Constructions of S-boxes with uniform sharing

Kerem Varici[1], Svetla Nikova[1], Ventzislav Nikov[2], and Vincent Rijmen[1]

[1] KU Leuven, imec-COSIC, Belgium, {name.surname}@esat.kuleuven.be
[2] NXP Semiconductors, Belgium, venci.nikov@gmail.com

Abstract. In this paper we focus on S-box constructions. We consider the uniformity property of an S-box which plays an important role in Threshold Implementations (TI). Most papers so far have studied TI sharings for given S-boxes. We proceed in the opposite way: starting from n-bit S-boxes with known sharings we construct new $(n+1)$-bit S-boxes from them with the desired sharings. In addition, we investigate the self-equivalency of S-boxes and show some interesting properties.

1 Introduction

Block ciphers are one of the key components in cryptography. Recently, lightweight cryptography became popular and it carried the design approach of a block cipher into different points. Depending on the requirements of the platform: low energy consumption, restriction on area, low multiplicative complexity, resistance against side-channel attacks etc., many new design approaches were proposed over the years.

In 1945, Shannon defined two properties of a block cipher: diffusion and confusion. Restrictions on lightweight cryptography pushes the search of a perfectly secure diffusion and confusion components to find suboptimal secure but efficient components. S-boxes are one of the main primitive in symmetric-key cryptography. They are the smallest component in an algorithm which provides non-linearity. The size of an S-box can change from three bit to n-bit but mostly four and eight bit length is preferred. In this work, we studied general n-bit S-boxes but we did our experiments for small sizes i.e. three to five bits.

The classification of all 3-bit and 4-bit S-boxes according to affine equivalency was first given in [3,6].

Definition 1. *Two S-boxes S_1 and S_2 are affine equivalent if there exists a pair of invertible affine permutations A and B, such that $S_1 = A \circ S_2 \circ B$.*

Note the ordering used in the paper for $A \circ B$ is first apply A, then apply B. It is well known that all invertible 2×2 S-boxes are affine, hence there is only one class. The set of invertible 3×3 S-boxes contains 4 equivalence classes: 3 classes containing quadratic functions, and one class containing the affine functions. The maximal algebraic degree of a balanced n-variable Boolean function is $n-1$ [4,7]. De Cannière lists 302 equivalence classes for the 4×4 bijections: the class of affine functions, 6 classes containing quadratic functions and the remaining 295 classes containing cubic functions. There is a transformation [2] which expands the 3-bit classes $\mathcal{Q}_1^3$, $\mathcal{Q}_2^3$, and $\mathcal{Q}_3^3$ into $\mathcal{Q}_4^4$, $\mathcal{Q}_{12}^4$ and $\mathcal{Q}_{300}^4$ correspondingly. That is, given a 3-bit permutation $S(x_1, x_2, x_3) = (y_1, y_2, y_3)$, its 4-bit extension is generated by $S(x_1, x_2, x_3, x_4) = (y_1, y_2, y_3, x_4)$.

Recently a classification of all quadratic 5 × 5 S-boxes was presented in [1]. The authors have also pointed out that the 5-bit classes $\mathcal{Q}_1^5$, $\mathcal{Q}_3^5$, $\mathcal{Q}_4^5$, $\mathcal{Q}_7^5$, $\mathcal{Q}_{13}^5$ and $\mathcal{Q}_{30}^5$ are extensions of the 4-bit quadratic classes $\mathcal{Q}_4^4$, $\mathcal{Q}_{294}^4$, $\mathcal{Q}_{12}^4$, $\mathcal{Q}_{299}^4$, $\mathcal{Q}_{293}^4$ and $\mathcal{Q}_{300}^4$ from [2] respectively. That is, given a 4-bit permutation $S(x_1, x_2, x_3, x_4) = (y_1, y_2, y_3, y_4)$, its 5-bit extension is generated by $S(x_1, x_2, x_3, x_4, x_5) = (y_1, y_2, y_3, y_4, x_5)$. Let $\bar{x} = (x_1, ..., x_n)$ then the method used in the above mentioned publications can be summarized as follows

$$\begin{aligned} S(\bar{x}, x_{n+1}) &= S_1(\bar{x}) \quad \text{for the first } n \text{ bits} \\ &= x_{n+1} \quad \text{for the } (n+1)\text{-st bit} \end{aligned} \tag{1}$$

Another well known construction is the so-called Shannon expansion.

Definition 2 (Shannon Expansion). *Let* $\bar{x} = (x_1, ..., x_n)$ *then*

$$F(\bar{x}) = x_i F_{x_i}(\bar{x}) + (x_i + 1) F_{x_i+1}(\bar{x}) \tag{2}$$

where $F_{x_i}(\bar{x}) = F(x_1, ..., x_i = 1, ..., x_n)$ *and* $F_{x_i+1}(\bar{x}) = F(x_1, ..., x_i = 0, ..., x_n)$ *i.e. these are two functions on* $n-1$ *variables* $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$.

The recent developments in the technology come with some new security requirements like physical security. Today, many ways exist for physically secure implementations. One of the most popular ways is Threshold Implementations (TI) [8]. The method uses the idea of secret sharing schemes and techniques from multiparty computation, and requires a sharing which needs to satisfy the following properties: Correctness, Non-completeness and Uniformity. In this paper we will mainly focus on the last property which simply can be stated as follows. Whenever a given n-bit S-box is a permutation, its sharing with k shares is a permutation on $GF(2^{kn})$. The number of shares k in d-th order TI depends on the algebraic degree t of the S-box - namely $k \geq td + 1$, hence for 1-st order security one needs at least $t + 1$ shares. We do not consider here the approach based on decomposition on the S-box into low degree S-boxes.

It is well known how to find a uniform sharing for all 3×3 and 4×4 S-boxes [2]. Recall that a uniform sharing with 3 shares exists for $\mathcal{Q}_1^3$, $\mathcal{Q}_2^3$, but not for $\mathcal{Q}_3^3$; and a uniform sharing with $4, 5$ and more shares exists for all 3 of them. Also recall that a uniform sharing with 3 shares exists for $\mathcal{Q}_4^4$, $\mathcal{Q}_{294}^4$, $\mathcal{Q}_{12}^4$, $\mathcal{Q}_{299}^4$, $\mathcal{Q}_{293}^4$, but not for $\mathcal{Q}_{300}^4$; and a uniform sharing with $4, 5$ and more shares exists for all 6 of them $\mathcal{Q}_4^4$, $\mathcal{Q}_{294}^4$, $\mathcal{Q}_{12}^4$, $\mathcal{Q}_{299}^4$, $\mathcal{Q}_{293}^4$, and $\mathcal{Q}_{300}^4$. Also a uniform sharing with 4 shares is known for $\mathcal{C}_1^4$, $\mathcal{C}_3^4$, $\mathcal{C}_{13}^4$ and $\mathcal{C}_{301}^4$. The remaining cubic 4-bit S-boxes have no uniform sharing with 4 shares. However, all of them have a uniform sharing with 5 shares. Recently, a 3-share uniform sharing for 30 of the 5-bit quadratic classes (namely classes $1-27$, 31, 33 and 34) was found [1]. Moreover, all 5-bit quadratic permutation classes have a uniform sharing with 4 and more shares. It is well known that any Boolean function of 3 or more variables has a uniform TI sharing with 3 and more sharings [9].

It is clear that method (1) is providing uniform sharing with k shares for S if and only if S_1 has a uniform sharing with k shares. In this work, we focus on the uniformity property of an S-box since it needs a special treatment and there does not exist a straightforward way of checking the property. We show under which conditions a uniform sharing of n-bit S-box can be used to construct uniform sharing of $(n+i)$-bit S-boxes where $i \geq 1$. The main reason to do this, is that we want to avoid an exhaustive uniformity check, whose complexity increases exponentially with the size of the S-box and the number of shares used. We used the idea of Shannon's expansion to generate bigger S-boxes and manage to show the cases that uniformity will hold for the newly generated S-boxes. Moreover, we showed for small sizes $n = 3$ and 4, which n-bit classes maps to which $(n+1)$-bit classes.

The original intention of the affine equivalence algorithm [3] was to discover equivalence relations between different S-boxes, but the algorithm can be applied for a single S-box S as well. In this case, the algorithm will return affine mappings A and B for the self-equivalent S. The number of different solutions for this equation can be seen as an indicator for the symmetry of the S-box [3]. S-boxes that have at least one non-trivial solution are called self-equivalent S-boxes [3].

A well known example is the 8-bit S-box S used in RIJNDAEL [5]. It has 2040 different self-equivalence relations. Although this number might seem surprisingly high at first, in [3] it is shown that it can easily be explained from the special algebraic structure of the S-box of RIJNDAEL and it can be generalized for the inversion in $GF(2^n)$. Let first introduce the notation $[a]$, which denotes the $n \times n$-bit matrix corresponding to a multiplication by a in $GF(2^n)$. Similarly, denote by Q the $n \times n$-bit matrix which performs the squaring operation in $GF(2^n)$ note this is a linear operation. Considering the fact that the RIJNDAEL S-box is defined as $S = inv \circ A$ with A a fixed affine mapping, we can derive a general expression for all pairs of affine mappings C and B that satisfy $B \circ S \circ C = S$

$$C = A^{-1} \circ [a] \circ Q^i \circ A, \quad \text{and} \quad B = [a] \circ Q^i$$

with $0 \leq i < n$ and $a \in GF(2^n) \setminus \{0\}$. Since i takes n different values and there are $2^n - 1$ different choices for a, one obtains exactly 2040 in the case $n = 8$ different solutions, while in general there are $n(2^n - 1)$ self-equivalent S-boxes of the inversion. It is easy to check that $Q^n = Id$ and thus $Q^i = Q^{n-i}$.

As noted in [3] these ideas also apply to a large extent to other ciphers that use S-boxes based on power functions, e.g., CAMELLIA, MISTY and KASUMI, whose S-boxes $S7$ and $S9$ are both designed to be affine equivalent to a power function over $GF(2^7)$ and $GF(2^9)$, respectively. In this paper we will investigate this property for any S-box, i.e. not only for the power functions.

Our contribution. In Section 2 we first study the Shannon expansion for S-boxes and derive a condition to obtain a new invertible S-box. We then show how to use the Shannon expansion to construct from given n-bit S-boxes with a uniform sharing $(n+1)$-bit S-boxes with a uniform sharing and $(n+2)$-bit S-boxes with a uniform sharing. We apply this for all 3-bit and 4-bit S-boxes to derive results on 4-bit and 5-bit S-boxes. Alas, we find that our method is not able to generate all $(n+1)$-bit S-boxes. In Section 3 we present some results on the presence of self-equivalent S-boxes and involutions in affine equivalence classes. We conclude in Section 4.

2 Constructions of S-boxes with uniform sharing

2.1 Increasing the size of an S-box by one

We start with a slightly modified definition of Shannon's expansion.

Definition 3. *Given two $n \times n$ S-boxes (bijections) $S_1(\bar{x}) = (t_1, t_2, \dots t_n)$ and $S_2(\bar{x}) = (u_1, u_2, \dots u_n)$, where $\bar{x} = (x_1, ..., x_n)$. Using Shannon's expansion, we get an $(n+1) \times (n+1)$ S-box (not always a bijection) $S(x_1, \dots, x_n, x_{n+1}) = (y_1, y_2, \dots, y_{n+1})$:*

$$\begin{aligned} y_i &= x_{n+1} t_i \quad + (1 + x_{n+1}) u_i, \qquad\qquad \textit{for } i = 1, \dots, n \qquad (3) \\ y_{n+1} &= x_{n+1} F(\bar{x}) + (1 + x_{n+1}) G(\bar{x}), \end{aligned}$$

where F and G are Boolean functions of n inputs.

It follows that the truth table of the constructed $(n+1) \times (n+1)$ S-box S has the form shown in Table 1.

Table 1. Truth table of S constructed by Definition 3

$(\bar{x}, x_{n+1} = 0)$	$(\bar{x}, x_{n+1} = 1)$
$(S_2(\bar{x}), G(\bar{x}))$	$(S_1(\bar{x}), F(\bar{x}))$

Theorem 1. *S is a bijection if and only if*

$$G(\bar{x}) = F(S_1^{-1}(S_2(\bar{x}))) + 1 \quad \textit{or equivalently} \quad G = S_2 \circ S_1^{-1} \circ F + 1\ . \qquad (4)$$

Proof. ($\Leftarrow$) Let the condition (4) hold and assume there are two different inputs $(\bar{x}', x'_{n+1})$ and $(\bar{x}'', x''_{n+1})$, such that $S(\bar{x}', x'_{n+1}) = S(\bar{x}'', x''_{n+1})$, i.e. assume S is not a bijection. Then we consider the following two cases:

a) if $x'_{n+1} = x''_{n+1}$ then it follows that $\bar{x}' \neq \bar{x}''$ since the inputs are different. From $S(\bar{x}', x'_{n+1}) = S(\bar{x}'', x''_{n+1})$ and $x'_{n+1} = x''_{n+1}$, it follows that either $S_1(\bar{x}') = S_1(\bar{x}'')$ or $S_2(\bar{x}') = S_2(\bar{x}'')$. But since both S_1 and S_2 are bijections, we arrive at a contradiction.

b) if $x'_{n+1} \neq x''_{n+1}$ then it follows either
b1) $S_1(\bar{x}') = S_2(\bar{x}'')$ and $F(\bar{x}') = G(\bar{x}'')$ or
b2) $S_1(\bar{x}'') = S_2(\bar{x}')$ and $F(\bar{x}'') = G(\bar{x}')$.

Let us consider case **b1)**: since $S_1(\bar{x}') = S_2(\bar{x}'')$ we get $\bar{x}' = S_1^{-1}(S_2(\bar{x}''))$ and hence $F(S_1^{-1}(S_2(\bar{x}''))) = G(\bar{x}'')$ but this contradicts to (4). Similarly one can get a contradiction for **b2)**.

Therefore, our assumption is incorrect and hence S is a bijection when the condition (4) holds.
($\Rightarrow$) Let S be a bijection. Choose $x'_{n+1} = 0$ and $x''_{n+1} = 1$, then for any n-tuple $\bar{x}'$ there exist n-tuple $\bar{x}''$, such that $S_1(\bar{x}') = S_2(\bar{x}'')$. Since $S(\bar{x}', x'_{n+1}) \neq S(\bar{x}'', x''_{n+1})$ this implies that $F(\bar{x}') \neq G(\bar{x}'')$ or in other words $F(\bar{x}') = G(\bar{x}'') + 1$. Again since $S_1(\bar{x}') = S_2(\bar{x}'')$ we get $\bar{x}' = S_1^{-1}(S_2(\bar{x}''))$ and thus $F(S_1^{-1}(S_2(\bar{x}''))) + 1 = G(\bar{x}'')$ which is exactly the condition (4). This completes the proof. □

We conclude that one has to choose only S_1, S_2 and F in order to build S. When S_1 and S_2 are fixed there are "only" 2^n choices for F in order to get S. We should stress that $\deg(S) = \max\{dA, dB\}$, where

$$dA = \begin{cases} \max\{\deg(S_1), \deg(S_2)\} & \deg(S_1 + S_2) < \max\{\deg(S_1), \deg(S_2)\} \\ \max\{\deg(S_1), \deg(S_2)\} + 1 & \deg(S_1 + S_2) = \max\{\deg(S_1), \deg(S_2)\}, \end{cases}$$

$$dB = \begin{cases} \max\{\deg(F), \deg(G)\} & \deg(F + G) < \max\{\deg(F), \deg(G)\} \\ \max\{\deg(F), \deg(G)\} + 1 & \deg(F + G) = \max\{\deg(F), \deg(G)\}. \end{cases}$$

Note that whenever $S_1 = S_2$ condition (4) can be simplified to: $F(\bar{x}) = G(\bar{x}) + 1$. In this case $\deg(S) = \max\{\deg(S_1), \deg(F)\}$, i.e. the simplified equation is:

$$\begin{aligned} y_i &= t_i \qquad\qquad \text{for } i = 1, \ldots, n \\ y_{n+1} &= x_{n+1} + F(\bar{x}) \end{aligned} \tag{5}$$

Here, instead of $F + 1$ we write F for simplicity. Note that compared to the construction (1) used in [2] to get from 3×3 an 4×4 S-box and similarly in [1] from 4×4 an 5×5 S-box, the construction (5) extends it to allow F to be any Boolean function on n variables.

Now we will show how the constructions (3) or (5) can be used to find uniform sharings. Let $\bar{1}_i$ denote the vector $(0, ..., 0, 1, 0, ..., 0)$ with 1 on the i-th position.

Theorem 2. *Let us consider any $n \times n$ S-box S_1 which has a uniform sharing with k shares and any Boolean function F with n variables which also has a uniform sharing with k shares. If S_2 is chosen in one of the $n + 1$ forms: $S_1(\bar{x})$, $S_1(\bar{x} + \bar{1}_i)$ for $i = 1, ..., n$, then the generated $(n + 1) \times (n + 1)$-bit S-box S by using S_1, S_2 and F has also a uniform sharing with k shares.*

Proof. When $S_1 = S_2$ it follows from Theorem 1 that S is a bijection if and only if $F(\bar{x}) = G(\bar{x}) + 1$ and in this case the equation (5) can be used. It is also clear in this case that S will have a uniform sharing with k shares. Now let us consider the general case when $S_2(\bar{x}) = S_1(\bar{x} + \bar{1}_i)$. According to Definition 2 we have:

$$\begin{aligned} S_1(\bar{x}) &= x_i S_{1,x_i}(\bar{x}) + (x_i + 1) S_{1,x_i+1}(\bar{x}) \\ F(\bar{x}) &= x_i F_{x_i}(\bar{x}) + (x_i + 1) F_{x_i+1}(\bar{x}) \end{aligned} \tag{6}$$

We rewrite (6) as follows:

$$\begin{aligned} S_1(\bar{x}) &= (x_i+1)[S_{1,x_i}(\bar{x}) + S_{1,x_i+1}(\bar{x})] + S_{1,x_i}(\bar{x}) \\ F(\bar{x}) &= (x_i+1)[F_{x_i}(\bar{x}) + F_{x_i+1}(\bar{x})] + F_{x_i}(\bar{x}) \end{aligned} \tag{7}$$

Note also that (6) implies:

$$\begin{aligned} S_1(\bar{x}+\bar{1}_i) &= (x_i+1)S_{1,x_i}(\bar{x}) + x_i S_{1,x_i+1}(\bar{x}) \\ F(\bar{x}+\bar{1}_i) &= (x_i+1)F_{x_i}(\bar{x}) + x_i F_{x_i+1}(\bar{x}) \end{aligned} \tag{8}$$

Next we replace $S_2(\bar{x})$ with $S_1(\bar{x}+\bar{1}_i)$ in the definition of S

$$\begin{aligned} S(\bar{x}, x_{n+1}) &= x_{n+1}S_1(\bar{x}) + (1+x_{n+1})S_1(\bar{x}+\bar{1}_i) \qquad \text{for } i = 1, \dots, n \\ &= x_{n+1}F(\bar{x}) + (1+x_{n+1})[F(\bar{x}+\bar{1}_i)+1] \end{aligned} \tag{9}$$

Here we used that condition (4) should hold in order for S to be a bijection. Then let's rewrite the first n-bits of (9) using (6) and (8)

$$\begin{aligned} S(\bar{x}, x_{n+1}) &= x_{n+1}S_1(\bar{x}) + (1+x_{n+1})S_1(\bar{x}+\bar{1}_i) \\ &= x_{n+1}x_i S_{1,x_i}(\bar{x}) + x_{n+1}(x_i+1)S_{1,x_i+1}(\bar{x}) \\ &+ (1+x_{n+1})(x_i+1)S_{1,x_i}(\bar{x}) + (1+x_{n+1})x_i S_{1,x_i+1}(\bar{x}) \\ &= [x_{n+1}+x_i+1]S_{1,x_i}(\bar{x}) + [x_{n+1}+x_i]S_{1,x_i+1}(\bar{x}) \\ &= [x_{n+1}+x_i][S_{1,x_i}(\bar{x}) + S_{1,x_i+1}(\bar{x})] + S_{1,x_i}(\bar{x}) \end{aligned} \tag{10}$$

Comparing (10) with (7), we notice that the change of variables, namely $[x_{n+1}+x_i]$ to $[x_i+1]$ gives a equivalence between the first n bits of S and S_1. Since (7) has a uniform sharing then (10) also has a uniform sharing, meaning that this is a $kn \times kn$ bijection. Finally, let's rewrite the last $(n+1)$-st bit of (9) using (6) and (8)

$$\begin{aligned} S(\bar{x}, x_{n+1}) &= x_{n+1}F(\bar{x}) + (1+x_{n+1})[F(\bar{x}+\bar{1}_i)+1] \\ &= x_{n+1}x_i F_{x_i}(\bar{x}) + x_{n+1}(x_i+1)F_{x_i+1}(\bar{x}) \\ &+ (1+x_{n+1})(x_i+1)F_{x_i}(\bar{x}) + (1+x_{n+1})x_i F_{x_i+1}(\bar{x}) + (1+x_{n+1}) \\ &= [x_{n+1}+x_i+1]F_{x_i}(\bar{x}) + [x_{n+1}+x_i]F_{x_i+1}(\bar{x}) + (1+x_{n+1}) \\ &= [x_{n+1}+x_i][F_{x_i}(\bar{x}) + F_{x_i+1}(\bar{x})] + F_{x_i}(\bar{x}) + (1+x_{n+1}) \end{aligned} \tag{11}$$

Now comparing (11) with (7) and using again the change of variables, namely $[x_{n+1}+x_i]$ to $[x_i+1]$, it gives "near" equivalence between the last bit of S and F except the term $x_{n+1}+1$. In other words, S can be rewritten as follows:

$$\begin{aligned} S(\bar{x}, x_{n+1}) &= S_1(\bar{x}) \qquad \text{for } j = 1, \dots, n \\ &= x_{n+1} + F(\bar{x}) + 1 \end{aligned} \tag{12}$$

upon the change of variables $[x_{n+1}+x_i]$ to $[x_i+1]$. This completes the proof, since this is equivalent to the case $S_2 = S_1$ where we know uniform sharings exist. □

To summarize: the case $S_2(\bar{x}) = S_1(\bar{x}+\bar{1}_i)$ reduces to the case $S_2(\bar{x}) = S_1(\bar{x})$ with the change of variables $[x_{n+1}+x_i]$ to $[x_i+1]$.

2.2 Application of Shannon's Expansion to S-boxes

We first investigate what happens when we use constructions (3) or (5) with affine equivalent S-boxes. Let S be constructed from S_1, S_2, F and S' be constructed from $S'_1 = A \circ S_1 \circ B$, $S'_2 = A \circ S_2 \circ B$ and $F' = A \circ F$. Then we have:

$$\begin{aligned} S(\bar{x}, x_{n+1}) &= x_{n+1} S_1(\bar{x}) + (1 + x_{n+1}) S_2(\bar{x}) \\ &= x_{n+1} F(\bar{x}) + (1 + x_{n+1}) G(\bar{x}), \end{aligned}$$

where $G(\bar{x}) = S_2 \circ S_1^{-1} \circ F(\bar{x}) + 1$ and

$$\begin{aligned} S'(\bar{x}, x_{n+1}) &= x_{n+1}[A \circ S1 \circ B(\bar{x})] + (1 + x_{n+1})[A \circ S_2 \circ B(\bar{x})] \\ &= x_{n+1}[A \circ F(\bar{x})] + (1 + x_{n+1}) G'(\bar{x}). \end{aligned}$$

Since $S_1'^{-1} = B^{-1} \circ S_1^{-1} \circ A^{-1}$ we have

$$G' = [A \circ S_2 \circ B] \circ [B^{-1} \circ S_1^{-1} \circ A^{-1}] \circ [A \circ F] = A \circ S_2 \circ S_1^{-1} \circ F = A \circ G.$$

If we define $A' = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix}$ and $B' = \begin{bmatrix} B & 0 \\ 0 & 1 \end{bmatrix}$ then we see that $A' \circ S \circ B' = S'$ i.e. the constructed S and S' are affine equivalent.

We will now explore two approaches that use this affine equivalence of the extended S-box to reduce the search complexity.

1. We go class per class by first fixing $S_1 = S_2$ to the class representative and then vary F over all possible Boolean functions to get S in its reduced form via the construction (5).
2. We go again class per class by first fixing S_1 to the class representative. However, next we run a second loop for S_2 varying it over all possible S-boxes and last we vary F over all possible Boolean functions. Then S is obtained in its most general form via the construction (3).

Applying the first approach over 3 × 3 S-boxes to obtain 4 × 4 S-boxes showed that (5) enriches the number of classes we can construct and the uniform sharings we can obtain as shown in Table 2.

Table 2. Extension of 3-bit S-box classes into 4-bit S-box classes

3-bit Class	4-bit Class		
$\mathcal{A}_0^3$	$\mathcal{A}_0^4$,	$\mathcal{C}_1^4$,	$\mathcal{Q}_4^4$
$\mathcal{Q}_1^3$	$\mathcal{C}_3^4$,	$\mathcal{Q}_4^4$,	$\mathcal{Q}_{294}^4$
$\mathcal{Q}_2^3$	$\mathcal{C}_{13}^4$,	$\mathcal{Q}_{12}^4$,	$\mathcal{Q}_{293}^4$
$\mathcal{Q}_3^3$	$\mathcal{C}_{301}^4$,	$\mathcal{Q}_{300}^4$	

These results show that in addition to the four corresponding classes $\mathcal{A}_0^4$, $\mathcal{Q}_4^4$, $\mathcal{Q}_{12}^4$ and $\mathcal{Q}_{300}^4$ which were already known from [2], we also get three additional quadratic classes and even four cubic classes. Using Theorem 2 we get a uniform sharing with k shares for S, whenever such sharing exists for S_1 and F. This explains the results obtained in [2] for the four cubic S-boxes which are the only ones among the cubic 4×4 S-boxes, which have uniform sharing with 4-shares. However, it also should be noted that we were not able to obtain class $\mathcal{Q}^4_{299}$ from any of the 3×3 S-boxes via this approach.

Similarly, applying the first approach over 4×4 S-boxes to obtain 5×5 but restricted only to the affine and quadratic S-boxes gives the results shown in Table 3. We obtain 23 out of the 75 quadratic classes given in [1]. In addition, it is clear from this construction why for the classes $\mathcal{Q}^5_{30}$ and $\mathcal{Q}^5_{32}$ no uniform sharing with 3 shares was found in [1]. Namely, they are extensions of class $\mathcal{Q}^4_{300}$ which has no uniform sharing itself.

Table 3. Extension of non-cubic 4-bit S-box classes into 5-bit S-box classes

4-bit Class	5-bit Class
$\mathcal{A}^4_0$	$\mathcal{Q}^5_0, \mathcal{Q}^5_1, \mathcal{Q}^5_{14}$
$\mathcal{Q}^4_4$	$\mathcal{Q}^5_1, \mathcal{Q}^5_2, \mathcal{Q}^5_3, \mathcal{Q}^5_{15}, \mathcal{Q}^5_{18}$
$\mathcal{Q}^4_{12}$	$\mathcal{Q}^5_4, \mathcal{Q}^5_6, \mathcal{Q}^5_{13}, \mathcal{Q}^5_{17}, \mathcal{Q}^5_{20}, \mathcal{Q}^5_{21}$
$\mathcal{Q}^4_{293}$	$\mathcal{Q}^5_{13}, \mathcal{Q}^5_{24}, \mathcal{Q}^5_{31}$
$\mathcal{Q}^4_{294}$	$\mathcal{Q}^5_3, \mathcal{Q}^5_5, \mathcal{Q}^5_{12}, \mathcal{Q}^5_{16}, \mathcal{Q}^5_{19}, \mathcal{Q}^5_{23}$
$\mathcal{Q}^4_{299}$	$\mathcal{Q}^5_7, \mathcal{Q}^5_{22}$
$\mathcal{Q}^4_{300}$	$\mathcal{Q}^5_{30}, \mathcal{Q}^5_{32}$

We finish this section by applying the second approach over 3×3 S-boxes to obtain 4×4 S-boxes. It is to be expected that the second approach based on the construction (3) generates more solutions than the first one based on construction (5), however, the complexity is much higher. We experimentally determined that by constructing 4-bit S-boxes from 3-bit S-boxes we obtain all the 4-bit classes except the 11 classes presented in Table 4. Notice that 8 out of these 11 exceptions belong to the Optimal Golden S-boxes [6]. Recall that there are 16 classes of the best 4-bit S-boxes i.e., $\{\mathrm{Diff}(S) = 4, \mathrm{Lin}(S) = 8\} = \{G_0, .., G_{15}\}$ and among them is the inversion G_3. Apparently the Shannon expansion has certain limitations, which make it impossible to obtain the 11 classes from Table 4. For one example, namely the class $\mathcal{Q}^4_{193}$, we did some kind of backward search to verify our results. For each S-box in this class we used (3) to determine all possible decompositions to 3-bit S-boxes. We tried with using each of the 4 variables x_1, x_2, x_3, x_4 as x_{n+1} in the formula and used every row once as "the last row". In this way, for any 4-bit S-box we derived 16 cases and for each of them obtained two 3-bit S-boxes. We found that for none of the S-boxes in this class and none of the 16 cases subsequently considered the obtained 3-bit S-boxes are permutations. This confirms that $\mathcal{Q}^4_{193}$ cannot be obtained via (3) when S_1 and S_2 are permutations.

Table 4. 4-bit S-box classes not obtained from 3-bit S-box classes

$\mathcal{Q}^4_{193}$	$\mathcal{Q}^4_{196}$	$\mathcal{Q}^4_{197}$	$\mathcal{Q}^4_{231}$	$\mathcal{Q}^4_{270}$	$\mathcal{Q}^4_{272}$	$\mathcal{Q}^4_{273}$	$\mathcal{Q}^4_{278}$	$\mathcal{Q}^4_{282}$	$\mathcal{Q}^4_{283}$	$\mathcal{Q}^4_{295}$
	G_7		G_{13}	G_4	G_6		G_5	G_3	G_{12}	G_{11}

2.3 Extending an S-box with two dimensions

Given four $n \times n$ S-boxes (bijections) $S_1(\bar{x}) = (t_1, t_2, \dots t_n), S_2(\bar{x}) = (u_1, u_2, \dots u_n)$, $S_3(\bar{x}) = (v_1, v_2, \dots v_n)$ and $S_4(\bar{x}) = (w_1, w_2, \dots w_n)$, where $\bar{x} = (x_1, ..., x_n)$ then using Shannon's expansion we get an $(n+2) \times (n+2)$ S-box (not always a bijection) $S(x_1, \dots, x_{n+1}, x_{n+2}) = (y_1, y_2, \dots, y_{n+2})$ with

$$\begin{aligned} y_i &= x_{n+2}[x_{n+1}t_i + (1 + x_{n+1})u_i] \\ &+ (1 + x_{n+2})[x_{n+1}v_i + (1 + x_{n+1})w_i] \qquad \text{for } i = 1, \dots, n \qquad (13) \\ y_{i+1} &= x_{n+2}[x_{n+1}F_1(\bar{x}) + (1 + x_{n+1})G_1(\bar{x})] \\ &+ (1 + x_{n+2})[x_{n+1}F_2(\bar{x}) + (1 + x_{n+1})G_2(\bar{x})] \\ y_{i+2} &= x_{n+2}[x_{n+1}F_3(\bar{x}) + (1 + x_{n+1})G_3(\bar{x})] \\ &+ (1 + x_{n+2})[x_{n+1}F_4(\bar{x}) + (1 + x_{n+1})G_4(\bar{x})] \end{aligned}$$

where F_1, F_2, F_3, F_4 and G_1, G_2, G_3, G_4 are Boolean functions of n inputs.
It follows from this definition that the truth table of the constructed $(n+2) \times (n+2)$ S-box S has the following form.

Table 5. Truth table of S

$(\bar{x}, x_{n+1} = 0, x_{n+2} = 0)$	$(\bar{x}, x_{n+1} = 0, x_{n+2} = 1)$	$(\bar{x}, x_{n+1} = 1, x_{n+2} = 0)$	$(\bar{x}, x_{n+1} = 1, x_{n+2} = 1)$
$(S_4(\bar{x}), G_2(\bar{x}), G_4(\bar{x}))$	$(S_3(\bar{x}), F_2(\bar{x}), F_4(\bar{x}))$	$(S_2(\bar{x}), G_1(\bar{x}), G_3(\bar{x}))$	$(S_1(\bar{x}), F_1(\bar{x}), F_3(\bar{x}))$

Theorem 3. *S is a bijection if and only if*

$$\begin{aligned} &F_1(S_1^{-1}(\bar{x})) = G_2(S_4^{-1}(\bar{x})) + 1 = F_2(S_3^{-1}(\bar{x})) = G_1(S_2^{-1}(\bar{x})) + 1 \quad \textit{and} \quad (14) \\ &F_3(S_1^{-1}(\bar{x})) = G_4(S_4^{-1}(\bar{x})) + 1 = F_4(S_3^{-1}(\bar{x})) + 1 = G_3(S_2^{-1}(\bar{x})). \end{aligned}$$

Proof (Sketch). The proof is similar to the proof of Theorem 1. First we consider the following subcases separately: $x_{n+1} = 0$ and $x_{n+2} = 0, 1$; $x_{n+1} = 1$ and $x_{n+2} = 0, 1$; $x_{n+2} = 0$ and $x_{n+1} = 0, 1$; $x_{n+2} = 0$ and $x_{n+1} = 0, 1$. From those four subcases we get four conditions accordingly:

$$\begin{aligned} G_1(\bar{z}) &= F_1(S_1^{-1}(S_2(\bar{z}))) + 1 \qquad (15) \\ G_2(\bar{z}) &= F_2(S_3^{-1}(S_4(\bar{z}))) + 1 \\ G_4(\bar{z}) &= G_3(S_2^{-1}(S_4(\bar{z}))) + 1 \\ F_4(\bar{z}) &= F_3(S_1^{-1}(S_3(\bar{z}))) + 1 \end{aligned}$$

or analogously (15) can be rewritten as:

$$\begin{aligned} G_1(S_2^{-1}(\bar{z})) &= F_1(S_1^{-1}(\bar{z}))+1 \\ G_2(S_4^{-1}(\bar{z})) &= F_2(S_3^{-1}(\bar{z}))+1 \\ G_4(S_4^{-1}(\bar{z})) &= G_3(S_2^{-1}(\bar{z}))+1 \\ F_4(S_3^{-1}(\bar{z})) &= F_3(S_1^{-1}(\bar{z}))+1 \end{aligned} \tag{16}$$

Next consider the special case when the first n-bits $\bar{z}$ are fixed

$$\begin{aligned} \bar{z} &= S_1(\bar{d}) \\ \bar{z} &= S_2(\bar{c}) \\ \bar{z} &= S_3(\bar{b}) \\ \bar{z} &= S_4(\bar{a}) \end{aligned}$$

and thus we have four different n-bit inputs $\bar{a}, \bar{b}, \bar{c}, \bar{d}$ as given above. Then the four tuples of the last two bits are formed by

$$\begin{aligned} G_2(\bar{a}), G_4(\bar{a}) \\ F_2(\bar{b}), F_4(\bar{b}) \\ G_1(\bar{c}), G_3(\bar{c}) \\ F_1(\bar{d}), F_3(\bar{d}) \end{aligned}$$

and we want to ensure that they are exactly the following four tuples $(0,0), (0,1), (1,0)$ and $(1,1)$ in any order. From this, four additional conditions can be derived

$$\begin{aligned} G_2(\bar{a}) &= F_1(\bar{d})+1 \\ G_4(\bar{a}) &= F_3(\bar{d})+1 \\ G_1(\bar{c}) &= F_2(\bar{b})+1 \\ G_3(\bar{c}) &= F_4(\bar{b})+1 \end{aligned}$$

Combined with the previous four conditions namely (16) written in a similar way

$$\begin{aligned} G_1(\bar{c}) &= F_1(\bar{d})+1 \\ G_2(\bar{a}) &= F_2(\bar{b})+1 \\ G_4(\bar{a}) &= G_3(\bar{c})+1 \\ F_4(\bar{b}) &= F_3(\bar{d})+1 \end{aligned}$$

we arrive at:

$$\begin{aligned} F_1(\bar{d}) = G_2(\bar{a})+1 = F_2(\bar{b}) = G_1(\bar{c})+1 \\ F_3(\bar{d}) = G_4(\bar{a})+1 = F_4(\bar{b})+1 = G_3(\bar{c}) \end{aligned}$$

which is exactly the condition (14) in the Theorem. This completes the proof. □

Therefore one has to choose only S_1, S_2, S_3, S_4 and F_1, F_3 in order to build S. When S_1, S_2, S_3 and S_4 are fixed there are "only" 2^{n+1} choices for F_1, F_3 in order to get S. In the general case the construction then becomes as follows:

$$\begin{array}{lcr}
y_i & = & x_{n+2}[x_{n+1}t_i + (1+x_{n+1})u_i] \\
 & + & (1+x_{n+2})[x_{n+1}v_i + (1+x_{n+1})w_i] \qquad \text{for } i = 1, \ldots, n \qquad (17) \\
y_{i+1} & = & x_{n+2}[x_{n+1}F_1(\bar{x}) + (1+x_{n+1})[F_1(S_1^{-1}(u_i)) + 1]] \\
 & + & (1+x_{n+2})[x_{n+1}[F_1(S_1^{-1}(v_i))] \\
 & + & (1+x_{n+1})[F_1(S_1^{-1}(w_i)) + 1]] \\
y_{i+2} & = & x_{n+2}[x_{n+1}F_3(\bar{x}) + (1+x_{n+1})[F_3(S_1^{-1}(u_i))]] \\
 & + & (1+x_{n+2})[x_{n+1}[F_3(S_1^{-1}(v_i)) + 1] \\
 & + & (1+x_{n+1})[F_3(S_1^{-1}(w_i)) + 1]]
\end{array}$$

The complexity to construct all 6×6 S-boxes given four 4×4 S-boxes will be 2^{32}.

3 Permutations and Self-Equivalence

If we rewrite Definition 1 as $A^{-1} \circ S = S \circ B$ we can say that one can "push" the affine S-box B through S (so moving it from "left to right") and obtain A^{-1}. We can thus associate with a self-equivalent S-box S the pair (A, B), where one of the pair entries completely determines the other one. Note that for a given S-box S more than one pair (A, B) can exist (here we do not consider the trivial case of $A = B = Id$) since $S = A \circ S \circ B$ can also be rewritten as $A^{-1} \circ S \circ B^{-1} = S$, for example. In other words when (A, B) is a pair for S then (A^{-1}, B^{-1}) is a pair too.

Lemma 1. *The number of self-equivalent pairs (A, B) is an affine invariant, i.e. all S-boxes in a given affine equivalent class has the same number of pairs.*

Proof. Let S' and S be two S-boxes which are affine equivalent (i.e. belong to the same class), then there exist affine permutations C and D, such that $S' = C \circ S \circ D$ or equivalently $S = C^{-1} \circ S' \circ D^{-1}$. Using Definition 1 it follows that $C^{-1} \circ S' \circ D^{-1} = A \circ C^{-1} \circ S' \circ D^{-1} \circ B$ or equivalently $S' = C \circ A \circ C^{-1} \circ S' \circ D^{-1} \circ B \circ D$. Denote by $A' = C \circ A \circ C^{-1}$ and by $B' = D^{-1} \circ B \circ D$ we get $S' = A' \circ S' \circ B'$. □

Definition 4. *An S-box S is called (affine) self-equivalent inverse if there exist affine permutations A and B such that $S^{-1} = A \circ S \circ B$ holds.*

Note that the inverse S-box in general may not belong to the same affine class, e.g. because it has different algebraic degree, however when this is the case we can establish some interesting properties. Again, if we rewrite the definition above as $A^{-1} \circ S^{-1} = S \circ B$ we can say that one can "push" the affine S-box B through S (so moving it from "left to right") and obtain A^{-1} for S^{-1}.

Similar to the self-equivalence case one can thus associate with a self-equivalent inverse S-box S the pair (A, B) where one of the pair entries completely determine the other. Note that for a given S-box S more than one pair (A, B) can exist (here we do not consider the trivial case of $A = B = Id$ and S involution). Indeed observe that if (A, B) is a pair for self-inverse S then (B, A) is pair too. Since $S^{-1} = A \circ S \circ B$ can be rewritten as $A^{-1} \circ S^{-1} \circ B^{-1} = S$ we obtain that $B \circ S \circ A = S^{-1}$ holds.

Lemma 2. *The number of self-equivalent inverse pairs (A, B) is an affine invariant, i.e. all S-boxes in a given affine equivalent class have the same number of pairs.*

Proof. Let S' and S be 2 S-boxes which are affine equivalent (i.e. belong to the same class) then there exist affine permutations C, D such that $S' = C \circ S \circ D$ or equivalently $S = C^{-1} \circ S' \circ D^{-1}$ and thus $S^{-1} = D \circ S'^{-1} \circ C$. Using Definition 4 it follows that $D \circ S'^{-1} \circ C = A \circ C^{-1} \circ S' \circ D^{-1} \circ B$. or equivalently $S'^{-1} = D^{-1} \circ A \circ C^{-1} \circ S' \circ D^{-1} \circ B \circ C^{-1}$. Denote by $A' = D^{-1} \circ A \circ C^{-1}$ and by $B' = D^{-1} \circ B \circ C^{-1}$ then we get $S'^{-1} = A' \circ S' \circ B'$. □

We should further check whether every self-equivalent inverse S-box gives an involution, i.e. $S^{-1} = S$ or equivalently $S \circ S = Id$.

Lemma 3. *If a self-equivalent inverse S-box S has a pair (A, B), such that $A = B$ then in that affine equivalent class there is an involution.*

Proof. We have shown in the previous lemma that by setting $A' = D^{-1} \circ A \circ C^{-1}$ and $B' = D^{-1} \circ B \circ C^{-1}$ one transforms an S-box S to another S'. Now when S' is an involution then $A' = B' = Id$ and thus $A = B = D \circ C$. □

Lemma 4. *If an affine equivalent class, except for the class of the affine permutations, has an involution then it has many involutions.*

Proof. If $S^{-1} = S$ then every affine permutation C will give a rise to another involution $S' = C \circ S \circ C^{-1}$. This directly follows from the transformation $A' = D^{-1} \circ A \circ C^{-1}$ and by $B' = D^{-1} \circ B \circ C^{-1}$ by taking into account that $A = B = Id$ and $D = C^{-1}$ so $A' = B' = Id$. □

Can we claim a relation between the number of self-equivalent pairs and the number of self-equivalent inverse pairs in a given class?

Theorem 4. *The number of self-equivalent pairs in an affine equivalent class is at least the number of self-equivalent inverse pairs in that class.*

Proof. Since (A, B) and (B, A) are both pairs for self-inverse S we have $S^{-1} = A \circ S \circ B$ and $S^{-1} = B \circ S \circ A$. From the first we get $S = A^{-1} \circ S^{-1} \circ B^{-1}$ and replacing S^{-1} in the second we get $S = A^{-1} \circ B \circ S \circ A \circ B^{-1}$ which is what we are aiming for: by setting $C = A^{-1} \circ B$ and $D = A \circ B^{-1}$ i.e $S = C \circ S \circ D$. Thus, the self-equivalent inverse pair (A, B) uniquely determines the self-equivalent pair (C, D).

For the opposite direction, given self-equivalent pair (C, D) we have $B = A \circ C$ then $B^{-1} = C^{-1} \circ A^{-1}$ which gives $D = A \circ C^{-1} \circ A^{-1}$. However, how many

A satisfy this last equation is not clear. Notice that for a fixed self-equivalent pair (C, D) and any self-equivalent inverse pair (A, B) one can obtain a “new” self-equivalent inverse pair $(A' = A \circ C, B' = D \circ B) = (B, A)$. □

Corollary 1. *When we have a self-equivalent inverse S in a given affine equivalent class then we have also an involution in that class.*

Proof. Since $(C = Id, D = Id)$ is the trivial self-equivalent pair for S, then it is easy to find the corresponding self-equivalent inverse pair (A, B). Using Theorem 4 we obtain that $A = B$ should hold. The latter implies that in the class of S there will be an involution (using Lemma 3). □

One approach to find an involution in a class (when exists) is to use the brute-force method to find the inverse self-equivalence $S^{-1} = A \circ S \circ B$ through all possible pairs (A, B). Instead, we can do better using Lemma 3 to look for a single affine permutation A, such that $S^{-1} = A \circ S \circ A$ holds. Then both $S' = A \circ S$ or $S'' = S \circ A$ are involutions. The complexity of this search is half of the brute-force complexity.

4 Conclusions

We have shown that Shannon's expansion can be used to construct uniform sharing for certain affine equivalent classes of S-boxes. We have also shown the limitations of this expansion, namely that not all 4-bit S-box classes can be obtained from the 3-bit S-box classes in this way.

References

1. D. Bozilov, B. Bilgin, HA. Sahin, A Note on 5-bit Quadratic Permutations Classification, IACR ToSC 2017 (1), pp. 398-404.
2. B. Bilgin, S. Nikova, V. Rijmen, V. Nikov, G. Stutz, Threshold Implementations of all 3×3 and 4×4 S-boxes, CHES 2012, LNCS 7428, pp. 76-91.
3. C. De Canniere, Analysis and Design of Symmetric Encryption Algorithms, PhD thesis, KU Leuven, 2007.
4. C. Carlet, Vectorial Boolean Functions for Cryptography, chapter of the volume “Boolean Methods and Models”, Cambridge University Press, Eds. Yves Crama and Peter Hammer.
5. J. Daemen, V. Rijmen, The Design of Rijndael: AES The Advanced Encryption Standard, Springer-Verlag, 2002.
6. G. Leander, A. Poschmann, On the classification of 4 bit S-boxes, WAIFI 2007, pp. 159176
7. R. Lidl, H. Niederreiter, Finite Fields, Encyclopedia of Mathematicsand its Applications, vol. 20, Addison-Wesley, 1983.
8. S. Nikova, C. Rechberger, V. Rijmen, Threshold Implementations Against Side-Channel Attacks and Glitches, ICICS 2006, LNCS 4307, Springer-Verlag pp. 529-545.
9. S. Nikova, V. Rijmen, M. Schlaffer, Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches, Journal of Cryptology (2011), Volume 24, Issue 2 pp. 292-321.