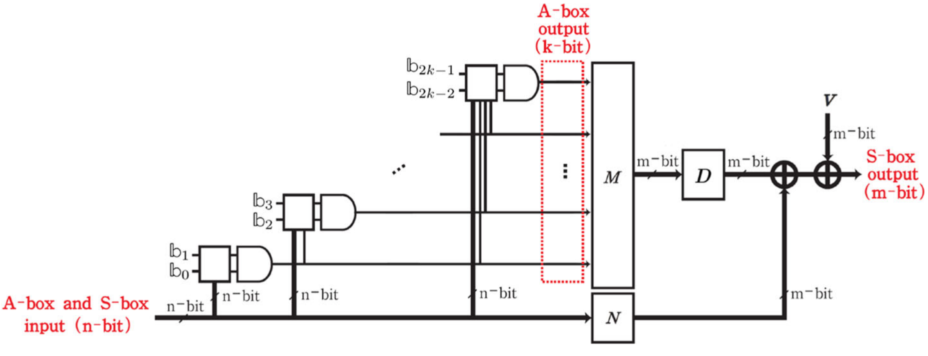


Fig. 2 Input and output of an (n, m)-bit S-box and an (n, k)-bit A-box

Theorem 1 *For any (n, m)-bit S-box S and $k \geq c_{\wedge}(S)$, there exist an $m \times k$ matrix M in RREF and an (n, k)-bit A-box S_A such that $\mathcal{T}_M \circ S_A$ is EA equivalent to S. If $k = c_{\wedge}(S)$, S_A is called* ***suitable*** *for S.*

Next, we consider the linear equivalence of an A-box. Let L and L' be linear functions on $\mathbb{F}_2^n$ and $\mathbb{F}_2^m$, respectively. Then, $\mathcal{T}_{L'} \circ S_A \circ \mathcal{T}_L$ is linear equivalent to S_A. Recall that our ultimate goal is to get the S-box, not the A-box. By substituting S_A for $\mathcal{T}_{L'} \circ S_A \circ \mathcal{T}_L$, the $\mathcal{T}_D \circ \mathcal{T}_M \circ S_A$ in (5) is changed to

$$\mathcal{T}_D \circ \mathcal{T}_M \circ \mathcal{T}_{L'} \circ S_A \circ \mathcal{T}_L = \mathcal{T}_{D'} \circ \mathcal{T}_{M'} \circ S_A \circ \mathcal{T}_L,$$

for $\mathcal{T}_{D'} \circ \mathcal{T}_{M'} = \mathcal{T}_D \circ \mathcal{T}_M \circ \mathcal{T}_{L'}$ where D' is an invertible $m \times m$ matrix, and M' is an $m \times k$ matrix in RREF. Thus, we just consider $\mathcal{T}_L$.

For any $n \times n$ invertible matrix $\mathcal{T}_L$, the $S_A \circ \mathcal{T}_L$ is *well defined* as A-box by the following theorem. This fact will be effectively used to search for linear classes of A-boxes in our algorithm (cf. Section 4).

Theorem 2 *For an (n, k)-bit A-box S_A and a linear permutation $\mathcal{T}_L : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$, let $S_A' = S_A \circ \mathcal{T}_L$, which is an A-box linear equivalent to S_A. If $(\mathbb{b}_{2k-1}, \cdots, \mathbb{b}_0)$ is a partner tuple of S_A, then the following $(\mathbb{b}'_{2k-1}, \cdots, \mathbb{b}'_0)$ is a partner tuple of S_A' such that*

$$\begin{cases} \mathbb{b}'_{2i} = \mathcal{T}_{L_i}(\mathbb{b}_{2i}), \\ \mathbb{b}'_{2i+1} = \mathcal{T}_{L_i}(\mathbb{b}_{2i+1}), \end{cases}$$

where $L_i = \begin{pmatrix} I_i & 0^{(i \times n)} \\ 0^{(n \times i)} & L \end{pmatrix}$ *for* $0 \leq i < k$.

Proof Suppose S_A has a partner tuple $(\mathbb{b}_{2k-1}, \cdots, \mathbb{b}_0)$. Furthermore, let $S_A(\mathbb{x}) = (y_{k-1}, \cdots, y_0)$ and $S_A \circ \mathcal{T}_L(\mathbb{x}) = (z_{k-1}, \cdots, z_0)$. As the definition of A-box is inductive, we treat z_0 first. We obtain

$$\begin{aligned} z_0 &= (\mathbb{b}_0 \cdot \mathcal{T}_L(\mathbb{x}))(\mathbb{b}_1 \cdot \mathcal{T}_L(\mathbb{x})) \\ &= (\mathcal{T}_{L^T}(\mathbb{b}_0) \cdot \mathbb{x})(\mathcal{T}_{L^T}(\mathbb{b}_1) \cdot \mathbb{x}). \end{aligned}$$

$\mathcal{T}_{L^T}(\mathbb{b}_0)$ and $\mathcal{T}_{L^T}(\mathbb{b}_1)$ become new partner vectors. We now denote $\mathbb{b}'_0$ and $\mathbb{b}'_1$ as follows.

$$\mathbb{b}'_0 = \mathcal{T}_{L^T}(\mathbb{b}_0), \; \mathbb{b}'_1 = \mathcal{T}_{L^T}(\mathbb{b}_1).$$

Then, we obtain

$$\begin{aligned} z_1 &= (\mathbb{b}_2 \cdot (z_0 || \mathcal{T}_L(\mathbb{x})))(\mathbb{b}_3 \cdot (z_0 || \mathcal{T}_L(\mathbb{x}))) \\ &= (\mathbb{b}_2 \cdot \mathcal{T}_{L_1}(z_0 || \mathbb{x}))(\mathbb{b}_3 \cdot \mathcal{T}_{L_1}(z_0 || \mathbb{x})) \\ &= (\mathcal{T}_{L_1^T}(\mathbb{b}_2) \cdot (z_0 || \mathbb{x}))(\mathcal{T}_{L_1^T}(\mathbb{b}_3) \cdot (z_0 || \mathbb{x})). \end{aligned}$$

where $L_1 = \begin{pmatrix} 1 & 0^{(1 \times n)} \\ 0^{(n \times 1)} & L \end{pmatrix}$. We denote $\mathbb{b}'_2$ and $\mathbb{b}'_3$ as follows.

$$\mathbb{b}'_2 = \mathcal{T}_{L_1^T}(\mathbb{b}_2), \qquad \mathbb{b}'_3 = \mathcal{T}_{L_1^T}(\mathbb{b}_3).$$

By repeating this process, the variables z_i for $i(< k)$ are shown below.

$$\begin{aligned}
L_i &= \begin{pmatrix} I_i & 0^{(i\times n)} \\ 0^{(n\times i)} & L \end{pmatrix}, \\
\mathbb{b}'_{2i} &= \mathcal{T}_{L_i^T}(\mathbb{b}_{2i}), \\
\mathbb{b}'_{2i+1} &= \mathcal{T}_{L_i^T}(\mathbb{b}_{2i+1}), \\
z_i &= (\mathbb{b}'_{2i} \cdot z_{i-1}||\cdots||z_0||\mathbb{x})(\mathbb{b}'_{2i+1} \cdot z_{i-1}||\cdots||z_0||\mathbb{x}).
\end{aligned}$$

Finally, we obtain an (n, k)-bit A-box $S'_A = (z_{k-1}, \cdots, z_0)$. Thus, the theorem holds. □

3 Theoretical lower bounds on the differential uniformity and linearity of S-boxes by MC

As mentioned before, the differential uniformity and linearity of S-boxes are invariant under EA transformation. Thus, we consider an (n, m)-bit S-box $S = \mathcal{T}_M \circ S_A$ with suitable A-box S_A where M is a matrix in RREF.

3.1 Bounds for differential uniformity

For a difference $\Delta a \in \mathbb{F}_2^n$, we obtain the following equation.

$$\begin{aligned}
&S(\mathbb{x}) \oplus S(\mathbb{x} \oplus \Delta a) \\
&= \mathcal{T}_M \circ S_A(\mathbb{x}) \oplus \mathcal{T}_M \circ S_A(\mathbb{x} \oplus \Delta a) \\
&= \mathcal{T}_M(S_A(\mathbb{x}) \oplus S_A(\mathbb{x} \oplus \Delta a)).
\end{aligned}$$

Therefore, for a difference $\Delta b \in \mathbb{F}_2^m$,

$$\begin{aligned}
&\{\mathbb{x} \in \{0,1\}^n | S_A(\mathbb{x}) \oplus S_A(\mathbb{x} \oplus \Delta a) = \Delta b\} \\
\subseteq\ &\{\mathbb{x} \in \{0,1\}^n | \mathcal{T}_M \circ S_A(\mathbb{x}) \oplus \mathcal{T}_M \circ S_A(\mathbb{x} \oplus \Delta a) = \mathcal{T}_M(\Delta b)\} \\
=\ &\{\mathbb{x} \in \{0,1\}^n | S(\mathbb{x}) \oplus S(\mathbb{x} \oplus \Delta a) = \mathcal{T}_M(\Delta b)\}.
\end{aligned}$$

As $\delta(\mathcal{T}_M \circ S_A) \geq \delta_S(\Delta a, \mathcal{T}_M(\Delta b)) \geq \delta_{S_A}(\Delta a, \Delta b)$ holds for all differences Δa and Δb by the above relation, the property $\delta(\mathcal{T}_M \circ S_A) \geq \delta(S_A)$ holds. Therefore, the differential uniformity of an (n, m)-bit S-box with MC k is greater than or equal to the differential uniformity of a suitable (n, k)-bit A-box. The lower bounds of differential uniformity of (n, k)-bit A-boxes become those of S-boxes with MC k.

The differential uniformity of the S-box, which has at least one input difference that induces only one output difference, is 2^n. In order to lower the differential uniformity, this input difference must be eliminated. In an A-box, if all AND gates have zero input differences, the differential uniformity becomes 2^n. The input differences make a space and we define the space as a *complementable space*. The word 'complementable' is taken from [12].

Lemma 1 *Let S_A be an (n, k)-bit A-box. Define the set $\mathcal{C}_{S_A}$ of Δa satisfying $\mathbb{b}_i|_n \cdot \Delta a = 0$ for all partner vectors $\mathbb{b}_i$ to be a* ***complementable space*** *of S_A. The complementable space $\mathcal{C}_{S_A}$ has the following properties.*

- *For $\Delta a \in \mathcal{C}_{S_A}$, $S_A(\Delta a) = 0^{(k)}$.*
- *For $\Delta a \in \mathcal{C}_{S_A}$ and $\mathbb{x} \in \mathbb{F}_2^n$, $S_A(\mathbb{x}) = S_A(\mathbb{x} \oplus \Delta a)$.*

– *If there is a nonzero difference in $\mathcal{C}_{S_A}$, then $\delta(S_A) = 2^n$.*

Proof Let Δa be a difference in $\mathcal{C}_{S_A}$ and $S_A = (f_{k-1}, \cdots, f_0)$. From $f_0(\Delta a) = (\mathbb{b}_0 \cdot \Delta a)(\mathbb{b}_1 \cdot \Delta a) = 0$, we obtain

$$f_i(\Delta a) = (\mathbb{b}_{2i} \cdot 0^{(i)}||\Delta a)(\mathbb{b}_{2i+1} \cdot 0^{(i)}||\Delta a) = (\mathbb{b}_{2i}|_n \cdot \Delta a)(\mathbb{b}_{2i+1}|_n \cdot \Delta a) = 0$$

by mathematical induction for all $i > 0$. Therefore, $S_A(\Delta a) = 0^{(k)}$ holds.

Let $S_A(\mathbb{x}) = (y_{k-1}, \cdots, y_0)$ and $S_A(\mathbb{x} \oplus \Delta a) = (y'_{k-1}, \cdots, y'_0)$. We use mathematical induction to prove that $y'_i = y_i$ for all i. In the base step, we obtain $y'_0 = y_0$ through the equation below.

$$\begin{aligned} y'_0 &= (\mathbb{b}_0 \cdot (\mathbb{x} \oplus \Delta a))(\mathbb{b}_1 \cdot (\mathbb{x} \oplus \Delta a)) \\ &= (\mathbb{b}_0 \cdot \mathbb{x} \oplus \mathbb{b}_0 \cdot \Delta a)(\mathbb{b}_1 \cdot \mathbb{x} \oplus \mathbb{b}_1 \cdot \Delta a) \\ &= (\mathbb{b}_0 \cdot \mathbb{x})(\mathbb{b}_1 \cdot \mathbb{x}) = y_0. \end{aligned}$$

In the inductive step, we assume $y'_i = y_i$ for all $i(< t-1)$ such that $t < k$, and we show that $y'_t = y_t$ through the equation below.

$$\begin{aligned} y'_t &= (\mathbb{b}_{2t} \cdot (y'_{t-1}|| \cdots ||y'_0||(\mathbb{x} \oplus \Delta a)))(\mathbb{b}_{2t+1} \cdot (y'_{t-1}|| \cdots ||y'_0||(\mathbb{x} \oplus \Delta a))) \\ &= (\mathbb{b}_{2t} \cdot (y_{t-1}|| \cdots ||y_0||(\mathbb{x} \oplus \Delta a)))(\mathbb{b}_{2t+1} \cdot (y_{t-1}|| \cdots ||y_0||(\mathbb{x} \oplus \Delta a))) \\ &= (\mathbb{b}_{2t} \cdot (y_{t-1}|| \cdots ||y_0||\mathbb{x}) \oplus \mathbb{b}_{2t}|_n \cdot \Delta a)(\mathbb{b}_{2t+1} \cdot (y_{t-1}|| \cdots ||y_0||\mathbb{x}) \oplus \mathbb{b}_{2t+1}|_n \cdot \Delta a) \\ &= (\mathbb{b}_{2t} \cdot (y_{t-1}|| \cdots ||y_0||\mathbb{x}))(\mathbb{b}_{2t+1} \cdot (y_{t-1}|| \cdots ||y_0||\mathbb{x})) = y_t. \end{aligned}$$

This fact indicates that $S_A(\mathbb{x}) = S_A(\mathbb{x} \oplus \Delta a)$ holds, regardless of $\mathbb{x}$. Therefore, $\delta(S_A) = \delta_{S_A}(\Delta a, 0^{(k)}) = 2^n$. □

We observe how large a k is needed to eliminate all nonzero elements in the complementable space. First, let us define a matrix below.

$$A = \begin{pmatrix} \mathbb{b}_{2k-1}|_n & \mathbb{b}_{2k-2}|_n & \cdots & \mathbb{b}_1|_n & \mathbb{b}_0|_n \end{pmatrix}^T = \begin{pmatrix} b_{2k-1}^{n-1} & b_{2k-1}^{n-2} & \cdots & b_{2k-1}^{1} & b_{2k-1}^{0} \\ b_{2k-2}^{n-1} & b_{2k-2}^{n-2} & \cdots & b_{2k-2}^{1} & b_{2k-2}^{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{1}^{n-1} & b_{1}^{n-2} & \cdots & b_{1}^{1} & b_{1}^{0} \\ b_{0}^{n-1} & b_{0}^{n-2} & \cdots & b_{0}^{1} & b_{0}^{0} \end{pmatrix}.$$

The complementable space $\mathcal{C}_{S_A}$ can be defined as a homogeneous linear system as follows.

$$\mathcal{C}_{S_A} = \{\Delta a \in \mathbb{F}_2^n | A(\Delta a) = 0^{(n)}\}.$$

As k increases by 1, the number of rows in A increases by 2. The size of the solution space of the system $A(\Delta a) = 0^{(n)}$ is at least 2^{n-2k}. In order to become $\mathcal{C}_{S_A} = \{0\}$, the number of rows in A must be n or more. Since $2^{n-2\lfloor \frac{n-1}{2} \rfloor} > 1$, an (n,k)-bit A-box with $k \leq \lfloor \frac{n-1}{2} \rfloor$ holds $\delta(S_A) = 2^n$ by Lemma 1. Therefore, we can derive Theorem 3 as follows.

Theorem 3 *Let S_A be an (n,k)-bit A-box. If $k \leq \lfloor \frac{n-1}{2} \rfloor$, then $\delta(S_A) = 2^n$.*

Consider an input difference Δa and an output difference Δb that comprise the differential uniformity of an A-box. When an AND gate is added, the best case is that Δa activates the last AND gate and halves the differential uniformity.

Theorem 4 *Let S_A be an (n, k)-bit A-box. If $k = \lfloor \frac{n-1}{2} \rfloor + l$, then $\delta(S_A) \geq 2^{n-l}$ for all $l \geq 0$.*

Proof We use mathematical induction to prove this theorem from $l = 0$. By Theorem 3, if $k = \lfloor \frac{n-1}{2} \rfloor$, then $\delta(S_A) = 2^n$. Assume that this theorem holds when $l = t$. Let S_A be an $(n, k+1)$-bit A-box for $k = \lfloor \frac{n-1}{2} \rfloor + t$. Then the (n, k)-bit A-box $S_A|_k$ satisfies $\delta(S_A|_k) \geq 2^{n-t}$ based on the assumption. There are two differences Δa and Δb such that $\delta_{S_A|_k}(\Delta a, \Delta b) \geq 2^{n-t}$. We obtain the following equation.

$$\begin{aligned}
&\{\mathbb{x} \in \mathbb{F}_2^n | S_A|_k(\mathbb{x}) \oplus S_A|_k(\mathbb{x} \oplus \Delta a) = \Delta b\} \\
= &\{\mathbb{x} \in \mathbb{F}_2^n | S_A(\mathbb{x}) \oplus S_A(\mathbb{x} \oplus \Delta a) = (0||\Delta b)\} \\
&\cup \{\mathbb{x} \in \mathbb{F}_2^n | S_A(\mathbb{x}) \oplus S_A(\mathbb{x} \oplus \Delta a) = (1||\Delta b)\}.
\end{aligned}$$

Either $\delta_{S_A}(\Delta a, 0||\Delta b) \geq 2^{n-t-1}$ or $\delta_{S_A}(\Delta a, 1||\Delta b) \geq 2^{n-t-1}$ holds by the pigeonhole principle. Thus, we have $\delta(S_A) \geq 2^{n-t-1}$. □

As previously considered, the bound of the A-box becomes the bound of the S-box. Corollary 1 follows from Theorems 1, 3, and 4.

Corollary 1 *Let S be an (n, m)-bit S-box. The following properties hold.*

- *If $c_\wedge(S) \leq \lfloor \frac{n-1}{2} \rfloor$, then $\delta(S) = 2^n$.*
- *If $c_\wedge(S) = \lfloor \frac{n-1}{2} \rfloor + l$, then $\delta(S) \geq 2^{n-l}$ for $l \geq 0$.*

3.2 Bounds for linearity

While the theoretical relationship between MC and differential uniformity has not been studied, the theoretical relationship between MC and linearity has been previously studied [10]. Boyar and Find studied the relationship using linear codes and $\Sigma\Pi\Sigma$ circuit. To have low linearity, more AND gates must be used than the length of the shortest corresponding linear code. For example, an (n, n)-bit S-box with linearity $2^{\frac{n-1}{2}}$ should use $L(n, \frac{n-1}{2})$ or more AND gates in the $\Sigma\Pi\Sigma$ circuit, where $L(n, \frac{n-1}{2})$ is the length of the shortest linear n-dimensional code over $\mathbb{F}_2$ with a distance $\frac{n-1}{2}$. For a sufficiently large n, $L(n, \frac{n-1}{2}) > 2.32n$. However, it remains to be determined how large n must be, which makes this result difficult to apply directly to the construction of an S-box in practice. Therefore, we will more clearly present the lower bounds of linearity by MC.

For two maskings $\Lambda a \in \mathbb{F}_2^n$ and $\Lambda b \in \mathbb{F}_2^m$, the linear equation of (n, m)-bit S-box S is followed.

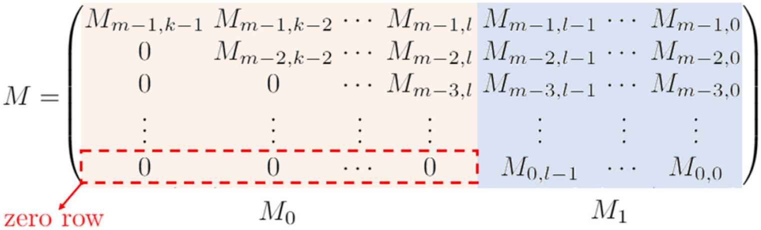
$$\begin{aligned}
\Lambda a \cdot \mathbb{x} \oplus \Lambda b \cdot S(\mathbb{x}) &= \Lambda a \cdot \mathbb{x} \oplus \Lambda b \cdot \mathcal{T}_M(S_A(\mathbb{x})) \\
&= \Lambda a \cdot \mathbb{x} \oplus \Lambda b \cdot \mathcal{T}_M(f_{k-1}(\mathbb{x}), \cdots, f_0(\mathbb{x})). \qquad (6)
\end{aligned}$$

If $k < m$, the last row of M is a zero row. In this case, it is easily shown that $\mathcal{L}_S(\Lambda a, \Lambda b) = 2^n$ when $\Lambda a = 0^{(n)}$ and $\Lambda b = 0^{(m-1)}||1$. The following theorem has been proved.

Theorem 5 *Let S be an (n, m)-bit S-box. If $c_\wedge(S) \leq m - 1$, then $\mathcal{L}(S) = 2^n$.*

Assume that $k = m + l - 1$ for $l \geq 1$. Then, M is decomposed as follows.

$$M = \begin{pmatrix} M_0 & M_1 \end{pmatrix}$$



$$M = \begin{pmatrix} M_{m-1,k-1} & M_{m-1,k-2} & \cdots & M_{m-1,l} & M_{m-1,l-1} & \cdots & M_{m-1,0} \\ 0 & M_{m-2,k-2} & \cdots & M_{m-2,l} & M_{m-2,l-1} & \cdots & M_{m-2,0} \\ 0 & 0 & \cdots & M_{m-3,l} & M_{m-3,l-1} & \cdots & M_{m-3,0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & M_{0,l-1} & \cdots & M_{0,0} \end{pmatrix}$$

Fig. 3 $m \times k$ matrix M decomposed by two partition matrices $m \times (m-1)$ M_0 and $m \times l$ M_1

where M_0 and M_1 are $m \times (m-1)$ and $m \times l$ partition matrices, respectively. Note that the last row of M_0 is a zero row (Fig. 3).

The (6) is expressed by

$$\begin{aligned} & \Lambda a \cdot \mathbb{x} \oplus \Lambda b \cdot S(\mathbb{x}) \\ = {} & \Lambda a \cdot \mathbb{x} \oplus \Lambda b \cdot \mathcal{T}_M(f_{k-1}(\mathbb{x}), \cdots, f_0(\mathbb{x})) \\ = {} & \Lambda a \cdot \mathbb{x} \oplus \Lambda b \cdot \mathcal{T}_{M_0}(f_{k-1}(\mathbb{x}), \cdots, f_l(\mathbb{x})) \oplus \Lambda b \cdot \mathcal{T}_{M_1}(f_{l-1}(\mathbb{x}), \cdots, f_0(\mathbb{x})) \\ = {} & \Lambda a \cdot \mathbb{x} \oplus (0^{(m-1)}||1) \cdot \mathcal{T}_{M_1}(f_{l-1}(\mathbb{x}), \cdots, f_0(\mathbb{x})) \end{aligned} \tag{7}$$

when $\Lambda b = 0^{(m-1)}||1$. We denote the n-variable Boolean function f as f as $f(\mathbb{x}) = (0^{(m-1)}||1) \cdot \mathcal{T}_{M_1}(f_{l-1}(\mathbb{x}), \cdots, f_0(\mathbb{x}))$. We refer to the following theorem, which is proved in [10].

Theorem 6 *Let f be an n-variable Boolean function. Then $\mathcal{L}(f) \geq 2^{n-c_\wedge(f)}$* [10].

By the definition of A-box, the MC of f is less than or equal to l. This means $\mathcal{L}(f) \geq 2^{n-l}$. For $\Lambda a \in \mathbb{F}_2^n$ such that $\mathcal{L}(f) = \mathcal{L}_f(\Lambda a, 1)$, we get

$$\mathcal{L}(S) \geq \mathcal{L}_S(\Lambda a, 0^{(m-1)}||1) = \mathcal{L}_f(\Lambda a, 1) = \mathcal{L}(f) \geq 2^{n-l}.$$

We have proved Theorem 7.

Theorem 7 *Let S be an (n, m)-bit S-box. If $c_\wedge(S) = m + l - 1$, then $\mathcal{L}(S) \geq 2^{n-l}$ for $l \geq 0$.*

4 Method for searching for S-boxes with low differential uniformity by MC

In [37], Zajac and Jókay presented an algorithm to construct bijective (4, 4)-bit S-boxes with minimal nonlinear gates. Since their method performs an exhaustive search for S-boxes regardless of their cryptographic properties such as differential uniformity and linearity, it would be computationally difficult to apply to search for S-boxes with a size larger than 4 bits. In order to search for S-boxes with a larger size, we focus on the S-boxes with low differential uniformity. We adopt the branch-and-bound technique for our algorithm to investigate S-boxes with larger sizes.

We say that an A-box (or S-box) has a theoretically optimal differential uniformity when its differential uniformity equals the lower bound presented in Corollary 1. That is, the theoretically optimal differential uniformity of an (n, k)-bit A-box is 2^{n-l} for $k = \lfloor \frac{n-1}{2} \rfloor + l$.

The phases for searching the S-box with k nonlinear gates, differential uniformity δ and linearity $\mathcal{L}$ are as follows.

Phase 1. Find an (n, k)-bit A-box S_A with the desired differential uniformity δ.

(a) Collect $(n, \lfloor\frac{n-1}{2}\rfloor+1)$-bit A-boxes with theoretically optimal differential uniformity 2^{n-1} (cf. Section 4.1).
(b) Extend them to $(n, \lfloor\frac{n-1}{2}\rfloor + l)$-bit A-boxes with theoretically optimal differential uniformity 2^{n-l}, where $l > 1$ (cf. Algorithm 1 in Section 4.2).
(c) Choose $(n, \lfloor\frac{n-1}{2}\rfloor + l)$-bit A-boxes with the lowest differential uniformity when Phase 1-(b) fails with respect to l. (cf. Section 4.2).

Phase 2. Find an $m \times k$ matrix M in RREF to make $\mathcal{T}_M \circ S_A$ with the desired differential uniformity δ and linearity $\mathcal{L}$ (cf. Section 4.3).

Phase 3. (Optional) Find an $n \times n$ matrix N to make an (n, n)-bit S-box $\mathcal{T}_M \circ S_A \oplus \mathcal{T}_N$ bijective, where $n = m$ (cf. Section 4.3).

The above process is described in Fig. 4.

4.1 $(n, \lfloor\frac{n-1}{2}\rfloor + 1)$-bit A-boxes with theoretically optimal differential uniformity 2^{n-1}

In order to have differential uniformity 2^{n-1}, the complementable space must be $\{0\}$. That is, the rows of matrix A in Section 3.1 span the dimension n. This fact induces the following lemma.

Lemma 2 *For $k = \lfloor\frac{n-1}{2}\rfloor+1$, let S_A be an (n, k)-bit A-box. If $\delta(S_A) = 2^{n-1}$, then restricted partner vectors $\{\mathbb{b}_0|_n, \cdots, \mathbb{b}_{2k-1}|_n\}$ span the dimension n.*

If n vectors span dimension n, the vectors are linearly independent. These vectors can be transformed on a standard basis by operating an appropriate matrix. Note Theorem 2. If we compose $\mathcal{T}_L$ on the input of the S_A, we can get a linear equivalent A-box that has the matrix-operated partner vectors. Specifically, the restricted partner vector $\mathbb{b}_i|_n$ is transformed into $\mathcal{T}_L(\mathbb{b}_i|_n)$ for all $i \geq 0$. In Lemma 2, if n is even, the restricted partner vectors $\{\mathbb{b}_0|_n, \cdots, \mathbb{b}_{n-1}|_n\}$ span the dimension n. All of the n vectors can be transformed on a standard basis through an appropriate $\mathcal{T}_L$. However, if n is odd, the $n + 1$ vectors span the dimension n, so they are linearly dependent. In this case, it can be resolved by removing the vector that causes the linear dependence. Owing to this difference, the partner vectors have a different form depending on whether n is even or odd.

Theorem 8 *Let $n = 2p$ for $p > 0$. For $k = p$, let S_A be an (n, k)-bit A-box such that $\delta(S_A) = 2^{n-1}$. Then, there is an A-box S'_A, which is linear equivalent to S_A and has a partner tuple $(\mathbb{b}'_{2k-1}, \cdots, \mathbb{b}'_0)$ such that*

$$\begin{cases} \mathbb{b}'_i|_n = e_i, & \text{for } 0 \leq i < n, \\ \mathbb{b}'_{2i}|^i \leq \mathbb{b}'_{2i+1}|^i, & \text{for } 0 < i < p, \end{cases}$$

where e_i is the n-bit value whose bits are all zeros except for the i^{th}-bit (e.g., $e_0 = (0, \cdots, 0, 1)$).

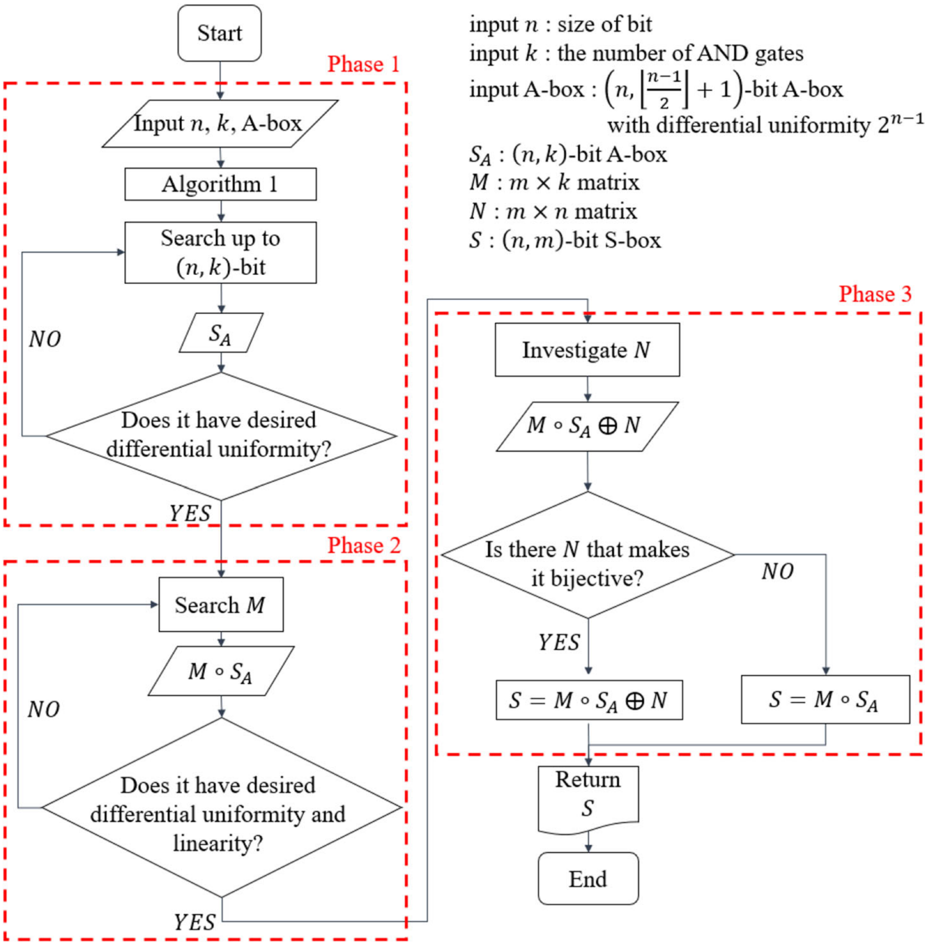

input n : size of bit
input k : the number of AND gates
input A-box : $\left(n, \left\lfloor\frac{n-1}{2}\right\rfloor + 1\right)$-bit A-box with differential uniformity 2^{n-1}
S_A : (n, k)-bit A-box
M : $m \times k$ matrix
N : $m \times n$ matrix
S : (n, m)-bit S-box
Start
Phase 1
Input n, k, A-box
Algorithm 1
Search up to (n, k)-bit
S_A
NO
Does it have desired differential uniformity?
YES
Phase 2
Search M
$M \circ S_A$
NO
Does it have desired differential uniformity and linearity?
YES
Phase 3
Investigate N
$M \circ S_A \oplus N$
Is there N that makes it bijective?
NO
YES
$S = M \circ S_A \oplus N$
$S = M \circ S_A$
Return S
End

Fig. 4 Constructing an S-box from an A-box

Proof Let us define an $n \times n$ matrix B as follows using the partner vectors $\{\mathbb{b}_{2k-1}, \cdots, \mathbb{b}_0\}$ of S_A.

$$B = \left(\mathbb{b}_{n-1}|_n \ \mathbb{b}_{n-2}|_n \ \cdots \ \mathbb{b}_1|_n \ \mathbb{b}_0|_n\right) = \begin{pmatrix} b_{n-1}^{n-1} & b_{n-2}^{n-1} & \cdots & b_1^{n-1} & b_0^{n-1} \\ b_{n-1}^{n-2} & b_{n-2}^{n-2} & \cdots & b_1^{n-2} & b_0^{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{n-1}^{1} & b_{n-2}^{1} & \cdots & b_1^{1} & b_0^{1} \\ b_{n-1}^{0} & b_{n-2}^{0} & \cdots & b_1^{0} & b_0^{0} \end{pmatrix}.$$

According to Lemma 2, $\{\mathbb{b}_0|_n, \cdots, \mathbb{b}_{n-1}|_n\}$ spans the dimension n and B is invertible. $B^{-1}(\mathbb{b}_i|_n) = e_i$ holds for all $i (< n)$, because it is the i^{th} column of the identity matrix. By Theorem 2, the A-box $S'_A = S_A \circ \mathcal{T}_{B^{-T}}$ is an A-box linear equivalent to S_A.

By the same theorem, we know that $\mathbb{b}'_{2i}|^i = \mathbb{b}_{2i}|^i$ and $\mathbb{b}'_{2i+1}|^i = \mathbb{b}_{2i+1}|^i$. Note that the A-box S_A is invariant when $\mathbb{b}_{2i}$ and $\mathbb{b}_{2i+1}$ are swapped. We Assume, without loss of generality, $\mathbb{b}_{2i}|^i < \mathbb{b}_{2i+1}|^i$ for $0 < i < p$ in integer form. S'_A satisfies all required conditions. □

Theorem 9 *Let $n = 2p + 1$ for $p > 0$. For $k = p + 1$, let S_A be an (n, k)-bit A-box such that $\delta(S_A) = 2^{n-1}$. Then, there is an A-box S'_A, which is linear equivalent to S_A and has a partner tuple $(\mathbb{b}'_{2k-1}, \cdots, \mathbb{b}'_0)$ such that*

$$\begin{cases} \mathbb{b}'_i|_n = e_i, & \text{for } 0 \le i < j, \\ \mathbb{b}'_i|_n = 0^{(n-j)}||\mathbb{d}, & \text{for } i = j, \\ \mathbb{b}'_i|_n = e_{i-1}, & \text{for } j + 1 \le i < n, \\ \mathbb{b}'_{2i}|^i \le \mathbb{b}'_{2i+1}|^i, & \text{for } i \ne j, j+1 \text{ and } 0 < i \le p, \end{cases}$$

where $\mathbb{d}$ is a j-bit string and j is the largest subscript that causes linear dependence for a set of $n + 1$ vectors: $\mathbb{b}_0|_n, \mathbb{b}_1|_n, \cdots, \mathbb{b}_j|_n, \cdots, \mathbb{b}_{n-1}|_n, \mathbb{b}_n|_n$.

Proof As the $n + 1$ vectors $\{\mathbb{b}_0|_n, \cdots, \mathbb{b}_n|_n\}$ are linearly dependent, there exist some $\mathbb{c} = (c_n, \cdots, c_0) \in \mathbb{F}_2^{n+1}$ such that

$$c_0\mathbb{b}_0|_n \oplus \cdots \oplus c_n\mathbb{b}_n|_n = 0^{(n)}. \tag{8}$$

Let c_j be the nonzero coefficient with the highest subscript j. We obtain the following:

$$\mathbb{b}_j|_n = c_0\mathbb{b}_0|_n \oplus \cdots \oplus c_{j-1}\mathbb{b}_{j-1}|_n.$$

If j is 0, then $\mathbb{b}_0|_n = \mathbb{b}_0 = 0^{(n)}$ and $y_0 = 0$. As this induces $c_\wedge(S_A) \le p - 1$, we obtain $\delta(S_A) \ge 2^{n+1}$ by Corollary 1. However, this contradicts the assumption. If $j = 1$, then $\mathbb{b}_1 = \mathbb{b}_1|_n = c_0\mathbb{b}_0|_n = c_0\mathbb{b}_0$. When $c_0 = 0$, we have the same case as when $j = 0$. $c_0 = 1$ makes y_0 a linear function, as shown below.

$$y_0 = (\mathbb{b}_0 \cdot \mathbb{x})(\mathbb{b}_1 \cdot \mathbb{x}) = (\mathbb{b}_0 \cdot \mathbb{x})^2 = \mathbb{b}_0 \cdot \mathbb{x}.$$

This induces $c_\wedge(S_A) \le p + l - 1$ and causes a contradiction, too. Therefore, we have found that $j \ge 2$.

$\{\mathbb{b}_{\rho_j(0)}|_n, \cdots, \mathbb{b}_{\rho_j(n-1)}|_n\}$ are linearly independent because they span the dimension n by Lemma 2. Let us make the following invertible matrix C

$$C = \left(\mathbb{b}_{\rho_j(n-1)}|_n \ \mathbb{b}_{\rho_j(n-2)}|_n \ \cdots \ \mathbb{b}_{\rho_j(1)}|_n \ \mathbb{b}_{\rho_j(0)}|_n\right)$$

$$= \begin{pmatrix} b^{n-1}_{\rho_j(n-1)} & b^{n-1}_{\rho_j(n-2)} & \cdots & b^{n-1}_{\rho_j(1)} & b^{n-1}_{\rho_j(0)} \\ b^{n-2}_{\rho_j(n-1)} & b^{n-2}_{\rho_j(n-2)} & \cdots & b^{n-2}_{\rho_j(1)} & b^{n-2}_{\rho_j(0)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b^{1}_{\rho_j(n-1)} & b^{1}_{\rho_j(n-2)} & \cdots & b^{1}_{\rho_j(1)} & b^{1}_{\rho_j(0)} \\ b^{0}_{\rho_j(n-1)} & b^{0}_{\rho_j(n-2)} & \cdots & b^{0}_{\rho_j(1)} & b^{0}_{\rho_j(0)} \end{pmatrix}$$

where $\rho_j(i) = \begin{cases} i & \text{for } i < j \\ i + 1 & \text{for } i \ge j \end{cases}$. We know that $C^{-1}(\mathbb{b}_{\rho_j(i)}|_n) = e_i$ for $i(< n)$. By Theorem 2, $S'_A = S_A \circ \mathcal{T}_{C^{-T}}$ is an A-box that is linear equivalent to S_A. S'_A also satisfies the first required condition.

Next, let us see what form $\mathbb{b}'_j$ becomes. Let $\mathbb{b}'_j|_n \cdot \mathbb{x} = d_{n-1}x_{n-1} \oplus \cdots \oplus d_0x_0$ for $(d_{n-1}, \cdots, d_0) \in \mathbb{F}_2^n$. Then, the property below follows.

$$\begin{aligned}
\mathbb{b}'_j|_n \cdot \mathbb{x} &= d_{n-1}x_{n-1} \oplus \cdots \oplus d_0x_0 \\
&= d_{n-1}(e_{n-1} \cdot \mathbb{x}) \oplus \cdots \oplus d_0(e_0 \cdot \mathbb{x}) \\
&= d_{n-1}((C^{-1}\mathbb{b}_{\rho_j(n-1)}|_n) \cdot \mathbb{x}) \oplus \cdots \oplus d_0((C^{-1}\mathbb{b}_{\rho_j(0)}|_n) \cdot \mathbb{x}) \\
&= (d_{n-1}(C^{-1}\mathbb{b}_{\rho_j(n-1)}|_n) \oplus \cdots \oplus d_0(C^{-1}\mathbb{b}_{\rho_j(0)}|_n)) \cdot \mathbb{x} \\
&= C^{-1}(d_{n-1}\mathbb{b}_{\rho_j(n-1)}|_n \oplus \cdots \oplus d_0\mathbb{b}_{\rho_j(0)}|_n) \cdot \mathbb{x}.
\end{aligned}$$

As the above property must hold for every $\mathbb{x}$, we obtain

$$\mathbb{b}'_j|_n = C^{-1}(\mathbb{b}_j|_n) = C^{-1}(d_{n-1}\mathbb{b}_{\rho_j(n-1)}|_n \oplus \cdots \oplus d_0\mathbb{b}_{\rho_j(0)}|_n).$$

Since j must be the highest subscript in the equation, we get

$$d_{n-1} = d_{n-2} = \cdots = d_j = 0.$$

Therefore,

$$\begin{aligned}
\mathbb{b}'_j|_n &= d_{j-1}e_{j-1} \oplus \cdots \oplus d_0e_0 \\
&= (0, \cdots, 0, d_{j-1}, \cdots, d_0).
\end{aligned}$$

As in the previous proof, assuming $\mathbb{b}_{2i}|^i < \mathbb{b}_{2i+1}|^i$ for $i \neq j, j+1$ and $0 < i \leq p$, S'_A satisfies all required conditions. □

These theorems enable us to significantly reduce the search space for possible $(n, \lfloor \frac{n-1}{2} \rfloor + 1)$-bit A-boxes with differential uniformity 2^{n-1} (Fig. 5). Table 3 presents the A-box search space, which excludes duplication, and the number of A-boxes with theoretically optimal differential uniformity.

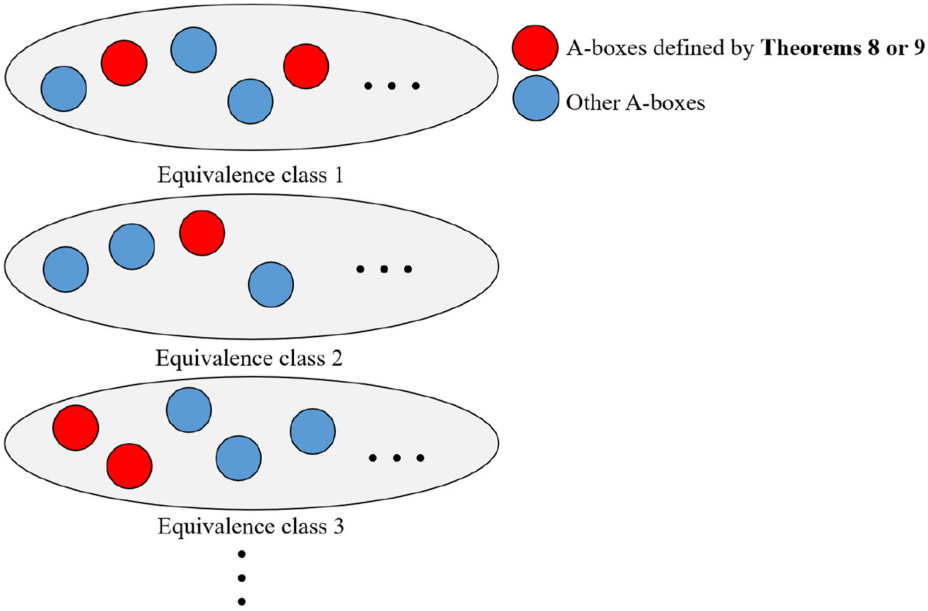


Fig. 5 A-boxes defined by Theorems 8 and 9

Table 3 $(n, \lfloor\frac{n-1}{2}\rfloor + 1)$-bit A-box search

n	3	4	5	6	7	8
Size of search space	44	3	2,720	30	466,080	1,080
Number of A-boxes with differential uniformity 2^{n-1}	9	3	990	30	220,320	1,080

4.2 Extending A-boxes with theoretically optimal differential uniformity by MC

We extend A-boxes by increasing AND gates step by step. Adding one AND gate to an (n, k)-bit A-box is the same as determining two additional partner vectors $\mathbb{b}_{2k}$ and $\mathbb{b}_{2k+1}$. In order to obtain an extended A-box with theoretically optimal differential uniformity, the A-box in a previous step must have theoretically optimal differential uniformity. This condition follows from the below theorem.

Theorem 10 *For $k = \lfloor\frac{n-1}{2}\rfloor + l$, let S_A be an (n, k)-bit A-box for $l \geq 0$. If $\delta(S_A) = 2^{n-l}$, then $\delta(S_A|_{k-p}) = 2^{n-l+p}$ for all $p \leq l$.*

Proof Note that the A-box $S_A|_{k-p}$ is constructed by removing the last p-bits (p MSBs) generated from the S_A. As $S_A|_{k-p}$ is an $(n, k-p)$-bit A-box, $\delta(S_A|_{k-p}) \geq 2^{n-l+p}$ holds by Theorem 4. Assume that $\delta(S_A|_{k-p}) \gneqq 2^{n-l+p}$. There are two differences Δa and Δb such that $\delta(S_A|_{k-p}) = \delta_{S_A|_{k-p}}(\Delta a, \Delta b)$. According to the definition of an A-box, the following equation holds.

$$\begin{aligned}&\{\mathbb{x} \in \mathbb{F}_2^n | S_A|_{k-p}(\mathbb{x}) \oplus S_A|_{k-p}(\mathbb{x} \oplus \Delta a) = \Delta b\}\\ = &\bigcup_{\mathbb{w} \in \mathbb{F}_2^p} \{\mathbb{x} \in \mathbb{F}_2^n | S_A(\mathbb{x}) \oplus S_A(\mathbb{x} \oplus \Delta a) = (\mathbb{w}||\Delta b)\}.\end{aligned}$$

Next, we obtain

$$\delta_{S_A|_{k-p}}(\Delta a, \Delta b) = \sum_{\mathbb{w} \in \mathbb{F}_2^p} \delta_{S_A}(\Delta a, \mathbb{w}||\Delta b).$$

As $\delta_{S_A|_{k-p}}(\Delta a, \Delta b) > 2^{n-l+p}$, there is a vector $\mathbb{w} \in \mathbb{F}_2^p$ that satisfies

$$\delta_{S_A}(\Delta a, \mathbb{w}||\Delta b) > \frac{2^{n-l+p}}{2^p} = 2^{n-l}$$

by the pigeonhole principle. As a result, we have $\delta(S_A) \geq \delta_{S_A}(\Delta a, \mathbb{w}||\Delta b) > 2^{n-l}$, but this contradicts the assumption. □

Therefore, we can accelerate the investigation by checking whether the differential uniformity of the A-boxes constructed in each step is theoretically optimal.

In Section 4.1, we obtained a set of $(n, \lfloor\frac{n-1}{2}\rfloor + 1)$-bit A-boxes with theoretically optimal differential uniformity 2^{n-1}. We call this set $\mathcal{A}$ and refer to A-boxes of $\mathcal{A}$ as parent nodes. A child node is an A-box with one AND gate added to a parent node. The number of child nodes per parent node is as many as two additional partner vectors $\mathbb{b}_{2k}$ and $\mathbb{b}_{2k+1}$ are possible for adding k^{th} AND gate (cf. Fig. 6). To count the number of A-boxes per depth, our algorithm performs a breadth-first search. The detailed process is described in Algorithm 1.

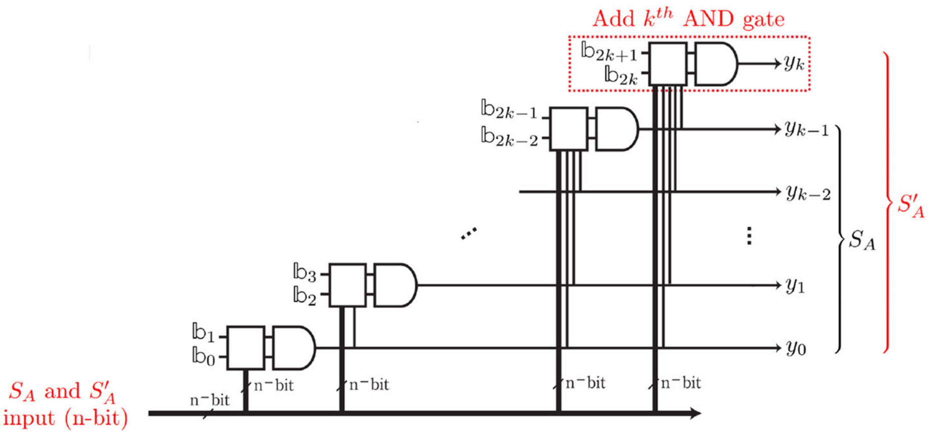


Fig. 6 A-box extension ($(n, k+1)$-bit S'_A extended from (n, k)-bit S_A)

Algorithm 1 Search algorithm for A-boxes with theoretically optimal differential uniformity.

```
Require: input size $n$, MC $K$, a set $\mathscr{A}$ of $(n, \lfloor \frac{n-1}{2} \rfloor + 1)$-bit A-boxes $S_A$
Ensure: a set of $(n, *)$-bit A-boxes with differential uniformity 2 or a set of $(n, K)$-bit A-boxes with
    theoretically optimal differential uniformity
$k \leftarrow \lfloor \frac{n-1}{2} \rfloor + 1$
$\delta_{min} \leftarrow 2^{n-1}$
while $k < K$ or $\delta_{min} > 2$ do
    $\mathscr{A}' \leftarrow \emptyset$
    for $S_A \in \mathscr{A}$ do
        for $\mathbb{b}_{2k} = 1, 2, 3, \cdots, 2^{n+k} - 2$ do
            for $\mathbb{b}_{2k+1} = \mathbb{b}_{2k} + 1, \mathbb{b}_{2k} + 2, \mathbb{b}_{2k} + 3, \cdots, 2^{n+k} - 1$ do
                Add $k^{th}$ AND gate to $S_A$ and construct an $(n, k+1)$-bit A-box $S'_A$
                if $\delta(S'_A) == \delta_{min}/2$ then
                    $\mathscr{A}' \leftarrow \mathscr{A}' \cup \{S'_A\}$
                end if
            end for
        end for
    end for
    $\delta_{min} \leftarrow \delta_{min}/2$
    $\mathscr{A} \leftarrow \mathscr{A}'$
    $k \leftarrow k + 1$
end while
return $\mathscr{A}$
```

In the algorithm, the partner vectors $\mathbb{b}_{2k}$ and $\mathbb{b}_{2k+1}$ are handled in integer form. There are two reasons why $\mathbb{b}_{2k+1}$ starts at $\mathbb{b}_{2k} + 1$:

- Since $(\mathbb{b}_{2k} \cdot \mathbb{x})(\mathbb{b}_{2k+1} \cdot \mathbb{x}) = (\mathbb{b}_{2k+1} \cdot \mathbb{x})(\mathbb{b}_{2k} \cdot \mathbb{x})$, generality is not lost even if $\mathbb{b}_{2k} \leq \mathbb{b}_{2k+1}$.
- Since $(\mathbb{b}_{2k} \cdot \mathbb{x})(\mathbb{b}_{2k} \cdot \mathbb{x}) = (\mathbb{b}_{2k} \cdot \mathbb{x})$, the case $\mathbb{b}_{2k} = \mathbb{b}_{2k+1}$ does not need to be investigated.

The number of A-boxes with theoretically optimal differential uniformity at each step is listed in Table 4. For 5-, 6-, and 8-bit sizes, our algorithm shows the following facts. The symbol '$*$' means that any number is possible.

- A $(5, *)$-bit S-box with differential uniformity 2 has MC at least 7.

Table 4 Number of partner tuples that construct A-boxes with theoretically optimal differential uniformity

MC	1	2	3	4	5	6	7
number of $(3, *)$-bit A-box	–	9	108	–	–	–	–
number of $(4, *)$-bit A-box	–	3	54	324	–	–	–
number of $(5, *)$-bit A-box	–	–	990	1,200,126	7,502,976	0	–
number of $(6, *)$-bit A-box	–	–	30	3,240	19,440	0	0
number of $(7, *)$-bit A-box	–	–	–	220,320	15,109,605,432	x	x
number of $(8, *)$-bit A-box	–	–	–	1,080	699,840	0	0

* '-' means that there is no need to investigate, and 'x' means full investigation is impossible because of huge computational complexity

- A $(6, *)$-bit S-box with differential uniformity 4 has MC at least 7.
- An $(8, *)$-bit S-box with differential uniformity 32 has MC at least 7.

This result also shows that Phase 1-(b) can fail. In this case, we chose the A-boxes with the lowest differential uniformity to construct the corresponding S-boxes.

4.3 Construction of S-boxes by A-boxes

Generating the $m \times k$ matrix M in RREF is simple. If $k \leq m$, M consists only of pivot columns. The differential uniformity of $\mathcal{T}_M \circ S_A$ equals the differential uniformity of S_A. The linearity is at most $2^n - 1$ as per Theorem 7. Let $k > m$. The columns of M are divided into m pivot columns and $k - m$ other columns. We randomly select the pivot columns and then randomly generate other columns. After arranging the pivot columns in order of subscript, the other columns are inserted randomly to generate a matrix. When M is generated, we calculate whether the differential uniformity and linearity of $\mathcal{T}_M \circ S_A$ are the desired values.

A bijective S-box is more useful than a non-bijective one when it has the same cryptographic properties. In particular, for the efficiency of side-channel masking, it is recommended to use an S-box with low MC as the primitive. For example, the block cipher Pyjamask [24], proposed in the recent NIST lightweight encryption competition, uses small S-boxes of $(3, 3)$-bit and $(4, 4)$-bit sizes to improve the side-channel masking efficiency. Fantomas and Robin [25] proposed LS-designs, which use bijective $(8, 8)$-bit S-boxes generated from an extension structure to reduce the number of nonlinear gates.

Let S be an (n, n)-bit S-box. Note that $S = \mathcal{T}_D \circ \mathcal{T}_M \circ S_A \oplus \mathcal{T}_N \oplus \mathbb{v}$ where N is an $n \times n$ matrix. We denote the entry of row i and column j of N by $N_{i,j}$. As $\mathcal{T}_D$ is bijective, the expression of S can be transformed as follows.

$$\begin{aligned} S &= \mathcal{T}_D \circ \mathcal{T}_M \circ S_A \oplus \mathcal{T}_N \oplus \mathbb{v} \\ &= \mathcal{T}_D \circ \mathcal{T}_M \circ S_A \oplus \mathcal{T}_D \circ \mathcal{T}_D^{-1} \circ \mathcal{T}_N \oplus \mathbb{v} \\ &= \mathcal{T}_D \circ (\mathcal{T}_M \circ S_A \oplus \mathcal{T}_D^{-1} \circ \mathcal{T}_N) \oplus \mathbb{v}. \end{aligned}$$

In the equation above, the bijectivity of S is expressed by $\mathcal{T}_M \circ S_A \oplus \mathcal{T}_D^{-1} \circ \mathcal{T}_N$. Finding $\mathcal{T}_D^{-1} \circ \mathcal{T}_N$ is the same as finding the $n \times n$ matrix, so we can regard it as $\mathcal{T}_N$. Therefore, our goal is to find the $\mathcal{T}_N$ that makes $\mathcal{T}_M \circ S_A \oplus \mathcal{T}_N$ bijective.

A bijective S-box has the characteristic that any combination of output bits is balanced. The new (n, d)-bit S-box $(f_{\sigma(d-1)}, \cdots, f_{\sigma(0)})$, generated by choosing d random f_i for the bijective (n, n)-bit S-box $S = (f_{n-1}, \cdots, f_0)$, is balanced ($\sigma$ is a permutation of $\mathbb{F}_2^n$). Let

$\mathcal{T}_M \circ S_A(\mathbb{x}) = (y_{n-1}, \cdots, y_0)$ and $\mathcal{T}_M \circ S_A \oplus \mathcal{T}_N(\mathbb{x}) = (z_{n-1}, \cdots, z_0)$. Then, for $i (< n)$, z_i is as follows:

$$z_i = y_i \oplus \bigoplus_{j=0}^{n-1} N_{j,i} x_j$$

We first investigate $(N_{n-1,n-1}, \cdots, N_{n-1,1}, N_{n-1,0})$ where $\mathbb{x} \mapsto z_{k-1}$ is balanced. Second, we investigate $(N_{n-1,n-1}, \cdots, N_{n-1,1}, N_{n-1,0})$ and $(N_{n-2,n-1}, \cdots, N_{n-2,1}, N_{n-2,0})$ where $\mathbb{x} \mapsto (z_{k-1}, z_{k-2})$ is balanced. By repeating this process, we can find the N where $\mathcal{T}_M \circ S_A \oplus \mathcal{T}_N(\mathbb{x})$ becomes bijective.

5 Conclusions

In this paper, we proved the theoretical lower bounds of differential uniformity and linearity of S-boxes by MC. We also presented an algorithm to search A-boxes with theoretically optimal differential uniformity by MC. The constructed A-boxes lead to S-boxes through our process. Some of the bijective S-boxes we found have better differential uniformity than those of existing bijective S-boxes with respect to the same nonlinear gates and linearity. Using our process, cryptography designers can make a trade-off between the implementation efficiency and security of the S-box, and they can reduce the complexity of S-box investigation because the minimum MC of the S-box having the desired security is known in advance based on this paper.

In future work, it would be interesting to investigate the following research topics:

- From a hardware point of view, is there a way to construct A-box to have high security but low AND depth?
- Is there a better way than a random process to construct S-boxes from a fixed A-box?
- How does bijectivity theoretically relate to A-boxes?
- How do the nonlinear gates of an A-box relate to other cryptographic properties such as algebraic degree, fixed points, or other properties?

Appendix A: Bitsliced implementations of our S-boxes

In this appendix, the method to implement the S-boxes presented in Table 2, which we found by experiments, is shown in Listing 1∼7.

```
// input MSB: X[4], LSB: X[0]
// output MSB: Y[4], LSB: Y[0]
T[0] = X[3]&X[4];
T[1] = X[0]&(X[2]^T[0]);
T[2] = (X[1]^T[1])&(X[2]^T[0]) ;
T[3] = (X[1]^X[2]^X[4]^T[0]^T[1])&(X[0]^X[3]);

Y[0] = X[0]^X[1]^X[2]^T[0]^T[1]^T[2];
Y[1] = X[1]^T[1]^T[3];
Y[2] = X[1]^X[2]^X[4]^T[0]^T[1];
Y[3] = X[0]^X[3];
Y[4] = X[1]^X[4]^T[1];
```

Listing 1 5-bit S-box with MC 4 (Differential uniformity 8, Linearity 32

These are written in the C language. In each listing, X is an input bit string, Y is an output bit string, and T is a temporary bit string.

```c
// input MSB: X[5], LSB: X[0]
// output MSB: Y[5], LSB: Y[0]
T[0] = X[0]&X[1];
T[1] = X[2]&X[3];
T[2] = X[4]&X[5];
T[3] = (X[0]^X[2]^X[4])&(X[1]^X[3]^X[5]);
T[4] = (X[0]^X[1]^X[3]^X[4])&(X[0]^X[2]^X[3]^X[5]);
T[5] = X[1]&X[4];

Y[0] = X[0]^T[0];
Y[1] = X[1]^T[1];
Y[2] = X[2]^T[2];
Y[3] = X[3]^T[3];
Y[4] = X[4]^T[4];
Y[5] = X[5]^T[5];
```

Listing 2 6-bit S-box with MC 6 (Differential uniformity 8, Linearity 32)

```c
// input MSB: X[5], LSB: X[0]
// output MSB: Y[5], LSB: Y[0]
T[0] = (X[0])&(X[1]);
T[1] = (X[2])&(X[3]);
T[2] = (X[4])&(X[5]);
T[3] = (X[1]^X[3]^X[5])&(X[0]^X[2]^X[4]^X[5]);
T[4] = (X[1]^X[2]^X[4])&(X[0]^X[2]^X[3]^X[5]);
T[5] = (X[1]^X[4])&(X[0]^X[5]);
T[6] = (X[0]^X[1]^X[3]^X[4])&(X[2]^X[3]^X[4]);

Y[0] = T[5]^T[6];
Y[1] = T[4]^T[6];
Y[2] = T[3]^T[6];
Y[3] = T[2]^T[6];
Y[4] = T[1]^T[6];
Y[5] = T[0]^T[6];
```

Listing 3 6-bit S-box with MC 7 (Differential uniformity 4, Linearity 16)

```
// input MSB: X[6], LSB: X[0]
// output MSB: Y[6], LSB: Y[0]
T[0] = X[0]&X[1];
T[1] = X[0]&X[2];
T[2] = X[3]&X[4];
T[3] = X[5]&X[6];
T[4] = (X[1]^X[3]^X[5])&(X[2]^X[4]^X[6]);

Y[0] = X[0]^T[0];
Y[1] = X[1]^T[1];
Y[2] = X[2]^T[2];
Y[3] = X[3]^T[3];
Y[4] = X[4]^T[4];
Y[5] = X[5];
Y[6] = X[6];
```

Listing 4 7-bit S-box with MC 5 (Differential uniformity 32, Linearity 128)

```
// input MSB: X[6], LSB: X[0]
// output MSB: Y[6], LSB: Y[0]
T[0] = (X[0])&(X[1]);
T[1] = (X[0])&(X[2]);
T[2] = (X[3])&(X[4]);
T[3] = (X[5])&(X[6]);
T[4] = (X[0]^X[1]^X[4]^X[5])&(X[2]^X[3]^X[4]^X[5]^X[6]);
T[5] = (X[1]^X[2]^X[3]^X[4]^X[5])&(X[0]^X[1]^X[3]^X[5]^X[6]);
T[6] = (X[1]^X[2]^X[4]^X[5]^X[6])&(X[1]^X[3]^X[4]^X[5]);
T[7] = (X[2]^X[3]^X[5]^T[1])&(X[3]^X[5]^T[1]);
T[8] = (X[1]^X[2]^X[3]^X[6])&(X[0]^X[2]^X[4]^X[5]);
T[9] = (X[0]^X[1]^X[3]^X[5]^X[6]^T[6])&(X[1]^X[4]^X[6]^T[6]);

Y[0] = T[9]^T[5]^T[3];
Y[1] = T[8]^T[5]^T[1];
Y[2] = T[7]^T[5];
Y[3] = T[6]^T[3]^T[1];
Y[4] = T[4]^T[1];
Y[5] = T[2]^T[5]^T[3]^T[1];
Y[6] = T[0]^T[5]^T[3]^T[1];
```

Listing 5 7-bit S-box with MC 10 (Differential uniformity 4, Linearity 32)

```
// input MSB: X[7], LSB: X[0]
// output MSB: Y[7], LSB: Y[0]
T[0] = (X[0])&(X[1])
T[1] = (X[2])&(X[3]);
T[2] = (X[4])&(X[5]^T[0]^T[1]);
T[3] = (X[6]^T[1]^T[2])&(X[7]^T[0]^T[1]^T[2]);
T[4] = (X[0]^X[2]^X[3]^X[5]^X[6]^X[7]^T[1])&(X[1]^X[3]^X[4]^X[6]\
^T[1]^T[2]);
T[5] = (X[0]^X[1]^X[2]^X[3]^X[5]^X[6]^T[0]^T[2])&(X[0]^X[2]^X[4]\
^X[7]^T[0]^T[1]^T[2]);
T[6] = (X[1]^X[5]^X[6]^X[7]^T[1])&(X[2]^X[6]^T[1]^T[2]);
T[7] = (X[0]^X[1]^X[3]^X[6]^X[7]^T[0]^T[3])&(X[1]^X[4]^X[5]^X[6]\
^T[0]^T[2]);

Y[0] = T[7]^X[1]^X[3]^X[4];
Y[1] = T[6]^X[3]^X[4];
Y[2] = T[5]^X[0]^X[1]^X[2]^X[3];
Y[3] = T[4]^X[0]^X[2];
Y[4] = T[3]^X[1]^X[2]^X[3];
Y[5] = T[2]^X[0]^X[1]^X[2]^X[4]^X[5]^X[7];
Y[6] = T[1]^X[1]^X[5]^X[6]^X[7];
Y[7] = T[0]^X[6]^X[7];
```

Listing 6 8-bit S-box with MC 8 (Differential uniformity 16, Linearity 128)

```
// input MSB: X[7], LSB: X[0]
// output MSB: Y[7], LSB: Y[0]
T[0] = (X[0])&(X[1]);
T[1] = (X[2])&(X[3]);
T[2] = (X[4])&(X[5]);
T[3] = (X[6]^T[2])&(X[7]^T[0]^T[2]);
T[4] = (X[0]^X[1]^X[2]^X[3]^X[5]^X[6]^X[7]^T[0])&(X[1]^X[3]^X[4]\
^X[5]^X[7]^T[0]^T[2]);
T[5] = (X[0]^X[1]^X[2]^X[4]^X[5])&(X[0]^X[3]^X[5]^X[7]^T[0]^T[2]);
T[6] = (X[0]^X[1]^X[2]^X[4]^X[7]^T[0]^T[2])&(X[4]^X[5]^X[6]^T[2]);
T[7] = (X[2]^X[3]^X[6]^X[7]^T[0])&(X[1]^X[2]^X[4]^X[6]^X[7]^T[0]);
T[8] = (X[1]^X[3]^X[6]^T[2])&(X[0]^X[3]^X[4]);
T[9] = (X[3]^X[4]^X[5]^X[6]^X[7]^T[0]^T[5])&(X[0]^X[1]^X[3]^X[4]\
^X[5]^X[7]^T[0]^T[2]);

Y[0] = T[9]^T[7];
Y[1] = T[8]^T[7];
Y[2] = T[6]^T[7];
Y[3] = T[5]^T[0];
Y[4] = T[4]^T[7]^T[0];
Y[5] = T[3]^T[7]^T[0];
Y[6] = T[2]^T[0];
Y[7] = T[1]^T[0];
```

Listing 7 8-bit S-box with MC 10 (Differential uniformity 8, Linearity 128)

Acknowledgements This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2017-0-00520, Development of SCR-Friendly Symmetric Key Cryptosystem and Its Application Modes)

References

1. Adomnicai, A., Berger, T.P., Clavier, C., Francq, J., Huynh, P., Lallemand, V., Le Gouguec, K., Minier, M., Reynaud, L., Thomas, G.: Lilliput-AE: a new lightweight tweakable block cipher for authenticated encryption with associated data. Submitted to NIST Lightweight Project (2019)
2. Albrecht, M.R., Rechberger, C., Schneider, T., Tiessen, T., Zohner, M.: Ciphers for MPC and FHE. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I, Lecture Notes in Computer Science, vol. 9056, pp. 430–454. Springer (2015). https://doi.org/10.1007/978-3-662-46800-5_17
3. Andreeva, E., Bilgin, B., Bogdanov, A., Luykx, A., Mendel, F., Mennink, B., Mouha, N., Wang, Q., Yasuda, K.: PRIMATEs v1.02. CAESAR submission. http://competitions.cr.yp.to/round2/primatesv102.pdf (2015)
4. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II, Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_5
5. Berger, T.P., Canteaut, A., Charpin, P., Laigle-Chapuy, Y.: On almost perfect nonlinear functions over F_2^n. IEEE Trans. Inf. Theory **52**(9), 4160–4170 (2006). https://doi.org/10.1109/TIT.2006.880036
6. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings, Lecture Notes in Computer Science, vol. 537, pp. 2–21. Springer (1990). https://doi.org/10.1007/3-540-38424-3_1
7. Bilgin, B., Meyer, L.D., Duval, S., Levi, I., Standaert, F.: Low AND depth and efficient inverses: a guide on s-boxes for low-latency masking. IACR Trans. Symmetric Cryptol. **2020**(1), 144–184 (2020). https://doi.org/10.13154/tosc.v2020.i1.144-184
8. Bilgin, B., Nikova, S., Nikov, V., Rijmen, V., Stütz, G.: Threshold implementations of all 3 ×3 and 4 ×4 s-boxes. In: Prouff, E., Schaumont, P. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings, Lecture Notes in Computer Science, vol. 7428, pp. 76–91. Springer (2012). https://doi.org/10.1007/978-3-642-33027-8_5
9. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings, Lecture Notes in Computer Science, vol. 4727, pp. 450–466. Springer (2007). https://doi.org/10.1007/978-3-540-74735-2_31
10. Boyar, J., Find, M.G.: Multiplicative complexity of vector valued Boolean functions. Theor. Comput. Sci. **720**, 36–46 (2018). https://doi.org/10.1016/j.tcs.2018.02.023
11. Boyar, J., Matthews, P., Peralta, R.: Logic minimization techniques with applications to cryptology. J. Cryptol. **26**(2), 280–312 (2013). https://doi.org/10.1007/s00145-012-9124-7
12. Boyar, J., Peralta, R.: Concrete multiplicative complexity of symmetric functions. In: Kralovic, R., Urzyczyn, P. (eds.) Mathematical Foundations of Computer Science 2006, 31st International Symposium, MFCS 2006, Stará Lesná, Slovakia, August 28-September 1, 2006, Proceedings, Lecture Notes in Computer Science, vol. 4162, pp. 179–189. Springer (2006). https://doi.org/10.1007/11821069_16
13. Boyar, J., Peralta, R., Pochuev, D.: On the multiplicative complexity of Boolean functions over the basis $(\wedge, \oplus, 1)$. Theor. Comput. Sci. **235**(1), 43–57 (2000). https://doi.org/10.1016/S0304-3975(99)00182-6
14. Bozilov, D., Bilgin, B., Sahin, H.A.: A note on 5-bit quadratic permutations' classification. IACR Trans. Symmetric Cryptol. **2017**(1), 398–404 (2017). https://doi.org/10.13154/tosc.v2017.i1.398-404
15. Canteaut, A., Perrin, L.: On CCZ-equivalence, extended-affine equivalence, and function twisting. Finite Fields Their Appl. **56**, 209–246 (2019). https://doi.org/10.1016/j.ffa.2018.11.008
16. Carlet, C., Ding, C.: Nonlinearities of S-boxes. Finite Fields Their Appl. **13**(1), 121–135 (2007). https://doi.org/10.1016/j.ffa.2005.07.003

17. Chabaud, F., Vaudenay, S.: Links between differential and linear cryptanalysis. In: Santis, A.D. (ed.) Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings, Lecture Notes in Computer Science, vol. 950, pp. 356–365. Springer (1994). https://doi.org/10.1007/BFb0053450
18. Chase, M., Derler, D., Goldfeder, S., Orlandi, C., Ramacher, S., Rechberger, C., Slamanig, D., Zaverucha, G.: Post-quantum zero-knowledge and signatures from symmetric-key primitives. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, pp. 1825–1842. ACM (2017). https://doi.org/10.1145/3133956.3133997
19. Courtois, N., Mourouzis, T., Hulme, D.: Exact logic minimization and multiplicative complexity of concrete algebraic and cryptographic circuits. Int. J. Adv. Intell. Syst. **6**(3), 165–176 (2013)
20. Courtois, N.T.: How fast can be algebraic attacks on block ciphers? In: Biham, E., Handschuh, H., Lucks, S., Rijmen, V. (eds.) Symmetric Cryptography, 07.01. - 12.01.2007, Dagstuhl Seminar Proceedings, vol. 07021. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany (2007). http://drops.dagstuhl.de/opus/volltexte/2007/1013
21. Courtois, N.T., Hulme, D., Mourouzis, T.: Solving Circuit Optimisation Problems in Cryptography and Cryptanalysis. IACR Cryptol. ePrint Arch. 2011, 475. http://eprint.iacr.org/2011/475 (2011)
22. Daemen, J., Rijmen, V.: The block cipher rijndael. In: Quisquater, J., Schneier, B. (eds.) Smart Card Research and Applications, This International Conference, CARDIS '98, Louvain-la-Neuve, Belgium, September 14-16, 1998, Proceedings, Lecture Notes in Computer Science, vol. 1820, pp. 277–284. Springer (1998). https://doi.org/10.1007/10721064_26
23. Giacomelli, I., Madsen, J., Orlandi, C.: ZKBoo: Faster zero-knowledge for boolean circuits. In: Holz, T., Savage, S. (eds.) 25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016, pp. 1069–1083. USENIX Association (2016). https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/giacomelli
24. Goudarzi, D., Jean, J., Kölbl, S., Peyrin, T., Rivain, M., Sasaki, Y., Sim, S.M.: Pyjamask: Block Cipher and Authenticated Encryption with Highly Efficient Masked Implementation. IACR Trans. Symmetric Cryptol. **2020**(S1), 31–59 (2020). https://doi.org/10.13154/tosc.v2020.iS1.31-59
25. Grosso, V., Leurent, G., Standaert, F., Varici, K.: LS-designs: Bitslice encryption for efficient masked software implementations. In: Cid, C., Rechberger, C. (eds.) Fast Software Encryption - 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers, Lecture Notes in Computer Science, vol. 8540, pp. 18–37. Springer (2014). https://doi.org/10.1007/978-3-662-46706-0_2
26. Halevi, S., Shoup, V.: Algorithms in HElib. In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I, Lecture Notes in Computer Science, vol. 8616, pp. 554–571. Springer (2014). https://doi.org/10.1007/978-3-662-44371-2_31
27. Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I., Manifavas, C.: A review of lightweight block ciphers. J. Cryptogr. Eng. **8**(2), 141–184 (2018). https://doi.org/10.1007/s13389-017-0160-y
28. Kim, H., Jeon, Y., Kim, G., Kim, J., Sim, B., Han, D., Seo, H., Kim, S., Hong, S., Sung, J., Hong, D.: A New Method for Designing Lightweight S-boxes with High Differential and Linear Branch Numbers, and Its Application. IACR Cryptol. ePrint Arch. 2020, 1582. https://eprint.iacr.org/2020/1582 (2020)
29. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings, Lecture Notes in Computer Science, vol. 1109, pp. 104–113. Springer (1996). https://doi.org/10.1007/3-540-68697-5_9
30. Kolesnikov, V., Schneider, T.: Improved garbled circuit: Free XOR gates and applications. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations, Lecture Notes in Computer Science, vol. 5126, pp. 486–498. Springer (2008). https://doi.org/10.1007/978-3-540-70583-3_40
31. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: Helleseth, T. (ed.) Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings, Lecture Notes in Computer Science, vol. 765, pp. 386–397. Springer (1993). https://doi.org/10.1007/3-540-48285-7_33
32. Songhori, E.M., Hussain, S.U., Sadeghi, A., Schneider, T., Koushanfar, F.: TinyGarble: Highly Compressed and Scalable Sequential Garbled Circuits. In: 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015, pp. 411–428. IEEE Computer Society (2015). https://doi.org/10.1109/SP.2015.32

33. Stoffelen, K.: Optimizing s-box implementations for several criteria using SAT solvers. In: Peyrin, T. (ed.) Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers, Lecture Notes in Computer Science, vol. 9783, pp. 140–160. Springer (2016). https://doi.org/10.1007/978-3-662-52993-5_8
34. Testa, E., Soeken, M., Amarù, L.G., Micheli, G.D.: Reducing the Multiplicative Complexity in Logic Networks for Cryptography and Security Applications. In: Proceedings of the 56th Annual Design Automation Conference 2019, DAC 2019, Las Vegas, NV, USA, June 02-06, 2019, p. 74. ACM (2019). https://doi.org/10.1145/3316781.3317893
35. Yao, A.C.: How to generate and exchange secrets (extended abstract). In: 27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986, pp. 162–167. IEEE Computer Society (1986). https://doi.org/10.1109/SFCS.1986.25
36. Zajac, P.: Constructing S-boxes with low multiplicative complexity. Stud. Sci. Math. Hung. **52**(2), 135–153 (2015)
37. Zajac, P., Jókay, M.: Multiplicative complexity of bijective 4×4 S-boxes. Cryptogr. Commun. **6**(3), 255–277 (2014). https://doi.org/10.1007/s12095-014-0100-y