Threshold Implementations of all 3×3 and 4×4 S-boxes *

Begül Bilgin[1,3], Svetla Nikova[1], Ventzislav Nikov[4], Vincent Rijmen[1,2], and Georg Stütz[2]

[1] KU Leuven, Dept. ESAT/SCD-COSIC and IBBT, Belgium
[2] Graz University of Technology, IAIK, Austria
[3] University of Twente, EEMCS-DIES, The Netherlands
[4] NXP Semiconductors, Belgium

Abstract. Side-channel attacks have proven many hardware implementations of cryptographic algorithms to be vulnerable. A recently proposed masking method, based on secret sharing and multi-party computation methods, introduces a set of sufficient requirements for implementations to be provably resistant against first-order DPA with minimal assumptions on the hardware. The original paper doesn't describe how to construct the Boolean functions that are to be used in the implementation. In this paper, we derive the functions for all invertible 3×3, 4×4 S-boxes and the 6×4 DES S-boxes. Our methods and observations can also be used to accelerate the search for sharings of larger (e.g. 8×8) S-boxes. Finally, we investigate the cost of such protection.
Keywords: DPA, masking, glitches, sharing, nonlinear functions, S-box, decomposition

1 Introduction

Side-channel analysis exploits the information leaked during the computation of a cryptographic algorithm. The most common technique is to analyze the power consumption of a cryptographic device using differential power analysis (DPA). This side-channel attack exploits the correlation between the instantaneous power consumption of a device and the intermediate results of a cryptographic algorithm.

Several countermeasures against side-channel attacks have been proposed. Circuit design approaches try to balance the power consumption

* This work was supported in part by the Research Council KU Leuven: GOA TENSE (GOA/11/007) and by the European Commission under contracts ICT-2007-216646 (ECRYPT II). V. Nikov was supported by the European Commission (FP7) within the Tamper Resistant Sensor Node (TAMPRES) project with contract number 258754 and the Internet of Things - Architecture (IoT-A) project with contract number 257521.

of different data values [31]. Another method is to randomize the intermediate values of an algorithm by masking them. This can be done at the algorithm level [1, 5, 12, 24], at the gate level [13, 27, 32] or even in combination with circuit design approaches [25].

Many of these approaches result in very secure *software* implementations. However, it has been shown that *hardware* implementations are much more difficult to protect against DPA [17]. The problem of most of these masking approaches is that they underestimate the amount of information that is leaked by hardware, for instance during glitches or other transient effects. The security proofs are based on an idealized hardware model, resulting in requirements on the hardware that are very expensive to meet in practice. The main advantages of the threshold implementation approach are that it provides provable security against first-order DPA attacks with minimal assumptions on the hardware technology, in particular, it is also secure in the presence of glitches, and that the method allows to construct realistic-size circuits [20, 22, 23].

1.1 Organization and contributions of this paper

The remainder of this paper is organized as follows. In Section 2 we introduce the notation and provide some background material. Section 2.6 contains our first contribution: a classification of S-boxes which simplifies the task to find implementations for all S-boxes. In Section 3 we present our second contribution: a method to decompose permutations as a composition of quadratic ones. We prove that all 4-bit S-boxes in the alternating group can be decomposed in this way. We extend the sharing method in Section 4 and show that all 3×3, 4×4 and DES 6×4 S-boxes can be shared with minimum 3 and/or 4 shares. We investigate the cost of an HW implementation of the shared S-boxes in Section 5. Some ideas for further improvements will be provided in the full version of the paper [2]. Finally, we conclude in Section 6.

2 Preliminaries

We consider n-bit permutations sometimes defined over a vector space $\mathcal{F}_2^n$ or over a finite field $GF(2^n)$. The degree of such a permutation F is the algebraic degree of the (n, n) vectorial Boolean function [6] or also called n-bit S-box. Any such function $F(x)$ can be considered as an n-tuple of Boolean functions $(f_1(x), \ldots, f_n(x))$ called the coordinate functions of $F(x)$.

2.1 Threshold implementations

Threshold implementations (TI), are a kind of side-channel attack countermeasures, based on secret sharing schemes and techniques from multiparty computation. The approach can be summarized as follows. Split a variable x into s additive shares x_i with $x = \sum_i x_i$ and denote the vector of the s shares x_i by $\mathbf{x} = (x_1, x_2, \dots, x_s)$. In order to implement a function $a = F(x, y, z, \dots)$ from $\mathcal{F}_2^m$ to $\mathcal{F}_2^n$, the TI method requires a *sharing*, i.e. a set of s functions F_i which together compute the output(s) of F. A sharing needs to satisfy three properties:

Correctness: $a = F(x, y, z, \dots) = \sum_i F_i(\mathbf{x}, \mathbf{y}, \mathbf{z}, \dots)$ for all $\mathbf{x}, \mathbf{y}, \mathbf{z}, \dots$ satisfying $\sum_i x_i = x, \sum_i y_i = y, \sum_i z_i = z, \dots$

Non-completeness: Every function is independent of at least one share of the input variables x, y, z. This is often translated to "F_i should be independent of $x_i, y_i, z_i, \dots$."

Uniformity (balancedness): For all $(a_1, a_2, \dots, a_s)$ satisfying $\sum_i a_i = a$, the number of tuples $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \dots) \in \mathcal{F}^{ms}$ for which $F_j(\mathbf{x}, \mathbf{y}, \mathbf{z}, \dots) = a_j, 1 \leq j \leq s$, is equal to $2^{(s-1)(m-n)}$ times the number of $(x, y, z, \dots) \in \mathcal{F}^m$ for which $a = F(x, y, z, \dots)$. Hence, if F is a permutation on $\mathcal{F}^m$, then the functions F_i define together a permutation on $\mathcal{F}^{ms}$. In other words, the sharing preserves the output distribution.

This approach results in combinational logic with the following properties. Firstly, since each F_i is completely independent of the unmasked values, also the subcircuits implementing them are, even in the presence of glitches. Because of the linearity of the expectation operator, the same holds true for the average power consumption of the whole circuit, or any linear combination of the power consumptions of the subcircuits. This implies perfect resistance against all first-order side-channel attacks [23]. The approach was recently extended and applied to Noekeon [23], Keccak [4], Present [26] and AES [19]. Whereas it is easy to construct for any function a sharing satisfying the first two properties, the uniformity property poses more problems. Hence reasonable questions to ask are: which functions (S-boxes) can be shared with this approach, how many shares are required and how can we construct such sharing?

A similar approach was followed in [28], where Shamir's secret sharing scheme is used to construct hardware secure against dth-order side-channel attacks in the presence of glitches. Instead of constructing dedicated functions F_i, they propose a general method which replaces every field multiplication by $4d^3$ field multiplications and $4d^3$ additions, using

$2d^2$ bytes of randomness. While the method is applicable everywhere, in principle, there are cases where it may prove too costly.

2.2 Decomposition as a tool to facilitate sharing

In order to share a nonlinear function (S-box) with algebraic degree d, at least $d+1$ shares are needed [20, Theorem 1]. Several examples of functions shared with 3 shares, namely quadratic Boolean function of two and three variables, multiplication on the extension field $GF(2^{2m})/GF(2^m)$ (e.g. multiplication in $GF(4)$), and the Noekeon S-box have been provided [20, 22, 23]. A realization of the inversion in $GF(16)$ with 5 shares was given in [20]. Since the area requirements of an implementation increase with the number of shares, it is desirable to keep the number of shares as low as possible.

The block ciphers Noekeon and Present have been designed for compact hardware implementations. They have S-boxes, which are not very complex 4×4 cubic permutations. Realizations for these two block ciphers have been presented for Noekeon in [22,23] and in [26] for Present. In order to decrease the algebraic degree of the functions for which sharings need to be found, these three realizations decompose the S-box into two parts. For the Present S-box, decompositions $S(x) = F(G(x))$ with $G(0) = 0$ have been found where $F(x)$ and $G(x)$ are quadratic permutations [26]. By varying the constant term $G(0)$ the authors found all possible decompositions of $S(X) = F(G(X))$. Both S-boxes $F(x), G(x)$ have been shared with three shares (F_1, F_2, F_3) and (G_1, G_2, G_3) that are correct, non-complete and uniform.

When the AES S-box (with algebraic degree seven) is presented using the tower field approach, the only nonlinear operation is the multiplication in $GF(4)$, which is a quadratic mapping [19]. This observation has lead to a TI for AES with 3 shares. In order to guarantee the uniformity, re-sharing (also called re-masking) has been used four times. Re-sharing is a technique where fresh uniform and random masks/shares are added inside a pipeline stage in order to make the shares follow an uniform distribution again.

A novel fault attack technique against several AES cores including one claimed to be protected with TI method has been proposed in [18]. But as the authors pointed out, contrary to the AES TI implementation in [19], their targeted core has been made without satisfying the non-completeness and uniformity properties by "sharing" the AND gates with 4 shares formula from [19,20]. Since the used method does not satisfy the TI properties it should not be called a TI implementation of AES. In

addition, the TI method was never claimed to provide protection against fault attacks.

2.3 Equivalence classes for $n = 2, 3, 4$

Definition 1 ([8]). *Two S-boxes $S_1(x)$ and $S_2(x)$ are* affine/linear equivalent *if there exists a pair of invertible affine/linear permutation $A(x)$ and $B(x)$, such that $S_1 = B \circ S_2 \circ A$.*

Every invertible affine permutation $A(x)$ can be written as $\mathrm{A} \cdot x + a$ with a an n-bit constant and A an $n \times n$ matrix which is invertible over $GF(2)$. It follows that there are $2^n \times \prod_{i=0}^{n-1}(2^n - 2^i)$ different invertible affine permutations.

The relation "being affine equivalent" can be used to define equivalence classes. We now investigate the number of classes of invertible $n \times n$ S-boxes for $n = 2, 3, 4$. Note that the algebraic degree is affine invariant, hence all S-boxes in a class have the same algebraic degree.

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 [8]: 3 classes containing quadratic functions, and one class containing the affine functions. We will provide a table with a representative of each class in the full version of the paper [2].

The maximal algebraic degree of a balanced 4-variable Boolean function is 3 [7, 16]. De Cannière uses an algorithm to search for the affine equivalent classes which guesses the affine permutation A for as few input points as possible, and then uses the linearity of A and B to follow the implications of these guesses as far as possible. This search is accelerated by applying the next observation, which follows from linear algebra arguments (change of basis):

Lemma 1 ([15]). *Let S be an $n \times n$ bijection. Then S is affine equivalent to an S-box $\tilde{S}$ with $\tilde{S}(0) = 0$, $\tilde{S}(1) = 1$, $\tilde{S}(2) = 2$, ..., $\tilde{S}(2^{n-1}) = 2^{n-1}$.*

In the case $n = 4$, this observation reduces the search space from $16! \approx 2^{44}$ to $11! \approx 2^{25}$.

De Cannière lists the 302 equivalence classes for the 4×4 bijections [8]: the class of affine functions, 6 classes containing quadratic functions and the remaining 295 classes containing cubic functions.[1] We will list the classes in the full version of the paper [2]. The numbering of the classes is derived from the lexicographical ordering of the truth tables of

[1] Independent of [8, 15], Saarinen classified the 4×4 S-boxes using a different equivalence relation [30].

the S-boxes. In order to increase readability, we introduce the following notation $\mathcal{A}_i^n$, $\mathcal{Q}_j^n$, $\mathcal{C}_k^n$ to denote the Affine class number i, Quadratic class number j and Cubic class number k of permutations of $\mathcal{F}_2^n$.

2.4 Order of a permutation

All bijections from a set X to itself (also called permutations) form the *symmetric group* on X denoted by S_X. A transposition is a permutation which exchanges two elements and keeps all others fixed. A classical theorem states that every permutation can be written as a product of transpositions [29], and although the representation of a permutation as a product of transpositions is not unique, the number of transpositions needed to represent a given permutation is either always even or always odd. The set of all even permutations form a normal subgroup of S_X, which is called the *alternating group* on X and denoted by A_X. The alternating group contains half of the elements of S_X. Instead of A_X and S_X, we will write here A_n and S_n, where n is the size of the set X.

2.5 Known S-boxes and their classes

There are only few cryptographically significant 3×3 S-boxes: the Inversion in $GF(2^3)$, the PRINTcipher, the Threeway and the Baseking S-boxes. They all belong to Class 3. There are many cryptographically significant 4×4 S-boxes. To mention some of them: Twofish, Gost, Serpent, Lucifer, Clefia, HB1, HB2, mCrypton, Klein, Khazad, Iceberg, Puffin, Present, Luffa, Hamsi, JH, Noekeon, Piccolo.

2.6 The inverse S-box

Note that S^{-1}, the inverse S-box, is not necessarily affine equivalent to S and in this case may not have the same algebraic degree. We know however, that the inverse of an affine permutation is always an affine permutation. In the case of 3×3 S-boxes it follows that the inverse of a quadratic permutation is again a quadratic permutation. Moreover, it can be shown that the 3 quadratic classes in S_8 are self-inverse, i.e. S^{-1} belongs to the same class as S. In the case $n = 4$, we can apply the following lemma.

Lemma 2 ([6]). *Let F be a permutation of $GF(2^n)$, then $deg(F^{-1}) = n - 1$ if and only if $deg(F) = n - 1$.*

Since the inverse of an affine S-box is affine, and, when $n = 4$, the inverse of a cubic S-box is cubic, it follows that in this case the inverse of a quadratic S-box is quadratic. The Keccak S-box ($n = 5$) is an example where the algebraic degree of the inverse S-box (3) is different from the algebraic degree of the S-box itself (2) [3].

We have observed that there are 172 self-inverse classes in S_{16}. The remaining 130 classes form 65 pairs, i.e., any S-box S of the first class has an inverse S-box S^{-1} in the second class (and vice versa). We will provide the list of the pairs of inverse classes in the full version of the paper [2].

3 Decomposition of 4×4 S-boxes

In this section we consider all 4×4 bijections, and investigate when a cubic bijection from S_{16} can be decomposed as a *composition of quadratic bijections*. We will refer to the minimum number of quadratic bijections in such a decomposition as *decomposition length*. Recall that the Noekeon S-box is cubic but defined as a composition of two quadratic S-boxes in $\mathcal{F}_2^4$: $S(x) = S_2(S_1(x))$. Similarly the Present S-box is cubic but has also been shown to be decomposable in two quadratic S-boxes.

Lemma 3. *If an S-box S can be decomposed into a sequence of t quadratic S-boxes, then all S-boxes which are affine equivalent to S can be decomposed into a sequence of t quadratic S-boxes.*

Lemma 4 ([33]). *For all n, the $n \times n$ affine bijections are in the alternating group.*

Lemma 5. *All 4×4 quadratic S-boxes belong to the alternating group A_{16}.*

Proof. Since all invertible affine transformations are in the alternating group (the previous Lemma), two S-boxes which are affine equivalent, are either both even or both odd. We have taken one representative of each of the 6 quadratic classes $\mathcal{Q}_i^4$ for $i \in \{4, 12, 293, 294, 299, 300\}$ [8] and have verified that their parities are even. □

Now we investigate which permutations we can generate by combining the affine and the quadratic permutations. We start with the following lemma.

Lemma 6. *Let Q_i be 6 arbitrarily selected representatives of the 6 quadratic classes $\mathcal{Q}_i^4$. (Hence $i \in \{4, 12, 293, 294, 299, 300\}$.) Then all cubic permutations S that have decomposition length 2, are affine equivalent to one*

of the cubic permutation that can be written as

$$\tilde{S}_{i\times j} = Q_i \circ A \circ Q_j, \tag{1}$$

where A is an invertible affine permutation and $i, j \in \{4, 12, 293, 294, 299, 300\}$.

It follows that we can construct all cubic classes of decomposition length 2 by running through the 36 possibilities of $i \times j$ and the 322560 invertible affine transformations in (1). This approach produces 30 cubic classes. In the remainder, we will denote the S-boxes $\tilde{S}_{i\times j}$ by $i \times j$ and refer to them as the *simple solutions*. In the full version of the paper [2] we provide the list of the simple solutions for all 30 decompositions with length 2. Note that if $Q_i \circ A \circ Q_j = S$, i.e. S can be decomposed as a product of $i \times j$, then $Q_j^{-1} \circ A^{-1} \circ Q_i^{-1} = S^{-1}$. Since for $n = 4$ all quadratics are affine equivalent to their inverse, it follows that S^{-1} is decomposed as a product of $j \times i$. Thus any self-inverse class has decomposition $i \times j$ and $j \times i$ as well. For the pairs of inverse classes we conclude that if $i \times j$ belongs to the first class then $j \times i$ belongs to the second class.

To obtain all decompositions with length 3 we use similar approach as for length 2 but the first permutation Q_i is cubic (instead of quadratic) and belongs to the already found list of cubic classes decomposable with length 2. It turns out that we can generate in this way the 114 remaining elements of A_{16}.

Summarizing, we can prove the following Theorem and Lemma (stated without proof in [9]).

Theorem 1. *A 4×4 bijection can be decomposed using quadratic bijections if and only if it belongs to the alternating group A_{16} (151 classes).*

Proof. ($\Rightarrow$) Let S be a bijection which can be decomposed with quadratic permutations say $Q_1 \circ Q_2 \circ \ldots \circ Q_t$. Since all $Q_i \in A_{16}$ (Lemma 5) and the alternating group is closed it follows that $S \in A_{16}$.
($\Leftarrow$) Lemma 3, Lemma 6 and the discussion following it imply that we can generate all elements of the alternating group using quadratic permutations. □

The left-hand-side columns of Table 1 list the decompositions of all 4×4 S-boxes. Theorem 1 implies that the classes which are not in the alternative group i.e. in $S_{16} \setminus A_{16}$, can't be decomposed as a product of quadratic classes. Now we make the following simple observation:

Lemma 7. *Let $\tilde{S}$ be a fixed permutation in $S_{16} \setminus A_{16}$ then any cubic permutation from $S_{16} \setminus A_{16}$ can be presented as a product of $\tilde{S}$ and a permutation from A_{16}.*

4 Sharing with 3, 4 and 5 shares

In this section we focus first on the permutations which can be shared with 3 shares, i.e. all S-boxes in $\mathcal{F}_2^3$ and half of the S-boxes in $\mathcal{F}_2^4$. Next we focus on those functions that can be shared with 4 shares, i.e. the other half of the S-boxes in $\mathcal{F}_2^4$. Then, we will show how to share all of these S-boxes in $\mathcal{F}_2^4$ with 5 shares without need of a decomposition.

4.1 A basic result

Theorem 2. *If we have a sharing for a representative of a class, then we can derive a sharing for all S-boxes from the same class.*

Proof. Let S be an $n \times n$ S-box which has a uniform, non-complete and correct sharing $\bar{S}$ using s shares S_i. Denote the input vector of S by x, and the shares by x_i. Each S_i contains n coordinate shared functions depending on at most $(s-1)$ of the x_i, such that the noncompleteness property is satisfied. We denote by $\mathbf{x}_i$ the vector containing the $s-1$ inputs of S_i.

We now construct a uniform, non-complete and correct sharing for any S-box $\tilde{S}$ which is affine equivalent to S. By definition, there exist two $n \times n$ invertible affine permutations A and B s.t. $\tilde{S} = B \circ S \circ A$. In order to lighten notation, we give the proof for the case that A and B are linear permutations. We define $\bar{A}, \bar{B}$ as the $ns \times ns$ permutations that apply A, respectively B, to each of the shares separately:

$$\bar{A}(x_1, x_2, \dots x_s) = (A(x_1), A(x_2), \dots A(x_s)),$$
$$\bar{B}(x_1, x_2, \dots x_s) = (B(x_1), B(x_2), \dots B(x_s)).$$

Denote $y_i = A(x_i), 1 \leq i \leq s$ and define $\mathbf{y}_i$ as the vector containing the $s-1$ shares y_i that we need to compute S_i. Consider $\bar{S}(\bar{A}(x_1, x_2, \dots, x_s)) = (S_1(\mathbf{y}_1), S_2(\mathbf{y}_2), \dots S_s(\mathbf{y}_s))$. By slight abuse of notation we can write $\mathbf{y}_i = \bar{A}(\mathbf{x}_i)$ and see that the noncompleteness of the $\bar{S}_i$ is preserved in $\bar{S} \circ \bar{A}$. Since $\bar{A}$ is a permutation, it preserves the uniformity of the input and since $\bar{S}$ is uniform so will be the composition $\bar{S} \circ \bar{A}$. The correctness follows from the fact that $\bar{S}$ is a correct sharing and that

$$y_1 + y_2 + \dots + y_s = A(x_1) + A(x_2) + \dots + A(x_s) = A(x_1 + x_2 + \dots x_s) = A(x).$$

Consider now $\bar{B}(\bar{S}(A(x))) = (B(S_1(\mathbf{y}_1)), B(S_2(\mathbf{y}_2)), \dots, B(S_s(\mathbf{y}_s)))$. Since $\bar{B}$ is a permutation, it preserves uniformity of the output and since $\bar{S}$

is uniform, the composition $\bar{B} \circ \bar{S}$ is uniform. The composition is non-complete since the $\bar{S}_i$ are non-complete and $\bar{B}$ doesn't combine different shares. Correctness follows from the fact that $\bar{S}$ is a correct sharing and hence

$$\begin{aligned} & B(S_1(\mathbf{y}_1)) + B(S_2(\mathbf{y}_2)) + \cdots + B(S_s(\mathbf{y}_s)) \\ = \; & B(S_1(\mathbf{y}_1) \; + S_2(\mathbf{y}_2) + \cdots + S_s(\mathbf{y}_s)) = B(S(A(x))). \end{aligned} \qquad \square$$

4.2 Direct sharing

The most difficult property to be satisfied when the function is shared is the uniformity. Assume that we want to construct a sharing for the function $F(x, y, z)$ with 3 shares. Then it is easy to produce a sharing which satisfies the correctness and the non-completeness requirements and is rotation symmetric, by means of a method that we call the *direct sharing method*, and that we now describe. First, we replace every input variable by the sum of 3 shares. The correctness is satisfied if we ensure that

$$F_1 + F_2 + F_3 = F(x_1 + x_2 + x_3, y_1 + y_2 + y_3, z_1 + z_2 + z_3).$$

In order to satisfy non-completeness, we have to divide the terms of the right hand side over the three F_j in such a way that F_j doesn't contain a term in x_j. We achieve this by assigning the linear terms containing an index j to F_{j-1}, the quadratic terms containing indices j and $j+1$ to F_{j-1} and the quadratic terms containing indices j only to F_{j-1}. For example,

$$\begin{aligned} F(x, y, z) = x + yz, \quad & \text{gives:} \\ F_1 &= x_2 + z_2y_2 + z_2y_3 + z_3y_2 \\ F_2 &= x_3 + z_3y_3 + z_3y_1 + z_1y_3 \\ F_3 &= x_1 + z_1y_1 + z_1y_2 + z_2y_1. \end{aligned}$$

Note that the uniformity of sharing produced in this way is not guaranteed. It has to be verified separately. The method can easily be generalized for larger number of shares.

Direct sharing has been used in [26] for the decomposition of the quadratic permutations F and G of the Present S-box S and similarly for Noekeon [23], Keccak [4].

With the direct sharing method we were able to find sharings respecting the uniformity condition for all 1344 permutations of $\mathcal{Q}_1^3$, but none of

$\mathcal{Q}_2^3$ and $\mathcal{Q}_3^3$. We were also able to find sharings for all 322560 permutations of $\mathcal{Q}_4^4$, $\mathcal{Q}_{294}^4$ and $\mathcal{Q}_{299}^4$, but none of $\mathcal{Q}_{12}^4$, $\mathcal{Q}_{293}^4$ and $\mathcal{Q}_{300}^4$. So, unfortunately half of the quadratic S-boxes can't be shared directly with length 1 but we still can find a sharing with length 2 by decomposing them as a composition of the already shared quadratic S-boxes. Thus, if we use only direct sharing we will be able to find sharings for all S-boxes in the alternating group but at the cost of longer path.

4.3 Correction terms

Since direct sharing not always results in an uniform sharing the use of *correction terms* (CT) has been proposed [20, 22]. Correction terms are terms that can be added in pairs to more than one share such that they satisfy the non-completeness rule. Since the terms in a pair cancel each other, the sharing still satisfies the correctness.

By varying the CT one can obtain all possible sharings of a given function. Consider a Boolean quadratic function with m variables (1 output bit), which we want to share with 3 shares. Note that the only terms which can be used as CT are x_i or $x_i y_i$ (or higher degree) for $i = 1, 2, 3$. Indeed terms like $x_i y_j$ for $i \neq j$ can't be used in the i-th and j-th share of the function because of the non-completeness rule and therefore such a term can be used in only 1 share, hence it can't be used as a CT.

Thus counting only the linear and quadratic CT and ignoring the constant terms, which will not influence the uniformity, for a quadratic function with m variables we obtain that there are $3(m+\binom{m}{2})$ CT. Taking into account all possible positions for the CT we get $2^{3(m+\binom{m}{2})}$ sharings. For example, for a quadratic function of 3 variables there are 2^{18} possible CT and therefore for a 3×3 S-boxes the search space will be 2^{54}. This makes the exhaustive search (to find a single good solution) over all CT unpractical, even for small S-boxes. For sharing with 4 shares even more terms can be used as CT.

4.4 A link between the 3×3 S-boxes and some quadratic 4×4 S-boxes

Lemma 8. *There is a transformation which expands $\mathcal{Q}_1^3$, $\mathcal{Q}_2^3$ and $\mathcal{Q}_3^3$ into $\mathcal{Q}_4^4$, $\mathcal{Q}_{12}^4$ and $\mathcal{Q}_{300}^4$ correspondingly.*

Proof. Starting from a 3×3 S-box S and adding a new variable we can obtain a 4×4 S-box $\tilde{S}$. Namely, the transformation is defined as follows: let $S(w, v, u) = (y1, y2, y3)$ and define $\tilde{S}(x, w, v, u) = (y1, y2, y3, x)$. It is

easy to check that this transformation maps the first 3 classes into the other 3 classes. □

The relation from Lemma 8 explains why if we have a sharing for a class in $\mathcal{F}_2^3$ we also obtain a sharing for the corresponding class in $\mathcal{F}_2^4$ and vice versa, i.e., if we can't share a class the corresponding class also can't be shared. The results we have obtained with 3 shares are summarized in Table 1 (middle columns).

Recall that if we use only direct sharing we will be able to share with 3 shares all S-boxes in the alternating group but at the cost of longer path than the one obtained by decomposition. However using CT we found sharing for classes: $\mathcal{Q}_1^3$, $\mathcal{Q}_2^3$, $\mathcal{Q}_4^4$, $\mathcal{Q}_{12}^4$, $\mathcal{Q}_{293}^4$, $\mathcal{Q}_{294}^4$ and $\mathcal{Q}_{299}^4$. So all quadratic classes except $\mathcal{Q}_3^3$ and $\mathcal{Q}_{300}^4$ can be shared with 3 shares and without decomposition. We want to pose an open question: find sharing without decomposition to classes $\mathcal{Q}_3^3$ and $\mathcal{Q}_{300}^4$ or show why they can't be shared with 3 shares in that way.

4.5 Sharing using decomposition

As an alternative to the search through a set of correction terms, we can also construct sharings after using decomposition: we try to decompose S-boxes into S-boxes for which we already have sharings. This decomposition problem is more restrained than the basic problem discussed in Section 3 for sharing with 3 shares, since we can use only the quadratic S-boxes for which we already have a sharing. It turns out that this extra requirement sometimes increases the decomposition length by one. For example, decomposition for $\mathcal{Q}_3^3$ is 1×2 and 2×1, i.e., we obtain a sharing for $\mathcal{Q}_3^3$ at the cost of length 2 (instead of length 1). Similarly $\mathcal{Q}_{300}^4$ can be decomposed as 4×12, 4×293, 12×4, 12×294, 293×4, 293×294, 294×12 and 294×293 so, again we obtain a sharing with length 2. Table 1 (right columns) gives the results.

Recall that one can't find a sharing with 3 shares for cubic functions outside the alternating group. Thus, 4 shares will be required in this case. Using direct sharing with 4 shares we obtain slightly better results for the quadratic S-boxes compared to 3 shares since we were able to share also class $\mathcal{Q}_{300}^4$ (and therefore $\mathcal{Q}_3^3$ too). The sharing of class $\mathcal{Q}_{300}^4$ has further improved the sharings of $\mathcal{C}_{130}^4$, $\mathcal{C}_{131}^4$ and $\mathcal{C}_{24}^4$ which have sharing with shorter length for 4 shares than for 3 shares. We have also found sharings with 4 shares for the cubic classes $\mathcal{C}_1^4$, $\mathcal{C}_3^4$, $\mathcal{C}_{13}^4$ and $\mathcal{C}_{301}^4$ from $S_{16} \setminus A_{16}$ using direct sharing. By using Lemma 7 we obtain sharings with 4 shares for all 4×4 S-boxes. Observe that the total length of the

sharing depends on the class we use ($\mathcal{C}_1^4$, $\mathcal{C}_3^4$, $\mathcal{C}_{13}^4$ and $\mathcal{C}_{301}^4$) and also on the class from the alternating group, which is used for the decomposition. For example, class $\mathcal{C}_7^4$ can be decomposed using $\mathcal{C}_1^4$ with length 4 but with classes $\mathcal{C}_3^4$ and $\mathcal{C}_{13}^4$ it can be decomposed with length 3. Note also that the number of solutions differ. We have found 10, 31 and 49 solutions when using $\mathcal{C}_1^4$, $\mathcal{C}_3^4$ and $\mathcal{C}_{13}^4$ classes, correspondingly. Surprisingly for the classes in the alternating group we have only slight improvement with 4 shares compared to 3 shares and only a few classes in $S_{16} \setminus A_{16}$ have direct sharing with 4 shares. However with 5 shares all classes can be shared directly without decomposition which is a big improvement compare to the situation with 4 shares.

Table 1: Overview of the numbers of classes of 4×4 S-boxes that can be decomposed and shared using 3 shares, 4 shares and 5 shares. The numbers are split up according to the decomposition length of the S-boxes (1, 2, 3, or 4), respectively their shares.

unshared			3 shares				4 shares			5 shares	remark
1	2	3	1	2	3	4	1	2	3	1	
6			5	1			6			6	quadratics
	30			28	2			30		30	cubics in A_{16}
		114			113	1			114	114	cubics in A_{16}
	–			–			4	22	125	151	cubics in $S_{16} \setminus A_{16}$

An open question is why for all S-boxes the sharing with 4 shares does not improve significantly the results compared to 3 shares and suddenly with 5 shares we can share all classes with length 1.

Recall that for the Present S-box, decompositions $S(x) = F(G(x))$ have been found in [26]. The authors also made an observation that exactly $\frac{3}{7}$ sharings out of the decompositions automatically satisfy the uniformity condition (i.e. without any correction terms). Recall that with the direct sharing method without CT we (as well as the authors of [26]) were able to share only 3 quadratic classes: $\mathcal{Q}_4^4$, $\mathcal{Q}_{294}^4$ and $\mathcal{Q}_{299}^4$. The Present S-box belongs to $\mathcal{C}_{266}^4$ and has 7 simple solutions but only 3 of them can be shared namely 294×299, 299×294, 299×299, which explains the authors' observation.

In the full version of the paper [2] we provide a complete list for the sharings with 3 and with 4 shares with their lengths. Recall that all classes can be shared with 5 shares with length 1 and that for the S-boxes

in $S_{16} \setminus A_{16}$ no solution with 3 shares exist. Note that the DES 6×4 S-boxes can be considered as an affine 2×2 selection S-box with four 4×4 S-boxes attached. Since we have sharings for both 2×2 and 4×4 S-boxes we conclude that we have sharings for the DES 6×4 S-boxes as well.

5 HW implementation of the sharings

In this section, our aim is to provide a fair comparison and prediction what the cost (ratio of area to a NAND gate referred to as GE) will be for a protected S-box in a specified library. For our investigations we used the TSMC 0.18μm standard cell library in the Synopsis development tool.

Quadratic classes and cubic classes with length 1 form the basis to all our implementations. Therefore, we concentrated our efforts on these classes. While considering 3×3 S-boxes we synthesized 840 affine equivalent S-boxes for each class. However the number of S-boxes in a class increases to more than 322560 as we move to 4×4 S-boxes. In that case, we choose 1000 S-boxes per class to synthesize.

Table 2: S_8: Quadratic S-boxes sharing

3×3 S-boxes Class # in S_8		Sharing Length (L)	Original S-box	Unshared Decomposed L reg	Shared 3 shares L reg	Shared 4 shares 1 reg	Shared 5 shares 1 reg
$\mathcal{Q}_1^3$	Min	1	27.66	-	98.66	138.00	148.00
	Max		29.66		121.66	150.00	185.66
$\mathcal{Q}_2^3$	Min	1	29.00	-	116.66	174.00	180.00
	Max		29.66		155.00	226.66	220.33
$\mathcal{Q}_3^3$	Min	2	30.00	50.00	194.33	140.00	167.00
	Max		32.00	51.00	201.00	194.33	228.66

In tables 2, 3 and 4 we show the implementation results for each class only the S-box with the *minimum* GE from the result of our original S-box synthesis (over the class), as well as the S-box with the *maximum* GE. However, note that the *Min* and *Max* values should only be taken as indications.

The area results listed in the column *original S-box* for an $n \times n$ S-box include one n-bit register. If a decomposition is necessary for a correct, non-complete and uniform sharing, then we included registers in between every pipelining operation as required [23] which increases the cost as expected.

Table 3: A_{16}: Quadratic S-boxes sharing

4×4 S-boxes Quadratic Class # in S_{16}		Sharing Length (L)	Original S-box	Unshared Decomposed L reg	Shared 3 shares L reg	Shared 4 shares 1 reg	Shared 5 shares 1 reg
$\mathcal{Q}_4^4$	Min	1	37.33	-	121.33	168.33	186.33
	Max		44.00		223.33	258.00	309.00
$\mathcal{Q}_{12}^4$	Min	1	36.66	-	139.33	204.00	218.00
	Max		48.00		253.33	290.33	340.66
$\mathcal{Q}_{293}^4$	Min	1	39.33	-	165.33	194.33	235.00
	Max		48.66		297.33	313.00	358.33
$\mathcal{Q}_{294}^4$	Min	1	40.00	-	141.33	170.33	210.33
	Max		49.66		261.00	240.00	255.00
$\mathcal{Q}_{299}^4$	Min	1	40.33	-	174.33	211.00	247.00
	Max		48.00		298.00	295.33	294.66
$\mathcal{Q}_{300}^4$	Min	2	33.66	58.00	207.33	209.66	249.33
	Max		52.66	70.00	346.00	295.00	342.33

For classes with decomposition length more than 1, we randomly choose a class representative i.e. an S-box. Then we implement the smallest amongst all possible decompositions of this S-box, namely the one which gives minimum GE. We saw that, classes $\mathcal{Q}_3^3$, $\mathcal{Q}_{300}^4$, $\mathcal{C}_{150}^4$, $\mathcal{C}_{151}^4$, $\mathcal{C}_{130}^4$, $\mathcal{C}_{131}^4$, $\mathcal{C}_{24}^4$, $\mathcal{C}_{204}^4$, $\mathcal{C}_{257}^4$ and $\mathcal{C}_{210}^4$ give relatively small results when implemented as 2×1, 12×4, 12×293, 293×12, $12 \times 4 \times 299$, $299 \times 12 \times 4$, $299 \times 12 \times 4 \times 299$, 3×294, 3×12 and $3 \times 293 \times 12$ respectively. The area figures for $\mathcal{C}_{204}^4$ and $\mathcal{C}_{257}^4$ differ significantly. Closer inspection reveals that this is due to the fact that their decompositions use different S-boxes from $\mathcal{C}_3^4$; the S-box used in the decomposition of $\mathcal{C}_{204}^4$ is smaller than the one in the decomposition of $\mathcal{C}_{257}^4$.

6 Conclusions

In this paper we have considered the threshold implementation method, which is a method to construct implementations of cryptographic functions that are secure against a large class of side-channel attacks, even when the hardware technology is not glitch-free.

We have analyzed which basic S-boxes can be securely implemented using 3, 4 or 5 shares. We have constructed sharings for all 3×3, 4×4 S-boxes and 6×4 DES S-boxes. Thus we have extended the threshold implementation method to secure implementations for any cryptographic algorithm which uses these S-boxes. Note that the mixing layer in the

Table 4: S_{16}: Cubic S-boxes sharing

4×4 S-boxes Cubic Class # in S_{16}		Sharing Length (L, L')	Original S-box	Unshared Decomposed L' reg	Shared 3 shares L reg	Shared 4 shares L' reg	Shared 5 shares 1 reg
$\mathcal{C}_1^4 \in S_{16} \setminus A_{16}$	Min	1,1	39.66		–	213.66	273.66
	Max		40.33		–	378.00	464.66
$\mathcal{C}_3^4 \in S_{16} \setminus A_{16}$	Min	1,1	40.33		–	230.33	286.33
	Max		43.00		–	413.66	500.66
$\mathcal{C}_{13}^4 \in S_{16} \setminus A_{16}$	Min	1,1	40.33		–	260.00	319.00
	Max		41.33		–	423.00	502.66
$\mathcal{C}_{301}^4 \in S_{16} \setminus A_{16}$	Min	1,1	39.33		–	289.33	350.33
	Max		59.33		–	526.33	605.66
$\mathcal{C}_{150}^4 \in A_{16}$		2,2	46.33	71.66	305.33	430.66	414.33
$\mathcal{C}_{151}^4 \in A_{16}$		2,2	47.33	69.66	286.00	410.00	390.00
$\mathcal{C}_{130}^4 \in A_{16}$		3,2	48.00	97.33	393.00	375.66	442.66
$\mathcal{C}_{131}^4 \in A_{16}$		3,2	50.00	99.00	386.00	363.33	435.66
$\mathcal{C}_{24}^4 \in A_{16}$		4,3	48.33	151.33	674.00	616.66	734.66
$\mathcal{C}_{204}^4 \in S_{16} \setminus A_{16}$		2,2	49.00	80.33	-	413.00	501.33
$\mathcal{C}_{257}^4 \in S_{16} \setminus A_{16}$		2,2	47.66	73.66	-	486.00	594.00
$\mathcal{C}_{210}^4 \in S_{16} \setminus A_{16}$		3,3	47.66	119.33	-	602.00	695.33

round function of a block cipher is a linear operation and thus it is trivially shared even with 2 shares. Finally, we have implemented several of the shared S-boxes in order to investigate the cost of the sharing as well as the additional cost due to the pipelining stages separated by latches or registers.

Table 5: Range for the ratio $\frac{\text{area of the } \textit{Shared with length } L \text{ S-box}}{\text{area of the } \textit{Original} \text{ S-box}}$

3 shares				4 shares			5 shares	remark
1	2	3	4	1	2	3	1	
3.6–5.2	6.3–6.5	–	–	5.0–7.6	–	–	5.4–7.4	quadratics in S_8
3.3–6.2	6.2–6.6	–	–	4.3–6.4	–	–	5.1–7.4	quadratics in S_{16}
–	6.0–6.6	7.7–8.2	13.9	–	7.3–9.3	12.8	8.2–15.2	cubics in A_{16}
–	–	–	–	5.4–10.2	8.4–10.2	12.6	10.2–14.6	cubics in $S_{16} \setminus A_{16}$

Our results summarized in Table 5 show that such secure implementation can also be made efficient. Note that we consider the cost of *sharing with L registers* which is the total price for the sharing (since it includes the sharing logic plus registers). Observe that the increase of the cost for sharing with 3 shares of a quadratic S-box is similar for $n = 3$ and $n = 4$.

As expected, the longer length a sharing has, the more costly it becomes (for 3 and 4 shares). It can be seen that sharings with 4 and 5 shares cost up to 50% more than sharings with 3 shares. However, there are several cases when using 4 or 5 shares reduces the cost by up to 30%, respectively 10%, compared to 3 shares with longer sharing length. For certain S-boxes using 5 shares may be even beneficial compared to 4 shares (up to 4%) but in general 5 shares are up to 30% more expensive than 4 shares.

An obvious conclusion is that the cost of the TI method heavily depends on the class the given S-box belongs to as well as the chosen number of shares and the associated sharing length. Therefore, in order to minimize the implementation cost the number of shares have to be carefully chosen. For all tested S-boxes we were able to find a sharing with cost ranging from 3.3 till 12.8 times the area of the original S-box. However, note that the area numbers are based on a few implementations from each class. The ratios may change significantly if the smallest/biggest S-boxes are found for every class.

Acknowledgements: We would like to thank Christophe De Cannière for the fruitful discussions and for sharing with us his toolkit for affine equivalent classes.

References

1. Akkar, M.L., Giraud, C.: "An Implementation of DES and AES, Secure against Some Attacks," CHES 2001, LNCS 2162, pp. 309–318.
2. B. Bilgin, S.Nikova, V.Nikov, V.Rijmen, G.Stütz: "Threshold Implementations of all 3×3 and 4×4 S-boxes", Cryptology ePrint Archive, Report 2012/300, `http://eprint.iacr.org/`.
3. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: "Keccak specifications," NIST SHA3 contest 2008.
4. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: "Building power analysis resistant implementations of Keccak", Round 3 finalist of the Cryptographic Hash Algorithm Competition of NIST, 2010.
5. Blömer, J., Guajardo, J., Krummel, V.: "Provably Secure Masking of AES," SAC 2004, LNCS 3357, pp. 69–83.
6. Boura, C., Canteaut, A.: "On the influence of the algebraic degree of F^{-1} on the algebraic degree of $G \circ F$", e-print archive 2011/503.
7. Carlet, C.,: "Vectorial Boolean Functions for Cryptography," to appear.
8. De Cannière, C.: "Analysis and Design of Symmetric Encrytption Algorithms," Ph.D. thesis, 2007.
9. De Cannière, C., Nikov, V., Nikova, S., Rijmen, V.: "S-box decompositions for SCA-resisting implementations," poster session of CHES 2011.
10. Daemen, J., Vandewalle, J.: "A New Approach Towards Block Cipher Design," FSE 1993, LNCS, pp. 18–33.
11. Daemen, J., Peeters, M., Van Assche, G.: "Bitslice Ciphers and Power Analysis Attacks,", FSE 2000, LNCS, pp. 10–12.

12. Golic, J.D., Tymen, C.: "Multiplicative Masking and Power Analysis of AES," CHES 2002, LNCS 2523, pp. 198–212.
13. Ishai, Y., Sahai, A., Wagner, D.: "Private Circuits: Securing Hardware against Probing Attacks," CRYPTO 2003, LNCS 2729, pp. 463–481.
14. Knudsen, L.R., Leander, G., Poschmann, A., Robshaw, M.: "PRINTcipher: A Block Cipher for IC-Printing", CHES 2010, LNCS 6225, pp. 16–32.
15. Leander, G., Poschmann, A.:"On the classification of 4 bit s-boxes", WAIFI 2007, LNCS 4547, pp. 159-–176.
16. Lidl, R., Niederreiter, H.: "Finite Fields", Encyclopedia of Mathematicsand its Applications, vol. 20, Addison-Wesley, 1983.
17. Mangard, S., Pramstaller, N., Oswald, E.: "Successfully Attacking Masked AES Hardware Implementations," CHES 2005, LNCS 3659, pp. 157–171
18. Moradi, A., Mischke, O., Paar, C., Li, Y., Ohta, K., Sakiyama, K.: "On the Power of Fault Sensitivity Analysis and Collision Side-Channel Attacks in a Combined Setting," CHES 2011, LNCS 6917, pp. 292–311.
19. Moradi, A., Poschmann, A., Ling, S., Paar, C., Wang, H.:"Pushing the Limits: A Very Compact and a Threshold Implementation of AES", Eurocrypt 2011, LNCS 6632, pp. 69–88.
20. Nikova, S., Rechberger, C., Rijmen, V.: "Threshold Implementations Against Side-Channel Attacks and Glitches," ICICS 2006, LNCS 4307, pp. 529–545.
21. Nikova, S., Rijmen, V., Schläffer, M.: "Using Normal Bases for Compact Hardware Implementations of the AES S-Box," SCN 2008, LNCS 5229, pp. 236–245.
22. Nikova, S., Rijmen, V., Schläffer, M.: "Secure Hardware Implementation of Non-linear Functions in the Presence of Glitches," ICISC 2008, LNCS 5461, pp. 218–234.
23. Nikova, S., Rijmen, V., Schläffer, M.: "Secure Hardware Implementation of Non-linear Functions in the Presence of Glitches," J. Cryptology 24 (2), pp. 292–321, 2011.
24. Oswald, E., Mangard, S., Pramstaller, N., Rijmen, V.: "A Side-Channel Analysis Resistant Description of the AES S-Box," FSE 2005, LNCS 3557, pp. 413–423.
25. Popp, T., Mangard, S.: "Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints," CHES 2005, LNCS 3659, pp. 172–186
26. Poschmann, A., Moradi, A., Khoo, K., Lim, C.W., Wang, H., Ling, S.: "Side-Channel Resistant Crypto for less than 2,300 GE," J. Cryptology 24 (2), pp. 322–345, 2011.
27. Rivain, M., Prouff, E.: "Provably Secure Higher-Order Masking of AES," CHES 2010, LNCS 6225, pp. 413-427, 2010.
28. Prouff, E., Roche, T.: "Higher-Order Glitches Free Implementation of the AES Using Secure Multi-party Computation Protocols," CHES 2011, LNCS 6917, pp. 63–78, 2011.
29. Rotman, J.:" An introduction to the theory of groups," Graduate texts in mathematics, Springer-Verlag, 1995.
30. Saarinen, M.-J. O.,: "Cryptographic analysis of all 4×4-bit s-boxes", SAC 2011, LNCS 7118, pp. 118–133, 2012.
31. Tiri, K., Verbauwhede, I.: "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation," DATE 2004, IEEE Computer Society, pp. 246–251.
32. Trichina, E., Korkishko, T., Lee, K.H.: "Small Size, Low Power, Side Channel-Immune AES Coprocessor: Design and Synthesis Results," 4th AES Conference, LNCS 3373, pp. 113–127, 2004.
33. Wernsdorf, R.: "The Round Functions of RIJNDAEL Generate the Alternating Group," FSE 2002, LNCS 2365, pp. 143–148.