Lightweight and Side-channel Secure 4×4 S-Boxes from Cellular Automata Rules

Ashrujit Ghoshal[1], Rajat Sadhukhan[1], Sikhar Patranabis[1], Nilanjan Datta[1], Stjepan Picek[2] and Debdeep Mukhopadhyay[1]

[1] Indian Institute of Technology, Kharagpur, India
[2] Delft University of Technology, The Netherlands
ashrujitg@iitkgp.ac.in,rajat.sadhukhan@iitkgp.ac.in,sikhar.patranabis@iitkgp.ac.in,nilanjan.datta@iitkgp.ac.in,s.picek@tudelft.nl,debdeep@cse.iitkgp.ac.in

Abstract. This work focuses on side-channel resilient design strategies for symmetric-key cryptographic primitives targeting lightweight applications. In light of NIST's lightweight cryptography project, design choices for block ciphers must consider not only security against traditional cryptanalysis, but also side-channel security, while adhering to low area and power requirements. In this paper, we explore design strategies for substitution-permutation network (SPN)-based block ciphers that make them amenable to low-cost threshold implementations (TI) - a provably secure strategy against side-channel attacks. The core building blocks for our strategy are cryptographically optimal 4×4 S-Boxes, implemented via repeated iterations of simple cellular automata (CA) rules. We present highly optimized TI circuits for such S-Boxes, that consume nearly 40% less area and power as compared to popular lightweight S-Boxes such as PRESENT and GIFT. We validate our claims via implementation results on ASIC using 180nm technology. We also present a comparison of TI circuits for two popular lightweight linear diffusion layer choices - bit permutations and MixColumns using almost-maximum-distance-separable (almost-MDS) matrices. We finally illustrate design paradigms that combine the aforementioned TI circuits for S-Boxes and diffusion layers to obtain fully side-channel secure SPN block cipher implementations with low area and power requirements.

Keywords: Lightweight · Block Ciphers · Side-channels · Threshold Implementation · Cellular Automata · Optimal S-Box.

1 Introduction

Lightweight cryptography has received great momentum with the proposal of a number of efficient symmetric-key cryptographic primitives in recent years. Design choices for lightweight cryptography typically focus on optimizing one or more essential implementation-based criteria, including (but not limited to) area, power, and throughput. At the same time, these primitives must also satisfy the basic security requirements against well-known cryptanalytic attacks such as linear [MY93] and differential [BS91] cryptanalysis. Lightweight block ciphers follow various design principles, amongst which *substitution-permutation network* (SPN) is highly popular. An SPN structure typically comprises several rounds, where each round has three operational layers - (a) a layer of nonlinear substitution-boxes (S-Boxes), (b) a linear permutation-layer, and (c) round-key-XOR. The impetus on lightweight cryptography has been further enhanced by NIST's recent announcement of a lightweight cryptography project [MBTM17], seeking design choices targeting a variety of devices and applications. In particular, the announcement lists resistance against *side-channel attacks* (SCAs) as a principal design criterion. This opens up the need to explore new design strategies for lightweight block ciphers that focus not only on security against traditional cryptanalysis but also side-channel security, while adhering to low area and power requirements. The aim of this paper is to address this issue with respect to the SPN block ciphers. In particular, our proposed strategies focus on protecting the two main components of any SPN block cipher, namely the S-Box layer and the permutation layer. A common protection strategy applied to both layers is the use of threshold implementation (TI) [NRR06], a provably secure technique against side-channels that has its roots in multi-party computation.

S-Boxes are essential components for any SPN block cipher, since they contribute to the protection against traditional cryptanalytic techniques. In order to do so, S-Boxes must fulfill certain cryptographic properties. The minimum set of criteria necessary to consider when designing S-Boxes for SPN designs includes bijectivity, high nonlinearity, and low differential uniformity. Naturally, in various ciphers, S-Boxes are of different sizes, which results in different values of cryptographic properties and can even lead to using S-Boxes with suboptimal properties (see e.g., the Keccak design where the S-Box (χ transformation) is suboptimal with respect to the nonlinearity and differential uniformity properties [BDPA11]).

When considering lightweight cryptography, the situation is simpler. The dominant S-Box size there is 4 × 4, which does not allow much difference in cryptographic properties, and in fact ciphers commonly use S-Boxes that are *optimal*. Optimal S-Boxes are those that are bijective, with nonlinearity equal to 4, and differential uniformity equal to 4 [LP07]. Such optimal S-Boxes are found in numerous popular designs like PRESENT [BKL$^+$07], Noekeon [JDR00], Piccolo [SIH$^+$11], Prince [BCG$^+$12], Rectangle [ZBL$^+$15], Skinny [BJK$^+$16], Midori [BBI$^+$15], etc. Some recently proposed block ciphers such as GIFT [BPP$^+$17] use cryptographically non-optimal lightweight 4×4 S-Boxes with special properties that allow combining them with bit permutations to achieve optimal diffusion characteristics. The small size of 4 × 4 S-Boxes has also enabled researchers to classify all optimal S-Boxes up to the affine equivalence where they show there are 16 optimal non-equivalent classes (commonly denoted G_0 to G_{15}) [LP07]. Existing works have also gone so far as to exhaustively enumerate all 4 × 4 bijective S-Boxes [Saa12].

Despite the existence of such classifications, it is largely an open problem to propose design strategies for S-Boxes that are low-area, low-power, and at the same-time, amenable to side-channel secure implementations (that is, the corresponding SCA-resistant implementations also optimize area and power as much as possible). One of the foremost techniques for securing S-Box implementations is the use of masking countermeasures [RP10, GPQ11, RBN$^+$15] that are provably secure up to a pre-determined attack order. In more recent times, TI seem to be the preferred choice owing to their enhanced security coverage, particularly against glitch-based SCAs. Thus, our aim is to design cryptographically optimal 4 × 4 nonlinear functions that support low-area and low-power implementations, while having low-cost side-channel protections in the form of TI circuits.

1.1 Overview of Our Contributions and Techniques

The main contributions of this paper are briefly summarized below:

- **Lightweight and Side-channel Secure Design Strategies for S-Boxes.**

  In this paper, we use *cellular automata* in order to design such nonlinear functions with inherently lightweight implementations. A cellular automaton is a finite state machine whose state transitions are based on simple local rules. Prior studies have extensively analyzed the scope of realizing complex functions via repeated iterations of this simple rules [Wol83, Wol84b, Wol84a]. A recent work by Picek et al. [PMY$^+$17] explores the possibility of designing cryptographically optimal 4 × 4 S-Boxes from such simple 4 × 1 CA-based rules. The idea is to iterate over a single instance of the CA rule, while cyclically shifting the input bits, to obtain one output bit of an S-Box at a time. In this work, we take a step further and explore the possibility of designing cryptographically optimal 4 × 4 S-Boxes from CA rules, while also ensuring that such S-Boxes give rise to side-channel secure TI circuits with low area footprint and power consumption. The main design principle for the TI circuit remains the same - we protect the core CA rule by decomposing the input and output bits into as few shares as possible, and then iterate over this core unit by cyclically permuting the input bits. We demonstrate that a significant proportion of the resulting S-Boxes achieve cryptographically optimal properties, and give rise to distinct classes based on their implementation overheads and amenability to TI designs. We also demonstrate additional optimizations on the most lightweight of these S-Box classes by exploiting the decomposability of its CA rule into smaller Boolean functions. Our implementation results on ASIC (180nm technology) show that the most lightweight TI circuit among all CA-based S-boxes has a 49.42% smaller area-footprint and consumes 52.3% less power as compared to the best-known TI of the PRESENT S-Box [PMK+11]. The same TI circuit also leads to a 35.36% smaller area-footprint and consumes 44.46% less power as compared to a highly optimized TI of the GIFT S-Box.

- **Lightweight and Side-channel Secure Design Strategies for Permutation Layers.** Permutation layers provide the much needed diffusion in any block cipher construction, and are hence important for side-channel security. Two main classes of permutation layers dominate nearly all lightweight SPN constructions - bit permutations and almost-maximum-distance-separable (almost-MDS) permutations. Examples of the former include PRESENT [BKL+07] and GIFT [BPP+17], while an example of the latter strategy is Midori [BBI+15]. In this paper, we present a comparative analysis of the area and power overheads corresponding to TI designs for both choices of permutations. Such a comparative analysis allows a designer to analyze the pros and cons of choosing either of these strategies with respect to a given application.
- **Combining it All Together.** Finally, we present a trade-off analysis between the design choices for the S-Box and permutation layers as components in an SPN structure. We first observe that our CA-based S-Boxes have a branch number of 2 (as opposed to 3 for the PRESENT S-Box), and also lack the bad-output-good-input (BOGI) property exhibited by the GIFT S-Box [BPP+17]. This makes it practically infeasible to combine these S-Boxes with bit-permutation layers in a full SPN structure and necessitates almost-MDS permutation layers. Interestingly, it turns out that the area and power savings from our CA-based S-Boxes outweigh the additional area and power requirements for an almost-MDS permutation layer over a bit permutation layer, particularly when implemented for side-channel security via TI. With these observations, we propose using CA-based S-Boxes in conjunction with almost-MDS mappings as a new design-for-security strategy for designing lightweight block ciphers that are amenable to low-area and low-power TI designs.

1.2 Paper Organization

The rest of this paper is organized as follows. In Section 2, we introduce the notation and present background material on cryptographic properties of S-Boxes, threshold implementations (TI), cellular automata (CA) and their properties, and relevant measurement units for area footprint and power consumption of CMOS devices. Section 3 presents direct-shared TI circuits for cryptographically optimal 4 × 4 S-Boxes obtained via repeated iterations of local CA rules, along with the area and power overheads for the same on ASIC platforms (180nm technology). Section 4 further refines these TI circuits by reducing the number of shares to achieve even lower area footprint and power consumption. Section 5 compares bit permutations and MixColumns using almost-MDS matrices in terms of their amenability to low-cost TI designs. This section also presents design paradigms for combining TI for S-Boxes and diffusion layers to achieve lightweight and fully side-channel secure block cipher implementations. Finally, Section 6 summarizes the major findings of the paper and discusses possible future research directions.

2 Preliminaries

2.1 Cryptographic Optimality and Representation of S-Boxes

In the standard cryptographic nomenclature, a substitution box (abbreviated as S-Box), is a nonlinear $n \times m$ Boolean function. In the rest of the paper, we consider only S-boxes that have the same number of inputs and outputs, i.e., $n \times n$ S-boxes. Here, we briefly describe some important cryptographic properties of S-boxes.

- Algebraic Degree. To define the algebraic degree of an S-Box, we use the algebraic normal form (ANF) representation of a Boolean function f represented by a polynomial in $\mathbb{F}_2\left[x_0, \ldots, x_{n-1}\right] / (x_0^2 - x_0, \ldots, x_{n-1}^2 - x_{n-1})$ [Car10a]. The algebraic degree deg_f of a Boolean function f is defined as the number of variables in the largest product term of the function's ANF having a non-zero coefficient [Car10a]. The algebraic degree deg_F of an S-Box F is the maximum algebraic degree of all non-zero linear combinations of the coordinate functions (i.e., component functions) of F [Car10b]. Ideally, a cryptographically useful S-Box should have high algebraic degree to resist algebraic attacks [MPC04].

- Balancedness. Let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. Then, F is balanced if it takes every value of $\mathbb{F}_2^n$ exactly once.

- Nonlinearity. Nonlinearity of an $n \times n$ S-Box F equals the minimum nonlinearity of all its component functions $v \cdot F$, where $v \in \mathbb{F}_2^{n*}$ [Nyb93, Car10b]:

  $$NL_F = 2^{n-1} - \frac{1}{2} \max_{\substack{a \,\in\, \mathbb{F}_2^n \\ v \,\in\, \mathbb{F}_2^{n*}}} |W_F(a, v)|,$$

  where
  $$W_F(a, v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{v \cdot F(x) + a \cdot x}, \;\; a, v \in \mathbb{F}_2^n,$$

  is the Walsh-Hadamard transform [Car10b] of the function F and $a \cdot b$ is the usual inner product of $a, b \in \mathbb{F}_2^n$ that equals $a \cdot b = \bigoplus_{i=1}^{n} a_i b_i$. We use the notation $\mathbb{F}_2^{n*}$ to denote the non-zero elements of the vector space $\mathbb{F}_2^n$. The nonlinearity of any (n, n) function F is bounded above by the covering radius bound:

  $$NL_F \leq 2^{n-1} - 2^{\frac{n}{2}-1}.$$

- Differential Uniformity. Let F be an S-Box from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ with $a \in \mathbb{F}_2^n$ and $b \in \mathbb{F}_2^n$. We define the *difference distribution table* of F with respect to a and b as:

  $$D_F(a, b) = \left\{ x \in \mathbb{F}_2^n : F(x) \oplus F(x \oplus a) = b \right\}.$$

  The entry at position (a, b) corresponds to the cardinality of the difference distribution table $D_F(a, b)$ and is denoted as $\delta_F(a, b)$. The *differential uniformity* δ_F is then defined as [Nyb94]:

  $$\delta_F = \max_{\substack{a \,\in\, \mathbb{F}_2^{n*} \\ b \,\in\, \mathbb{F}_2^n}} \delta_f(a, b).$$

- Differential Branch Number. Let F be an S-Box from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. We define the differential branch number of F as:

  $$BN_F = \min_{x \neq y} \; wt(x \oplus y) + wt(F(x) \oplus F(y)),$$

  where $wt(a)$ denotes the Hamming weight of a. Throughout this paper we use the term branch number to denote the differential branch number.

In order to resist linear and differential cryptanalysis attacks, a balanced S-Box should ideally have high nonlinearity and low differential uniformity. In particular, a 4 × 4 S-Box is said to be *cryptographically optimal* if it is bijective, has nonlinearity equal to 4, and differential uniformity equal to 4 [LP07].

2.2 Threshold Implementation: A Countermeasure to SCA

Here, we provide a brief overview of Threshold Implementation along with a simple example and a brief discussion on the importance of this countermeasure to resist side-channel attacks.

2.2.1 Countermeasures against SCA

There exist various countermeasures against side-channel power attacks which have been proposed over the years. A general approach focuses on decreasing the information gathered from traces.

- Noise Addition. Introducing external noise in the side-channel, shuffling the operations or inserting dummy operations in cryptographic implementations are often used as a countermeasure against side-channel attacks. The basic objective is to reduce the signal-to-noise ratio (SNR), and thereby decrease the information gathered from traces. Many works on this topic explicitly focus on improving the statistical distribution of these delays. Still, as shown by Durvaux et al. [DRS$^+$12], these countermeasures become insecure with increasing attack time.

- Dynamic and Differential CMOS Logic. Tiri et al. [TV04a] proposed Sense Amplifier Based Logic (SABL), a logic style that uses a fixed amount of charge for every transition, including the degenerated events in which a gate does not change state. In every cycle, a SABL gate charges a total capacitance with a constant value. SABL is based on two principles: (i) it is a *Dynamic and Differential Logic* (DDL) and therefore has exactly one switching event per cycle (independent of the input value and sequence) and (ii) during a switching event, it discharges and charges the sum of all the internal node capacitances together with one of the balanced output capacitances. Some special constant power implementation like Wave Dynamic Digital Logic (WDDL) [TV04b] are based on SABL and have a close to constant power consumption. However, this comes at a huge overhead costs of area, time, and power consumption.

- Leakage Resilience. Another countermeasure, typically applied at the system level, focuses on restricting the number of usages of the same key for an algorithm. However, generation and synchronization of new keys has a major practical issue. Dziembowski et al. introduced a technique called leakage resilience [DP08], which relocates this problem to the protocol level by introducing an algorithm to generate these keys. This approach can be extended such that several different keys (chunks) are used with the same input text. Nevertheless, both of these techniques drastically decrease the performance of a system, and hence are not practical for real-world implementations.

- Masking. One of the most efficient and powerful approaches to thwart DPA is Masking [CJRR99, GP99], which targets to break the correlation between the power traces and the intermediate values of the computations. This powerful method achieves security by randomizing the intermediate values using secret sharing and carrying out all the computations on the shared values.

2.2.2 Threshold Implementation: A Brief Overview

Threshold Implementation (TI) is a widely used masking technique proposed by Nikova et al. [NRR06] as a countermeasure against Differential Power Attacks (DPA) [KJJ99]. What sets TI apart from most masking techniques is the security it guarantees even in non-ideal circuits where glitches have shown to result in leakage in more conventional masking schemes [MPO05]. Initially, the proposals on TI dealt solely with the first-order DPA security, but it was later extended to protect against higher-order DPA attacks as well [BGN+14, RBN+15]. More recently, the pitfalls in the multivariate setting of the higher-order TI scheme were solved in [RBN+15]. TI works under extremely relaxed assumptions on the underlying leakage which are more achievable in practical scenarios. It offers provable security and allows to construct secure circuits which are practical in size. Additionaly, designing TI does not require many design iterations in practice. TI is a Boolean masking technique based on secret sharing and secure multi-party computation. In order to achieve the mentioned security a TI design must satisfy the following properties:

- Uniformity. All intermediate shares are required to be uniformly distributed. This ensures decoupling of intermediate states from the mean of the leakages, which is essential requirement to counteract the first-order DPA. It suffices to check uniformity at the inputs and the outputs of each of the functions [Bil15]. In case no direct uniform sharing is found, uniformity can be either achieved through correction terms by using more input shares, or by re-masking i.e., adding randomness after the non-uniform computation.

- Non-completeness. Any combination of d or fewer component functions f_i of $\mathbf{f}$ must be independent of at least one input share x_i in order to achieve d^{th}-order non-completeness. For protection against the first-order DPA, 1^{st}-order non-completeness is required, i.e., every function must be independent of at least one input share. Non-completeness ensures that the side-channel security of the final circuit is not affected by glitches. Since glitches can only occur in component functions and each individual component function f_i lacks knowledge of at least one share x_i, glitches cannot reveal any additional information.

- Correctness. Applying the component functions to a valid shared input must always yield a valid sharing of the correct output.

2.2.3 A Simple Example of a Threshold Implementation

We illustrate the concept of TI using a simple example of a two-bit multiplier circuit computing $\mathbf{a} = \mathbf{xy}$. The following is a uniform sharing of the circuit [GDC17] with 1^{st}-order non-completeness using four input and output shares.

$$\mathbf{x} = (x_1 \oplus x_2 \oplus x_3 \oplus x_4)$$
$$\mathbf{y} = (y_1 \oplus y_2 \oplus y_3 \oplus y_4)$$
$$\mathbf{a} = (a_1 \oplus a_2 \oplus a_3 \oplus a_4)$$

where the output shares a_1, a_2, a_3, a_4 are computed as:

$$\begin{aligned}
a_1 &= (x_2 \oplus x_3 \oplus x_4)(y_2 \oplus y_3) \oplus y_3 \oplus y_4\\
a_2 &= (x_1 \oplus x_3)(y_1 \oplus y_4) \oplus x_1 y_3 \oplus x_4\\
a_3 &= (x_2 \oplus x_4)(y_1 \oplus y_4) \oplus x_4 \oplus y_4\\
a_4 &= x_1 y_2 \oplus y_3
\end{aligned}$$

The number of input and output shares can be further reduced using random bits (see [Bil15] for details).

2.3 Cellular Automata

Cellular Automata (CA) are parallel computational models used in order to simulate and analyze various discrete complex systems. A cellular automaton consists of a regular grid (lattice) of cells. The grid may be in any finite number of dimensions. For each cell, a set of cells called its neighborhood is defined relative to the specified cell. Each cell is in one of a finite number of states. Typically, at every time step all the cells update their states synchronously. The state update is governed by a local rule which is applied to the neighborhood of every cell.

CA as Vectorial Boolean Function. In this paper, we restrict ourselves to periodic boundary one-dimensional Boolean cellular automata i.e., the case where every cell is in state 0 or 1 and the lattice is a linear array. A Periodic Boundary CA (PBCA) with n input cells $F : \mathbb{F}_2{}^n \to \mathbb{F}_2{}^n$ is defined for all $x \in \mathbb{F}_2{}^n$ as:

$$F(x_1, x_2, \cdots, x_n) = (f(x_1, \cdots, x_d), \cdots, f(x_{n-d+2}, \cdots, x_1), \cdots, f(x_n, \cdots, x_{d-1}))$$

where f is a Boolean function on d variables($d \leq n$) is called a local rule. Thus, a CA can be seen as a vectorial Boolean function (S-box) where each coordinate function f_i corresponds to the local rule f applied to the neighborhood $(x_i, \cdots, x_{i+d-1})$. The vectorial Boolean function F of a CA is also called the CA global rule.

We note that cellular automata based S-Boxes are actually widely used today, since the nonlinear transformation χ in Keccak is actually a PBCA with $n = 5$ cells and local rule f defined as:

$$f(x_1, x_2, x_3) = x_1 \oplus x_2 x_3 \oplus x_3 \quad . \tag{1}$$

Besides being used in Keccak, the same rule is also used in Panama [DC98], Radio-Gatún [BDPA06], Subterranean [CDGP93], and 3Way [DGV94] ciphers. Unfortunately, despite being very small rule that can be efficiently implemented, it results in optimal S-Boxes only for dimension 3 × 3 and is bijective only for odd dimensions. Finally, Picek et al. recently showed that CA-based S-boxes can be very efficient when considering power and area [PMY+17].

2.4 Area Overhead and Power Consumption Results

The CMOS technology used for all ASIC implementation results reported in this paper is 180nm. Each implemented circuit is taken through the RTL-to-GDS2 flow to estimate the area overhead and power consumption. We used Synopsys Design Compiler version I-2013.12-SP5-4 for synthesis and Synopsys IC-Compiler version J-2014.09-SP1 for placement and routing of the design. For simulation we used Synopsys VCS version I-2014.03-SP1-1. Standard cell library TSL18FS120 from Tower Semiconductor Ltd. is used for physical design. The area overhead for all implemented circuits are measured in terms of gate equivalents (GE), where a GE in our case is equal to the lowest area occupied by a 2-input NAND gate of 1x drive of 180nm technology.

The total power consumption of a CMOS device is given by:

$$P_{total} = P_{static} + P_{dynamic},$$

where P_{static} and $P_{dynamic}$ denote the static and dynamic power consumption of the device. In this paper, we concentrate on the dynamic power consumption that originates from the switching activity of the circuit:

$$P_{dynamic} = \alpha C V^2 f,$$

where α is the switching factor (the probability of a bit switching from 0 to 1), C is the switched capacitance, V is the voltage, and f is the clock frequency. In our approach, we aim to use a simple structure of CA-based elements, which reduces the area and consequently the capacitance (since capacitance depends on the area). As the capacitance reduces, $P_{dynamic}$ also reduces since the other factors do not increase.

3 Lightweight S-Boxes from Cellular Automata Rules

In this section, we illustrate our cellular automata (CA)-based design strategies for obtaining 4×4 S-Boxes that are area and power-efficient, and also amenable to low-cost TI. The idea is to choose a local CA rule, which is essentially a 4×1 Boolean function, such that it has a low-cost equivalent implementation in hardware. The 4×4 S-Box mapping is obtained by applying the same CA rule to four different (cyclic) permutations of the input bits. This allows for an iterative implementation in hardware, with the CA rule implemented once in the data-path, and the control unit applying a cyclically shifted variant of the input bits in each clock cycle to obtain the corresponding output bit. We first describe the De Bruijn graph-based technique to choose the local CA rule, and subsequently enumerate certain cryptographically optimal S-Boxes obtained with this procedure. We also classify these S-Boxes in terms of their amenability to low-area and low-power TI, and present optimized TI designs for representatives from each class.

3.1 Choosing the CA Rule

Given a 4×1 CA rule f, the corresponding 4×4 S-Box is given by:

$$S(X, Y, Z, W) = (f(X, Y, Z, W), f(Y, Z, W, X), f(Z, W, X, Y), f(W, X, Y, Z))$$

We focus on choosing such CA rules that ensure that the corresponding S-box is bijective. The test for injectivity of the global map of a one-dimensional CA was shown to be decidable in [AP72], while the test for surjectivity for the same was shown to have a quadratic-time algorithm in [Sut91], using De Bruijn graphs. These graphs provide a convenient way to describe configurations of linear CAs. We follow these principles to identify local 4×1 CA rules, which in turn guarantee that the resultant 4×4 S-Box is bijective. The detailed technique for choosing such a CA rule is as follows.

3.1.1 De Bruijn Graph Representation

For any CA with an n-variable local rule $f : \mathbb{F}_2{}^n \rightarrow \mathbb{F}_2$, the associated De Bruijn graph is a directed graph $G = (V, E)$, where every vertex $v \in V$ is labeled with an $(n-1)$-bit string. There exists an edge e from vertex v_1 to vertex v_2 if the first $(n-2)$ bits of the label of v_2 are the same as the last $(n-2)$ bits of the label of v_1. For example, the De Bruijn graph with $n = 4$ has an edge from $v_1 = 010$ to $v_2 = 100$ as the first two bits of v_2 are 10, which is same as the last 2 bits of v_1. Quite evidently, $|V| = 2^{n-1}$, and $|E| = 2 \cdot 2^{n-1} = 2^n$ (observe that each vertex has exactly two incoming and two outgoing edges).

3.1.2 Generating Optimal 4×4 S-Boxes from De Bruijn Graphs

Given a De-Bruijn graph $G = (V, E)$ with $|V| = 2^{n-1}$, a CA local rule may be derived by associating each edge of this graph with a bit $b \in \{0, 1\}$. Since there are 2^n edges, the total number of possible CA rules that can be associated with this graph is 2^{2^n}. In particular, for $n = 4$, the total number of such CA rules is $2^{2^4} = 2^{16}$. Each such rule gives rise to a unique 4×4 function. An exhaustive search of these functions yields 1 536 bijective functions, which are our candidate S-Boxes. Finally, we test these functions for cryptographic optimality in terms of their nonlinearity and differential uniformity, which narrows down our search space to 512 candidate S-Boxes, which may be further sub-classified into four affine-equivalent classes - namely, G_3, G_4, G_5, and G_6. Details of these S-Boxes have been reported previously in [MPLJ17].

We would like to point out that the number of possible CA-based rules is 2^{16} (as $n = 4$) in our case. Hence, instead of De Bruijin graph representation, we could have also used a simple brute-force approach. However, for higher values of n, where brute force search may not be feasible, a systematic approach with De Bruijin graph is a good choice.

3.2 Classification of Cryptographically Optimal CA-based 4×4 S-Boxes

Our next step is to classify the 512 cryptographically optimal CA-based 4×4 S-Boxes into certain classes, such that each category comprises S-Boxes that are expected to have similar area and power overhead in hardware, as well as similar TI circuit representations. As it turns out, each of these quantities are closely related to the nature of the algebraic normal form (ANF) representation of the S-Boxes. Given that each S-Box under consideration has optimal algebraic degree 3, we use the following facts from [BGN+14]:

- CA-based S-Boxes with the same number of cubic, quadratic, and linear terms in their ANF form have similar area footprint and expected power consumption in hardware.
- CA-based S-Boxes with the same number of cubic, quadratic, and linear terms in their ANF form have nearly identical TI circuits owing to their nearly identical algebraic structure.

Based on this rationale, we classify the S-Boxes depending on the number of linear, quadratic, and cubic terms present in the ANF of the S-Box. According to this classification, we have obtained 12 S-Box classes as shown in Table 1.

We also list the CA rules corresponding to representative optimal S-Boxes for each class. Note that class (a, b, c) comprises optimal S-Boxes with a cubic terms, b quadratic terms, and c linear terms, respectively. We also summarize the cryptographic properties of these representative S-Boxes in Table 2, and compare them with the cryptographic properties of popular 4×4 S-Boxes that include the S-Boxes of PRESENT, GIFT, Skinny, Piccolo, Noekeon, Midori and Prince.

3.3 Threshold Implementations of CA-based S-Boxes

We now describe direct sharing-based TI circuits for the aforementioned classes of CA-based S-boxes, and compare their relative area overheads and power consumption results.

3.3.1 TI of CA-based S-Boxes with Examples

Since each of the representative S-Boxes listed above has algebraic degree equal to 3, we adopt the direct 4-to-4 non-complete sharing method for cubic functions originally

Table 1: Grouping S-Boxes into classes by ANF properties

S-Box Class	**Representative CA Rule**
(1,2,2)	$f(X,Y,Z,W) = XZW \oplus XY \oplus YW \oplus Y \oplus Z$
(1,3,1)	$f(X,Y,Z,W) = YZW \oplus XZ \oplus YZ \oplus YW \oplus X$
(1,3,3)	$f(X,Y,Z,W) = YZW \oplus XY \oplus XZ \oplus YW \oplus Y \oplus Z \oplus W$
(1,4,2)	$f(X,Y,Z,W) = YZW \oplus XY \oplus XZ \oplus XW \oplus ZW \oplus X \oplus W$
(1,5,1)	$f(X,Y,Z,W) = XYW \oplus XY \oplus XZ \oplus XW \oplus YW \oplus ZW \oplus Z$
(1,5,3)	$f(X,Y,Z,W) = XYW \oplus XY \oplus XZ \oplus XW \oplus YZ \oplus YW \oplus Y \oplus Z \oplus W$
(3,2,2)	$f(X,Y,Z,W) = XYZ \oplus XZW \oplus YZW \oplus XZ \oplus YZ \oplus X \oplus Y$
(3,3,1)	$f(X,Y,Z,W) = XYZ \oplus XZW \oplus YZW \oplus XZ \oplus XW \oplus YW \oplus Z$
(3,3,3)	$f(X,Y,Z,W) = XYW \oplus XZW \oplus YZW \oplus XY \oplus XZ \oplus YW \oplus X \oplus Z \oplus W$
(3,4,2)	$f(X,Y,Z,W) = XYZ \oplus XYW \oplus XZW \oplus XY \oplus XZ \oplus XW \oplus YZ \oplus Z \oplus W$
(3,5,1)	$f(X,Y,Z,W) = XYZ \oplus XYW \oplus YZW \oplus XZ \oplus XW \oplus YZ \oplus YW \oplus ZW \oplus Y$
(3,5,3)	$f(X,Y,Z,W) = XYZ \oplus XYW \oplus XZW \oplus XY \oplus XZ \oplus YZ \oplus YW \oplus ZW \oplus X \oplus Y \oplus W$

Table 2: Cryptographic properties of the considered S-boxes. All the properties listed here corresponding to the Skinny, Piccolo, Noekeon, Midori, and Prince S-Boxes are the same, and we represent all of them by the symbol X.

S-Box	**Nonlinearity**	**Differential Uniformity**	**Optimality**	**Balancedness**	**Algebraic Degree**	**Branch Number**
CA-based	4	4	Yes	Yes	3	2
PRESENT	4	4	Yes	Yes	3	3
GIFT	4	6	No	Yes	3	2
X	4	4	Yes	Yes	3	2

proposed in [Bil15] to obtain the corresponding TI circuits for each of the corresponding CA rules. We explicitly depict two of the most area-efficient and low-power TI circuits below. These correspond to the representative CA-rules for the S-Box classes $(1,2,2)$ and $(1,3,1)$, respectively. Note that $\{X_j, Y_j, Z_j, W_j\}_{j\in[1,4]}$ denote the shares for the input bits X, Y, Z and W, respectively, while $\{f_j\}_{j\in[1,4]}$ denotes the shares for the output f of the CA rule.

Class:(1,2,2) , CA-Rule: $\boldsymbol{f = XZW \oplus YW \oplus XY \oplus Y \oplus Z}$

$$
\begin{aligned}
f_1 &= (X_1Z_2W_3) \oplus (X_1Z_3W_2) \oplus (X_2Z_1W_3) \oplus (X_2Z_3W_1) \oplus (X_3Z_1W_2) \oplus (X_3Z_2W_1) \oplus Y_1 \oplus Z_1 \\
f_2 &= ((X_2 \oplus X_3 \oplus X_4)(Z_2 \oplus Z_3 \oplus Z_4)(W_2 \oplus W_3 \oplus W_4)) \oplus ((X_2 \oplus X_3 \oplus X_4)(Y_2 \oplus Y_3 \oplus Y_4)) \\
&\quad \oplus ((Y_2 \oplus Y_3 \oplus Y_4)(W_2 \oplus W_3 \oplus W_4)) \oplus Y_2 \oplus Z_2 \\
f_3 &= (X_1(Z_3 \oplus Z_4)(W_3 \oplus W_4)) \oplus (Z_1(X_3 \oplus X_4)(W_3 \oplus W_4)) \oplus (W_1(X_3 \oplus X_4)(Z_3 \oplus Z_4)) \\
&\quad \oplus (X_1Z_1(W_3 \oplus W_4)) \oplus (X_1W_1(Z_3 \oplus Z_4)) \oplus (Z_1W_1(X_3 \oplus X_4)) \oplus (X_1Z_1W_1) \\
&\quad \oplus (X_1(Y_3 \oplus Y_4)) \oplus (Y_1(X_3 \oplus X_4)) \oplus (X_1Y_1) \oplus (Y_1(W_3 \oplus W_4)) \oplus (W_1(Y_3 \oplus Y_4)) \\
&\quad \oplus (Y_1W_1) \oplus Y_3 \oplus Z_3 \\
f_4 &= (X_1Z_1W_2) \oplus (X_1Z_2W_1) \oplus (X_2Z_1W_1) \oplus (X_1Z_2W_2) \oplus (X_2Z_1W_2) \oplus (X_2Z_2W_1) \\
&\quad \oplus (X_1Z_2W_4) \oplus (X_2Z_1W_4) \oplus (X_1Z_4W_2) \oplus (X_2Z_4W_1) \oplus (X_4Z_1W_2) \oplus (X_4Z_2W_1) \\
&\quad \oplus (X_1Y_2) \oplus (Y_1X_2) \oplus (Y_1W_2) \oplus (W_1Y_2) \oplus Y_4 \oplus Z_4
\end{aligned}
$$

Class:(1,3,1) , CA-Rule: $\boldsymbol{f = YZW \oplus YW \oplus YZ \oplus XZ \oplus X}$

$$
\begin{aligned}
f_1 = {} & (Y_1Z_2W_3) \oplus (Y_1Z_3W_2) \oplus (Y_2Z_1W_3) \oplus (Y_2Z_3W_1) \oplus (Y_3Z_1W_2) \oplus (Y_3Z_2W_1) \oplus X_1 \\
f_2 = {} & ((Y_2 \oplus Y_3 \oplus Y_4)(Z_2 \oplus Z_3 \oplus Z_4)(W_2 \oplus W_3 \oplus W_4)) \oplus ((X_2 \oplus X_3 \oplus X_4)(Z_2 \oplus Z_3 \oplus Z_4)) \\
& \oplus ((Y_2 \oplus Y_3 \oplus Y_4)(Z_2 \oplus Z_3 \oplus Z_4)) \oplus ((Y_2 \oplus Y_3 \oplus Y_4)(W_2 \oplus W_3 \oplus W_4)) \oplus X_2 \\
f_3 = {} & (Y_1(Z_3 \oplus Z_4)(W_3 \oplus W_4)) \oplus (Z_1(Y_3 \oplus Y_4)(W_3 \oplus W_4)) \oplus (W_1(Y_3 \oplus Y_4)(Z_3 \oplus Z_4)) \\
& \oplus (Y_1Z_1(W_3 \oplus W_4)) \oplus (Y_1W_1(Z_3 \oplus Z_4)) \oplus (Z_1W_1(Y_3 \oplus Y_4)) \oplus (Y_1Z_1W_1) \\
& \oplus (X_1(Z_3 \oplus Z_4)) \oplus (Z_1(X_3 \oplus X_4)) \oplus (X_1Z_1) \oplus (Y_1(Z_3 \oplus Z_4)) \oplus (Z_1(Y_3 \oplus Y_4)) \oplus \\
& (Y_1Z_1) \oplus (Y_1(W_3 \oplus W_4)) \oplus (W_1(Y_3 \oplus Y_4)) \oplus (Y_1W_1) \oplus X_3 \\
f_4 = {} & (Y_1Z_1W_2) \oplus (Y_1Z_2W_1) \oplus (Y_2Z_1W_1) \oplus (Y_1Z_2W_2) \oplus (Y_2Z_1W_2) \oplus (Y_2Z_2W_1) \oplus (Y_1Z_2W_4) \\
& \oplus (Y_2Z_1W_4) \oplus (Y_1Z_4W_2) \oplus (Y_2Z_4W_1) \oplus (Y_4Z_1W_2) \oplus (Y_4Z_2W_1) \oplus (X_1Z_2) \oplus (Z_1X_2) \\
& \oplus (Y_1Z_2) \oplus (Z_1Y_2) \oplus (Y_1W_2) \oplus (W_1Y_2) \oplus X_4
\end{aligned}
$$

Figure 3.1 illustrates the hardware architecture for the direct-sharing based TI circuit corresponding to a given CA rule. The main components of the architecture are the shift registers (cyclic) for the shares corresponding to the input variables, the core block implementing the TI circuit for the CA rule, and the demultiplexer gates that are used to output one bit per clock cycle. Note that the counter bits are dependent only on the clock signal; in particular, they are independent of the other intermediate share values, and hence need not themselves be shared. A comparison of the area and power consumption for the direct sharing-based TI circuits for all representative S-Boxes is given in Table 3. The following trend is evident from the hardware implementation results:

Observation 1. *If (i)* $a_1 < a_2$ *or (ii)* $a_1 = a_2$, $b_1 + c_1 < b_2 + c_2$ *then TI of an S-Box belonging to class* (a_1, b_1, c_1) *has lower area and power consumption than an S-Box of class* (a_2, b_2, c_2).

On the other hand, in the case where $a_1 = a_2$ and $(b_1 + c_1) = (b_2 + c_2)$, there is no such obvious trend. This could be attributed to certain optimizations made by the design compiler during synthesis.

3.3.2 Comparison with Direct-Shared TI for Other Popular S-Boxes

Now, we provide a comparative study of our S-Boxes with a class of lightweight S-Boxes that includes PRESENT, GIFT, Skinny, Piccolo, Noekeon, Midori, and Prince. Note that the first six CA-based S-Box representatives (for classes $(1, 2, 2)$ through $(1, 5, 3)$) in Table 3 have TI circuits with lower area footprint as compared to all the other S-Boxes. Additionally, the power consumption for nearly all CA-based TI circuits is significantly lower.

Note that in the direct-shared TI, each input and output variable is four-shared, which leads to a significant area overhead. It is possible to minimize the area overheads of these circuits even further by reducing the number of shares in each case. This is achieved by a technique referred to as *composite TI*, which we describe in the next section.

4 Composite TI: Optimizing TI Circuits for Low Area and Power

In this section, we present *composite TI* - a generic technique that allows for highly optimized TI designs of CA rules, in comparison to direct sharing techniques. A similar

Table 3: TI of CA-based S-Box representatives: area and power consumption (ASIC Technology: 180nm)

S-Box		**Area (GE)**	**Dynamic Power (μ W)**
	Class		
CA-Based	(1,2,2)	265.03	232.51
	(1,3,1)	259.23	222.36
	(1,3,3)	276.06	247.78
	(1,4,2)	288.35	254.89
	(1,5,1)	276.55	244.97
	(1,5,3)	298.7	284.19
	(3,2,2)	378.98	349.76
	(3,3,1)	393.83	357.6
	(3,3,3)	415.21	398.51
	(3,4,2)	405.57	381.00
	(3,5,1)	397.10	381.46
	(3,5,3)	418.16	413.14
GIFT		303.81	380.44
PRESENT		450.54	490.18
Skinny		370.59	433.9
Piccolo		375.88	424.8
Noekeon		454.54	495.45
Midori		408.00	457.06
Prince		516.99	559.92

technique has been used in [PMK+11] to obtain a highly optimized TI for the PRESENT S-Box. The idea is to express each 4 × 1 CA rule of algebraic degree 3 as a composition of Boolean sub-functions of degree 2 each. We then proceed by identifying uniform and non-complete sharing for these degree 2 sub-functions, and subsequently cascading them. In order to maintain non-completeness, the cascading must ensure that the TI circuits for the two sub-functions are separated by using registers. This can be illustrated using the following instance. Suppose that a CA-rule $f(X)$ can be expressed as a composition of two sub-rules $g(A)$ and $h(X)$, where A denotes the intermediate output of $h(X)$. Now, consider a uniform first-order 3-sharing of h, denoted as $A_1 = h_1(X_1, X_2)$ and $A_2 = h_2(X_2, X_3)$, that are fed subsequently to the sharing of g. Here $h(X) = h_1(X_1, X_2) \oplus h_2(X_2, X_3)$. Note that the share function $g_1\left(A_1, A_2\right)$ can also be written as $g_1(X_1, X_2, X_3)$, in which case, a glitch in this function produces a leakage dependent on all the shares of X. This is avoided by partitioning the nonlinear operations with a register that disallows the propagation of a glitch affecting all the shares of an unmasked value. We illustrate the decomposition strategy for the representative S-Boxes of the classes $(1, 2, 2)$ and $(1, 3, 1)$, which are the most area and power-efficient among all the S-Box classes (see Table 3).

4.1 Decomposition for CA-based S-Box Class $(1, 2, 2)$

We begin by illustrating a decomposition of the representative CA-rule for the S-Box class $(1, 2, 2)$. While the original rule f has algebraic degree 3, each of the decomposed functions b_1, b_2 and b_3 have degree 2.

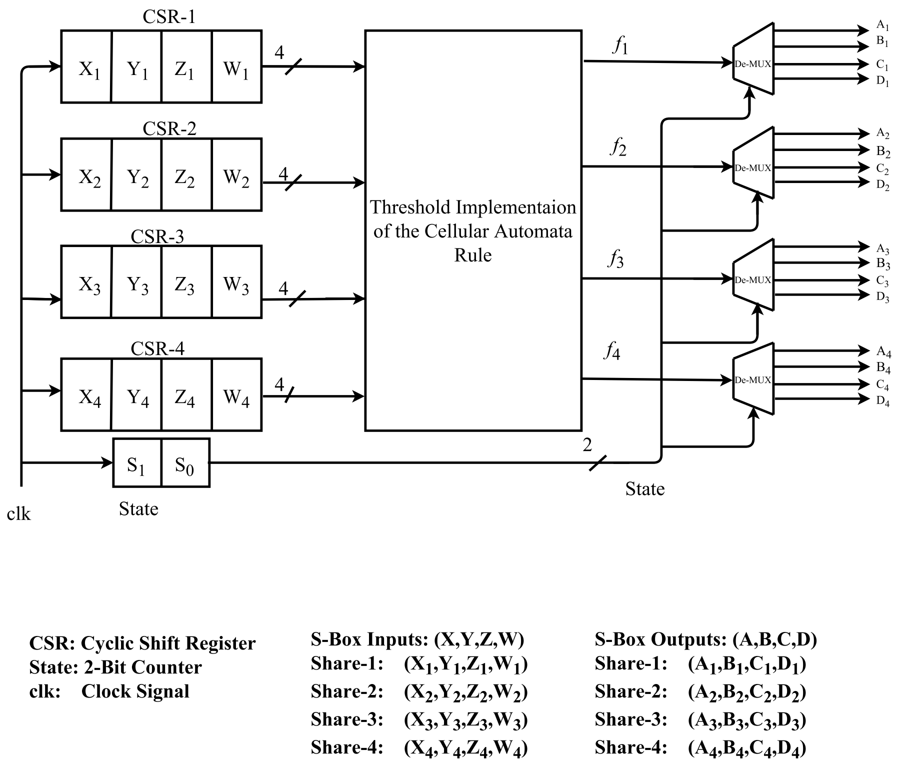

Figure 3.1: Architecture for TI circuits corresponding to CA-based S-Boxes

$$\boldsymbol{f = XZW \oplus YW \oplus XY \oplus Y \oplus Z}$$

$$\begin{aligned}
b_1(X,Y,W) &= X \oplus Y \oplus XW \oplus YW \\
b_2(X,Y,Z) &= Z \oplus XY \oplus XZ \\
b_3(X,Z,W) &= X \oplus W \oplus XZ \oplus ZW \\
f(X,Y,Z,W) &= b_1 \oplus b_2 \oplus b_1b_3 \oplus b_2b_3 = b_1(b_1,b_2,b_3)
\end{aligned}$$

The next step is to obtain a uniform three-sharing for the decomposed functions b_1, b_2, and b_3. We first present a nomenclature of the shares for the various input variables and decomposed functions.

$$\begin{aligned}
b_1 &= b_{11} \oplus b_{12} \oplus b_{13} \\
b_2 &= b_{21} \oplus b_{22} \oplus b_{23} \\
b_3 &= b_{31} \oplus b_{32} \oplus b_{33} \\
X &= X_1 \oplus X_2 \oplus X_3 \\
Y &= Y_1 \oplus Y_2 \oplus Y_3 \\
Z &= Z_1 \oplus Z_2 \oplus Z_3 \\
W &= W_1 \oplus W_2 \oplus W_3
\end{aligned}$$

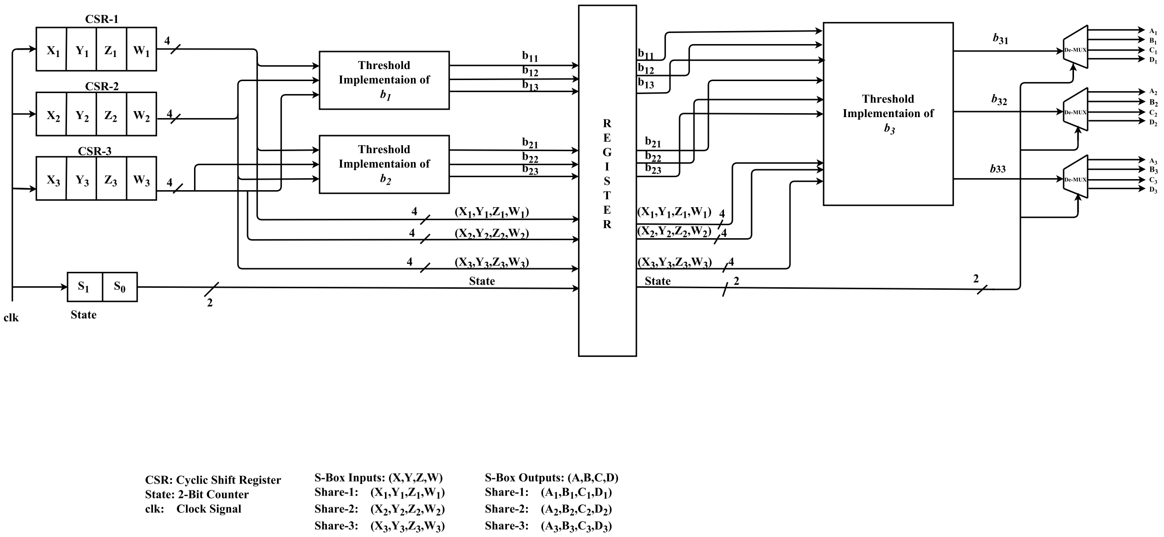

Figure 4.1: Architecture for TI circuits corresponding to CA-based S-Boxes

The three-shared TI circuit is now illustrated below:

$$\boxed{\begin{aligned}
b_{11} &= X_1 \oplus Y_2 \oplus (Y_1W_1) \oplus (Y_1W_2) \oplus (Y_2W_1) \oplus (X_1W_1) \oplus (X_1W_2) \oplus (X_2W_1)\\
b_{12} &= X_2 \oplus Y_3 \oplus (Y_2W_2) \oplus (Y_2W_3) \oplus (Y_3W_2) \oplus (X_2W_2) \oplus (X_2W_3) \oplus (X_3W_2)\\
b_{13} &= X_3 \oplus Y_1 \oplus (Y_3W_3) \oplus (Y_3W_1) \oplus (Y_1W_3) \oplus (X_3W_3) \oplus (X_3W_1) \oplus (X_1W_3)\\
b_{21} &= Z_1 \oplus (Z_1X_2) \oplus (Z_2X_1) \oplus (Y_1X_2) \oplus (Y_2X_1) \oplus (Z_1X_1) \oplus (Y_1X_1)\\
b_{22} &= Z_2 \oplus (Z_2X_3) \oplus (Z_3X_2) \oplus (Y_2X_3) \oplus (Y_3X_2) \oplus (Z_2X_2) \oplus (Y_2X_2)\\
b_{23} &= Z_3 \oplus (Z_1X_3) \oplus (Z_3X_1) \oplus (Y_1X_3) \oplus (Y_3X_1) \oplus (Y_3X_3) \oplus (Z_3X_3)\\
b_{31} &= X_1 \oplus W_2 \oplus (Z_1W_1) \oplus (Z_1W_2) \oplus (Z_2W_1) \oplus (X_1Z_1) \oplus (X_1Z_2) \oplus (X_2Z_1)\\
b_{32} &= X_2 \oplus W_3 \oplus (Z_2W_2) \oplus (Z_2W_3) \oplus (Z_3W_2) \oplus (X_2Z_2) \oplus (X_2Z_3) \oplus (X_3Z_2)\\
b_{33} &= X_3 \oplus W_1 \oplus (Z_3W_3) \oplus (Z_3W_1) \oplus (Z_1W_3) \oplus (X_3Z_3) \oplus (X_3Z_1) \oplus (X_1Z_3)
\end{aligned}}$$

4.2 Decomposition for CA-based S-Box Class $(1, 3, 1)$

We now illustrate a decomposition of the representative CA-rule for the S-Box class $(1, 3, 1)$. Once again, while the original rule f has algebraic degree 3, each of the decomposed functions b_1, b_2, and b_3 have degree 2.

$$\boxed{\begin{array}{r}
\boldsymbol{f = YZW \oplus XZ \oplus YW \oplus YZ \oplus X}\\
\hline
\\
b_1(X,Y,W) = X \oplus YW \qquad\qquad\qquad\qquad\\
b_2(X,Y,Z) = YZ \oplus YW \qquad\qquad\qquad\quad\\
f(X,Y,Z,W) = b_2 \oplus b_1W \oplus X = b_3(b_1,b_2,X,W)
\end{array}}$$

We now present a uniform three-sharing for the decomposed functions b_1, b_2, and b_3. The nomenclature of the shares for the various input variables and decomposed functions is the same as described above.

Table 4: Hardware overhead of highly optimized composite TI of CA-Based S-Boxes and Comparison with popular S-Boxes

S-Box		**Area (GE)**	**Dynamic Power (μW)**
CA-Based	**Class**		
	(1,2,2)	212.61	170.2
	(1,3,1)	140.62	113.3
GIFT		217.57	207.75
PRESENT		278.00	237.4
Skinny		321.24	282.3
Piccolo		324.75	281.1
Noekeon		348.54	298.1
Midori		367.29	331.5
Prince		475.55	411.8

$$\begin{aligned}
b_{11} &= X_1 \oplus (Y_1W_2) \oplus (W_1Y_2)\\
b_{12} &= X_2 \oplus (W_2W_3) \oplus (Y_2Y_3)\\
b_{13} &= X_3 \oplus (Y_3W_1) \oplus (Y_3W_1) \oplus (Y_1W_3) \oplus (Y_3W_1)\\
b_{21} &= (Z_1Y_2) \oplus (Z_2Y_1) \oplus (W_1Y_2) \oplus (W_2Y_1) \oplus (Z_1Y_1) \oplus (W_1Y_1) \oplus (Z_1W_1) \oplus (Z_2W_2)\\
b_{22} &= (Z_2Y_3) \oplus (Z_3Y_2) \oplus (W_2Y_3) \oplus (W_3Y_2) \oplus (Z_2Y_2) \oplus (W_2Y_2) \oplus (Z_2W_2) \oplus (Z_3W_3)\\
b_{23} &= (Z_1Y_3) \oplus (Z_3Y_1) \oplus (W_1Y_3) \oplus (W_3Y_1) \oplus (W_3Y_3) \oplus (Z_3Y_3) \oplus (Z_3W_3) \oplus (Z_1W_1)\\
b_{31} &= (Z_1b_{12}) \oplus (b_{11}Z_2) \oplus (b_{21}) \oplus (X_1)\\
b_{32} &= ((Z_2 \oplus Z_3)(b_{12} \oplus (b_{13}))) \oplus (b_{22}) \oplus (X_2)\\
b_{33} &= (Z_1b_{13}) \oplus (b_{11}Z_3) \oplus (b_{13}b_{11}) \oplus (b_{23}) \oplus (X_3)
\end{aligned}$$

4.3 Hardware Results for Composite TI of CA-based S-Boxes

In this section, we compare the area and power requirements of the composite TI circuits described above. We also compare these results with composite TI for all the other lightweight S-Boxes mentioned in the previous section. The architecture for the composite TI circuit is illustrated in Figure 4.1. As mentioned before, the counter bits need not be shared as they are independent of all other intermediate share values. For the PRESENT S-Box, we implement the same composite TI circuit reported in [PMK+11], while for all the other S-Boxes (GIFT, Skinny, Piccolo, Noekeon, Midori, and Prince) we present new results for composite TI that have not been reported in existing literature. The comparison presented in Table 4 reveals that the smallest composite TI circuit among CA-based S-boxes has the smallest area footprint and consumes lowest power. In fact, our CA-based has a 35.36% smaller area-footprint and consumes 44.46% less power as compared to the highly optimized composite TI of the GIFT S-Box, which is the best among all the existing lightweight S-Boxes.

4.4 Side-channel Leakage Resistance Evaluation using TVLA

We conclude this section by presenting a side-channel evaluation of the best TI circuit among all CA-based S-Boxes, corresponding to the representative CA rule for the class $(1, 3, 1)$. The evaluation was performed by implementing the TI circuit on a Virtex-5 FPGA on a SASEBO-GII board. The programming file for our design was generated using Xilinx ISE 14.7; the "Keep Hierarchy" constraint was kept on while generating the programming file in order to prevent optimizations over module boundaries. We

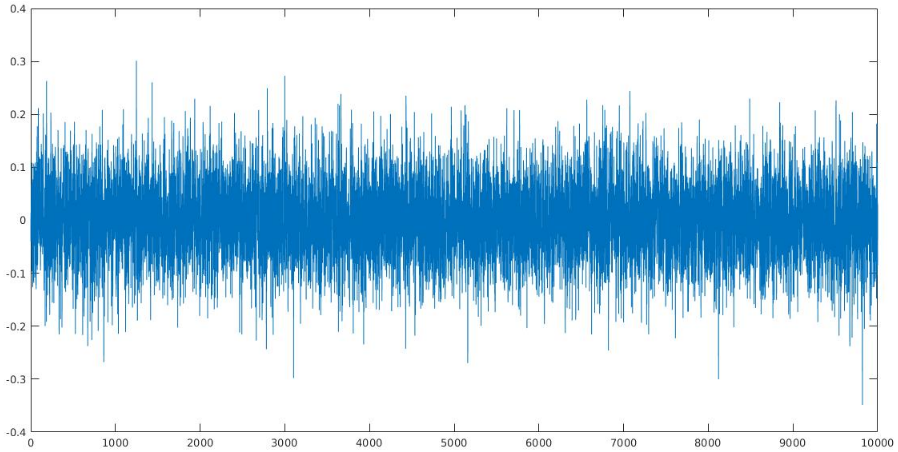

Figure 4.2: TVLA of Composite-TI circuit for CA-Based S-Box representing class $(1, 3, 1)$

collected 1 000 000 power trace samples from the target FPGA device, and performed a fixed-versus-random statistical test vector leakage assessment (TVLA) test on these collected traces. The fixed class for the test was chosen as the all-zero input in all our evaluations. Figure 4.2 demonstrates the result of the TVLA analysis on the power traces. The outcome of the statistical test consists of values in the range $(-0.4, 0.3)$, which is well within the permissible range of $(-4.5.4.5)$ [SM15].

5 Area and Power Efficient Threshold Implementations for SPN Block Ciphers

In this section, we provide a brief discussion on lightweight TI designs for the other major component of an SPN block cipher, namely, the linear diffusion layer. We then discuss how our CA-based S-Boxes may be combined with such diffusion layers to achieve lightweight TI circuits for full block ciphers.

5.1 Lightweight TI circuits for Linear Diffusion Layers

Popular diffusion layer choices in SPN block ciphers include bit-permutation (as in PRESENT and GIFT), MixColumns using MDS matrices (as in AES [DR00]), and MixColumns using almost-MDS matrices (as in Midori [BBI^{+}15]). Of these, MDS matrices are typically avoided in ciphers targeting lightweight applications owing to their high area footprints and power requirements. Bit permutations are obviously the most efficient choice for hardware implementations, since they have the minimal area footprint and power consumption. However, bit-permutation based block ciphers require greater number of rounds to achieve security against standard cryptanalytic attacks. Almost-MDS matrices constitute a somewhat intermediate alternative, in the sense that they lead to slightly more expensive implementations, but provide better throughput by reducing the number of required rounds. In this section, we compare the area and power requirements for TI circuits of bit permutations and almost-MDS matrices:

TI for Bit Permutations. As a bit permutation is essentially simple wiring of bits, and does not require any mathematical operations, there is no extra overhead for TI of bit permutation. Note that permutation layers like Shift-Row (used in AES) or Shuffle-Cell (used in Midori) are essentially bit permutations, and hence no additional overhead is required during TI design of these operations (see Table 5).

MixColumns using Almost-MDS. Another lightweight choice for obtaining diffusion is MixColumns operation using almost-MDS matrices. Following is the most lightweight 4×4 almost-MDS matrix:

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

This matrix is used in the block cipher Midori. We implemented a TI circuit for multiplying a 4×1 state vector with the aforementioned matrix. Note that a straightforward TI circuit must protect 8 XOR gates (two per row of the matrix). In our implementation, we reduce the overhead to 7 XOR gates as follows: we first compute the XOR of all input vector elements (this requires 3 XORs), and then XOR one element per row to obtain the desired output. The area and power requirements for the same are reported in Table 5.

Table 5: TI circuits for diffusion layer choices (ASIC Technology: 180nm)

Diffusion Layer	Area (GE)	Dynamic Power (mW)
Bit Permutation	3.15	0
Almost-MDS	213.47	1.47

5.2 Combining it All Together

In this section, we propose two design paradigms for combining the CA-based optimal S-Boxes with the aforementioned diffusion layer choices to achieve SPN block ciphers with low-area and low-power TI circuits. The first of these paradigms focuses only on optimizing area and power of the TI circuit, without caring for the throughput. In the second design paradigm, we also incorporate the throughput as an additional performance criteria for the TI circuit.

5.2.1 Design Paradigm-1: Focus on Area and Power Only

In this design paradigm, we adopt the SPN block cipher structure of GIFT (which is conceptually identical to that of PRESENT), in the sense that a layer of n 4×4 S-Boxes (typically, $n = 16$) is followed by a bit permutation layer. The S-Box is chosen to be one of the two CA-based S-Boxes (corresponding to classes $(1, 2, 2)$ and $(1, 3, 1)$), or is the original GIFT/PRESENT S-Box. Note that these CA-based S-Boxes (i) have branch number equal to 2 and (ii) do not posses the BOGI (Bad Output Good Input) property[1] defined in [BPP+17]. This observation essentially tells us that, to sustain against linear and differential cryptanalysis, the number of rounds required for an SPN block cipher using our CA-based S-Box with bit permutations would be considerably higher than an equivalent cipher using the GIFT/PRESENT S-Box with bit permutations. More specifically, to achieve linear and differential probability less than 2^{-80} (assuming 80 bit key size as used in PRESENT), we would require 40 rounds. This is due to the fact that in 40 rounds, there is at least 40 many active S-Boxes and the maximum differential probability of the S-Box is 2^{-2}.

We note that the aforementioned derivation of the number of rounds is an estimation based solely on the resistance of the cipher against linear and differential analysis. In order to achieve security against other advanced cryptanalytic techniques, additional rounds

[1] In fact, all 4×4 CA-based optimal S-Boxes have branch number 2 and none of them possess BOGI property.

may be necessary. However, such additions would primarily affect the throughput of the design rather than the area or power consumption. This does not violate the principles the first design paradigm, which primarily targets efficiency in terms of area and power, without much restrictions on throughput. In other words, our CA-based S-Boxes act as viable alternatives to the GIFT/PRESENT S-Boxes in applications where area and power consumption are the primary targets for optimization.

Following the implementation results summarized in Table 6[2], one can observe that the area requirement and power consumption for SPN block ciphers with CA-based S-Box representing class $(1, 3, 1)$ and bit-permutation is optimal in this design paradigm.

Table 6: Lightweight TI for SPN block cipher: area and power (ASIC Technology: 180nm)

S-Box		**Diffusion Layer**	**Area (GE)**			**Power (mW)**
	Class		**16 S-Boxes**	**Diffusion Layer**	**Total**	
CA-Based	$(1, 2, 2)$	Bit permutation	3 401.76	3.15	3 404.91	2.72
		Almost-MDS		216.62	3 618.38	4.19
	$(1, 3, 1)$	Bit permutation	2 249.92	3.15	2 253.07	1.81
		Almost-MDS		216.62	2 466.54	3.28
PRESENT		Bit permutation	4 448.00	3.15	4 451.15	3.79
GIFT		Bit permutation	3 481.12	3.15	3 484.27	3.32
Skinny		Almost-MDS	5 139.84	216.62	5 356.46	5.99
Midori		Almost-MDS	5 876.64	216.62	6 093.26	7.35

Table 7: Area, Power, and Throughput Comparison for TI of SPN block cipher across different choices of S-Boxes and design paradigms (ASIC Technology: 180nm).For throughput calculation, we have used the following: (i) the number of clock cycles required for CA-based S-Box is 8, where as for all the other S-Boxes it is 2; (ii) the critical operating frequencies for CA-based S-Box and the PRESENT S-Box are 526.2 MHz and 476 MHz, respectively; all other S-Boxes have critical operating frequencies close to 500 MHz.

S-Box	**Diffusion**	**Rounds**	**Area (GE)**	**Power (mW)**	**Throughput (MBps)**
CA-Based $(1, 3, 1)$	Bit Permutation	≥ 40	2 253.07	1.81	≤ 17.54
CA-Based $(1, 3, 1)$	Almost-MDS	16	2 466.54	3.28	43.85
PRESENT	Bit Permutation	31	4 448.00	3.79	61.41
GIFT	Bit Permutation	28	3 484.27	3.32	71.42
Skinny	Almost-MDS	32	5 356.46	5.99	62.50
Midori	Almost-MDS	16	6 093.26	7.35	125.00

5.2.2 Design Paradigm-2: Focus on Area and Power with Reasonable Throughput

In this design paradigm, we adopt an SPN block cipher structure with the following design choices:

- We use standard bit permutations in conjunction with the S-Boxes of PRESENT and GIFT.
- We use a standard bit permutation *followed by a MixColumns operation using an almost-MDS matrix* in conjunction with our CA-based S-Boxes, and the S-Box of Midori and Skinny.

Note that use of MixColumns operation with an almost-MDS matrix achieves significant diffusion in each round, ensuring a significant reduction in the number of rounds (and hence, an improved throughput) as compared to the previous design paradigm. If we use

[2] In Table 6, we restrict the comparison of our proposed CA-based optimal S-Boxes with the S-Boxes for PRESENT, GIFT, Skinny and Midori. The remaining candidate S-Box choices, namely Piccolo, Noekeon and PRINCE, are either significantly more area consuming or are parts of block ciphers that do not adhere to the SPN design paradigm.

the same bit permutation and almost-MDS matrix as used in Midori, exactly 16 rounds would be sufficient to achieve the desired security. This analysis essentially follows from the analysis of Midori itself, which has 16 rounds, uses an S-Box with identical branch number (= 2), linear and differential characteristics as our $(1, 3, 1)$ S-Box, and the same almost-MDS matrix. Hence, in this case, it is natural to expect that 16 rounds would provide the same cryptanalytic resistance as Midori. Following Table 7[3], we observe that block ciphers with CA-based S-Box representing class $(1, 3, 1)$ and bit-permutation followed by almost-MDS MixColumns, retain a reasonable throughput of 43.85 MBps, which is comparable with the throughputs of PRESENT and GIFT (61.41 MBps and 71.42 MBps respectively). On the other hand, even though the CA-based S-Box is used in conjunction with the almost-MDS matrix, the area and power savings from the choice of S-Box make up for the additional overhead due to the MixColumns layer. In fact, the overall area requirement for this CA-based S-Boxes with almost-MDS MixColumns as diffusion is 2 466.54 GE, which is lowest among all the constructions considered here.

5.3 Scope for Non-optimal CA-Based S-Boxes: An Exploration

In the aforementioned analysis, we have primarily focused on CA-based S-Boxes that have optimal cryptographic properties with respect to their nonlinearity and differential uniformity. Cryptographic optimality is typically essential for good diffusion: intuitively, using an optimal S-Box in a block cipher construction (as opposed to a non-optimal one) reduces the overall number of rounds required to achieve the desired linear and differential probabilities. This often outweighs the potential area savings afforded by non-optimal S-Box variants. An exception to this intuitive rule is the GIFT S-Box [BPP+17], which is non-optimal yet allows high throughput, while also being significantly more lightweight as compared to the PRESENT S-Box. The reason for this is the existence of a unique BOGI permutation that compensates for the non-optimality of the GIFT S-Box itself. To explore similar possibilities with respect to CA-based S-Boxes, we explored each of the 1 024 possible non-optimal bijective 4×4 CA-based S-Boxes. Our exploration led to the following observations:

- Out of the 1 024 non-optimal bijective CA-based S-Boxes, 112 S-Boxes have comparable area overhead with the most lightweight candidate among their optimal counterparts.

- Each of the 1 024 non-optimal bijective CA-based S-Boxes lacks in strong cryptographic properties. To be more specific, either these S-Boxes have nonlinearity 0 or 2 (which is highly undesirable) or linear and differential characteristics greater than or equals to $2^{-1.414}$.

- Finally, and most crucially, *none* of the 1 024 non-optimal CA-based S-Boxes exhibit the BOGI property of the GIFT S-Box.

From the aforementioned observations, we conclude that with respect to CA-based S-Boxes, optimality is an essential criteria with respect to both cryptanalytic resistance and throughput. In other words, non-optimal CA-based S-Boxes seem to offer no benefits over their optimal counterparts.

[3] In Table 7, we again restrict the comparison of our proposed CA-based optimal S-Boxes with the same set of S-Boxes considered in Table 6, namely PRESENT, GIFT, Skinny and Midori. The reason for this restriction has been discussed earlier.

6 Conclusions and Discussions

In this paper, we present highly optimized TI circuits for cryptographically optimal 4 × 4 S-Boxes, obtained from CA rules. We classify such CA-based S-Boxes into 12 categories based on their amenability to low-area and low-power TI, and present direct-sharings for representative S-Boxes from the each class. The architecture for our implementation direct-shares the local CA rule, and iterates over the same to obtain SCA resistant S-Box implementations. Subsequently, we reduce the number of shares further via functional decomposition of CA-rules, to obtain composite TI-circuits with even lower area footprint and power consumption. Our implementation results on ASIC (180nm technology) show that the most lightweight TI circuit among all CA-based S-boxes has a 49.42% smaller area-footprint and consumes 52.3% less power as compared to the best-known TI of the PRESENT S-Box. The same TI circuit also leads to a 35.36% smaller area-footprint and consumes 44.46% less power as compared to a highly optimized TI of the GIFT S-Box. Finally, this TI circuit also passes the TVLA test over 1 000 000 power traces.

Subsequently, we present TI circuits for bit permutations and MixColumns using almost-MDS matrices, with hardware results naturally favoring the former for lightweight applications. We finally present design paradigms for SPN block ciphers that combine TI circuits for our CA-based S-Boxes with TI circuits for bit permutations (and optionally, for MixColumns operations) for full-fledged side-channel resistance. In particular, the use of TI-protected MixColumns operation offers a practical trade-off between area and power savings, and reasonable throughput requirements.

An apparent disadvantage inherent to any CA-based S-Box design strategy is the reduction in throughput due to its iterative nature. One possible workaround is to operate the target device at higher clock frequencies, keeping in mind that local CA rules are usually simple combinatorial circuits, and hence afford designs with higher critical frequencies. Additionally, with respect to TI circuits, iterative architectures seem to minimize the possibility of additional leakages resulting from correlations among the output bits, since they are processed in different clock cycles. A more thorough exploration of the pros and cons of such iterative S-Box design principles can be an interesting direction of future work. Extensions of our design principles to TI circuits for 5 × 5 and 8 × 8 S-Boxes seem to be an intriguing direction of future research.

Acknowledgements. We would like to thank Carlos Cid for his invaluable comments and suggestions in preparing the final draft. We would also like to thank all the anonymous reviewers of FSE 2019 for helping improve the work. Debdeep would also like to thank his DST Swarnajayanti fellowship (2015-16) for partial support. He would also like to thank DRDO, India for funding the project, “Secure Resource - constrained communication Framework for Tactical Networks using Physically Unclonable Functions (SeRFPUF)" for partially supporting the research. He would also like to thank Information Security Education Awareness (ISEA), DIT, India for encouraging research in the area of computer security. Sikhar would like to thank Qualcomm India Innovation Fellowship 2017-18.

References

[AP72] Serafino Amoroso and Yale N. Patt. Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures. *Journal of Computer and System Sciences*, 6(5):448–464, 1972.

[BBI$^+$15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In *Proceedings, Part II, of the 21st International*

Conference on Advances in Cryptology — ASIACRYPT 2015 - Volume 9453, pages 411–436, New York, NY, USA, 2015. Springer-Verlag New York, Inc.

[BCG+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, pages 208–225, 2012.

[BDPA06] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Radiogatún, a belt-and-mill hash function. *IACR Cryptology ePrint Archive*, 2006:369, 2006.

[BDPA11] Guido Bertoni, Joan Daemen, Michäel Peeters, and Gilles Van Assche. The Keccak reference, January 2011. http://keccak.noekeon.org/.

[BGN+14] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Higher-order threshold implementations. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 326–343. Springer, 2014.

[Bil15] Begül Bilgin. Threshold implementations: as countermeasure against higher-order differential power analysis. 2015.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 123–153, 2016.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In *CHES 2007*, pages 450–466, 2007.

[BPP+17] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. Gift: A small present. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems – CHES 2017*, pages 321–345, Cham, 2017. Springer International Publishing.

[BS91] Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. In *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '90, pages 2–21, London, UK, UK, 1991. Springer-Verlag.

[Car10a] Claude Carlet. Boolean Functions for Cryptography and Error Correcting Codes. In Yves Crama and Peter L. Hammer, editors, *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, pages 257–397. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[Car10b] Claude Carlet. Vectorial Boolean Functions for Cryptography. In Yves Crama and Peter L. Hammer, editors, *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, pages 398–469. Cambridge University Press, New York, USA, 1st edition, 2010.

[CDGP93] L. Claesen, J. Daemen, M. Genoe, and G. Peeters. Subterranean: A 600 Mbit/sec cryptographic VLSI chip. In *Computer Design: VLSI in Computers and Processors, 1993. ICCD '93. Proceedings., 1993 IEEE International Conference on*, pages 610–613, Oct 1993.

[CJRR99] Suresh Chari, Charanjit Jutla, Josyula Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *Advances in Cryptology—CRYPTO'99*, pages 791–791. Springer, 1999.

[DC98] Joan Daemen and Craig S. K. Clapp. Fast Hashing and Stream Encryption with PANAMA. In *Fast Software Encryption, 5th International Workshop, FSE '98, Paris, France, March 23-25, 1998, Proceedings*, pages 60–74, 1998.

[DGV94] Joan Daemen, René Govaerts, and Joos Vandewalle. A new approach to block cipher design. In Ross Anderson, editor, *Fast Software Encryption: Cambridge Security Workshop Cambridge, U. K.,1993 Proceedings*, pages 18–32, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 293–302. IEEE, 2008.

[DR00] Joan Daemen and Vincent Rijmen. Rijndael for AES. In *AES Candidate Conference*, pages 343–348, 2000.

[DRS$^+$12] François Durvaux, Mathieu Renauld, François-Xavier Standaert, Loic van Oldeneel tot Oldenzeel, and Nicolas Veyrat-Charvillon. Cryptanalysis of the ches 2009/2010 random delay countermeasure. *IACR Cryptology ePrint Archive*, 2012:38, 2012.

[GDC17] Ashrujit Ghoshal and Thomas De Cnudde. Several masked implementations of the boyar-peralta aes s-box. In *International Conference in Cryptology in India*, pages 384–402. Springer, 2017.

[GP99] Louis Goubin and Jacques Patarin. Des and differential power analysis the "duplication" method. In *Cryptographic Hardware and Embedded Systems*, pages 728–728. Springer, 1999.

[GPQ11] Laurie Genelle, Emmanuel Prouff, and Michaël Quisquater. Thwarting higher-order side channel analysis with additive and multiplicative maskings. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2011: 13th International Workshop, Nara, Japan, September 28 – October 1, 2011. Proceedings*, pages 240–255, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[JDR00] Gilles Van Assche Joan Daemen, Michael Peeters and Vincent Rijmen. The NOEKEON Block Cipher. 2000. http://gro.noekeon.org/Noekeon-spec.pdf.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *Proceedings of CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer-Verlag, 1999.

[LP07] G. Leander and A. Poschmann. On the Classification of 4 Bit S-Boxes. In Claude Carlet and Berk Sunar, editors, *Arithmetic of Finite Fields*, volume 4547 of *Lecture Notes in Computer Science*, pages 159–176. Springer Berlin Heidelberg, 2007.

[MBTM17] Kerry A. McKay, Larry Bassham, Meltem Sönmez Turan, and Nicky Mouha. Report on Lightweight Cryptography. 2017. http://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.pdf.

[MPC04] Willi Meier, Enes Pasalic, and Claude Carlet. Algebraic attacks and decomposition of boolean functions. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, pages 474–491, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[MPLJ17] Luca Mariot, Stjepan Picek, Alberto Leporati, and Domagoj Jakobovic. Cellular automata based s-boxes. Cryptology ePrint Archive, Report 2017/1055, 2017. https://eprint.iacr.org/2017/1055.

[MPO05] Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully attacking masked aes hardware implementations. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 157–171. Springer, 2005.

[MY93] Mitsuru Matsui and Atsuhiro Yamagishi. A new method for known plaintext attack of FEAL cipher. In *Proceedings of the 11th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'92, pages 81–91, Berlin, Heidelberg, 1993. Springer-Verlag.

[NRR06] Svetla Nikova, Christian Rechberger, and Vincent Rijmen. Threshold implementations against side-channel attacks and glitches. In *International Conference on Information and Communications Security*, pages 529–545. Springer, 2006.

[Nyb93] Kaisa Nyberg. On the construction of highly nonlinear permutations. In RainerA. Rueppel, editor, *Advances in Cryptology - EUROCRYPT' 92*, volume 658 of *Lecture Notes in Computer Science*, pages 92–98. Springer Berlin Heidelberg, 1993.

[Nyb94] Kaisa Nyberg. S-boxes and round functions with controllable linearity and differential uniformity. In *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, pages 111–130, 1994.

[PMK+11] Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling. Side-channel resistant crypto for less than 2,300 ge. *Journal of Cryptology*, 24(2):322–345, 2011.

[PMY+17] Stjepan Picek, Luca Mariot, Bohan Yang, Domagoj Jakobovic, and Nele Mentens. Design of s-boxes defined with cellular automata rules. In *Proceedings of the Computing Frontiers Conference, CF'17, Siena, Italy, May 15-17, 2017*, pages 409–414, 2017.

[RBN+15] Oscar Reparaz, Begül Bilgin, Svetla Nikova, Benedikt Gierlichs, and Ingrid Verbauwhede. Consolidating Masking Schemes. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 764–783, 2015.

[RP10] Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of aes. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010: 12th International Workshop, Santa Barbara, USA, August 17-20, 2010. Proceedings*, pages 413–427, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[Saa12] Markku-Juhani O. Saarinen. Cryptographic analysis of all 4 x 4-bit s-boxes. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography*, pages 118–133, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[SIH$^+$11] Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: An ultra-lightweight blockcipher. In *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, pages 342–357, 2011.

[SM15] Tobias Schneider and Amir Moradi. Leakage assessment methodology. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 495–513. Springer, 2015.

[Sut91] Klaus Sutner. De bruijn graphs and linear cellular automata. *Complex Systems*, 5(1):19–30, 1991.

[TV04a] K. Tiri and I. Verbauwhede. A logic level design methodology for a secure dpa resistant asic or fpga implementation. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, volume 1, pages 246–251 Vol.1, Feb 2004.

[TV04b] Kris Tiri and Ingrid Verbauwhede. A logic level design methodology for a secure dpa resistant asic or fpga implementation. In *Proceedings of the conference on Design, automation and test in Europe-Volume 1*, page 10246. IEEE Computer Society, 2004.

[Wol83] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of modern physics*, 55(3):601, 1983.

[Wol84a] Stephen Wolfram. Cellular automata as models of complexity. *Nature*, 311(5985):419, 1984.

[Wol84b] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):1–35, 1984.

[ZBL$^+$15] WenTao Zhang, ZhenZhen Bao, DongDai Lin, Vincent Rijmen, BoHan Yang, and Ingrid Verbauwhede. Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms. *Science China Information Sciences*, 58(12):1–15, 2015.