An Optimal Universal Construction for the Threshold Implementation of Bijective S-boxes

Enrico Piccione[1], Samuele Andreoli[1], Lilya Budaghyan[1], Claude Carlet[1,2], Siemen Dhooghe[3], Svetla Nikova[1,3], George Petrides[4], Vincent Rijmen[1,3]

[1] University of Bergen, Bergen, Norway, {name.surname}@uib.no
[2] University of Paris 8, Saint-Denis, France, claude.carlet@gmail.com
[3] KU Leuven, Leuven, Belgium, {name.surname}@esat.kuleuven.be
[4] Univeristy of Cyprus, Nicosia, Cyprus g.petrides@yahoo.com

Abstract. Threshold implementation is a method based on secret sharing to secure cryptographic ciphers (and in particular S-boxes) against differential power analysis side-channel attacks which was proposed by Nikova, Rechberger, and Rijmen in 2006. Until now, threshold implementations were only constructed for specific types of functions and some small S-boxes, but no generic construction was ever presented. In this paper, we present the first universal threshold implementation with $t+2$ shares that is applicable to any bijective S-box, where t is its algebraic degree (or is larger than the algebraic degree). While being universal, our construction is also optimal with respect to the number of shares, since the theoretically smallest possible number, $t+1$, is not attainable for some bijective S-boxes. Our results enable low latency secure hardware implementations without the need for additional randomness. In particular, we apply this result to find two uniform sharings of the AES S-box. The first sharing is obtained by using the threshold implementation of the inversion in $\mathbb{F}_{2^8}$ and the second by using two threshold implementations of two cubic power permutations that decompose the inversion. Area and performance figures for hardware implementations are provided.

Keywords: AES, DPA, Glitches, Masking, Permutation Polynomials, Sharing, Threshold Implementations, Vectorial Boolean Functions

1 Introduction

In 1999, Kocher et al. [1] introduced Differential Power Analysis (DPA), an attack which uses information emanating from a physical device such as its power consumption to retrieve the secret keys of the embedded cryptographic algorithms, originally demonstrated on DES. The weakness is not in the design of DES, but rather in its implementation. As a result, side-channel countermeasures were developed, aiming for the secure implementation of symmetric primitives, including standards such as the Advanced Encryption Standard (AES) [2]. Since then, a lot of research has been done to improve the side-channel attacks and to develop protection mechanisms against them. The most prominent countermeasure in both academia and industry is called *sharing* (or also called *masking*) and was introduced by Goubin and Patarin [3] and Chari et al. [4] independently in the same year. In s-share Boolean sharing, a variable $x \in \mathbb{F}_2$ is shared as s values $(x_1, ..., x_s) \in \mathbb{F}_2^s$, such that $\sum_{i=1}^{s} x_i = x$ (and similarly one can do the same for $x \in \mathbb{F}_2^n$). For each function F in the algorithm (in particular, each S-box) we shall map a sharing of the input x to a sharing of the output $F(x)$. While masking helps to protect algorithms against formally defined adversary models, the method's protection is not evident when translated to a physical device. For example, the glitching of values on hardware can undermine a naive sharing's security. This led to several attacks on shared AES implementations in hardware by Mangard et al. [5]. The next year, in 2006, Nikova, Rechberger, and Rijmen [6] published a methodology to build a countermeasure, called *Threshold Implementations* (TI), which addresses this physical behaviour on hardware. The TI method aims to construct a function $\mathcal{F}$ that acts as the implementation of the target function F (commonly a bijective S-box). The function $\mathcal{F}$ must satisfy three essential properties. First, it requires that the function $\mathcal{F}$ is *correct* with respect to the target function F. This means that every sharing of a secret x is mapped by $\mathcal{F}$ to a sharing of $F(x)$. Second, it requires that it is *non-complete*. In its simplest form, non-completeness is satisfied if each output share generated by $\mathcal{F}$ can be computed with at most $s-1$ shares where s is the number of shares in input. Third, it has to be *uniform* which, if the target function F is bijective, means that $\mathcal{F}$ is bijective. Given the above properties, the function $\mathcal{F}$ ensures protection against *first-order attacks* where the mean of the power traces is used as a distinguisher. A lower bound on the number of shares $s \geq t+1$ to achieve a threshold implementation is given in [6,7], where t is the algebraic degree of the S-box.

The TI approach has been applied to many symmetric primitives. For example, on the AES by De Cnudde et al. [8] and by Moradi et al. [9], on PRESENT by Poschmann et al. [10], on Keccak by Groß et al. [11], etc. Each of these papers have implemented the countermeasure and verified its order of security in practice. The trust in the TI method as a countermeasure against side-channel analysis is based on the significant quantity of works with observed practical results.

Threshold implementations have proven to be an effective countermeasure, but constructing one is non-trivial. There are three known methods. The first is using *direct sharing* [12] to guarantee the non-completeness and correctness property, then apply *correction terms* [6] for uniformity. However, this method does not guarantee success for any arbitrary bijective S-box. The second method uses direct sharing, but adds additional randomness as a re-sharing step to guarantee the uniformity. The third method is introduced by Daemen [13] and is called *the changing of the guards*. It embeds the non-complete and correct function in a Feistel construction to guarantee the uniformity.

There is currently no known universal construction of a threshold implementation for an arbitrary bijective S-box of any size. Instead, research on threshold implementations focuses on finding solutions for small sizes [12,14]. In particular, there are several known TI sharings of the AES S-box, but the most relevant for our considerations is the one started by Wegener and Moradi [15] who gave a decomposition of the AES S-box into two cubic power functions, namely $x \mapsto x^{26}$ and $x \mapsto x^{49}$. The following year, the list of all possible decompositions on quadratic and cubic power functions for the inversion over any binary field $\mathbb{F}_{2^n}$ up to $n = 16$ was given by Nikova et al. [16]. The list has recently been extended up to $n = 32$ by Petrides [17]. In particular, for the inversion in $\mathbb{F}_{2^8}$, the decomposition in power functions up to algebraic degree three was presented.

In this paper, we present the first universal construction of a threshold implementation for bijective S-boxes. It provides a TI with $t+2$ shares for every bijective S-box of any algebraic degree $t \geq 2$. Since the theoretically smallest number of shares $t+1$ is not possible for all bijective S-boxes, as proven in [18], then there does not exist a universal construction with $t+1$ shares. Hence, the construction presented in this paper is an optimal universal TI construction. It enables low latency hardware implementations without the need for additional randomness to guarantee uniformity. We demonstrate this by providing the first threshold implementations (with and without a decomposition) of the AES S-box that are uniform by construction, comparing them to the state of the art.

The results of this paper also contribute to the theory of permutation polynomials, since with this TI construction we can take any permutation polynomial in $\mathbb{F}_{2^n}[x]$ with algebraic degree t and construct a new permutation polynomial in $\mathbb{F}_{2^{n(t+2)}}[x]$ of algebraic degree t. In particular, taking any infinite family of permutation polynomials in $\mathbb{F}_{2^n}[x]$

we can construct a new infinite family of permutation polynomials in $\mathbb{F}_{2^{n(t+2)}}[x]$. We demonstrate it on the cube function x^3 which is a permutation polynomial in $\mathbb{F}_{2^n}[x]$ for any n odd, constructing a new permutation polynomial in $\mathbb{F}_{2^{4n}}[x]$.

Paper Outline. Section 2 introduces notations and main concepts used in the paper, such as Boolean functions and threshold implementations. Section 3 covers the main result of the paper, which is an optimal universal construction for the threshold implementation of bijective S-boxes. In the same section, we discuss the importance of this construction in the theory of permutation polynomials using the cube function as an example. Section 4 applies the construction to achieve two uniform sharings of the AES S-box (one direct and the other with decomposition), where performance results in hardware are given. In Section 5 we discuss the known cases of permutations of algebraic degree t admitting TI with $t+1$ shares and we present conjectures on the non-existence of threshold implementations with $t+1$ shares for power and APN permutations. Finally, Section 6 concludes the paper and provides additional observations and perspectives.

2 Preliminaries

In this section, we provide the necessary background on Boolean functions, secret sharing, and threshold implementation.

2.1 Boolean Functions

Let n and s be positive integers. We denote by $\mathbb{F}_2$, respectively, by $\mathbb{F}_{2^n}$ the finite field with 2, respectively, 2^n elements and by $\mathbb{F}_2^s$, respectively, by $\mathbb{F}_{2^n}^s$ the s-dimensional vector space over $\mathbb{F}_2$, respectively, over $\mathbb{F}_{2^n}$. For any field $\mathbb{F}$, we denote by $\mathbb{F}[x]$ the set of all (univariate) polynomials with coefficients in $\mathbb{F}$ and by $\mathbb{F}[x_1,\ldots,x_k]$ the set of all multivariate polynomials with coefficients in $\mathbb{F}$ with k variables.

A *Boolean function* f over n bits is an $\mathbb{F}_2$-valued function on $\mathbb{F}_2^n$. The unique representation of f as a multivariate polynomial in $\mathbb{F}_2[x_1,\ldots,x_n]$ of the form

$$f(x_1,\ldots,x_n)=\sum_{u\in\mathbb{F}_2^n} c(u)\left(\prod_{i=1}^n x_i^{u_i}\right),\quad c(u)\in\mathbb{F}_2$$

is called the *algebraic normal form* of f. The global degree of the algebraic normal form of f is called the *algebraic degree* of the function f [19].

Any function F from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$ can be considered as a *vectorial Boolean function*, i.e. F can be presented in the form

$$F(x_1,\ldots,x_n)=(f_1(x_1,\ldots,x_n),\ldots,f_m(x_1,\ldots,x_n)),$$

where the Boolean functions $f_1,\ldots,f_m$ are called the *coordinate functions*. A *component function* of F is any nonzero linear combination of its coordinate functions. The algebraic degree of F is equal to the maximum algebraic degree of the coordinate functions of F (see [19]). A vectorial Boolean function F is *affine*, *quadratic*, or *cubic* if its algebraic degree is respectively less than or equal to 1, 2, or 3. Moreover, F is is *linear* if it is affine and $F(0)=0$.

If we identify $\mathbb{F}_2^n$ with the finite field $\mathbb{F}_{2^n}$, then any function $F\colon\mathbb{F}_{2^n}\to\mathbb{F}_{2^n}$ is also uniquely represented as a univariate polynomial in $\mathbb{F}_{2^n}[x]$ of the form

$$F(x)=\sum_{i=0}^{2^n-1} c_i x^i,\quad c_i\in\mathbb{F}_{2^n}.$$

For any integer k, $0 \le k \le 2^n - 1$, the number $w_2(k)$ of non-zero coefficients $k_s \in \{0,1\}$, in the binary expansion $\sum_{s=0}^{n-1} 2^s k_s$ of k is called the 2-weight of k. The algebraic degree of a function $F\colon \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ is equal to the maximum 2-weight of the exponents i of the polynomial $F(x)$ such that $c_i \neq 0$ (see [20]), that is

$$\max_{\substack{0 \le i \le 2^n - 1 \\ c_i \neq 0}} w_2(i).$$

In particular, F is linear if and only if $F(x)$ is a *linearized polynomial* in $\mathbb{F}_{2^n}[x]$ that has the form:

$$\sum_{i=0}^{n-1} c_i x^{2^i}, \quad c_i \in \mathbb{F}_{2^n}.$$

Let F be a function from $\mathbb{F}_2^n$ to itself. Then, F is a *permutation* if it is bijective. A function $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ is called *balanced* if $n \ge m$ and F takes every value of $\mathbb{F}_2^n$ the same number 2^{n-m} of times. A balanced function from $\mathbb{F}_2^n$ to itself is a *permutation*.

Let $A_1\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ and $A_2\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ be affine permutations, then the functions $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ and $A_1 \circ F \circ A_2$ are called *affine equivalent*. We have that two affine equivalent functions have the same algebraic degree.

Let $N = ns$ and $M = ms'$, then we can represent a function $\mathcal{F}$ from $\mathbb{F}_2^N$ to $\mathbb{F}_2^M$ as a function from $(\mathbb{F}_2^n)^s$ to $(\mathbb{F}_2^m)^{s'}$ in the following way

$$\mathcal{F}(x_1, \ldots, x_s) = (\mathcal{F}_1(x_1, \ldots, x_s), \ldots, \mathcal{F}_{s'}(x_1, \ldots, x_s)),$$

where the functions $\mathcal{F}_1, \ldots, \mathcal{F}_{s'}\colon (\mathbb{F}_2^n)^s \to \mathbb{F}_2^m$ are called the *coordinate functions* of the function $\mathcal{F}$.

A function $\mathcal{F}\colon \mathbb{F}_2^N \to \mathbb{F}_2^n$ where $N = ns$ can be represented as a function from $\mathbb{F}_{2^n}^s$ to $\mathbb{F}_{2^n}$ as a multivariate polynomial in $\mathbb{F}_{2^n}[x_1, \ldots, x_s]$ of the following form:

$$\mathcal{F}(x_1, \ldots, x_s) = \sum_{u \in \{0, \ldots, 2^n-1\}^s} c(u) \left(\prod_{i=1}^{s} x_i^{u_i} \right), \quad c(u) \in \mathbb{F}_{2^n}.$$

2.2 Threshold Implementations

Given $x \in \mathbb{F}_2^n$ and a positive integer s, we define the set of *Boolean s-sharings* of x as

$$\mathrm{Sh}_s(x) \coloneqq \left\{ (x_1, \ldots, x_s) \in (\mathbb{F}_2^n)^s \mid \sum_{i=1}^{s} x_i = x \right\}.$$

It follows directly from the definition that $\mathrm{Sh}_s(x)$ is an affine subspace of $(\mathbb{F}_2^n)^s$ of dimension $n(s-1)$ over $\mathbb{F}_2$ (it can be also viewed as an affine hyperplane in $\mathbb{F}_{2^n}^s$ over $\mathbb{F}_{2^n}$), and consequently $|\mathrm{Sh}_s(x)| = 2^{n(s-1)}$.

Let s and s' be positive integers, and $\mathcal{F}\colon (\mathbb{F}_2^n)^s \to (\mathbb{F}_2^m)^{s'}$ and $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ be vectorial Boolean functions. We say that $\mathcal{F}$ is a *threshold implementation* (TI) of F if $\mathcal{F}$ is correct with respect to F, non-complete, and uniform, as defined below.

Correctness We say that $\mathcal{F}$ is *correct*, or equivalently that it has the *correctness* property, with respect to F if for all $x \in \mathbb{F}_2^n$, it maps any Boolean s-sharing of x to a Boolean s'-sharing of $F(x) \in \mathbb{F}_2^m$. More precisely, for all $x \in \mathbb{F}_2^n$ and $\underline{x} \in \mathrm{Sh}_s(x)$ we have

$$\mathcal{F}(\underline{x}) \in \mathrm{Sh}_{s'}(F(x)). \tag{1}$$

Note that the notion of *correctness* given in [21], namely that for all $x \in \mathbb{F}_2^n$ we have that $\underline{x} \in \mathrm{Sh}_s(x)$ and $\sum_{j=1}^{s'} \mathcal{F}_j(\underline{x}) = F(\sum_{i=1}^{s} x_i)$, immediately follows from (1) and the definition of Boolean s-sharing.

Non-completeness For $i \in \{1, \ldots, s\}$ and $j \in \{1, \ldots, s'\}$, we say that $\mathcal{F}_j$, the j-th coordinate function of $\mathcal{F}$, is independent of its i-th input coordinate if the latter does not affect the output of the former, or in other words, for all $(x_1, \ldots, x_s) \in (\mathbb{F}_2^n)^s$ and $a \in \mathbb{F}_2^n$,

$$\mathcal{F}_j(x_1, \ldots, x_s) = \mathcal{F}_j(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_s) \ .$$

We say that $\mathcal{F}$ is *non-complete*, or equivalently that it has the *non-completeness* property, if for all $j \in \{1, \ldots, s'\}$ there exists at least one $i \in \{1, \ldots, s\}$ such that $\mathcal{F}_j$ is independent of its i-th input coordinate.

Together, correctness and non-completeness impose a restriction on s based on the algebraic degree t of F, namely that $s \geq t + 1$ [6, 7, 22].

Uniformity Let $\mathcal{F}$ be correct with respect to F. We say that $\mathcal{F}$ is *uniform*, or equivalently that it has the *uniformity* property, if for all $x \in \mathbb{F}_2^n$ the restriction of $\mathcal{F}$ as a function $\mathrm{Sh}_s(x) \to \mathrm{Sh}_{s'}(F(x))$ is balanced, that is for all $\underline{y} \in \mathrm{Sh}_{s'}(F(x))$ we have

$$\left|\{\underline{x} \in \mathrm{Sh}_s(x) \mid \mathcal{F}(\underline{x}) = \underline{y}\}\right| = \frac{2^{n(s-1)}}{2^{m(s'-1)}} \ .$$

We note that according to [21], if $s = s'$ and $\mathcal{F}$ is correct with respect to a permutation F, we have that $\mathcal{F}$ being uniform is equivalent to $\mathcal{F}$ being a permutation. To the best of our knowledge, no formal proof of this statement exists in the literature, only some arguments in [23]. Here we are going to prove the following proposition, which is stronger.

Proposition 1. *Let $F \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ and $\mathcal{F} \colon (\mathbb{F}_2^n)^s \to (\mathbb{F}_2^n)^s$ be a function that is correct with respect to F. Then $\mathcal{F}$ is a permutation if and only if $\mathcal{F}$ is uniform and F is a permutation.*

Proof. First, let $\mathcal{F}$ be uniform and F be a permutation. We claim that $\mathcal{F}$ is surjective, and since the domain of $\mathcal{F}$ is equal to its codomain, this implies that $\mathcal{F}$ is a permutation as required. Let $y \in \mathbb{F}_2^n$ and $\underline{y} \in \mathrm{Sh}_s(y)$. Since F is surjective, there exists an $\underline{x} \in (\mathbb{F}_2^n)^s$ such that $\mathcal{F}(\underline{x}) = \underline{y}$. Because of correctness, there exists an $x \in \mathbb{F}_2^n$ such that $\underline{x} \in \mathrm{Sh}_s(x)$ and $F(x) = y$, and thus $\mathcal{F}$ is surjective as claimed.

Next, let $\mathcal{F}$ be a permutation. We make two claims. Firstly, we claim that F is surjective, and since the domain of F is equal to its codomain, this implies that F is a permutation, which is the first requirement. Let $\underline{y} \in (\mathbb{F}_2^n)^s$ and $y \in \mathbb{F}_2^n$ be such that $\underline{y} \in \mathrm{Sh}_s(y)$. Since $\mathcal{F}$ is surjective, there exists an $\underline{x} \in (\mathbb{F}_2^n)^s$ such that $\mathcal{F}(\underline{x}) = \underline{y}$. Since $\mathcal{F}$ is correct, then there exists $x \in \mathbb{F}_2^n$ such that $\underline{x} \in \mathrm{Sh}_s(x)$ and $F(x) = y$. Hence, we conclude that F is surjective, as stated in the first claim. Secondly, we claim that $\mathcal{F}$ is uniform, which is the second requirement. Let $x \in \mathbb{F}_2^n$ and $\underline{y} \in \mathrm{Sh}_s(F(x))$. Since F is a permutation, there exists a unique $\underline{x} \in (\mathbb{F}_2^n)^s$ such that $\mathcal{F}(\underline{x}) = \underline{y}$. Because of correctness, there exists an $x' \in \mathbb{F}_2^n$ such that $\underline{x} \in \mathrm{Sh}_s(x')$ and $F(x') = F(x)$. Since F is a permutation, we have that $x' = x$. This implies that $\mathcal{F}$ is uniform, as stated in the second claim. $\square$

In this paper, we focus on constructing threshold implementation of permutations in the specific case $s = s'$. We will be referring to these simply as *threshold implementation with s shares*. Moreover, we say that a function F *admits a threshold implementation with s shares* if there exists at least one threshold implementation of F with s shares.

Additional randomness The problem of finding $\mathcal{F}$ with the three properties described above can be relaxed by adding an extra input k, called *additional randomness*. The deterministic TI $\mathcal{F}$ is replaced by a family $\mathcal{F}_k$ indexed by the random value k. Uniformity can now be defined as a property that is achieved *on average* over all members of the family $\mathcal{F}_k$. This construction relaxes the problem, but for the application it comes with an extra cost: for every computation of F, a new random number k needs to be generated and provided as extra input to the implementation. The construction that we present in this paper does not require additional randomness.

3 A Universal Construction for Threshold Implementation with $t+2$ shares

In this section, we construct a threshold implementation $\mathcal{F}$ with $t+2$ shares for every permutation F over $\mathbb{F}_2^n$ of algebraic degree at most t, for $t \geq 2$. The condition on the algebraic degree is not very restrictive because it is already known how to construct threshold implementations of affine permutations. In addition, the construction does not depend on the dimension n of the vector space $\mathbb{F}_2^n$ that is permuted by F. Using as an example the cube function, we will be discussing in detail some aspects of the main result in the context of permutation polynomial theory. At the end (Subsection 3.3), we will show how the main construction can be modified in order to obtain a different threshold implementation of the function F. This procedure allows the designer, who has to implement the TI in practice, to potentially improve desired cryptographic properties.

We introduce the following notation to define the main construction in a compact way. For any $k \geq 1$ we denote by $\mathcal{P}_k = \mathcal{P}(\{1,\dots,k\})$ the set of all subsets of $\{1,\dots,k\}$ (including $\emptyset$). We will also use $\mathcal{P}_k^* = \mathcal{P}_k \setminus \{\{1,\dots,k\}\}$. Moreover, we use the convention that if $x_1,\dots,x_k \in \mathbb{F}_2^n$ then $\sum_{i\in\emptyset} x_i = 0$ (with abuse of notation when we write $i \in \emptyset$).

We state the main result of this section in the theorem below. The proof of this theorem is based on Propositions 2 and 3 proved in Subsection 3.1 and 3.2, respectively.

Theorem 1. *Let F be a permutation over $\mathbb{F}_2^n$ with algebraic degree at most $t \geq 2$. Let $\mathcal{F}\colon (\mathbb{F}_2^n)^{t+2} \to (\mathbb{F}_2^n)^{t+2}$ be defined for every $\underline{x} = (x_1,\dots,x_{t+2}) \in (\mathbb{F}_2^n)^{t+2}$ as*

$$\mathcal{F}(\underline{x}) = \begin{pmatrix} \mathcal{F}_1(\underline{x}) & = & x_1 \\ \mathcal{F}_2(\underline{x}) & = & \sum_{i=3}^{t+2} x_i + F\left(\sum_{i=2}^{t+2} x_i\right) \\ \mathcal{F}_j(\underline{x}) & = & x_j + \sum_{I\in\mathcal{P}_{j-2}} F\left(\sum_{i\in I} x_i + \sum_{i=j}^{t+2} x_i\right) \\ & & j = 3,\dots,t+1 \\ \mathcal{F}_{t+2}(\underline{x}) & = & x_{t+2} + x_1 + \sum_{I\in\mathcal{P}_t} F\left(\sum_{i\in I} x_i\right) \end{pmatrix}^T. \tag{2}$$

Then $\mathcal{F}$ is a threshold implementation of F with $t+2$ shares.

Proof. We observe that the function $\mathcal{F}$ defined in (2) is non-complete by construction. Indeed, $\mathcal{F}_1$ is independent of its i-th input coordinate for $i = 2,\dots,t+2$, and $\mathcal{F}_j$ is independent of its $(j-1)$-th input coordinate for $j = 2,\dots,t+2$. Furthermore, by Proposition 2 function $\mathcal{F}$ is correct with respect to F and by Proposition 3 it is uniform. With this, we conclude that $\mathcal{F}$ is a threshold implementation of F. □

Remark 1. Regarding Theorem 1, the condition that F has algebraic degree at most t implies that if the algebraic degree is exactly τ, then F admits a threshold implementation with s shares where $s = t+2 \geq \tau+2$. There are instances where the flexibility with the number of shares might be useful. For instance, when composing functions with different algebraic degrees.

Example 1. In this example, we identify $\mathbb{F}_2^n$ with the field $\mathbb{F}_{2^n}$. Consider the vectorial Boolean function $F\colon \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ defined using the univariate representation as $F(x) = x^3$. The function F is called the Cube function. It is known that F is a permutation if and only if n is odd, in which case it also possesses the best resistance against linear and differential cryptanalysis. We choose this permutation for its simplicity to demonstrate how the TI construction in Theorem 1 works. The algebraic degree of F is $t = 2$ because $3 = 2+1$, so we construct $\mathcal{F}\colon (\mathbb{F}_{2^n})^4 \to (\mathbb{F}_{2^n})^4$ as in (2):

$$\mathcal{F}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}^{\mathrm{T}} = \begin{pmatrix} x_1 \\ x_3 + x_4 + (x_2+x_3+x_4)^3 \\ x_3 + (x_1+x_3+x_4)^3 + (x_3+x_4)^3 \\ x_4 + x_1 + 0^3 + x_1^3 + x_2^3 + (x_1+x_2)^3 \end{pmatrix}^{\mathrm{T}}.$$

We can observe that $\mathcal{F}$ is non-complete. In the following subsections we will see also that correctness and uniformity are clearly illustrated by this example.

3.1 Proving the Correctness Property

By definition, to prove the correctness property we need to show that the function $\mathcal{F}$ defined in (2) satisfies

$$F\left(\sum_{i=1}^{t+2} x_i\right) = \sum_{j=1}^{t+2} \mathcal{F}_j(\underline{x})$$

for all $\underline{x} = (x_1, \ldots, x_{t+2}) \in (\mathbb{F}_2^n)^{t+2}$. We observe that when we sum all the coordinate functions of $\mathcal{F}$ the linear terms cancel out. Hence, proving correctness follows from the following proposition.

Proposition 2. *Let $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ be of algebraic degree at most $t \geq 2$. Then for every $x_1, x_2, \ldots, x_{t+2} \in \mathbb{F}_2^n$ we have that*

$$F\left(\sum_{i=1}^{t+2} x_i\right) = F\left(\sum_{i=2}^{t+2} x_i\right) + \sum_{j=3}^{t+1} \sum_{I \in \mathcal{P}_{j-2}} F\left(\sum_{i\in I} x_i + \sum_{i=j}^{t+2} x_i\right) + \sum_{I\in\mathcal{P}_t} F\left(\sum_{i\in I} x_i\right). \quad (3)$$

Moreover, if $n = m$ then the function $\mathcal{F}$ defined in (2) is correct with respect to function F.

We can observe that the correctness property does not depend on the condition that F is a permutation.

To prove Proposition 2 we need the following three lemmas. The first lemma allows us to take any expression of the form $F\left(\sum_{i=1}^{s} x_i\right)$ for any $s \geq t+1$ and write it as the sum of evaluations of F that depends at most on t shares where t is greater or equal than the algebraic degree of F.

Lemma 1 ([24, Corollary 1])**.** *Let $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ be of algebraic degree at most $t \geq 1$ and let $s > t$. Then for every $x_1, x_2, \ldots, x_s \in \mathbb{F}_2^n$ we have that*

$$F\left(\sum_{i=1}^{s} x_i\right) = \sum_{j=0}^{t} \mu_{s,t}(j) \sum_{I\in\mathcal{P}_s,\, |I|=j} F\left(\sum_{i\in I} x_i\right)$$

where $\mu_{s,t}(j) = \binom{s-j-1}{t-j} \mod 2$ for every $j = 0, \ldots, t$ (with the convention that $\binom{0}{0} = 1$).

One can notice that the expression in $t+2$ variables we want to prove in Proposition 2 can be turned into an expression in $t+1$ variables by setting $z = x_{t+1} + x_{t+2}$. With the following two lemmas, we want to find that expression starting from the one we get with $s = t+1$ from Lemma 1.

Lemma 2. *Let t be a positive integer. Set $\mathcal{J}_2 = \{2, \ldots, t\}$, $\mathcal{J}_j = \{I \cup \{j, \ldots, t\} \mid I \in \mathcal{P}_{j-2}\}$ for $2 < j < t+1$, and $\mathcal{J}_{t+1} = \mathcal{P}_{t-1}$. Then*

1. *$\mathcal{P}_{t+1}^* = \mathcal{P}_t \cup \{I \cup \{t+1\} \mid I \in \mathcal{P}_t^*\}$.*

2. $\mathcal{P}_t^* = \bigcup_{j=2}^{t+1} \mathcal{J}_j$.

3. $\mathcal{J}_{j_1} \cap \mathcal{J}_{j_2} = \emptyset$ *for all* $j_1, j_2 \in \{2, \ldots, t+1\}$ *with* $j_1 \neq j_2$.

Proof. The first statement follows directly from the definition. Let us prove the second statement. Obviously, we have that $\mathcal{P}_t^* \supseteq \bigcup_{j=2}^{t+1} \mathcal{J}_j$. Let $I \in \mathcal{P}_t^*$ and $j \in \{2, \ldots, t\}$. If $\{j, \ldots, t\} \subseteq I$, then $I \in \mathcal{J}_j$. Suppose $\{j, \ldots, t\} \not\subseteq I$ for all $j \in \{2, \ldots, t\}$, then $t \notin I$ and this implies that $I \in \mathcal{J}_{t+1}$. Since all the conditions for I to belong to a certain $\mathcal{J}_j$ are mutually exclusive, the third statement follows as a consequence. □

Lemma 3. *Let* $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ *be of algebraic degree at most* $t \geq 2$. *Then for every* $x_1, x_2, \ldots, x_{t+1} \in \mathbb{F}_2^n$ *we have that*

$$F\left(\sum_{i=1}^{t+1} x_i\right) = F\left(\sum_{i=2}^{t+1} x_i\right) + \sum_{j=3}^{t+1} \sum_{I \in \mathcal{P}_{j-2}} F\left(\sum_{i \in I} x_i + \sum_{i=j}^{t+1} x_i\right) + \sum_{I \in \mathcal{P}_t} F\left(\sum_{i \in I} x_i\right).$$

Proof. By using Lemma 1 for the case $s = t+1$, we obtain

$$F\left(\sum_{i=1}^{t+1} x_i\right) = \sum_{I \in \mathcal{P}_{t+1}^*} F\left(\sum_{i \in I} x_i\right),$$

since $\mu_{t+1,t}(j) = \binom{t-j}{t-j} \mod 2 = 1$ for all $j = 0, \ldots, t$ (recall that we use the convention $\binom{0}{0} = 1$).

By using the first statement of Lemma 2 we have that

$$\sum_{I \in \mathcal{P}_{t+1}^*} F\left(\sum_{i \in I} x_i\right) = \sum_{I \in \mathcal{P}_t} F\left(\sum_{i \in I} x_i\right) + \sum_{I \in \mathcal{P}_t^*} F\left(\sum_{i \in I} x_i + x_{t+1}\right).$$

By using the second and the third statements of the same lemma we have that

$$\begin{aligned}
&\sum_{I \in \mathcal{P}_t^*} F\left(\sum_{i \in I} x_i + x_{t+1}\right) \\
=&F\left(\sum_{i=2}^{t} x_i + x_{t+1}\right) + \sum_{j=3}^{t} \sum_{I \in \mathcal{P}_{j-2}} F\left(\sum_{i \in I} x_i + \sum_{i=j}^{t} x_i + x_{t+1}\right) + \sum_{I \in \mathcal{P}_{t-1}} F\left(\sum_{i \in I} x_i + x_{t+1}\right) \\
=&F\left(\sum_{i=2}^{t+1} x_i\right) + \sum_{j=3}^{t+1} \sum_{I \in \mathcal{P}_{j-2}} F\left(\sum_{i \in I} x_i + \sum_{i=j}^{t+1} x_i\right)
\end{aligned}$$

and so we have concluded the proof. □

Now we can prove Proposition 2.

Proof (of Proposition 2). Set $z_{t+1} = \sum_{j=t+1}^{t+2} x_j$ and $z_i = x_i$ for $i = 1, \ldots, t$. By using Lemma 3 on $z_1, \ldots, z_{t+1}$, we obtain

$$\begin{aligned}
F\left(\sum_{i=1}^{t+2} x_i\right) &= F\left(\sum_{i=1}^{t+1} z_i\right) \\
=&F\left(\sum_{i=2}^{t+1} z_i\right) + \sum_{j=3}^{t+1} \sum_{I \in \mathcal{P}_{j-2}} F\left(\sum_{i \in I} z_i + \sum_{i=j}^{t+1} z_i\right) + \sum_{I \in \mathcal{P}_t} F\left(\sum_{i \in I} z_i\right) \\
=&F\left(\sum_{i=2}^{t+2} x_i\right) + \sum_{j=3}^{t+1} \sum_{I \in \mathcal{P}_{j-2}} F\left(\sum_{i \in I} x_i + \sum_{i=j}^{t+2} x_i\right) + \sum_{I \in \mathcal{P}_t} F\left(\sum_{i \in I} x_i\right).
\end{aligned}$$

□

Example 2. We continue the discussion about the cube function stated in Example 1. Let $z_1 = x_1$, $z_2 = x_2$, and $z_3 = x_3 + x_4$. First, we show how to get the expression of Lemma 1 for $(z_1 + z_2 + z_3)^3$:

$$
\begin{aligned}
&(z_1 + z_2 + z_3)^3 = z_1^3 + z_2^3 + z_3^3 + \sum_{1 \le i,j \le 3} z_i^2 z_j \\
=&z_1^3 + z_2^3 + z_3^3 + (z_1 + z_2)^3 + (z_1 + z_3)^3 + (z_2 + z_3)^3 = \sum_{I \in \mathcal{P}_3^*} \left(\sum_{i \in I} x_i \right)^3 .
\end{aligned}
$$

Then we show that we can derive the expression of Lemma 3:

$$
\begin{aligned}
&(z_1 + z_2 + z_3)^3 = z_1^3 + z_2^3 + z_3^3 + (z_1 + z_2)^3 + (z_1 + z_3)^3 + (z_2 + z_3)^3 \\
=&(z_2 + z_3)^3 + (z_1 + z_3)^3 + z_3^3 + z_1^3 + z_2^3 + (z_1 + z_2)^3 \\
=&\left(\sum_{i=2}^{3} z_i \right)^3 + \sum_{I \in \mathcal{P}_1} \left(\sum_{i \in I} z_i + z_3 \right)^3 + \sum_{I \in \mathcal{P}_2} \left(\sum_{i \in I} z_i \right)^3 .
\end{aligned}
$$

Finally, we show the expression in terms of x_1, x_2, x_3, x_4 :

$$
\begin{aligned}
&(x_1 + x_2 + x_3 + x_4)^3 = (z_1 + z_2 + z_3)^3 \\
=&(z_2 + z_3)^3 + (z_1 + z_3)^3 + z_3^3 + z_1^3 + z_2^3 + (z_1 + z_2)^3 \\
=&(x_2 + x_3 + x_4)^3 + (x_1 + x_3 + x_4)^3 + (x_3 + x_4)^3 + x_1^3 + x_2^3 + (x_1 + x_2)^3.
\end{aligned}
$$

Using the last expression, we can show that $\mathcal{F}$ is correct with respect to F:

$$
\begin{aligned}
&\sum_{i=1}^{4} \mathcal{F}_i(x_1, x_2, x_3, x_4) = x_1 + \left(x_3 + x_4 + (x_2 + x_3 + x_4)^3\right) \\
&+ \left(x_3 + (x_1 + x_3 + x_4)^3 + (x_3 + x_4)^3\right) + \left(x_4 + x_1 + x_1^3 + x_2^3 + (x_1 + x_2)^3\right) \\
=&(x_2 + x_3 + x_4)^3 + (x_1 + x_3 + x_4)^3 + (x_3 + x_4)^3 + x_1^3 + x_2^3 + (x_1 + x_2)^3 \\
=&(x_1 + x_2 + x_3 + x_4)^3.
\end{aligned}
$$

3.2 Proving the Uniformity Property

If F is a permutation over $\mathbb{F}_2^n$, in order to prove the uniformity property of the function $\mathcal{F}$ defined in (2), it suffices to show that $\mathcal{F}$ is a permutation over $(\mathbb{F}_2^n)^{t+2}$ (see Proposition 1).

Proposition 3. *Let F be a permutation over $\mathbb{F}_2^n$ of algebraic degree at most $t \ge 2$. Then the function $\mathcal{F}$ as defined in (2) is a permutation over $(\mathbb{F}_2^n)^{t+2}$. Hence, $\mathcal{F}$ is uniform.*

Proof. Let $\mathcal{F} \colon (\mathbb{F}_2^n)^{t+2} \to (\mathbb{F}_2^n)^{t+2}$ be as defined in (2). Let $\underline{x} = (x_1, \ldots, x_{t+2}) \in (\mathbb{F}_2^n)^{t+2}$. We introduce variables $y_i = \mathcal{F}_i(\underline{x})$ over $\mathbb{F}_2^n$ for $i = 1, \ldots, t+2$ to define a system of equations:

$$
\begin{cases}
y_1 = x_1 \\
y_2 = \sum_{i=3}^{t+2} x_i + F\left(\sum_{i=2}^{t+2} x_i\right) \\
y_j = x_j + \sum_{I \in \mathcal{P}_{j-2}} F\left(\sum_{i \in I} x_i + \sum_{i=j}^{t+2} x_i\right) \quad j = 3, \ldots, t+1 \\
y_{t+2} = x_{t+2} + x_1 + \sum_{I \in \mathcal{P}_t} F\left(\sum_{i \in I} x_i\right).
\end{cases}
\tag{4}
$$

Let $\underline{y} = (y_1, \dots, y_{t+2})$. Then $\mathcal{F}$ is a permutation if and only if for every $i = 1, \dots, t+2$ there exists a function $\mathcal{G}_i \colon (\mathbb{F}_2^n)^{t+2} \to \mathbb{F}_2^n$ such that $x_i = \mathcal{G}_i(\underline{y})$. Since $\mathcal{F}$ is correct with respect to F due to Proposition 2, then we have that $F\left(\sum_{i=1}^{t+2} x_i\right) = \sum_{i=1}^{t+2} y_i$ and

$$\sum_{i=1}^{t+2} x_i = F^{-1}\left(\sum_{i=1}^{t+2} y_i\right). \tag{5}$$

using the hypothesis that F is a permutation over $\mathbb{F}_2^n$. We claim that for every $2 \le j \le t+1$ we have that $x_k = \mathcal{G}_k(\underline{y})$ for all $1 \le k \le j$ and $\sum_{i=j+1}^{t+2} x_i = \mathcal{H}_{j+1}(\underline{y})$ for some function $\mathcal{H}_{j+1}$. We are going to prove it by induction j. Consider the case $j = 2$. Then

$$x_1 = y_1 = \mathcal{G}_1(\underline{y})$$

using the first equation of System (4). By using the second equation of System (4), we have that

$$\sum_{i=3}^{t+2} x_i = y_2 + F\left(\sum_{i=2}^{t+2} x_i\right) = y_2 + F\left(\mathcal{G}_1(\underline{y}) + F^{-1}\left(\sum_{i=1}^{t+2} y_i\right)\right) = \mathcal{H}_3(\underline{y})$$

for some function $\mathcal{H}_3$. By using Equality (5), we have that

$$x_2 = x_1 + \sum_{i=3}^{t+2} x_i + F^{-1}\left(\sum_{i=1}^{t+2} y_i\right) = \mathcal{G}_1(\underline{y}) + \mathcal{H}_3(\underline{y}) + F^{-1}\left(\sum_{i=1}^{t+2} y_i\right) = \mathcal{G}_2(\underline{y}).$$

Now we continue the proof by induction for $3 \le j \le t+1$, assuming it is true for $j-1$. By using the j-th equation of System (4), we have that

$$\begin{aligned} x_j =& y_j + \sum_{I \in \mathcal{P}_{j-2}} F\left(\sum_{i \in I} x_i + \sum_{i=j}^{t+2} x_i\right) \\ =& y_j + \sum_{I \in \mathcal{P}_{j-2}} F\left(\sum_{i \in I} \mathcal{G}_i(\underline{y}) + \mathcal{H}_j(\underline{y})\right) = \mathcal{G}_j(\underline{y}). \end{aligned}$$

By using Equality (5), we have that

$$\sum_{i=j+1}^{t+2} x_i = \sum_{i=1}^{j} x_i + F^{-1}\left(\sum_{i=1}^{t+2} y_i\right) = \sum_{i=1}^{j} \mathcal{G}_i(\underline{y}) + F^{-1}\left(\sum_{i=1}^{t+2} y_i\right) = \mathcal{H}_{j+1}(\underline{y}).$$

Hence, we get $x_k = \mathcal{G}_k(\underline{y})$ for all $k \le t+1$ and $x_{t+2} = \sum_{i=t+2}^{t+2} x_i = \mathcal{H}_{t+2}(\underline{y}) = \mathcal{G}_{t+2}(\underline{y})$. □

Remark 2. We observe that we never used the last equation of System (4) in the proof of Proposition 3. The reason being that the last coordinate function can be deduced from the first $t+1$ coordinates using the correctness property because

$$\mathcal{F}_{t+2}(\underline{x}) = \sum_{i=1}^{t+1} \mathcal{F}_i(\underline{x}) + F\left(\sum_{i=1}^{t+2} x_i\right).$$

Example 3. We continue the discussion of Example 2. We are going to follow the same steps as in the proof of Proposition 3 for the special case of the Cube function. Now, we have to assume that n is odd and we will write $\frac{1}{3}$ to indicate the inverse of 3 modulo $2^n - 1$. We construct the same system as in (4). First, we observe that

$$x_1 + x_2 + x_3 + x_4 = (y_1 + y_2 + y_3 + y_4)^{\frac{1}{3}}.$$

We have that $x_1 = y_1 = \mathcal{G}_1(y_1, y_2, y_3, y_4)$ using the first equation. We continue with

$$\begin{aligned} x_3 + x_4 =& (x_2 + x_3 + x_4)^3 + y_2 = \left(x_1 + (y_1 + y_2 + y_3 + y_4)^{\frac{1}{3}}\right)^3 + y_2 \\ =& \left(y_1 + (y_1 + y_2 + y_3 + y_4)^{\frac{1}{3}}\right)^3 + y_2 = \mathcal{H}_3(y_1, y_2, y_3, y_4) \end{aligned}$$

and then we get

$$\begin{aligned} x_2 =& x_1 + x_3 + x_4 + (y_1 + y_2 + y_3 + y_4)^{\frac{1}{3}} = y_1 + \mathcal{H}_3(y_1, y_2, y_3, y_4) + (y_1 + y_2 + y_3 + y_4)^{\frac{1}{3}} \\ =& \mathcal{G}_2(y_1, y_2, y_3, y_4). \end{aligned}$$

Since $t = 2$, in this case, we just need to do one induction step of the original proof. We have

$$\begin{aligned} x_3 =& y_3 + (x_1 + x_3 + x_4)^3 + (x_3 + x_4)^3 \\ =& y_3 + (\mathcal{G}_1(y_1, y_2, y_3, y_4) + \mathcal{H}_3(y_1, y_2, y_3, y_4))^3 + (\mathcal{H}_3(y_1, y_2, y_3, y_4))^3 \\ =& \mathcal{G}_3(y_1, y_2, y_3, y_4) \end{aligned}$$

and

$$\begin{aligned} x_4 =& x_1 + x_2 + x_3 + (y_1 + y_2 + y_3 + y_4)^{\frac{1}{3}} \\ =& \mathcal{G}_1(y_1, y_2, y_3, y_4) + \mathcal{G}_2(y_1, y_2, y_3, y_4) + \mathcal{G}_3(y_1, y_2, y_3, y_4) + (y_1 + y_2 + y_3 + y_4)^{\frac{1}{3}} \\ =& \mathcal{G}_4(y_1, y_2, y_3, y_4). \end{aligned}$$

Hence, $\mathcal{F}$ is uniform and we can conclude that $\mathcal{F}$ is a threshold implementation of F with 4 shares.

3.2.1 A new secondary construction of permutation polynomials providing infinite families of them

We have proved that for any permutation F over $\mathbb{F}_{2^n}$ the function $\mathcal{F}$ constructed using F defined in (2) is again a permutation. We have used the Cube function in odd dimension as an example for that. We recall that for $F(x) = x^3$ in $\mathbb{F}_{2^n}$, we have that

$$\mathcal{F}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}^{\mathrm{T}} = \begin{pmatrix} x_1 \\ x_3 + x_4 + (x_2 + x_3 + x_4)^3 \\ x_3 + (x_1 + x_3 + x_4)^3 + (x_3 + x_4)^3 \\ x_4 + x_1 + x_1^3 + x_2^3 + (x_1 + x_2)^3 \end{pmatrix}^{\mathrm{T}}.$$

There are general arguments showing that x^3 and $\mathcal{F}$ are different functions. Although both are quadratic, F is clearly different from $\mathcal{F}$ since the first one is a permutation in odd dimensions only while the second is defined in even dimensions (precisely, in dimensions divisible by 4). Moreover, $\mathcal{F}$ is CCZ-inequivalent to x^3 when x^3 is considered over $\mathbb{F}_{2^{4n}}$ too. CCZ-equivalence is the most general known equivalence relation for vectorial Boolean functions preserving the differential and nonlinear properties (in particular APN property) and it corresponds to affine equivalence of graphs of functions (for a function F over $\mathbb{F}_{2^n}$ it is the set of all pairs of inputs and outputs of the function $\{(x, F(x)) \ : \ x \in \mathbb{F}_{2^n}\}$). It is known that x^3 is APN, that is, it is optimal with respect to differential cryptanalysis [25]. If $\mathcal{F}$ was CCZ-equivalent to x^3 (or any other APN function) then $\mathcal{F}$ would be a quadratic APN permutation over the field of even extension, which is not possible according to [26]. Hence, with similar arguments for any quadratic APN permutation F we can formulate the following proposition.

Proposition 4. *Let F be a quadratic APN permutation over $\mathbb{F}_{2^n}$ and $\mathcal{F}$ be the permutation defined in (2). Then the function $\mathcal{F}$ is CCZ-inequivalent to any APN function. In particular, if $F(x) = x^3$ over $\mathbb{F}_{2^n}$ with n odd then $\mathcal{F}$ is CCZ-inequivalent to x^3 over $\mathbb{F}_{2^{4n}}$.*

To represent $\mathcal{F}$ as a polynomial in $\mathbb{F}_{2^{4n}}[X]$ we shall take two bijective and $\mathbb{F}_{2^n}$-linear functions $\phi\colon \mathbb{F}_{2^{4n}} \to (\mathbb{F}_{2^n})^4$ and $\psi\colon (\mathbb{F}_{2^n})^4 \to \mathbb{F}_{2^{4n}}$ (both represent a change of basis). We can represent ϕ by $\phi(X) = (\phi_1(X), \phi_2(X), \phi_3(X), \phi_4(X))$ where $\phi_1, \phi_2, \phi_3, \phi_4$ are polynomials in $\mathbb{F}_{2^{4n}}[X]$ and ψ by a multivariate polynomial $\psi(x_1, x_2, x_3, x_4) = \sum_{i=1}^{4} \lambda_i x_i \in \mathbb{F}_{2^{4n}}[x_1, x_2, x_3, x_4]$. For simplicity, we write $\phi_{i,j} = \phi_i + \phi_j$ and $\phi_{i,j,k} = \phi_i + \phi_j + \phi_k$ for all $i, j, k \in \{1, 2, 3, 4\}$. So we have that $(\psi \circ \mathcal{F} \circ \phi)(X) = Q(X) + L(X)$ where

$$\begin{aligned}
L(X) =& \lambda_1\phi_1(X) + \lambda_2\phi_{3,4}(X) + \lambda_3\phi_3(X) + \lambda_4\phi_{1,4}(X),\\
Q(X) =& \lambda_2 \left(\phi_{2,3,4}(X)\right)^3 + \lambda_3 \left(\phi_{1,3,4}(X)\right)^3 + \lambda_3 \left(\phi_{3,4}(X)\right)^3\\
&+ \lambda_4 \left(\phi_1(X)\right)^3 + \lambda_4 \left(\phi_2(X)\right)^3 + \lambda_4 \left(\phi_{1,2}(X)\right)^3.
\end{aligned}$$

3.3 Obtaining many threshold implementations using correction terms

The use of correction terms has been proposed first in [6,23] as a method to make a direct sharing [12] uniform. We define mathematically what correction terms are without concern about their original scope. Correction terms are, informally speaking, coordinate terms that are added in pairs to more than one share, such that the new function obtained still satisfies the correctness and non-completeness property. Our aim is to take the threshold implementation $\mathcal{F}$ as constructed in (2) and add some correction terms so that the new function obtained is still a threshold implementation. This procedure is important to study because it gives the possibility to construct new threshold implementations that may have better cryptographic properties than the one described in (2).

In the following proposition we present functions $\mathcal{C}$, such that $\mathcal{F} + \mathcal{C}$ is still a threshold implementation with $t + 2$ shares.

Proposition 5. *Let F be a permutation over $\mathbb{F}_2^n$ of algebraic degree at most $t \geq 2$ and $\mathcal{F}$ be the function defined in (2). Let $\mathcal{C}\colon (\mathbb{F}_2^n)^{t+2} \to (\mathbb{F}_2^n)^{t+2}$ be defined as*

$$\mathcal{C}(\underline{x}) = \begin{pmatrix}
\mathcal{C}_1(\underline{x}) & = & x_1 + P_1(x_1)\\
\mathcal{C}_2(\underline{x}) & = & \sum_{i=3}^{t+2} x_i + P_2\left(\sum_{i=3}^{t+2} x_i\right) + C_2\left(\sum_{i=2}^{t+2} x_i\right)\\
\mathcal{C}_j(\underline{x}) & = & x_j + P_j(x_j) + C_j(x_1, \ldots, x_{j-2}, \sum_{i=j}^{t+2} x_i)\\
 & & j = 3, \ldots, t+1\\
\mathcal{C}_{t+2}(\underline{x}) & = & C_{t+2}(x_1, \ldots, x_t, x_{t+2})
\end{pmatrix}^T,$$

where function P_j is a permutation over $\mathbb{F}_2^n$ for all $j = 1, \ldots, t+1$; C_j is a function from $(\mathbb{F}_2^n)^{j-1}$ to $\mathbb{F}_2^n$ for $j = 2, \ldots, t+2$, such that $\sum_{i=1}^{t+2} \mathcal{C}_i(\underline{x}) = 0$. Then $\mathcal{F} + \mathcal{C}$ is a permutation over $(\mathbb{F}_2^n)^{t+2}$.

Proof. The proof is very similar to the one of Proposition 3, so we will not give too many details. First, we have that

$$\sum_{i=1}^{t+2} \mathcal{F}_i(\underline{x}) = \sum_{i=1}^{t+2} \mathcal{F}_i(\underline{x}) + \sum_{i=1}^{t+2} \mathcal{C}_i(\underline{x}).$$

We introduce variables $y_i = \mathcal{F}_i(\underline{x}) + \mathcal{C}_i(\underline{x})$ for $i = 1, \ldots, t+2$.

Since $\mathcal{F}_1(\underline{x}) + \mathcal{C}_1(\underline{x}) = P_1(x_1)$ and P_1 is a permutation, we can write x_1 and $\sum_{i=2}^{t+2} x_i$ in terms of the y's.

Since $\mathcal{F}_2(\underline{x}) + \mathcal{C}_2(\underline{x}) = P_2\left(\sum_{i=3}^{t+2} x_i\right) + C_2\left(\sum_{i=2}^{t+2} x_i\right) + F\left(\sum_{i=2}^{t+2} x_i\right)$ and P_2 is a permutation, we can write $\sum_{i=3}^{t+2} x_i$ in terms of the y's and consequently x_2 in terms of the y's.

We continue using a similar induction argument as in the proof of Proposition 3. We will prove that, for $2 \leq j \leq t+1$, we can write $x_1, \ldots, x_j$, and $\sum_{i=j+1}^{t+2} x_i$ in terms of the y's. For $j = 2$, it is true. Now, assuming it is true for $j-1$, we can prove it for $j \geq 3$. Since $\mathcal{F}_j(\underline{x}) + \mathcal{C}_j(\underline{x}) = P_j(x_j) + C_j(x_1, \ldots, x_{j-2}, \sum_{i=j}^{t+2} x_i)$ and P_j is a permutation, we can write x_j in terms of the y's and consequently $\sum_{i=j+1}^{t+2} x_i$ in terms of the y's. $\square$

Example 4. We continue the discussion of Example 3. We will show some ways to modify the original construction using Proposition 5. One possibility is

$$(\mathcal{F}+\mathcal{C})\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}^{\mathrm{T}} = \begin{pmatrix} L(x_1) \\ L(x_3+x_4) + (x_2+x_3+x_4)^3 \\ L(x_3) + (x_1+x_3+x_4)^3 + (x_3+x_4)^3 \\ L(x_4+x_1) + x_1^3 + x_2^3 + (x_1+x_2)^3 \end{pmatrix}^{\mathrm{T}}.$$

where L is a linear permutation over $\mathbb{F}_2^n$. In fact, $L(x_1)+L(x_3+x_4)+L(x_3)+L(x_4+x_1) = 0$. One may improve the cryptographic properties of the function $\mathcal{F}$ in the following way. We can take a permutation G over $\mathbb{F}_2^n$ with good cryptographic properties and use it to construct:

$$(\mathcal{F}+\mathcal{C}')\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}^{\mathrm{T}} = \begin{pmatrix} G(x_1) \\ x_3+x_4 + (x_2+x_3+x_4)^3 \\ x_3 + (x_1+x_3+x_4)^3 + (x_3+x_4)^3 \\ x_4 + G(x_1) + x_1^3 + x_2^3 + (x_1+x_2)^3 \end{pmatrix}^{\mathrm{T}}.$$

Clearly, the functions $\mathcal{F}$ and $\mathcal{F}+\mathcal{C}$ have the same cryptographic properties since the second can be obtained from the first by adding a linear function. For $n = 3$ and $G(x) = x^3$, we checked that the nonlinearity (the parameter measuring the resistance of an S-box against linear attacks) improves from 0 to 1024 when adding $\mathcal{C}'$ to $\mathcal{F}$.
Note also that for $F(x) = x^3$ over $\mathbb{F}_{2^n}$, with similar arguments as in Proposition 4, whatever quadratic $\mathcal{C}$ we use in Proposition 5, the function $\mathcal{F}+\mathcal{C}$ is CCZ-inequivalent to x^3 over $\mathbb{F}_{2^{4n}}$.

4 Two Uniform Implementations of the AES S-box

We use the construction (2) from Section 3 to find the first threshold implementations of the AES S-box without the use of the changing of the guards construction by Daemen [13]. We implement the sharings and provide an area cost in hardware.

The first implementation is a direct application of the construction (2) from Section 3 on the AES S-box. Since this S-box has an algebraic degree of seven, we use nine shares. The result is a sharing with a large area cost, but which requires only one cycle to compute.

The second implementation uses the decomposition given in the work by Wegener and Moradi [15]. There, the inversion over $\mathbb{F}_{2^8}$ is represented as a composition of two cubic functions, namely x^{26} and x^{49}. Each of the two cubic functions is then shared using the construction (2) of Section 3. The result is a sharing with a much smaller area overhead compared to the first implementation, but it requires two cycles to compute.

We have estimated the hardware cost of these two uniform masked S-boxes. The area is measured in gate equivalences (GE), i.e., the S-box area normalised to the area of a 2-input `NAND` gate in a given standard cell library. In this work, we use the `NANGATE` 45nm Open Cell Library [27] where the synthesis results are obtained with the Synopsis Design Compiler v2021.06. The results of the synthesis and its comparison with sharings of the AES S-box in the literature, using the same standard cell library, are shown in Table 4.1. The latency of the masked S-boxes is given in the number of cycles, denoted *cc*, and we

Table 4.1: Hardware cost of the masked AES S-box in the `NANGATE` 45nm library.

Design	Shares	Area [kGE]	Latency [cc]	Randomness [$bits$]
This work [without decomposition]	9	166.37	1	0
This work [with decomposition]	5	22.05	2	0
Wegener-Moradi [15][1]	4	4.20	16	0
Sugawara [28]	3	3.50	4	0
Gross et al. [29]	2	60.76	1	2 048
Gross et al. [29]	2	6.74	2	416

1. Wegener and Moradi wrote that without serialisation their design costs will be "more than 20 kGE" which is comparable to our design's cost.

provide the total number of random bits which are needed in the computation of the masked AES S-box.

The first implementation on the AES S-box provides a sharing costing 166.37 kGE which is the first sharing of the AES S-box in one cycle requiring no additional randomness. The second implementation of the decomposed shared AES S-box is an improvement over the S-box design by Wegener and Moradi who used the changing of the guards method by Daemen to ensure the uniformity of their four-share non-complete sharing of the cubic functions x^{26} and x^{49}. Instead, this work's sharing method provides both non-completeness and uniformity. Note that the difference in implementation results of this work and Wegener and Moradi's is in the architecture. Whereas their work made a highly serialised implementation, we went with a rolled-out design of each cubic function. The result is a low-latency sharing at the cost of an increased area. The sharing requires only two cycles and is randomness-free.

Considering the other comparable works. The work by Sugawara [28] provides a sharing of the AES S-box using the changing of the guards approach [13] for each shared multiplier in the tower-field decomposed S-box. Since the multiplication is a quadratic function, it requires three shares to be non-complete. The work by Gross et al. [29] uses a low-latency masking technique by increasing the number of output shares in function of the algebraic degree of the masked function. As a result, they are able to provide trade-offs between area and latency allowing them to achieve very low-latency maskings of the AES S-box but at a higher area cost. Since their masking technique does not make the sharings uniform, they require several re-masking steps with additional randomness causing a high randomness cost for their designs.

We note that by using the correction terms from Subsection 3.3, one could get improved area costs. However, we did not pursue this direction and leave this possible optimisation for future investigation.

5 On the existence of threshold implementations with t+1 shares

In this section, we discuss permutations of algebraic degree $t \geq 2$ which are known to admit a threshold implementation with $t+1$ shares. They all can be placed into the three categories discussed below. These are computational results for small dimensions, Feistel functions, and permutations in $n+1$ and $n+2$ bits constructed in a specific way from permutations in n variables admitting a threshold implementation with $t+1$ shares. Further we also make some observations about the cryptographic properties of those functions and we try to identify cases where we strongly believe that a threshold implementations with $t+1$ shares does not exists. It seems functions admitting $t+1$ shares have bad cryptographic properties (such as resistance to linear and differential attacks) while strong cryptographic functions are among those for which $t+2$ shares are optimal.

5.1 Discussing known cases

It has been proven by Bilgin et al. [12] that threshold implementation is preserved by affine equivalence. Then, using the classification of the affine equivalent classes for 3 and 4 bit permutations, Bilgin et al. provided threshold implementations for all of these classes. Later, Bozilov et al. [14] and De Meyer and Bilgin [30] provided threshold implementations of 5 and 6 bit quadratic permutations. In Table 5.1 we summarise the known results on threshold implementations for 3 and 4 bit permutations and 5 bit quadratic permutations (excluding linear permutations which are known to admit a TI with $t+1=2$ shares) [12,14]. For each size, we report the number of S-boxes (up to affine equivalence) with respect to the smallest number of shares for known threshold implementations. For 3 bits we have three permutations in total, all of which are quadratic: two of them admit a threshold implementation with $t+1=3$ shares while for the third one, corresponding to the inversion (which in this specific case is equivalent to the Cube function x^3), the minimum number of shares is provably 4. For 4 bits the number of permutations with known TIs with $t+1$ shares is 9 out of 301. For 5 bits, the number of quadratic permutations with known TIs with $t+1$ shares is 30 out of 45. This might be an evidence that the proportion of permutations admitting a threshold implementation with $t+1$ shares significantly decreases for larger numbers of bits.

Table 5.1: The smallest number of shares for known threshold implementations of bijective S-boxes of 3 and 4 bits and bijective quadratic S-boxes of 5 bits

size	degree	3 shares	4 shares	5 shares
3	2	2	1	
4	2	5	1	
	3	-	4	291
5	2	30	45	

A threshold implementation with $t+1$ shares for each Feistel function was constructed in [31]. Consider a Feistel function (or a Feistel scheme) $F\colon \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n \times \mathbb{F}_2^n$ defined as $F(x,y) = (x, y+G(x))$ where $G\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$. It provides a well known method in cryptography to construct larger S-boxes using smaller ones. Suppose without loss of generality that the algebraic degree t of F satisfies $t \geq 2$, and note that the algebraic degree of G must also be t. To construct a threshold implementation with $t+1$ shares use a function $\mathcal{G}\colon (\mathbb{F}_2^n)^{t+1} \to (\mathbb{F}_2^n)^{t+1}$ that is non-complete and correct with respect to G. For example, $\mathcal{G}$ can be chosen to be the direct sharing of G. Then the function $\mathcal{F}\colon (\mathbb{F}_2^n)^{t+1} \times (\mathbb{F}_2^n)^{t+1} \to (\mathbb{F}_2^n)^{t+1} \times (\mathbb{F}_2^n)^{t+1}$ defined as $\mathcal{F}(\underline{x}, \underline{y}) = (\underline{x}, \underline{y} + \mathcal{G}(\underline{x}))$ is a threshold implementation of F with $t+1$ shares [31].

Starting from n-bit bijective S-boxes, Varici et al. [32] construct new $(n+1)$-bit and $(n+2)$-bit bijective S-boxes. The authors show that, if a threshold implementation for the n-bit bijective S-boxes exist, then the constructed $(n+1)$-bit and $(n+2)$-bit bijective S-boxes also have a threshold implementation with the same number of shares. These results imply, in particular, that for any $n \geq 3$ there exist permutations over $\mathbb{F}_2^n$ admitting a threshold implementation with $t+1$ shares.

5.2 Two conjectures on nonexistence of threshold implementations with t+1 shares

Analysing all the data available in [12,14,30], we observed that all permutations admitting $t+1$ shares, except one, have 0 nonlinearity (have linear component functions making them the worst with respect to linear cryptanalysis). The only exception C_{301}^4 (notation used in [12]) has nonlinearity 2 which is not good either. All these functions are very bad with respect to differential attacks too: they have differential uniformity greater or equal to 2^{n-1}.

We believe that the fact that the 3-bit Cube function does not admit such a threshold implementation is not a coincidence. There have been many examples in the vectorial Boolean function theory where properties of the Cube functions reflected general results [19]. The fact that the function x^3 over $\mathbb{F}_{2^3}$, being the simplest non-linear power permutation and the simplest case of APN functions does not admit $t+1$ shares, leads us to the conjectures presented below.

Conjecture 1. *No power permutation of algebraic degree $t \geq 2$ admits a threshold implementation with $t+1$ shares.*

Conjecture 2. *No APN permutation of algebraic degree t admits a threshold implementation with $t+1$ shares.*

These conjectures are supported by all computational data available nowadays [12, 14, 30].

6 Conclusion

In this paper, we have presented a universal construction for the threshold implementation of permutations (bijective S-boxes) with $t+2$ shares, where t is the algebraic degree of the S-box. This is the first construction that applies to all bijective S-boxes of any size (that is, in any number of variables). This result is a significant advance with respect to the state of the art on the construction of threshold implementations, which were either computational searches for small sizes using direct sharing [12], or techniques to achieve uniformity. For instance, the use of correction terms [6], fresh randomness, or the changing of the guards [13]. It was also noted that this construction yields a threshold implementation with $t+2$ shares in the case of the 3-bit inversion. Such an implementation was proven to be optimal for this S-box by Bilgin et al. [18]. This means that this construction is optimal as a universal one for all S-boxes; of course, being universal, it cannot achieve the theoretical lower bound $t+1$ on the number of shares, since some S-boxes are known not to allow TI with such number of shares.

We observed that the construction is rather flexible, allowing to change the form of the constructed threshold implementation using correction terms and providing a description of the terms that can be used for this purpose.

We applied this construction to obtain the first uniform sharing of the AES S-box. We have analysed the cost of the implementation using this construction, both directly to the AES S-box and to a decomposition of the S-box using cubic power permutations. The results include the first design of a randomness-free AES S-box in one cycle and a direct improvement on the S-box design by Wegener and Moradi [15]. We noted that it might be possible to improve the presented implementation using correction terms. However, we leave this investigation as future work.

The result is also of importance for the research on permutation polynomials, as it provides a method for the construction of new infinite families of permutations.

This result is a very important advance in the understanding of the general theory of threshold implementation. Regarding other aspects of this topic, very little is known. In some cases, it is very hard to find even computational results. Achieving the uniformity property for non-permutations is very challenging, since there is no characterisation that is computationally faster to verify than the definition. Moreover, we do not know the exact reasons why some permutations do not admit a threshold implementation with $t+1$ shares. In fact, we have very little data to study the non-existence because it is only feasible to run the exhaustive search of correction terms [12] for 3-bit S-boxes. However, we could still observe that the only examples of permutations that admit a threshold implementation with $t+1$ shares are ones with bad cryptographic properties. We conjecture that bijective S-boxes such as power permutations or APN permutations, which are of particular interest for cryptography, do not admit a TI with $t+1$ shares.

Acknowledgements We thank Ventzislav Nikov for useful discussions. We thank Zhenda Zhang for the help with the hardware design. Siemen Dhooghe is supported by a PhD Fellowship from the Research Foundation – Flanders (FWO). The research of this paper is supported by the Norwegian Research Council.

References

[1] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology - CRYPTO '99 Proceedings*, 1999, pp. 388–397. [Online]. Available: https://doi.org/10.1007/3-540-48405-1_25

[2] National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES)," FIPS PUB 197, U.S. Department of Commerce, Nov. 2001.

[3] L. Goubin and J. Patarin, "DES and differential power analysis (the "duplication" method)," in *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, ser. Lecture Notes in Computer Science, Ç. K. Koç and C. Paar, Eds., vol. 1717. Springer, 1999, pp. 158–172. [Online]. Available: https://doi.org/10.1007/3-540-48059-5_15

[4] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, ser. Lecture Notes in Computer Science, M. J. Wiener, Ed., vol. 1666. Springer, 1999, pp. 398–412. [Online]. Available: https://doi.org/10.1007/3-540-48405-1_26

[5] S. Mangard, N. Pramstaller, and E. Oswald, "Successfully attacking masked AES hardware implementations," in *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, ser. Lecture Notes in Computer Science, J. R. Rao and B. Sunar, Eds., vol. 3659. Springer, 2005, pp. 157–171. [Online]. Available: https://doi.org/10.1007/11545262_12

[6] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold implementations against side-channel attacks and glitches," in *Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006, Proceedings*, ser. Lecture Notes in Computer Science, P. Ning, S. Qing, and N. Li, Eds., vol. 4307. Springer, 2006, pp. 529–545. [Online]. Available: https://doi.org/10.1007/11935308_38

[7] G. Petrides, "On non-completeness in threshold implementations," in *Proceedings of ACM Workshop on Theory of Implementation Security, TIS@CCS 2019, London, UK, November 11, 2019*, B. Bilgin, S. Petkova-Nikova, and V. Rijmen, Eds. ACM, 2019, pp. 24–28. [Online]. Available: https://doi.org/10.1145/3338467.3358951

[8] T. D. Cnudde, O. Reparaz, B. Bilgin, S. Nikova, V. Nikov, and V. Rijmen, "Masking AES with d+1 shares in hardware," in *Cryptographic Hardware and Embedded*

Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings, ser. Lecture Notes in Computer Science, B. Gierlichs and A. Y. Poschmann, Eds., vol. 9813. Springer, 2016, pp. 194–212. [Online]. Available: https://doi.org/10.1007/978-3-662-53140-2_10

[9] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the limits: A very compact and a threshold implementation of AES," in *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, ser. Lecture Notes in Computer Science, K. G. Paterson, Ed., vol. 6632. Springer, 2011, pp. 69–88. [Online]. Available: https://doi.org/10.1007/978-3-642-20465-4_6

[10] A. Poschmann, A. Moradi, K. Khoo, C. Lim, H. Wang, and S. Ling, "Side-channel resistant crypto for less than 2, 300 GE," *J. Cryptol.*, vol. 24, no. 2, pp. 322–345, 2011. [Online]. Available: https://doi.org/10.1007/s00145-010-9086-6

[11] H. Groß, D. Schaffenrath, and S. Mangard, "Higher-order side-channel protected implementations of KECCAK," in *Euromicro Conference on Digital System Design, DSD 2017, Vienna, Austria, August 30 - Sept. 1, 2017*, H. Kubátová, M. Novotný, and A. Skavhaug, Eds. IEEE Computer Society, 2017, pp. 205–212. [Online]. Available: https://doi.org/10.1109/DSD.2017.21

[12] B. Bilgin, S. Nikova, V. Nikov, V. Rijmen, and G. Stütz, "Threshold implementations of all 3 ×3 and 4 ×4 s-boxes," in *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, ser. Lecture Notes in Computer Science, E. Prouff and P. Schaumont, Eds., vol. 7428. Springer, 2012, pp. 76–91. [Online]. Available: https://doi.org/10.1007/978-3-642-33027-8_5

[13] J. Daemen, "Changing of the guards: A simple and efficient method for achieving uniformity in threshold sharing," in *Cryptographic Hardware and Embedded Systems - CHES 2017 Proceedings*, 2017, pp. 137–153. [Online]. Available: https://doi.org/10.1007/978-3-319-66787-4_7

[14] D. Bozilov, B. Bilgin, and H. A. Sahin, "A note on 5-bit quadratic permutations' classification," *IACR Trans. Symmetric Cryptol.*, vol. 2017, no. 1, pp. 398–404, 2017. [Online]. Available: https://doi.org/10.13154/tosc.v2017.i1.398-404

[15] F. Wegener and A. Moradi, "A first-order SCA resistant AES without fresh randomness," in *Constructive Side-Channel Analysis and Secure Design - 9th International Workshop, COSADE 2018, Singapore, April 23-24, 2018, Proceedings*, ser. Lecture Notes in Computer Science, J. Fan and B. Gierlichs, Eds., vol. 10815. Springer, 2018, pp. 245–262. [Online]. Available: https://doi.org/10.1007/978-3-319-89641-0_14

[16] S. Nikova, V. Nikov, and V. Rijmen, "Decomposition of permutations in a finite field," *Cryptogr. Commun.*, vol. 11, no. 3, pp. 379–384, 2019. [Online]. Available: https://doi.org/10.1007/s12095-018-0317-2

[17] G. Petrides, "On decompositions of permutation polynomials into quadratic and cubic power permutations," *Cryptography and Communications*, 2022. [Online]. Available: https://doi.org/10.1007/s12095-022-00600-8

[18] B. Bilgin, S. Nikova, V. Nikov, V. Rijmen, N. N. Tokareva, and V. Vitkup, "Threshold implementations of small s-boxes," *Cryptogr. Commun.*, vol. 7, no. 1, pp. 3–33, 2015. [Online]. Available: https://doi.org/10.1007/s12095-014-0104-7

[19] C. Carlet, Ed., *Boolean Functions for Cryptography and Coding Theory.* Cambridge University Press, 2021. [Online]. Available: https://doi.org/10.1017/9781108606806

[20] C. Carlet, P. Charpin, and V. A. Zinoviev, "Codes, bent functions and permutations suitable for des-like cryptosystems," *Des. Codes Cryptogr.*, vol. 15, no. 2, pp. 125–156, 1998. [Online]. Available: https://doi.org/10.1023/A:1008344232130

[21] B. Bilgin, "Threshold implementations : as countermeasure against higher-order differential power analysis," Ph.D. dissertation, University of Twente, Enschede, Netherlands, 2015. [Online]. Available: http://purl.utwente.nl/publications/95796

[22] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "Higher-order threshold implementations," in *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, ser. Lecture Notes in Computer Science, P. Sarkar and T. Iwata, Eds., vol. 8874. Springer, 2014, pp. 326–343. [Online]. Available: https://doi.org/10.1007/978-3-662-45608-8_18

[23] S. Nikova, V. Rijmen, and M. Schläffer, "Secure hardware implementation of non-linear functions in the presence of glitches," in *Information Security and Cryptology - ICISC 2008, 11th International Conference, Seoul, Korea, December 3-5, 2008, Revised Selected Papers*, ser. Lecture Notes in Computer Science, P. J. Lee and J. H. Cheon, Eds., vol. 5461. Springer, 2008, pp. 218–234. [Online]. Available: https://doi.org/10.1007/978-3-642-00730-9_14

[24] C. Carlet, E. Prouff, M. Rivain, and T. Roche, "Algebraic decomposition for probing security," in *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, ser. Lecture Notes in Computer Science, R. Gennaro and M. Robshaw, Eds., vol. 9215. Springer, 2015, pp. 742–763. [Online]. Available: https://doi.org/10.1007/978-3-662-47989-6_36

[25] K. Nyberg, "Differentially uniform mappings for cryptography," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, vol. 765. Springer, 1993, pp. 55–64.

[26] ——, "S-boxes and round functions with controllable linearity and differential uniformity," in *FSE*, ser. Lecture Notes in Computer Science, vol. 1008. Springer, 1994, pp. 111–130.

[27] NANGATE, "The NanGate 45nm Open Cell Library," version: PDKv1.3 v2010 12.Apache.CCL, https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts/tree/master/flow/platforms/nangate45.

[28] T. Sugawara, "3-share threshold implementation of AES s-box without fresh randomness," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 1, pp. 123–145, 2019. [Online]. Available: https://doi.org/10.13154/tches.v2019.i1.123-145

[29] H. Groß, R. Iusupov, and R. Bloem, "Generic low-latency masking in hardware," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 2, pp. 1–21, 2018. [Online]. Available: https://doi.org/10.13154/tches.v2018.i2.1-21

[30] L. D. Meyer and B. Bilgin, "Classification of balanced quadratic functions," *IACR Trans. Symmetric Cryptol.*, vol. 2019, no. 2, pp. 169–192, 2019. [Online]. Available: https://doi.org/10.13154/tosc.v2019.i2.169-192

[31] E. Boss, V. Grosso, T. Güneysu, G. Leander, A. Moradi, and T. Schneider, "Strong 8-bit sboxes with efficient masking in hardware extended version," *J. Cryptogr. Eng.*, vol. 7, no. 2, pp. 149–165, 2017. [Online]. Available: https://doi.org/10.1007/s13389-017-0156-7

[32] K. Varici, S. Nikova, V. Nikov, and V. Rijmen, "Constructions of s-boxes with uniform sharing," *Cryptogr. Commun.*, vol. 11, no. 3, pp. 385–398, 2019. [Online]. Available: https://doi.org/10.1007/s12095-018-0345-y