Cryptogr. Commun. (2014) 6:255–277
DOI 10.1007/s12095-014-0100-y

Multiplicative complexity of bijective 4 × 4 S-boxes

Pavol Zajac · Matúš Jókay

Received: 20 September 2012 / Accepted: 9 April 2014 / Published online: 9 May 2014

Abstract Multiplicative complexity of S-box is the minimum number of 2-input AND-gates required to implement the S-box in AND, XOR, NOT logic. We show that under an affine equivalence there is only a single class of bijective $n \times n$ S-boxes with multiplicative complexity 1. Furthermore, we show that each bijective 4×4 S-box has multiplicative complexity at most 5. Finally, we refine the bounds on the multiplicative complexity of each affine class of bijective 4×4 S-boxes.

Keywords S-box · Multiplicative complexity · Affine equivalence

Mathematics Subject Classifications (2010) 94A60 · 06E30

1 Introduction

A basic building block providing the security of many cipher designs is an S-box. From the mathematical point of view, it is simply a vectorial Boolean function, with specific properties induced by security goals. In practice, an S-box can be implemented as a look-up table, a (relatively complex) electronic circuit, or as a specific sequence of (logic) instructions. When designing ciphers we must balance the security and effectiveness of the implementation. Although larger S-boxes have better resistance against linear and differential cryptanalysis, they are more difficult to implement: they take larger chip area, more operations operations in software and more memory for lookup tables. Another concern is the

This research was supported by grants APVV-0513-10 and APVV-0586-11.

P. Zajac (✉) · M. Jókay
Institute of Computer Science and Mathematics, FEI STU, Ilkovičova 3, 812-19 Bratislava, Slovakia
e-mail: pavol.zajac@stuba.sk

presence of side-channel attacks, as it seems more difficult to protect designs with larger S-boxes.

Implementation concerns are especially important in lightweight cryptography [10]. Lightweight cryptography deals with implementations in resource constrained environments, such as smartcards or RFID tags. Lightweight cryptography might be especially important for securing solutions based on intelligent sensors, such as body sensor networks in eHealth and telemedicine solutions. The lightweight cipher Present [3] is standardized in ISO/IEC 29192-2:2012. A core building block of Present is a special 4×4 bijective S-box optimized from both security and implementation aspects.

Minimum size of a bijective non-linear S-box is 3×3, i.e., the S-box operates on 3-bit inputs and produces 3-bit outputs. A more practical size for an S-box is 4×4 due to the accepted word sizes in prevailing hardware. There are $16! \approx 2^{44}$ bijective 4×4 S-boxes. It is possible to study various properties of all of these S-boxes [1, 12, 16, 18] using fast affine equivalence algorithms [2]. We are interested in one specific property: multiplicative complexity. This property is important for various problems connected to S-boxes, such as logic circuit minimization, algebraic cryptanalysis, and optimal masking against higher order power analysis attacks.

Boyar and Peralta introduced a new technique for logic synthesis and circuit minimization based on the notion of multiplicative complexity. They define multiplicative complexity (MC) of the circuit as a minimum number of AND gates required to implement a circuit in (AND, XOR) algebra (all other logic gates can be constructed with these two). Although MC addresses only a part of an overall Gate complexity (XOR gates are not counted) Boyar and Peralta formulate a hypothesis that

> it is plausible that a two-step process, which first reduces multiplicative complexity and then optimizes linear components, leads to small circuits [5].

Courtois et. al. [9] introduced new tools to compute MC for small S-boxes. They also conjectured that MC of whole ciphers plays a significant role in algebraic cryptanalysis. In [19] we described a new method suitable for algebraic cryptanalysis that has a complexity closely related only to the number of non-linear operations (and thus MC of the related circuit).

The number of non-linear operations in hardware realization of S-boxes is also important for implementations resistant against the first-order DPA [1]. However, in this area the complexity is usually expressed in the number of $GF(2^n)$ multiplications instead of just $GF(2)$ multiplications (AND gates) [7, 15]. We discuss the problem of connection between $GF(2^n)$-multiplicative complexity and $GF(2)$-multiplicative complexity in Section 6.

It is easy to show that MC is invariant under affine transforms of the S-box. Thus it is possible to study MC of affine classes of S-boxes instead of individual ones. We investigate S-boxes with low multiplicative complexity. Our contribution is two-fold. First, we show that there is only a single affine class of $n \times n$ S-boxes with MC equal to 1 for any n. Then we explore small 3×3 S-boxes, and show that all 4 affine classes can be generated as a composition of the single MC1 S-box. We show that this result does not hold for larger n's.

Our additional results for $n = 4$ are mostly computer generated. We have implemented an algorithm that allows us to enumerate affine classes of S-boxes up to a given MC (feasible up to 4). Then, using composition of S-boxes of MC at most 3, we have found that the limit on MC in case $n = 4$ is 5. I.e., each 4×4 S-boxes can be realized using at most 5 AND-gates. We provide the statistics and the list of representatives of the affine classes along with their multiplicative complexity.

2 Preliminaries

In the article, the term (n-bit) S-box will denote a bijective vectorial Boolean function $S : GF(2)^n \rightarrow GF(2)^n$. An affine mapping is a bijective vectorial Boolean function $A : GF(2)^n \rightarrow GF(2)^n$, $A(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x} + \mathbf{c}$, where $\mathbf{c} \in GF(2)^n$, and $\mathbf{A} \in GF(2)^{(n \times n)}$ is an invertible $n \times n$ matrix over $GF(2)$. If $\mathbf{c} = \mathbf{0}$, mapping A is linear. As usual, we denote the set of affine mappings over $GF(2)^n$ by $Aff(2, n)$.

Let us define a relation: $S_1 \sim S_2$ iff there exist two affine mappings A_1, A_2, such that $A_1 \circ S_1 = S_2 \circ A_2$. It is easy to show that $\sim$ is an equivalence relation. We will call $S_1 \sim S_2$ affinely equivalent.

We will call an S-box with the property $S(\mathbf{0}) = \mathbf{0}$ a *constant-free* S-box. From any S-box we can get an affine equivalent constant-free S-box by using the affine mapping $S(\mathbf{x}) \mapsto S(\mathbf{x}) + S(\mathbf{0})$. Every bijective S-box is affinely equivalent to a function that keeps the canonical basis invariant (see Lemma 1), i.e., $S(\mathbf{e}^{(i)}) = \mathbf{e}^{(i)}$, where $\mathbf{e}^{(i)}$ denotes a vector with a single one on i-th position. We call a bijective constant-free function that keeps canonical basis invariant a *normalized* S-box. Each affine class of bijective S-boxes contains a normalized S-box.

Lemma 1 *Let $S : GF(2)^n \rightarrow GF(2)^n$ be a bijective function. Then there exist two non-singular matrices $\mathbf{A}, \mathbf{B} \in GF(2)^{(n \times n)}$, and a constant vector $\mathbf{c} \in GF(2)^n$, such that the function $F : GF(2)^n \rightarrow GF(2)^n$, $F(\mathbf{x}) = \mathbf{A} \cdot S(\mathbf{B} \cdot \mathbf{x}) + \mathbf{c}$ is a normalized S-box. That is, $F(\mathbf{0}) = \mathbf{0}$, and $F(\mathbf{e}^{(i)}) = \mathbf{e}^{(i)}$ for $i = 1, 2, \ldots, n$.*

Proof Let us define a new notation useful for the proof: let $\mathbf{M}$ be an arbitrary $n \times n$ matrix. We denote the i-the column of $\mathbf{M}$ by $\mathbf{M}^{(i)}$. Note that $\mathbf{M}^{(i)} = \mathbf{M} \cdot \mathbf{e}^{(i)}$.

Let $S_{\mathbf{B}}$ denote a vectorial Boolean function given by a choice of $n \times n$ matrix $\mathbf{B}$:

$$S_{\mathbf{B}} = S(\mathbf{B} \cdot \mathbf{x}) + S(\mathbf{0}).$$

It is easy to see that $S_{\mathbf{B}}$ is a bijection only if $\mathbf{B}$ is non-singular. We also note that $S_{\mathbf{B}}(\mathbf{0}) = \mathbf{0}$. Let us construct an $n \times n$ matrix $\mathbf{M_B}$ with columns given by $\mathbf{M}_{\mathbf{B}}^{(i)} = S_{\mathbf{B}}(\mathbf{e}^{(i)})$. If matrix $\mathbf{M_B}$ is non-singular, then we can define a vectorial Boolean function F as $F(\mathbf{x}) = \mathbf{M}_{\mathbf{B}}^{-1} \cdot S_{\mathbf{B}}(\mathbf{x}) = \mathbf{M}_{\mathbf{B}}^{-1} \cdot S(\mathbf{B} \cdot \mathbf{x}) + \mathbf{M}_{\mathbf{B}}^{-1} \cdot S(\mathbf{0})$. It is easy to verify that F has the desired properties $F(\mathbf{0}) = \mathbf{0}$, and $F(\mathbf{e}^{(i)}) = \mathbf{e}^{(i)}$. Thus, for a given $\mathbf{B}$, we get $\mathbf{A} = \mathbf{M}_{\mathbf{B}}^{-1}$, and $\mathbf{c} = \mathbf{M}_{\mathbf{B}}^{-1} \cdot S(\mathbf{0})$, respectively.

To prove Lemma 1, we must show that for an arbitrary S-box S we can always find a suitable non-singular matrix $\mathbf{B}$ producing a corresponding non-singular matrix $\mathbf{M_B}$.

Recall that columns of $\mathbf{B}$ and $\mathbf{M_B}$ are connected by the following equation:

$$\mathbf{M}_{\mathbf{B}}^{(i)} = S_{\mathbf{B}}(\mathbf{e}^{(i)}) = S(\mathbf{B} \cdot \mathbf{e}^{(i)}) + S(\mathbf{0}) = S(\mathbf{B}^{(i)}) + S(\mathbf{0}). \tag{1}$$

Because S is a bijection, there is a single vector $\mathbf{M}_{\mathbf{B}}^{(i)}$ corresponding to a given $\mathbf{B}^{(i)}$ (and vice-versa). Moreover, since $S_{\mathbf{B}}$ has a fixed point at zero, a non-zero column $\mathbf{B}^{(i)}$ always corresponds to a non-zero column $\mathbf{M}_{\mathbf{B}}^{(i)}$.

Let $\mathbf{I}$ be the $n \times n$ identity matrix. Let $\mathbf{M_I}$ have $t < n$ linearly independent columns (if $t = n$, we can set $\mathbf{B} = \mathbf{I}$ and we are done). We can swap columns of $\mathbf{M_I}$, along with the corresponding columns of $\mathbf{I}$ to produce a permutation matrix $\mathbf{B}_t$, for which exactly the first t columns of $\mathbf{M}_{\mathbf{B}_t}$ are linearly independent (recall that $\mathbf{M_I} = S(\mathbf{e}^{(i)}) + S(\mathbf{0})$, and $\mathbf{M}_{\mathbf{B}_t}^{(i)} = S(\mathbf{M}_{\mathbf{B}_t}^{(i)}) + S(\mathbf{0})$, respectively). We now have two corresponding matrices $\mathbf{B}_t, \mathbf{M}_{\mathbf{B}_t}$, with first $t < n$ columns linearly independent.

Let $\mathcal{S}$ denote a set of vectors $\mathbf{v}$ for which $\mathbf{u} = S(\mathbf{v}) + S(\mathbf{0})$ (see (1)) is not a linear combination of vectors $\mathbf{M}_{\mathbf{B}_t}^{(1)}, \mathbf{M}_{\mathbf{B}_t}^{(2)}, \ldots, \mathbf{M}_{\mathbf{B}_t}^{(t)}$. There are 2^t forbidden vectors $\mathbf{u}$, thus $|\mathcal{S}| = 2^n - 2^t$. Note that $\mathbf{0} \notin \mathcal{S}$. Let us further exclude from $\mathcal{S}$ all non-zero linear combinations of vectors $\mathbf{B}_t^{(1)}, \mathbf{B}_t^{(2)}, \ldots, \mathbf{B}_t^{(t)}$, giving us set $\mathcal{S}'$. There are $2^t - 1$ non-zero linear combinations of vectors $\mathbf{B}_t^{(1)}, \mathbf{B}_t^{(2)}, \ldots, \mathbf{B}_t^{(t)}$, thus $|\mathcal{S}'| \geq |\mathcal{S}| - (2^t - 1) = 2^n - 2^{t+1} + 1$. Using $t < n$, we get $|\mathcal{S}'| \geq 1$, so there exists at least one vector in $|\mathcal{S}'|$. Now let $\mathbf{B}_{t+1}^{(i)} = \mathbf{B}_t^{(i)}$ for each $i \neq t + 1$, and let us select any vector $\mathbf{B}_{t+1}^{(t+1)} \in \mathcal{S}'$. This choice ensures that the first $t + 1$ columns of both $\mathbf{B}_{t+1}$, and the corresponding $\mathbf{M}_{\mathbf{B}_{t+1}}$ are linearly independent.

We can now repeat this procedure with matrices $\mathbf{B}_{t+1}, \mathbf{B}_{t+2} \ldots, \mathbf{B}_{n-1}$. Final pair $\mathbf{B}_n$, $\mathbf{M}_{\mathbf{B}_n}$ will be a pair of required non-singular matrices, which completes the proof. □

There are different systems of representatives for known affine classes of S-boxes [1, 6, 12, 16]. We propose to use a combination of [16], and [1]: As a natural representative we use the first normalized S-box in a lexicographic order. In the appendix we use the numbering of classes from [1], and our system of representatives.

We denote a normalized representative of S-box S by S^*. The following conditions hold:

1. $S \sim S^*$,
2. S^* is normalized,
3. for each $S_1 \sim S$: $S^* \leq_{\text{lex}} S_1$.

All affine mappings are affinely equivalent with the identity mapping. For $n > 2$, all affine mappings are even permutations, thus all permutations in an affine class are either odd or even.

Definition 1 Multiplicative complexity of Boolean function $F : GF(2)^n \to GF(2)^n$ is the (minimum) number of $GF(2)$ multiplications sufficient and necessary to compute $F(\mathbf{x})$ for any $\mathbf{x}$.

We will denote multiplicative complexity of function F by $MC(F)$. Equivalent definition of multiplicative complexity is based on the number of (2-input) AND-gates in $(\wedge, \oplus, 1)$ algebra, where $\wedge$ (AND) is multiplication in $GF(2)$, $\oplus$ is addition in $GF(2)$, and negation is computed by adding a constant, i.e., $x \oplus 1$. We do not need to consider the addition of constants for constant-free S-boxes (i.e., $S(\mathbf{0}) = \mathbf{0}$) (see Lemma 2 in [11]).

Multiplicative complexity has been intensively studied in the context of quadratic forms [14, 17], simple Boolean predicates [11], and symmetric functions [4]. However, not much is known in the case of vectorial Boolean functions (and S-boxes).

It is easy to see that multiplicative complexity of all S-boxes in an affine class is the same. Thus it is sufficient to compute the multiplicative complexity of the selected representatives of affine classes. One possible practical approach to computing multiplicative complexity was presented by Courtois [9]. In this case, a problem of computing $MC(S)$ is converted to an instance of SAT, and verified by a SAT solver (top-down approach). Another practical approach is presented by Ullrich et. al [18]. The main idea is to go through all possible combinations of an instruction set (in case of constant-free S-boxes these consists only of AND, and XOR instructions) by a search, until each class is enumerated (bottom-up approach). We use the bottom-up approach, however, unlike in [18], our approach presented in Section 5 focuses strictly on multiplications. Furthermore, it is complemented by a different approach based on the composition, using the fact that $MC(S_1 \circ S_2) \leq MC(S_1) + MC(S_2)$.

3 Bijective S-boxes with multiplicative complexity 1

The purpose of this section is to show that there is only a single affine class of bijective S-boxes with multiplicative complexity 1 for any $n \geq 3$. Moreover, we can choose a very specific representative of this class, which can be realized by a generalization of the circuit depicted in Fig. 1. We formalize this result in Theorem 1, and the rest of this section contains the proof of this theorem.

Theorem 1 *Let $S : GF(2)^n \to GF(2)^n$ be a bijective vectorial Boolean function with multiplicative complexity 1. Then for $n \geq 3$ it is affinely equivalent to $\Lambda_n : GF(2)^n \to GF(2)^n$,*

$$\Lambda_n(\mathbf{x}) = (x_1 + x_{n-1}x_n, x_2, \ldots, x_{n-1}, x_n).$$

Proof To prove the Theorem 1, we will first restrict the search using Lemma 2. Then we need to prove Lemma 4 that provides a generic formula S-boxes with multiplicative complexity 1. For the proof of Lemma 4, we will use Lemma 3. Afterwards, we finish the proof by using affine equivalence. The first two lemmas are rather trivial, but we include them for the sake of completeness. Lemma 4 is also relatively straightforward, as there are not many choices we can make if we can only use a single AND gate to construct a Boolean function. However, a special care is needed to show the conditions that guarantee that this function is bijective.

Lemma 2 *Any S-box S is affinely equivalent to S-box $S_\mathbf{c}$ given by $S_\mathbf{c}(x) = S(x) + \mathbf{c}$.* □

Lemma 2 is trivially derived from the definition of affine equivalence.

If we choose $\mathbf{c} = S(\mathbf{0})$, we get a constant free S-box with $S_\mathbf{c}(\mathbf{0}) = \mathbf{0}$. Using Lemma 2, and the transitivity of affine equivalence, we only need to prove Theorem 1 for constant-free S-boxes.

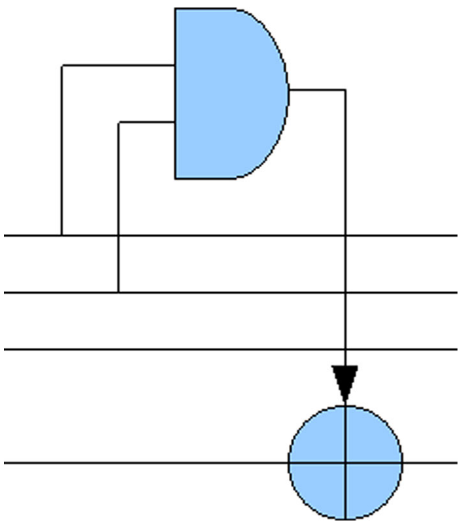

Fig. 1 The representative Λ_4 of the affine class of bijective 4×4 S-boxes with $MC(S) = 1$. Inputs and outputs are numbered from bottom up

Lemma 3 *Let $f, g, h : GF(2)^n \to GF(2)$, and let $h(\mathbf{x}) = f(\mathbf{x}) \cdot g(\mathbf{x})$. If f, g are distinct balanced Boolean functions, then their product h is not balanced.*

Proof Function f is balanced, so there are exactly 2^{n-1} points where $f(\mathbf{x}) = 1$. We can only get $h(\mathbf{x}) = 1$, if both $f(\mathbf{x}) = 1$, and $g(\mathbf{x}) = 1$. Functions f, and g are distinct, so there is at least one point such that $f(\mathbf{x}) = 1$, and $g(\mathbf{x}) = 0$. Thus there are at most $2^{n-1} - 1$ points, where $h(\mathbf{x}) = 1$, which means that h is not balanced. □

It is a well known that all (non-constant) affine Boolean functions are balanced. Thus, a corollary of Lemma 3 is that a product of two affine Boolean functions is not balanced. We use this fact in the proof of Lemma 4.

Lemma 4 *Any bijective n-bit S-box S with $S(\mathbf{0}) = \mathbf{0}$, and multiplicative complexity $MC(S) = 1$, can be written in the form*

$$S(\mathbf{x}) = \mathbf{Mx} + \left((\mathbf{a}^T\mathbf{x}) \cdot (\mathbf{b}^T\mathbf{x})\right)\mathbf{d}, \tag{2}$$

where $\mathbf{a}, \mathbf{b}, \mathbf{d} \in GF(2)^n \setminus \{\mathbf{0}\}$, $\mathbf{a} \neq \mathbf{b}$, $\mathbf{M}$ is an invertible $n \times n$ matrix over $GF(2)$, and $\mathbf{a}^T\mathbf{M}^{-1}\mathbf{d} = \mathbf{b}^T\mathbf{M}^{-1}\mathbf{d} = 0$.

Proof It is easy to see that any S given by (2) has MC at most 1, and that (2) covers any Boolean function that can be realized by a single $GF(2)$ multiplication. We must show that conditions of Lemma 4 are necessary and sufficient for S to be a non-linear bijection.

If $\mathbf{a} = \{\mathbf{0}\}$, $\mathbf{b} = \{\mathbf{0}\}$, or $\mathbf{d} = \{\mathbf{0}\}$, formula (2) is reduced to $S(\mathbf{x}) = \mathbf{Mx}$, which means that S is a linear function. Similarly, if $\mathbf{a} = \mathbf{b}$, we get

$$S(\mathbf{x}) = \mathbf{Mx} + (\mathbf{a}^T\mathbf{x})\mathbf{d} = \left(\mathbf{M} + \mathbf{da}^T\right)\mathbf{x},$$

which is also a linear function. In any other case the non-linear terms provided by $\left((\mathbf{a}^T\mathbf{x}) \cdot (\mathbf{b}^T\mathbf{x})\right)\mathbf{d}$ cannot be cancelled out. Thus S is non-linear (with $MC(S) = 1$), if and only if the following conditions hold: $\mathbf{a}, \mathbf{b}, \mathbf{d} \neq \{\mathbf{0}\}$, and $\mathbf{a} \neq \mathbf{b}$, respectively.

The remaining conditions of Lemma 4 are needed to ensure that S is a bijective function. First we will show that if $\mathbf{M}$ is singular, then S cannot be bijective. Let $\mathbf{u}$ be any non-zero vector from the kernel of the mapping $\mathbf{Mx}$. We can find 2^{n-1} pairs of vectors $(\mathbf{x}_1, \mathbf{x}_2 = \mathbf{x}_1 + \mathbf{u})$, such that $\mathbf{Mx}_1 = \mathbf{Mx}_2$. If S is a bijection than for each pair $\mathbf{x}_1 \neq \mathbf{x}_2$ we must get $S(\mathbf{x}_1) \neq S(\mathbf{x}_2)$, or equivalently $S(\mathbf{x}_1) + S(\mathbf{x}_2) \neq \mathbf{0}$.

We can rewrite this using (2) to:

$$S(\mathbf{x}_1) + S(\mathbf{x}_2) = \left((\mathbf{a}^T\mathbf{x}_1) \cdot (\mathbf{b}^T\mathbf{x}_1) + (\mathbf{a}^T\mathbf{x}_2) \cdot (\mathbf{b}^T\mathbf{x}_2)\right)\mathbf{d} \neq \mathbf{0},$$

and thus

$$\left((\mathbf{a}^T\mathbf{x}_1) \cdot (\mathbf{b}^T\mathbf{x}_1) \neq (\mathbf{a}^T\mathbf{x}_2) \cdot (\mathbf{b}^T\mathbf{x}_2)\right). \tag{3}$$

Let $g(\mathbf{x}) = (\mathbf{a}^T\mathbf{x}) \cdot (\mathbf{b}^T\mathbf{x})$. Condition (3) means that g must be a balanced Boolean functions, because we must choose the pairs $(\mathbf{x}_1, \mathbf{x}_2)$ in such a way that $g(\mathbf{x}_1) = 0$, and $g(\mathbf{x}_2) = 1$. On the other hand, g is a product of two distinct linear functions $\mathbf{a}^T\mathbf{x}$, and $\mathbf{b}^T\mathbf{x}$, which are balanced, and according to Lemma 3, g is not balanced. So there is no suitable function g, and thus S cannot be a bijection if there is a non-zero vector $\mathbf{u}$ in the kernel of the mapping $\mathbf{Mx}$, i.e., if $\mathbf{M}$ is singular. Thus if S is bijective, $\mathbf{M}$ must be an invertible matrix. On the other hand, just the condition that $\mathbf{M}$ is invertible is not sufficient for bijective S.

Finally, we must show (by contradiction) that last two conditions $\mathbf{a}^T\mathbf{M}^{-1}\mathbf{d} = 0$, and $\mathbf{b}^T\mathbf{M}^{-1}\mathbf{d} = 0$, are necessary and sufficient for bijective S. Without the loss of generality, let $\mathbf{a}^T \cdot \mathbf{M}^{-1}\mathbf{d} = 1$ (similarly for $\mathbf{b}$). Let us consider function $h(\mathbf{x}) = \mathbf{a}^T\mathbf{M}^{-1}S(\mathbf{x})$. If S is bijection then h must be a balanced Boolean function [13].

However, if we rewrite h using formula (2):

$$\begin{aligned} h(\mathbf{x}) &= \mathbf{a}^T\mathbf{M}^{-1}\mathbf{M}x + \left((\mathbf{a}^T\mathbf{x}) \cdot (\mathbf{b}^T\mathbf{x})\right)\mathbf{a}^T \cdot \mathbf{M}^{-1}\mathbf{d} \\ &= \mathbf{a}^T\mathbf{x} + (\mathbf{a}^T\mathbf{x}) \cdot (\mathbf{b}^T\mathbf{x}) \\ &= (\mathbf{a}^T\mathbf{x}) \cdot (1 \oplus \mathbf{b}^T\mathbf{x}), \end{aligned}$$

we can see that h is a product of two distinct affine functions. According to Lemma 3, h cannot not balanced. Thus we get a contradiction, so both $\mathbf{a}^T\mathbf{M}^{-1}\mathbf{d} = 0$, and $\mathbf{b}^T\mathbf{M}^{-1}\mathbf{d} = 0$, must hold to get bijective S.

On the other hand, let us suppose that S is not bijective, i.e., $S(\mathbf{x}_1) = S(\mathbf{x}_2)$ for some $\mathbf{x}_1 \neq \mathbf{x}_2$. From $S(\mathbf{x}_1) = S(\mathbf{x}_2)$ we can derive that

$$\mathbf{x}_1 + \mathbf{x}_2 = \left((\mathbf{a}^T\mathbf{x}_1) \cdot (\mathbf{b}^T\mathbf{x}_1) + (\mathbf{a}^T\mathbf{x}_1) \cdot (\mathbf{b}^T\mathbf{x}_1)\right)\mathbf{M}^{-1}\mathbf{d}$$

Multiplying by $\mathbf{a}^T$, and $\mathbf{b}^T$, we get that

$$\mathbf{a}^T(\mathbf{x}_1 + \mathbf{x}_2) = \mathbf{b}^T(\mathbf{x}_1 + \mathbf{x}_2) = \mathbf{0}, \tag{4}$$

or equivalently

$$\mathbf{a}^T\mathbf{x}_1 = \mathbf{a}^T\mathbf{x}_2, \quad \text{and} \quad \mathbf{b}^T\mathbf{x}_1 = \mathbf{b}^T\mathbf{x}_2. \tag{5}$$

Rewriting $S(\mathbf{x}_1 + \mathbf{x}_2)$ using (2) yields

$$S(\mathbf{x}_1 + \mathbf{x}_2) = \left((\mathbf{a}^T\mathbf{x}_1) \cdot (\mathbf{b}^T\mathbf{x}_2) + (\mathbf{a}^T\mathbf{x}_2) \cdot (\mathbf{b}^T\mathbf{x}_1)\right)\mathbf{d}.$$

Using (5), we finally get $S(\mathbf{x}_1 + \mathbf{x}_2) = \mathbf{0}$. But S is constant-free, so $\mathbf{x}_1 + \mathbf{x}_2 = \mathbf{0}$. This is a contradiction, so S must be a bijection. □

Now, let us return the the proof of Theorem 1. Conditions of Lemma 4 are easy to verify for S-box Λ_n. We have $\mathbf{M} = \mathbf{I}$ (identity matrix), and $\mathbf{a} = \mathbf{e}^{(n-1)}$, $\mathbf{b} = \mathbf{e}^{(n)}$, $\mathbf{d} = e^{(1)}$, where vector $\mathbf{e}^{(i)} \in GF(2)^n$ has only a single one on i-th position. Using $(\mathbf{e}^{(i)})^T\mathbf{e}^{(j)} = 0$, for $i \neq j$, it is easy to see that $\mathbf{a}^T\mathbf{M}^{-1}\mathbf{d} = \mathbf{b}^T\mathbf{M}^{-1}\mathbf{d} = 0$.

Lemma 4 tells us how all bijective constant-free Boolean functions with multiplicative complexity 1 look like. Now we would like to show that for any permissible choice of parameters we can find two invertible matrices $\mathbf{A}, \mathbf{B}$, such that $S(\mathbf{x}) = \mathbf{B}\Lambda_n(\mathbf{A}\mathbf{x})$.

Let $\mathbf{A}$ be an invertible matrix chosen (at first) arbitrarily, with rows denoted by $\mathbf{u}_1^T, \ldots, \mathbf{u}_n^T$. We remark that $(e^{(i)})^T\mathbf{A} = \mathbf{u}_i^T$. Let $\mathbf{B} = \mathbf{M}\mathbf{A}^{-1}$. For linearly equivalent S-box S we can write

$$S(\mathbf{x}) = \mathbf{B}\Lambda_n(\mathbf{A}\mathbf{x}) = \mathbf{B}\mathbf{A}\mathbf{x} + \left((\mathbf{u}_{n-1}^T\mathbf{x}) \cdot (\mathbf{u}_n^T\mathbf{x})\right)\mathbf{B}\mathbf{e}^{(1)} \tag{6}$$

$$= \mathbf{M}\mathbf{x} + \left((\mathbf{u}_{n-1}^T\mathbf{x}) \cdot (\mathbf{u}_n^T\mathbf{x})\right)(\mathbf{M}\mathbf{A}^{-1}\mathbf{e}^{(1)}). \tag{7}$$

Comparing (6) with (2), we require that $\mathbf{u}_{n-1} = \mathbf{a}$, $\mathbf{u}_n = \mathbf{b}$, and $\mathbf{M}\mathbf{A}^{-1}\mathbf{e}^{(1)} = \mathbf{d}$, respectively. We must show that under these conditions it is still possible to construct matrix $\mathbf{A}$ for any permissible $\mathbf{a}$, $\mathbf{b}$, $\mathbf{M}$, $\mathbf{d}$.

Conditions of Lemma 4 $\mathbf{a}, \mathbf{b} \neq \mathbf{0}$, and $\mathbf{a} \neq \mathbf{b}$ guarantee that the last two rows of matrix $\mathbf{A}$ are linearly independent.

Matrix $\mathbf{M}$ is invertible, so we can rewrite $\mathbf{M}\mathbf{A}^{-1}\mathbf{e}^{(1)} = \mathbf{d}$ as $\mathbf{A}^{-1}\mathbf{e}^{(1)} = \mathbf{M}^{-1}\mathbf{d}$. In other words, the first column of matrix A^{-1} must be equal to $\mathbf{M}^{-1}\mathbf{d}$.

Using identity $\mathbf{A}\mathbf{A}^{-1} = I$ we get

$$\mathbf{u}_i^T \cdot \mathbf{M}^{-1}\mathbf{d} = \begin{cases} 1 \text{ if } i = 1, \\ 0 \text{ otherwise.} \end{cases} \tag{8}$$

Conditions $\mathbf{a}^T\mathbf{M}^{-1}\mathbf{d} = \mathbf{b}^T\mathbf{M}^{-1}\mathbf{d} = 0$ of Lemma 4 guarantee that these conditions hold for prescribed vectors $\mathbf{u}_{n-1} = \mathbf{a}$, and $\mathbf{u}_n = \mathbf{b}$, respectively. We can always choose the set of remaining $n-3$, such that all $\mathbf{u}_i$ are linearly independent, and conditions (8) hold. E.g., we can choose $\mathbf{u}_1 = \mathbf{e}^{(j)}$, where j is the position of the first non-zero bit of $\mathbf{M}^{-1}\mathbf{d}$. For other vectors, we can try to use remaining basis vectors. If we get a conflict $\mathbf{u}_i\mathbf{M}^{-1}\mathbf{d} = 1$, we replace the offending vector by $\mathbf{u}_i + \mathbf{u}_1$.

This completes the proof that any *constant-free* S-box S with $n \geq 3$, and $MC(S) = 1$, if affinely (even linearly) equivalent with Λ_n. Now using Lemma 2 we can also drop the condition $S(\mathbf{0}) = \mathbf{0}$, and thus finish the proof of Theorem 1.

4 Bijective 3 x 3 S-boxes

In case of $n = 3$, there are only 4 affine classes of 3×3 bijective S-boxes [1]. Thus, the situation with multiplicative complexity can be examined easily:

1. All affine S-boxes are in the same class $\mathcal{A}_0^3$ as the identity permutation `01234567`.
2. Permutation Λ_3 (`01234576`), class $\mathcal{Q}_1^3$ is given by a single swap, thus it is an odd permutation. According to Theorem 1, Λ_3 is the only affine class with multiplicative complexity 1.
3. A representative `01234756` of class $\mathcal{Q}_2^3$ can be written as $\Lambda_3 \circ \mathrm{rot}_{-1} \circ \Lambda_3 \circ \mathrm{rot}_1$, where rot_n denotes a rotation of a bit vector by n positions ($x_i \mapsto x_{i+n}$), giving $MC(\mathcal{Q}_2^3) \leq 2$. Using Theorem 1, we can see that $MC(\mathcal{Q}_2^3) = 2$.
4. Finally, representative `01254736` of class $\mathcal{Q}_3^3$ can be constructed as depicted in Fig. 2. Thus $MC(\mathcal{Q}_3^3) \leq 3$. Theorem 1 gives $MC(\mathcal{Q}_3^3) > 1$.

We remark that in the case $n = 3$, each class can be generated using the construction $S = (\Lambda_3 \circ \mathrm{rot}_{-1})^c$, where c is the desired multiplicative complexity.

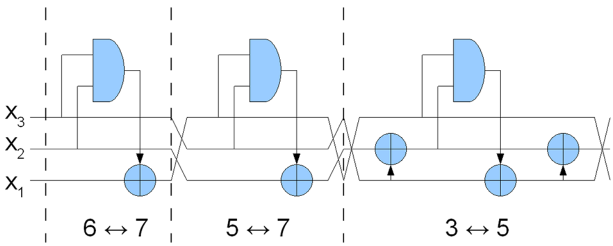

Fig. 2 Construction of representatives of affine classes of 3×3 S-boxes

5 Multiplicative complexity of bijective 4 × 4 S-boxes

The situation for $n = 4$ is more complicated, as there are are 16! S-boxes (11! normalized) in 302 affine classes of 4×4 bijective S-boxes [1]. It is still feasible to examine multiplicative complexities of 4×4 bijective S-boxes using a reasonable amount of computing power. Recall that we only need to compute multiplicative complexity of the representatives of affine classes.

First, let us extend the $*$-notation to sets of S-boxes. Let $\mathcal{S}$ be a set of S-boxes. By $\mathcal{S}^*$ we denote a set of representatives of affine classes of S-boxes in $\mathcal{S}$. That is, $\mathcal{S}^* = \{S^*; S \in \mathcal{S}\}$.

We compute the multiplicative complexity of affine classes of S-boxes using the following idea. Let $\mathcal{M}_c$ be a set of all S-boxes with $MC(S) \leq c$. Then clearly $MC(S) = c$ for each $S \in \mathcal{M}_c \setminus \mathcal{M}_{c-1}$. Set $\mathcal{M}_c$ is defined by these 2 conditions:

1. for each $S \in \mathcal{M}_c$, $MC(S) \leq c$;
2. if $MC(S) \leq c$, then $S \in \mathcal{M}_c$;

A set that fulfils condition 1 can be constructed by defining a set of circuits that use at most c 2-input AND gates. However, it is more difficult to ensure that condition 2 holds. As $\mathcal{M}_c$ is large, in practice we want to work with the set $\mathcal{M}_c^*$ instead.

Affine transformations do not require any multiplications, and non-linear transformations require at least one multiplication. Thus $\mathcal{M}_0^* = Aff(2, n) = \{id\}$. In Section 3 we have shown that there is only one class of S-boxes with multiplicative complexity 1, so we also know that[1]

$$\mathcal{M}_1^* = \{id, \Lambda_n\}.$$

For larger c, it is more difficult to ensure the construction of representatives directly. Instead, we will construct a set $\mathcal{C}$ with the following properties:

- for each $S \in \mathcal{C}$, $MC(S) \leq c$;
- for each S with $MC(S) \leq c$, there exist $S_1 \in \mathcal{C}$ such that $S \sim S_1$;

Then $\mathcal{C}^* = \mathcal{M}_c^*$.

To produce sets $\mathcal{C}$ we use two constructions based on Lemma 5 (composition), and Lemma 6 (expansion and compression), respectively.

Lemma 5 *Let*

$$\mathcal{C}_{i,j} = \{S_2 \circ A \circ S_1; S_1 \in \mathcal{M}_i^*, S_2 \in \mathcal{M}_j^*, A \in Aff(2, n)\}.$$

Then for each $S \in \mathcal{C}_{i,j}$: $MC(S) \leq i + j$.

Proof S_1 from the definition of $\mathcal{C}_{i,j}$ can be constructed by using i $GF(2)$ multiplications, and S_2 using j multiplications, respectively. Affine transformation does not require multiplications. Thus any S-box in $\mathcal{C}_{i,j}$ can be constructed using at most $i + j$ multiplications. □

As seen in Section 4, Lemma 5 can be used to construct all affine classes of bijective 3×3 S-boxes: $\mathcal{M}_2^* = \mathcal{C}_{1,1}^*$, $\mathcal{M}_3^* = \mathcal{C}_{2,1}^*$. The situation is different for $n > 3$, as $\mathcal{M}_1^*$

[1] Λ_n is a proper representative of its class, if x_1/f_1 denotes the least significant bit of the corresponding encoding of inputs/outputs.

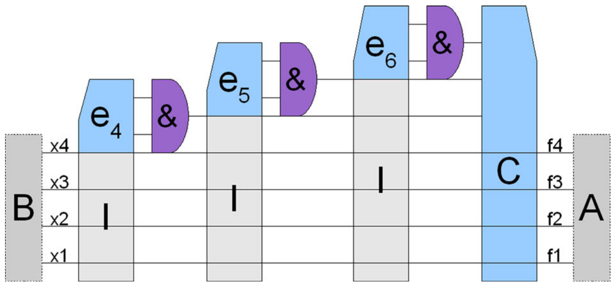

Fig. 3 Illustration of Lemma 6, the construction of a Boolean function with a multiplicative complexity 3. Matrices **A**, **B** denote the place of linear equivalence

contains only even permutations, and thus we cannot construct odd permutations using only the iteration process based on Lemma 5.

Lemma 6 *Let* $E_n : GF(2)^n \to GF(2)^{n+1}$,

$$E_n(\mathbf{x}) = \left(x_1, x_2, \ldots, x_n, (\mathbf{b}_1^T \cdot \mathbf{x}) \cdot (\mathbf{b}_2^T \cdot \mathbf{x})\right).$$

Let $C_{m,n} : GF(2)^m \to GF(2)^n$ *be a linear function. Any Boolean function* $F : GF(2)^n \to GF(2)^n$ *with* $F(\mathbf{0}) = \mathbf{0}$, *and multiplicative complexity* $MC(F) \leq c$ *can be written as a composition*

$$F = C_{n+c,n} \circ E_{n+c-1} \circ \cdots \circ E_{n+1} \circ E_n.$$

Proof Let $c = 0$. Any function with $F(\mathbf{0}) = \mathbf{0}$, and $MC(F) = 0$ is a linear function, which can be written as $F = C_{n,n}$.

Let $c = 1$. Circuit to implement function F with $MC(F) = 1$ must contains a single AND-gate. Circuit have n inputs $x_1, \ldots, x_n$. A circuit can implement any number of affine functions, i.e., functions $h_i(\mathbf{x}) = \mathbf{a}_i \cdot \mathbf{x} + c_i$. At most $n + 1$ of these functions are linearly independent. Let one of the independent functions be $h_0 = 1$, and the n others $h_i = \mathbf{e}^{(i)} \cdot \mathbf{x}$, i=$1, \ldots, n$.

Two inputs of the AND-gate can be consist of any affine transformation of available inputs, and the AND-gate provides a single output. The AND-gate can be expressed by the function $g(\mathbf{x}) = (\mathbf{b}_1^T \cdot \mathbf{x} + d_1) \cdot (\mathbf{b}_2^T \cdot \mathbf{x} + d_2)$. We can move constants d_1, d_2 to the linear part of the circuit by constructing g as a sum $g(\mathbf{x}) = g_1((x)) + g_2((x)) + d_1 d_2$, where

$$g_1(\mathbf{x}) = (\mathbf{b}_1^T \cdot \mathbf{x}) \cdot (\mathbf{b}_2^T \cdot \mathbf{x}),$$

and

$$g_2(\mathbf{x}) = (d_1 \mathbf{b}_2^T + d_2 \mathbf{b}_1^T) \cdot \mathbf{x}.$$

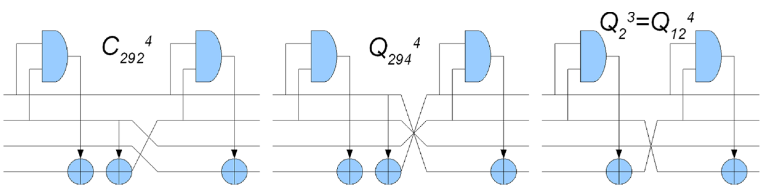

Fig. 4 Three affine classes of even S-boxes with multiplicative complexity 2 (one of the classes has degree 3)

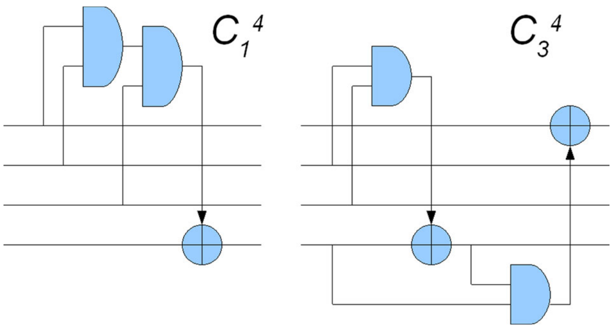

Fig. 5 Two affine classes of odd S-boxes with multiplicative complexity 2

Function g_1 is linearly independent from any of h_i's (g_2 is a linear combination of h_i's). Finally, we can construct any function with n outputs by using for each output any linear combination of $n+2$ linearly independent functions $\{1, h_1, \ldots, h_n, g_1\}$. However, $F(\mathbf{0}) = \mathbf{0}$, so each output f_i must be constant-free. Thus h_0 is not used, and we get F as a linear combination of $n+1$ functions $\{h_1, \ldots, h_n, g_1\}$, which is exactly construction $F = C_{n+1,n} \circ E_n$.

Similarly, for larger c, each E is used to construct the next function $g_2, g_3, \ldots$, that is the output of the additional AND-gate. The input of the new c-th gate can be any linear[2] combination of the previous functions $h_1, \ldots, h_n, g_1, g_2, \ldots, g_{n+c-1}$ (so we use E_{n+c-1} as the expansion function). Finally, we use n linear combinations of $h_1, \ldots, h_n, g_1, g_2, \ldots, g_{n+c-1}, g_{n+c}$ (compression function $C_{n+c,n}$) to construct outputs of F. □

Lemma 6 can be used to compute $\mathcal{M}_c$ by computing the set (Fig. 3)

$$\{F = C_{n+c,n} \circ E_{n+c-1} \circ \cdots \circ E_{n+1} \circ E_n;\ F \text{ is bijection}\}^*.$$

Unfortunately, this direct approach is quite impractical, as the number of options when constructing F's is too large even for small c, n. E.g. for $n = 4, c = 2$ we need to choose $(4 \times 2) + (5 \times 4)$ bits to go through all E's and C's, which is 2^{28} S-boxes (many repeated). For $n = 4, c = 4$ we get 2^{76} S-boxes (for which we do not have enough computing power).

Figures 4, and 5, respectively, denote five classes of bijective 4×4 S-boxes. Three even classes can be decomposed using Λ_4, two odd classes cannot be decomposed in this way.

The search can be sped up if we use affine equivalence, and search only for $\mathcal{M}_c^*$. We can use the following properties of the construction:

1. Multiplication is commutative, and $(\mathbf{b}^T \cdot \mathbf{x}) \cdot (\mathbf{b}^T \cdot \mathbf{x}) = \mathbf{b}^T \cdot \mathbf{x}$. Thus we can restrict the search to $\mathbf{b}_1 < \mathbf{b}_2$ (halves the search space for each E).
2. We only search for a single S-box in each affine class. We can suppose that there is an input linear transformation given by an invertible matrix $\mathbf{B}$ (so we compute $F(\mathbf{Bx})$ instead of $F(\mathbf{x})$). Now we can replace $\mathbf{b}_1$, $\mathbf{b}_2$ in E_n by $\mathbf{e}^{(1)}, \mathbf{e}^{(2)}$, and move $\mathbf{b}_1$, $\mathbf{b}_2$ into $\mathbf{B}$ instead (as the first two rows). Inner transformation $\mathbf{Bx}$ can be "removed" in compression function C. (A similar construction is used in the proof of Theorem 1).

[2] All possible constants can be moved to the linear part of the circuit.

Table 1 Combinations of the vectors for the first two expansions explored in construction of $\mathcal{M}_c^*$ ($n = 4$)

Description	E_4		E_5	
	$\mathbf{b}_1$	$\mathbf{b}_2$	$\mathbf{b}_1$	$\mathbf{b}_2$
4 independent rows of **B**	$\mathbf{e}^{(1)}$	$\mathbf{e}^{(2)}$	$\mathbf{e}^{(3)}$	$\mathbf{e}^{(4)}$
Linear combination	$\mathbf{e}^{(1)}$	$\mathbf{e}^{(2)}$	$\mathbf{e}^{(3)}$	$\mathbf{e}^{(1)}$
of first two rows of **B**	$\mathbf{e}^{(1)}$	$\mathbf{e}^{(2)}$	$\mathbf{e}^{(3)}$	$\mathbf{e}^{(2)}$
multiplied by the third one	$\mathbf{e}^{(1)}$	$\mathbf{e}^{(2)}$	$\mathbf{e}^{(3)}$	$\mathbf{e}^{(1)} + \mathbf{e}^{(2)}$
3 independent rows of **B**, and g_1	$\mathbf{e}^{(1)}$	$\mathbf{e}^{(2)}$	$\mathbf{e}^{(3)}$	$\mathbf{e}^{(5)}$
3 independent rows of **B**,	$\mathbf{e}^{(1)}$	$\mathbf{e}^{(2)}$	$\mathbf{e}^{(3)}$	$\mathbf{e}^{(5)} + \mathbf{e}^{(1)}$
g_1, and linear combination	$\mathbf{e}^{(1)}$	$\mathbf{e}^{(2)}$	$\mathbf{e}^{(3)}$	$\mathbf{e}^{(5)} + \mathbf{e}^{(2)}$
of the first two rows	$\mathbf{e}^{(1)}$	$\mathbf{e}^{(2)}$	$\mathbf{e}^{(3)}$	$\mathbf{e}^{(5)} + \mathbf{e}^{(1)} + \mathbf{e}^{(2)}$

3. Similarly, we can replace $\mathbf{b}_1$, $\mathbf{b}_2$ in E_{n+1} by the choice of the next two rows of **B**. However, we must take into account possible linear combination with the inputs of the first AND-gate, e.g. to produce linearly independent functions $g_1(\mathbf{x}) = x_1x_2$, and $g_2(\mathbf{x}) = x_1x_3$. For $n = 4$, $c = 2$ we get nine options (see Table 1). For each option there are roughly 2^{24} matrices **C** that should be explored (e.g. if we require $MC(S) = 2$, we can skip C's that do not use the outputs of the AND-gates). This process can be extended even further, but it is impractical to implement.
4. Using outer linear transform we can also reduce the number of C's we need to explore. $C_{n+c,n}$ is given by $n \times (n + c)$ matrix **C**. We can write $\mathbf{C} = \mathbf{AT}$, where **T** is an upper triangular matrix, and **A** is an invertible $n \times n$ matrix that can be removed as outer linear transform of affine equivalence.
5. In each class we focus our attention to normalized S-boxes. Each normalized S-box can be written in the form $S(\mathbf{x}) = \mathbf{x} + F(\mathbf{x})$, where the component functions of F do not contain any linear terms in their ANF's. Due to this fact, we suppose (an unproven hypothesis, which were verified by computer search for $n = 4$, $c = 2$, $c = 3$), that we only need to concentrate on functions for which $\mathbf{C} = \left(\mathbf{I}|\mathbf{c}^{n+1} \cdots \mathbf{c}^{n+c}\right)$, where **I** is an identity matrix, so that

$$\mathbf{C} \cdot (E_{n+c-1} \circ \cdots \circ E_n) = \mathbf{x} + (\mathbf{c}^{n+1} \cdots \mathbf{c}^{n+c}) \begin{pmatrix} g_1 \\ \vdots \\ g_c \end{pmatrix}.$$

Table 2 Statistics of S-boxes according to multiplicative complexity

MC	Classes	NormRep	Classes [%]	NormRep [%]
0	1	1	0.33	0.00
1	1	85	0.33	0.00
2	5	5250	1.66	0.01
3	25	471560	8.28	1.18
4	140	18515360	46.36	46.38
5	130	20924544	43.05	52.42

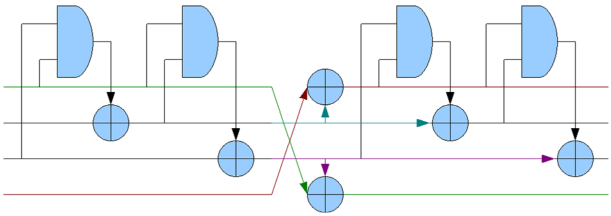

Fig. 6 S-box construction which is affinely equivalent to PRESENT S-box, class $\mathcal{C}^4_{266}$

We have used the above method to compute (for $n = 4$) $\mathcal{M}^*_2$, $\mathcal{M}^*_3$, and $\mathcal{M}^*_4$, respectively. Computing $\mathcal{M}^*_2$, $\mathcal{M}^*_3$ is relatively fast. For $c = 4$, we have reduced the search space (using the above reductions) to $9 \cdot 2^{25} \cdot 2^{16} \doteq 2^{44}$ S-boxes. For each function thus generated, we verify whether it is a permutation. If it is a permutation, it is normalized, and using a large lookup table (11! entries) its affine class is determined. The computation was distributed to 16 computing cores[3], and took between 6 and 7 days in real time to finish.

Using the computed sets $\mathcal{M}^*_2$, $\mathcal{M}^*_3$, we have further computed the set $\mathcal{D} = [\mathcal{C}_{2,3} \cap \mathcal{C}_{3,2}]^*$. This set contains all 302 affine classes[4], thus $\mathcal{D} = \mathcal{M}^*_5$, and $MC(S) \leq 5$ for each bijective 5×5 S-box.

5.1 Computational results

Computational results ($n = 4$) are summarized in Appendix. For each S-box we give a proof of construction with the given number of multiplications. It can be viewed as an upper bound on its multiplicative complexity. Unfortunately, it should not be considered as the proven multiplicative complexity, mainly due to computer generated results.

In this section we highlight some of the observations: S-boxes with multiplicative complexity 2, statistics of the S-box classes, and finally the results for S-box classes with optimal linear and differential properties (including the PRESENT S-box).

Table 2 summarizes the statistics of the multiplicative complexity as presented in Appendix. For each multiplicative complexity we list the number of classes, and the number of normalized representatives of the class. Although the number of classes with multiplicative complexity 4 is higher, classes with multiplicative complexity 5 have a larger number of representatives (an average number of representatives grows with multiplicative complexity).

An important requirement for an S-boxes is its resistance against linear, and differential cryptanalysis, respectively. There are 16 affine classes of optimal 4×4 S-boxes [12]. Out of these classes, six classes have multiplicative complexity 4 (including PRESENT [3] S-box, see Fig. 6):

1. even permutations: $\mathcal{C}^4_{296}, \mathcal{C}^4_{266}, \mathcal{C}^4_{297}, \mathcal{C}^4_{223}$ (G_0, G_1, G_2, G_8)
2. odd permutations: $\mathcal{C}^4_{209}, \mathcal{C}^4_{210}$ (G_{14}, G_{15})

All other optimal classes have multiplicative complexity 5. We remark, that $n \times n$ S-box with $MC(S) < n$ should not be used in the (classical SPN-like) cipher design, as we can

[3]Each node had a different fixed value c^{n+1}, but each node produced the same set, so there are still potential reductions of the search space.

[4]We remark that this set also contains all 3374 classes of constant-free S-boxes under linear equivalence.

always find a combination of inputs and outputs that sums to a constant (as there are only $n-1$ linearly independent non-linear component functions).

6 Conclusions and open questions

We show that there is a single class of bijective $n \times n$ S-boxes under affine equivalence ($n \geq 3$), represented by the permutation Λ_n. As Λ_3 is an odd permutation, it can be used to construct all affine classes of 3×3 S-boxes by composition, in such a way that multiplicative complexity corresponds to the number of Λ_3's composed. For larger n's Λ_n is an even permutation, and the composition based construction is not possible. We remark, that even if we add an odd permutation to the possible compositions, not all S-boxes can be decomposed in a similar way (such that the multiplicative complexity of the final S-box is given directly as a sum of multiplicative complexities of the composed S-boxes). However, the composition construction might be useful to prove the upper bounds on multiplicative complexity for a specific class of S-boxes. Using composition of S-boxes with multiplicative complexity 2, and 3, respectively, we have shown that multiplicative complexity of all 4×4 bijective S-boxes is at most 5. Combined with the SAT-solver based proofs of Courtois [9], we can be quite confident that some affine classes have multiplicative complexity exactly 5.

Using construction based on non-linear expansion and linear compression, we have computed the bounds for multiplicative complexities for each affine class of 4×4 S-boxes. Knowing S-box multiplicative complexity is useful for the optimal hardware implementation of the S-box, but it might also be used in algebraic cryptanalysis. Our construction can also be used for larger n's to construct S-boxes with low multiplicative complexity. Unfortunately, in this case the number of possibilities, as well as the number of affine classes is much larger (already for $n = 5$ the number is approx. 2^{61} [6]), so we cannot cover all classes (with the available computing power).

An interesting open question is the connection of multiplicative complexity based on $GF(2)$ multiplications with masking complexity in $GF(2^n)$. In general, masking complexity is defined (Definition 3, [7]) as the number of non-linear multiplications required to evaluate polynomial representation of S-box over $GF(2^k)$. Thus the multiplicative complexity is just a special case with $k = 1$. On the other hand, in $GF(2^n)$ terms, multiplicative complexity expresses the minimum number of operations in the form $Tr(\alpha_1 x)Tr(\alpha_2 x)$ required to evaluate the polynomial along with an unlimited number of linear operations.

The question of $GF(2)$-multiplicative complexity of multiplication in extension fields is intensively studied in the complexity theory area. E.g., it is known that to implement a $GF(2^4)$ multiplication we need at most eight $GF(2)$ multiplications [8]. An important research question is to determine the optimal k (and the related circuit design) for general n (or a specific affine class of S-boxes) with respect to masking against DPA attacks. Our hypothesis is that optimum is obtained always at $k = 1$: Let us suppose that we need at most $M_2(n)$ $GF(2)$ multiplications to implement a single $GF(2^n)$ multiplication. Furthermore let the minimum number of non-linear $GF(2^n)$ multiplications to implement some S-box be k. Our hypothesis is that the multiplicative complexity of the S-box is significantly lower than $k \cdot M_2(n)$. E.g., the masking complexity of PRESENT S-boxes and Serpent S-boxes in $GF(2^4)$ is 3 [15], but their multiplicative complexity (masking complexity over $GF(2)$) is significantly lower than the expected $3 \cdot 8 = 24$ (Serpent S-boxes 3 and 7 have MC=5, PRESENT S-box and other Serpent S-boxes have MC=4). If it is cheaper to mask 4 or 5 non-linear single-bit operations instead of 3 non-linear four-bit operations, than the choice of $k = 1$ for evaluating masking complexity is more suitable.

Appendix

List of S-boxes

For each class of S-box we list its number according to [1] (we only list class number), its representative in hexadecimal notation (first normalized S-box in lexicographic order, (upper bound on) multiplicative complexity, and the constructive proof of MC. The proof is either by composition of two S-boxes with lower MC, or by writing down coefficients of expansion-compression construction for an S-box in the given class (we do not provide a proof of affine equivalence with the representative). The expansion-compression proof was required for 2 S-boxes with $MC(S) = 2$, 5 S-boxes with $MC(S) = 3$, and 25 S-boxes with $MC(S) = 4$.

The format of expansion-compression proof:

1. Four (single-digit) hex numbers encode vectors $\mathbf{c}^{n+1} \cdots \mathbf{c}^{n+4}$,
2. Eight (two-digit) hex numbers encode vectors $\mathbf{b}_1, \mathbf{b}_2$ for E_4, E_5, E_6, E_7.

Class	Representative	MC	Proof
0	0123456789ABCDEF	0	$Aff(2, n)$
4	0123456789ABDCFE	1	Λ_4 (Theorem 1)
1	0123456789ABCDFE	2	0800 01 02 04 10 00 00 00 00
3	0123456789ABDEFC	2	4800 01 02 04 10 00 00 00 00
2	0123456789ABCEFD	3	0210 04 08 01 10 03 10 00 00
54	012345798A6BDECF	3	4210 01 02 09 14 26 31 00 00
241	012345768A9CDEBF	3	4210 02 08 07 13 24 38 00 00
242	012345768A9CDFEB	3	4210 01 02 08 15 0b 30 00 00
291	012345768A9BCFED	3	0210 02 04 08 13 0c 30 00 00
41	012345768A9CBFED	4	4941 01 02 10 04 15 1a 0d 48
43	012345798ABE6CFD	4	8261 01 02 18 04 1b 2d 02 24
44	012345798ABFCED6	4	4821 01 02 11 04 15 1d 0a 4c
49	012345798AEDC6BF	4	8294 01 02 18 04 28 36 07 28
50	012345798AFDC6EB	4	1294 01 02 18 04 07 36 08 40
63	012345798AC6EBFD	4	8241 01 02 18 04 28 31 02 24
64	012345798ADF6CEB	4	4821 01 02 11 04 09 15 47 59
70	012345798E6CFBDA	4	8261 01 02 18 04 19 2c 25 76
71	012345798EACDF6B	4	8294 01 02 18 04 0e 1b 09 7b
76	012345798ABFEDC6	4	1295 01 02 18 04 07 32 08 46
82	0123457986ACFBED	4	8294 01 02 18 04 19 2f 06 30
85	012345798ABD6EFC	4	8241 01 02 18 04 29 32 06 26
86	012345798ABFC6ED	4	4821 01 02 11 04 17 1f 0a 4c
92	012345798AE6BDCF	4	8295 01 02 18 04 1a 30 0f 40
114	012345798FAEDBC6	4	8295 01 02 18 04 1d 2c 1b 6d
115	012345798F6CBEDA	4	71b4 01 02 18 04 1f 35 05 31
126	012345798AC6DFBE	4	1695 01 02 18 04 2e 31 05 37

Class	Representative	MC	Proof
127	012345798AFD6CBE	4	8294 01 02 18 04 31 36 19 2b
147	012345798AE6DBFC	4	8241 01 02 18 04 28 31 06 24
148	012345798AE6CFBD	4	4821 01 02 11 04 15 1d 47 4d
187	012345798ABFEC6D	4	1294 01 02 18 04 07 32 2a 34
201	012345798E6DAFBC	4	8295 01 02 18 04 1d 31 27 4e
202	012345798A6DECBF	4	8295 01 02 18 04 07 30 2f 35
209	0123469A85EDC7FB	4	e261 01 02 18 04 27 3d 06 48
210	0123469B87DE5FAC	4	8a94 01 02 18 04 0f 16 0b 70

The format of composition proof $S_2 \circ A \circ S_1$:

1. S_1 is representative of the first listed class;
2. A is always linear transformation, encoded by 4 hexadecimal numbers, the images of 1,2,4,8 in this order (1 encodes $\mathbf{e}^{(1)}$).
3. S_2 is representative of the second listed class;

Class	Representative	MC	Proof
12	0123456789CDEFAB	2	C: 4, 4218, 4
292	012345768A9BCEFD	2	C: 4, 4168, 4
294	0123456789BAEFDC	2	C: 4, 8429, 4
5	0123456789ACDBFE	3	C: 4, 8421, 1
6	0123456789ACBDFE	3	C: 4, 2481, 292
11	0123456789BCEFDA	3	C: 4, 8341, 3
13	0123456789CDEFBA	3	C: 4, 8241, 3
39	012345768A9CBEFD	3	C: 4, 8421, 3
40	012345768A9CBFDE	3	C: 3, 8421, 4
233	0123459A8B67CEFD	3	C: 4, C241, 292
234	0123459A8EF6BDC7	3	C: 4, 4A21, 292
236	0123459A87B6CEFD	3	C: 4, A421, 292
243	012345768ACF9BDE	3	C: 4, 8241, 294
244	012345768ACE9BFD	3	C: 4, 4821, 294
258	0123459A8BCFED76	3	C: 4, 8521, 12
259	0123459A8B67CFDE	3	C: 4, 8421, 12
260	0123459A8BCF76ED	3	C: 4, 8421, 294
264	0123459A8BCDE67F	3	C: 4, E421, 292
287	012345768A9CDFBE	3	C: 4, 4281, 292
288	0123456789CEFBDA	3	C: 4, 2C41, 292
293	0123457689CDEFBA	3	C: 4, 8142, 294
299	012345678ACEB9FD	3	C: 4, 4281, 294
300	012345AB89CDEF67	3	C: 4, 8142, 12
7	0123456789ACBEFD	4	C: 4, 1842, 241
8	0123456789ACDEFB	4	C: 4, 8241, 2
9	0123456789ACDEBF	4	C: 4, 1A42, 40
10	0123456789BCAEFD	4	C: 4, 4281, 6

Class	Representative	MC	Proof
17	0123456987CDEFAB	4	C: 4, 8241, 5
18	0123456987ACDBFE	4	C: 4, 8421, 6
21	0123456987ACBDFE	4	C: 4, 4281, 39
22	0123456987ACEFBD	4	C: 4, 4821, 6
25	0123456987ABCEFD	4	C: 4, 4381, 39
38	0123456987ABDEFC	4	C: 4, 8721, 6
46	012345798ABDFC6E	4	C: 3, C421, 3
52	012345798EBCAF6D	4	C: 4, 8142, 287
53	012345798FADBEC6	4	C: 4, 1A42, 234
55	012345798AB6FCED	4	C: 4, 2841, 287
57	012345798ABEC6DF	4	C: 4, A142, 287
58	012345798ACFDE6B	4	C: 4, 8341, 11
59	012345798AECFDB6	4	C: 4, 8341, 40
65	012345798ADFC6EB	4	C: 4, 6281, 54
66	012345798F6DEABC	4	C: 4, 8241, 264
67	012345798A6DCFBE	4	C: 4, 8241, 287
68	0123457986ACDEBF	4	C: 4, 8142, 241
69	0123457986ACDFEB	4	C: 4, 4281, 54
72	012345798E6CFDBA	4	C: 4, A421, 54
77	012345798ACB6EDF	4	C: 4, A142, 40
79	0123457986ACBEFD	4	C: 4, 5281, 54
80	0123457986ACBFDE	4	C: 4, 8241, 241
81	0123457986ACFEBD	4	C: 4, A341, 287
93	012345798F6DEACB	4	C: 4, 8142, 40
94	012345798AD6CEFB	4	C: 4, A241, 40
95	012345798ADFC6BE	4	C: 4, 4281, 233
96	012345798ADEC6FB	4	C: 4, A241, 287
97	0123459A87EDCFB6	4	C: 4, 8421, 241
98	0123459A87DFB6CE	4	C: 4, C341, 241
100	0123459A87BE6CDF	4	C: 4, A621, 39
101	0123459A87ECFD6B	4	C: 4, 8421, 287
108	0123459A8D6FE7BC	4	C: 4, 8621, 241
109	0123459A8B6CFD7E	4	C: 4, E241, 242
116	0123459A8DC6BFE7	4	C: 4, 8521, 241
117	0123459A8C6DFB7E	4	C: 4, E241, 39
118	0123459A8EC6F7DB	4	C: 4, 8521, 241
122	0123459A87ED6BFC	4	C: 4, E241, 39
128	0123459A87BEDF6C	4	C: 4, 6B21, 54
129	0123459A87E6BCDF	4	C: 4, A421, 39
130	0123459A8E7FDB6C	4	C: 4, 1842, 234
131	0123459A8F6C7EBD	4	C: 4, 5281, 233
132	0123459A8EF7DB6C	4	C: 4, 8621, 40
133	0123459A8EC6F7BD	4	C: 4, C241, 40
150	0123459A87ECDFB6	4	C: 4, 3481, 260
151	0123459A87EC6BFD	4	C: 4, 3481, 258
152	0123459A87ECD6BF	4	C: 4, A621, 54
153	0123459A87BEFDC6	4	C: 4, 8621, 242

Class	Representative	MC	Proof
158	0123459A87ECF6BD	4	C: 4, 5821, 236
159	0123459A87ED6FBC	4	C: 4, 9142, 234
161	0123459A876EFDCB	4	C: 4, 5921, 236
162	0123459A876ECFDB	4	C: 4, 3481, 264
164	0123459A876ECBFD	4	C: 4, 8341, 264
165	0123459A876EDBCF	4	C: 4, 8341, 233
166	0123459E8F6CADB7	4	C: 4, 8421, 40
167	0123459A8CBF76ED	4	C: 4, 8721, 11
168	0123459A8CE7FB6D	4	C: 4, 6381, 259
169	012345798ACFB6ED	4	C: 4, 8241, 242
170	012345798AECB6FD	4	C: 4, 8341, 241
171	0123459A8CE7BFD6	4	C: 4, 5281, 259
172	0123459A8CD67EFB	4	C: 4, 8341, 243
173	012345798AFDC6BE	4	C: 4, 8241, 40
176	0123457986ACEFDB	4	C: 4, 8341, 287
178	0123459A8BDFC76E	4	C: 4, 9142, 236
181	0123459A8CFBE7D6	4	C: 4, 4A21, 54
182	0123459A8CEB76FD	4	C: 4, C142, 241
183	0123459E8AB6FDC7	4	C: 4, A421, 287
184	0123459A8BCEDF76	4	C: 4, 8721, 287
185	0123459A8B6ECFD7	4	C: 4, C421, 40
186	0123459A8BECDF76	4	C: 4, 8621, 11
190	0123459A8DBC7EF6	4	C: 4, C621, 241
191	0123459A8BDF6CE7	4	C: 4, E621, 242
195	0123459A87E6DFBC	4	C: 4, A521, 54
199	0123459A87ECBFD6	4	C: 4, 6A21, 54
200	0123459A87EC6FBD	4	C: 4, C241, 242
203	012345798ACF6BDE	4	C: 4, 8241, 11
204	012345798AEC6BDF	4	C: 4, A421, 40
206	0123459A8EF7BDC6	4	C: 4, A621, 40
207	0123459A8C67DFEB	4	C: 4, C142, 40
213	0123459A8CFB76DE	4	C: 4, 8521, 11
214	0123459A8BCFDE67	4	C: 4, 8421, 299
215	012345AB86CE79FD	4	C: 4, 8421, 293
216	0123457986CDEFBA	4	C: 4, 8142, 11
220	0123469B85CF7AED	4	C: 4, C521, 242
222	0123467985EDBFAC	4	C: 4, 8521, 242
223	0123469B87CF5AED	4	C: 4, E241, 244
229	0123459A8C6D7EFB	4	C: 4, 5A21, 54
230	0123459A8D7EBF6C	4	C: 4, C241, 241
232	0123459A8EC67FDB	4	C: 4, 8521, 40
235	0123459A86B7CFDE	4	C: 4, 8421, 291
237	0123459A86B7CEFD	4	C: 4, 6821, 40
238	0123457689CEAFBD	4	C: 4, 4821, 288
239	0123457689CEAFDB	4	C: 4, 5821, 40
240	012345768A9CDEFB	4	C: 4, 1842, 236

Class	Representative	MC	Proof
245	012345768ACF9BED	4	C: 4, C241, 291
246	0123456987BAEFDC	4	C: 4, 2841, 5
247	012345698AB7CDFE	4	C: 4, 8421, 2
248	0123456987CEFBDA	4	C: 4, 4A21, 39
252	0123459A86CFEB7D	4	C: 4, 9241, 264
256	012345AB86CF79ED	4	C: 4, 8421, 13
257	0123459A8BCF76DE	4	C: 4, 8421, 11
262	0123459A8BCDF76E	4	C: 4, A621, 244
263	0123459A87E6FDCB	4	C: 4, A721, 39
265	0123459A87CDE6BF	4	C: 4, C621, 40
266	0123469B87CFA5DE	4	C: 4, A241, 244
267	012345798F6CBEAD	4	C: 4, 6381, 54
286	0123459A86EF7BCD	4	C: 4, 8421, 288
289	0123456789CEBFDA	4	C: 4, 1C42, 40
290	0123456789BCEAFD	4	C: 4, 1842, 40
296	0123469B87CFDEA5	4	C: 4, 8241, 244
297	0123469A8D5E7FBC	4	C: 4, A421, 244
301	012345AB89CDEF76	4	C: 4, 8142, 13
14	0123456987CDEFBA	5	C: 294, A512, 287
15	012345698ABEC7DF	5	C: 294, A162, 40
16	012345698ABEC7FD	5	C: 294, 5821, 6
19	0123456987ACDEFB	5	C: 294, 8142, 241
20	0123456987ACDFEB	5	C: 3, 1964, 241
23	0123456987BCFEDA	5	C: 294, 4391, 54
24	012345698ABCFED7	5	C: 3, 1A42, 40
26	0123456987BCDEFA	5	C: 3, 1862, 241
27	012345698ABCDEF7	5	C: 294, A124, 40
28	0123456987BCFADE	5	C: 3, 2841, 40
29	012345698ABCEFD7	5	C: 294, 2941, 39
30	012345698ACB7EFD	5	C: 294, 2841, 39
31	0123456987ACBEFD	5	C: 3, 2954, 241
32	0123456987ACEBFD	5	C: 294, 8A14, 241
33	0123456987BCFEAD	5	C: 294, 8521, 6
34	0123456987BCEFDA	5	C: 294, 8D21, 6
35	0123456987CEAFDB	5	C: 294, 4821, 39
36	0123456987CEAFBD	5	C: 3, 5F32, 39
37	0123456987ACDFBE	5	C: 294, A214, 40
42	012345798A6CEDBF	5	C: 3, 1942, 236
45	012345798AFBC6ED	5	C: 292, 4C13, 241
47	012345798ADE6CFB	5	C: 3, 1862, 236
48	012345798AED6CBF	5	C: 3, 2851, 287
51	012345798EADFB6C	5	C: 3, 8A41, 287
56	012345798ABF6CED	5	C: 3, 6A31, 287
60	012345798AFCB6ED	5	C: 294, A631, 287
61	012345798AFCEDB6	5	C: 12, AB24, 287
62	012345798ABFC6DE	5	C: 3, A851, 287

Class	Representative	MC	Proof
73	012345798E6CDFBA	5	C: 3, 2C54, 241
74	012345798F6CEDBA	5	C: 3, 5821, 40
75	012345798AFB6CDE	5	C: 3, 1A64, 236
78	0123457986ACBFED	5	C: 3, 2851, 40
83	0123457986ACFBDE	5	C: 3, 2854, 241
84	012345798F6DCABE	5	C: 3, 2A51, 287
87	012345798F6DECBA	5	C: 294, E421, 39
88	012345798E6DCFBA	5	C: 294, A531, 39
89	012345798F6DCEBA	5	C: 3, 2D54, 40
90	012345798E6CFBAD	5	C: 3, 2841, 287
91	012345798FACDE6B	5	C: 3, 2A41, 287
99	0123459A87CBDE6F	5	C: 3, 4A32, 241
102	012345798AEFC6BD	5	C: 294, 8E12, 40
103	0123459A86C7DEBF	5	C: 294, 1582, 236
104	0123459A86C7EDFB	5	C: 294, 8A51, 287
105	0123459A867CBFDE	5	C: 3, 6A21, 287
106	0123459A867CBEFD	5	C: 3, 2954, 287
107	0123459A86CF7EBD	5	C: 294, A241, 287
110	0123459A86CED7FB	5	C: 3, 7821, 242
111	0123459A86BC7EFD	5	C: 3, 6C21, 40
112	0123459A8CBF7ED6	5	C: 294, A421, 39
113	0123459A8BECD6F7	5	C: 294, C241, 40
119	012345798A6CDEFB	5	C: 3, 5932, 236
120	0123459A8DCBE6F7	5	C: 294, 8241, 287
121	0123459A8BFD6E7C	5	C: 294, 8251, 287
123	0123459A86BCF7ED	5	C: 294, A851, 287
124	0123459A86CEDFB7	5	C: 294, 1482, 236
125	0123459A86CB7FDE	5	C: 3, 2854, 287
134	0123459A86C7DFEB	5	C: 3, 6821, 40
135	0123459A86CDF7BE	5	C: 294, A431, 39
136	0123459A86C7EBDF	5	C: 3, 3C54, 39
137	0123459A87EFB6DC	5	C: 294, 8962, 40
138	0123459A86EFB7DC	5	C: 294, 4821, 288
139	0123456987BCAFDE	5	C: 294, 2851, 39
140	012345698ABECFD7	5	C: 3, 2D54, 39
141	0123459A86EF7BDC	5	C: 294, C162, 40
142	012345798E6DFCBA	5	C: 3, 2C51, 40
143	0123459A86CEB7FD	5	C: 294, 8612, 40
144	0123459A86CEDF7B	5	C: 294, 8521, 40
145	0123459A86BCE7DF	5	C: 3, 6831, 40
146	0123459A86CB7EFD	5	C: 3, 3798, 264
149	012345798AE6FDCB	5	C: 294, 8162, 40
154	0123467985EDFACB	5	C: 3, 6E31, 242
155	0123467985EDAFBC	5	C: 3, 7832, 242
156	0123459A86CEF7DB	5	C: 3, 2E51, 234
157	0123459A86CFEBD7	5	C: 3, 5C32, 234
160	0123469A85FD7CEB	5	C: 294, C421, 241

Class	Representative	MC	Proof
163	0123469C85FAEDB7	5	C: 294, C621, 40
174	0123459A86CE7FDB	5	C: 3, 6932, 242
175	012345798F6CDEBA	5	C: 294, 8621, 242
177	0123457986BCFADE	5	C: 294, 8214, 40
179	012345798AEDC6FB	5	C: 3, 1962, 236
180	0123459A8D6CFB7E	5	C: 3, 6821, 242
188	012345798ABEFC6D	5	C: 3, 3691, 233
189	0123457986CEFBDA	5	C: 3, 4F32, 39
192	0123459A8DBF6CE7	5	C: 3, 3D41, 39
193	0123459A8CBEF67D	5	C: 292, 4812, 241
194	012345798AFCDE6B	5	C: 294, 8A14, 287
196	0123469C85FAED7B	5	C: 292, 1862, 236
197	0123459A8F7E6CDB	5	C: 292, 6813, 40
198	0123459A86C7BFDE	5	C: 294, 4128, 264
205	012345798AEFB6CD	5	C: 294, 4821, 6
208	0123459A86C7BEFD	5	C: 3, 6932, 40
211	0123459A86CFD7BE	5	C: 294, 4C21, 264
212	0123459A86BC7FDE	5	C: 294, A621, 287
217	012345698ACBE7FD	5	C: 294, 8421, 2
218	0123457986BCDFAE	5	C: 294, 8124, 241
219	012345798AFC6BDE	5	C: 294, 8251, 236
221	0123457986CEAFDB	5	C: 3, 2C41, 40
224	0123457986CFBEAD	5	C: 294, C731, 242
225	0123456987CFBEAD	5	C: 3, 1C42, 40
226	012345798E6DAFCB	5	C: 3, 3498, 233
227	012345798AD6FCEB	5	C: 3, 3491, 233
228	0123459A86BCFD7E	5	C: 3, 6A31, 40
231	0123469A85FDBCE7	5	C: 292, 8251, 236
249	0123459A86CEFBD7	5	C: 3, 2C41, 39
250	0123459A86CD7EBF	5	C: 294, 8A51, 40
251	0123459A86CFE7BD	5	C: 3, 2C41, 234
253	0123459A86CF7EDB	5	C: 3, 2B54, 287
254	0123459A86BCEDF7	5	C: 294, 5618, 264
255	012345798EBCAFD6	5	C: 3, 6932, 287
261	0123459A86CFB7DE	5	C: 294, C421, 291
268	0123459A86CEBF7D	5	C: 3, 4E21, 39
269	0123459A86CFBE7D	5	C: 3, 8142, 234
270	0123469A8C5D7EFB	5	C: 292, 7921, 287
271	0123469A8C5DF7BE	5	C: 3, BE31, 234
272	0123469A8DBC5EF7	5	C: 292, 6E54, 40
273	0123459A8DBFE76C	5	C: 292, 2A53, 236
274	0123459A87EDBF6C	5	C: 3, 8162, 234
275	0123459A8CEBD6F7	5	C: 3, 2C54, 39
276	0123459A8DFB7C6E	5	C: 3, 6832, 287
277	0123459A87EBDF6C	5	C: 294, 8421, 260
278	0123469A8FBEC75D	5	C: 292, 2917, 236
279	0123459A87E6CFDB	5	C: 3, 3841, 264

Class	Representative	MC	Proof
280	0123457986ACFDEB	5	C: 3, 3781, 54
281	0123457986ACEBFD	5	C: 3, 2841, 241
282	0123469A8DBCF75E	5	C: 292, 6E72, 242
283	0123469A8C5DBEF7	5	C: 292, 7C62, 241
284	0123459A87ECFB6D	5	C: 294, 8631, 40
285	0123457986CEAFBD	5	C: 294, EA21, 39
295	0123469C8A7DE5FB	5	C: 292, 6A23, 40
298	012345698ACEB7FD	5	C: 294, 8214, 5

References

1. 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.) CHES, Lecture Notes in Computer Science, vol. 7428, pp. 76–91. Springer (2012)
2. Biryukov, A., Cannière, C.D., Braeken, A., Preneel, B.: A toolbox for cryptanalysis: Linear and affine equivalence algorithms. In: Biham, E. (ed.) Advances in Cryptology – EUROCRYPT 2003, Lecture Notes in Computer Science, vol. 2656, pp. 33–50. Springer-Verlag. doi:10.1007/3-540-39200-9_3 (2003)
3. 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.) CHES, Lecture Notes in Computer Science, vol. 4727, pp. 450–466. Springer (2007)
4. Boyar, J., Peralta, R.: Tight bounds for the multiplicative complexity of symmetric functions. Theor. Comput. Sci. **396**(1–3), 223–246 (2008). doi:10.1016/j.tcs.2008.01.030
5. Boyar, J., Peralta, R.: A new combinational logic minimization technique with applications to cryptology. SEA, 178–189 (2010)
6. Cannière, C.D.: Analysis and design of symmetric encryption algorithms. Ph.D. thesis, Katholieke Universiteit Leuven (2007)
7. Carlet, C., Goubin, L., Prouff, E., Quisquater, M., Rivain, M.: Higher-order masking schemes for s-boxes. In: Fast Software Encryption, pp. 366–384. Springer (2012)
8. Cenk, M., Özbudak, F.: On multiplication in finite fields. J. Complex. **26**(2), 172–186 (2010). doi:10.1016/j.jco.2009.11.002, http://www.sciencedirect.com/science/article/pii/S0885064X09001095
9. Courtois, N., Hulme, D., Mourouzis, T.: Solving circuit optimisation problems in cryptography and cryptanalysis. Cryptology ePrint Archive. Report 2011/475 (2011)
10. Eisenbarth, T., Kumar, S.: A survey of lightweight-cryptography implementations, Vol. 24, pp. 522–533 (2007)
11. Fischer, M., Peralta, R.: Counting predicates of conjunctive complexity one. Tech. Rep. YALEU/DCS/TR1222, Yale University (2001)
12. Leander, G., Poschmann, A.: On the classification of 4 bit S-boxes. In: Carlet, C., Sunar, B. (eds.) Arithmetic of Finite Fields, Lecture Notes in Computer Science, vol. 4547, pp. 159–176. Springer Berlin / Heidelberg (2007), doi:10.1007/978-3-540-73074-3_13
13. Lidl, R., Niederreiter, H.: Finite Fields, Encyclopedia of Mathematics and its Applications, Vol. 20. Addison-Wesley, Reading, Massachussetts (1983)
14. Mirwald, R., Schnorr, C.: The multiplicative complexity of quadratic boolean forms. Theor. Comput. Sci. **102**(2), 307–328 (1992). doi:10.1016/0304-3975(92)90235-8, http://www.sciencedirect.com/science/article/pii/0304397592902358
15. Roy, A., Vivek, S.: Analysis and improvement of the generic higher-order masking scheme of fse 2012. Cryptology ePrint Archive, Report 2013/345. http://eprint.iacr.org/ (2013)
16. Saarinen, M.J.O.: Cryptographic analysis of all 4×4 - bit S-boxes. In: Miri, A., Vaudenay, S. (eds.) Selected Areas in Cryptography, Lecture Notes in Computer Science, vol. 7118, pp. 118–133. Springer (2011)

17. Schnorr, C.: The multiplicative complexity of boolean functions. In: Mora, T. (ed.) Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Lecture Notes in Computer Science, vol. 357, pp. 45–58. Springer Berlin / Heidelberg. 10.1007/3-540-51083-4_47 (1989)
18. Ullrich, M., Cannière, C.D., Indesteege, S., Küçük, O., Mouha, N., Preneel, B.: Finding optimal bitsliced implementations of 4 x 4-bit S-boxes. In: Symmetric Key Encryption Workshop. 20 (2011)
19. Zajac, P.: A new method to solve mrhs equation systems and its connection to group factorization. J. Math. Cryptol. **7**(4), 279–381 (2013). doi:10.1515/jmc-2013-5012