Comparing the Cost of Protecting Selected Lightweight Block Ciphers Against Differential Power Analysis in Low-Cost FPGAs

William Diehl, Abubakr Abdulgadir, Jens-Peter Kaps and Kris Gaj
ECE Department, George Mason University, Fairfax, U.S.A.
{wdiehl, aabdulga, jkaps, kgaj}@gmu.edu

Abstract— **Lightweight block ciphers are an important topic in the Internet of Things (IoT), since they provide moderate security, while requiring fewer resources than AES. Ongoing cryptographic contests and standardization efforts evaluate lightweight block ciphers on their resistance to power analysis side channel attack (SCA), and the ability to apply countermeasures. While some ciphers have been individually evaluated, a large scale comparison of resistance to side channel attack and formulation of the relative cost of implementing countermeasures is difficult, since researchers typically use varied architectures, optimization strategies, technologies, and evaluation techniques. In this research we leverage the t-test leakage detection methodology and an open-source side channel analysis suite (FOBOS) to compare FPGA implementations of AES, SIMON, SPECK, PRESENT, LED, and TWINE, using a choice of architecture targeted to optimize throughput-to-area (TP/A) ratio, for resistance to differential power analysis (DPA). We then apply an equivalent level of protection to the above ciphers using 3-share threshold implementations (TI), and verify improved resistance to DPA. We find that SIMON has the highest TP/A ratio of protected versions, followed by PRESENT, TWINE, LED, AES, and SPECK. However, PRESENT uses the least energy in terms of nJ-per-bit.**

Index Terms— **Cipher, cryptography, encryption, field programmable gate array, side channel attack, countermeasure**

I. Introduction

CRYPTOGRAPHIC services, such as confidentiality, integrity, and authentication, are required in many of the billions of small devices constituting the "Internet of Things" (IoT). Such devices could include cyber-physical sensors and actuators, wireless sensors, biometric devices, driverless cars, etc. These devices are often heavily constrained by size, weight, and power (SWaP), and are often located apart from secure data facilities, and thus, more vulnerable to physical compromise.

Existing standards for cryptographic block ciphers such as DES, Triple-DES (3DES), and AES, are primarily intended for information-intensive applications, and are optimized for throughput and use in high-speed communication protocols. However, with the migration of applications away from mainframe servers and personal computers to embedded and wireless remote devices in the IoT, there is growing emphasis on providing solutions that are less power and resource-intensive at the cost of somewhat relaxed security margins. Many such solutions can be realized using lightweight cryptographic algorithms.

Adversaries who can gain physical access to cryptographic devices can attempt to recover sensitive variables (such as a secret key) through "side-channel attacks" (SCA). While cryptanalytic attacks on well-constructed cryptographic algorithms are generally infeasible using current computing capabilities, real ciphers must still exist on physical devices and are vulnerable to information leakage. Differential power analysis (DPA) is one SCA technique that can be used to target cryptographic implementations (including lightweight block ciphers) to recover sensitive information.

Several current cryptographic contests and standards development projects have targeted improvements in lightweight cryptography. One example is the Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR), currently in Round Three with expected selection of final portfolio in 2018 [1]. The CAESAR committee specified use cases for which candidates would be optimized and ultimately selected during Round Three and the Final Round [2]. One of these use cases is lightweight applications (resource constrained environments), for which desired characteristics include "natural ability to protect against side-channel attacks" [2].

A second example is the National Institute of Standards and Technology's (NIST) Lightweight Cryptography Project, which will develop new recommendations using an open call for proposals to standardize and evaluate algorithms based on several characteristics, including side-channel resistance [3, 4].

In this work, we support the above efforts by measuring the resistance of six secret-key block ciphers to DPA using the t-test leakage detection methodology and the Flexible Open-source workBench fOr Side-channel analysis (FOBOS) [5, 6]. We then apply an equivalent level of protection against 1st order DPA for all six ciphers using threshold implementations (TI)[7], and verify improved resistance to DPA using FOBOS. Next, we evaluate the protected implementations in terms of area, throughput, and throughput-to-area (TP/A) ratio on two FPGAs. Finally, we measure actual power and energy usage during cipher operation on the Spartan 3E FPGA.

TABLE I
Block cipher variants implemented in this research

Cipher	Block Size	Key Size	Rnds	Type
AES-128	128	128	10	SPN
SIMON 96/96	96	96	52	Feistel, ARX
SPECK 96/96	96	96	28	Feistel, ARX
PRESENT-80	64	80	31	SPN
LED-80	64	80	48	SPN
TWINE-80	64	80	36	SPN

978-1-5090-4825-0/17/$31.00 ©2017 IEEE

II. Background and Previous Work

A. Block ciphers implemented in this research

The major characteristics of the block cipher variants implemented in this research are shown in Table I. The reader is referred to [8 – 12] for the detailed specifications of all implemented ciphers. Five of these ciphers are used as cryptographic primitives for authenticated ciphers being evaluated in the CAESAR third round competition, including CLOC-AES, CLOC-TWINE, AES-JAMBU, SIMON-JAMBU, SILC-AES, SILC-PRESENT, and SILC-LED [13, 14].

B. T-test Leakage Detection Methodology

DPA is used to recover sensitive variables, such as all or a portion of a secret key, by statistically comparing differences between observed power measurements (e.g., collected in "power traces"), and the presumed contents of a sensitive intermediate variable, according to a hypothetical power model [15, 16]. However, the authors of [5] recognized that traditional DPA is time- and resource-intensive, in that the attacker must have access to the underlying architecture, and conduct expert analysis (often through trial-and-error) to develop an accurate power model.

In cases where we desire to show that a cryptographic implementation is leaking information, or determine whether or not our power-analysis countermeasures are effective, we can employ an expedited leakage assessment methodology called the t-test. As described in [5, 17], the Welch's t-test determines whether two distributions are different from one another. In contrast to attack-based testing, the t-test finds leakage of information without mounting an attack, does not rely on knowledge of the underlying architecture, and can quickly reveal when information leaks and when a countermeasure has failed. However, it does provide information about the difficulty of mounting an attack, and cannot be used to recover sensitive intermediate values, such as the secret key.

In the Welch's t-test, a figure of merit *t* is calculated as $t = (\mu_0 - \mu_1)/\sqrt{{s_0}^2/n_0 + {s_1}^2/n_1}$, where μ_0 and μ_1 are means of distributions Q_0 and Q_1, s_0 and s_1 are standard deviations, and n_0 and n_1 are the cardinality of the distributions, or the number of samples. *t* also depends on degrees of freedom (v), however, *v* can be omitted in cases of equal cardinality, i.e., $n_0 = n_1$, and the number of samples is sufficiently large, e.g., $n > 1000$.

A t-test is performed on the populations Q_0 and Q_1, which are characterized by normal distributions, and where the probability p of a sample belonging to both Q_0 and Q_1 is calculated as $p = 2\int_{|t|}^{\infty} f(t)dt$, where $f(t)$ is a probability distribution function (pdf). We assume the null hypothesis, i.e., that "samples are from the same population," and that we cannot differentiate between populations. Since $|t|$ is a limit of the definite integral of the two-tailed pdf, we choose a threshold (e.g., $|t| > 4.5$) so that p is sufficiently small (e.g., $p < 10^{-5}$), that we can reject the null hypothesis. If, during our t-test, we encounter points in the time domain (i.e., "samples") where $|t| > 4.5$, we reject the null hypothesis that "the samples are from the same distribution" and conclude that "we can distinguish between Q_0 and Q_1," i.e., "the device is leaking information."

One method of evaluating leakage on a device, before and after application of countermeasures, is the "non-specific t-test." In one type of non-specific t-test, called a "fixed-versus-random" t-test, we preselect some "fixed" sensitive data D (e.g., message). Then we randomly interleave the feeding of D, or random data, to the victim cipher. The power traces collected from the fixed data or random data are used to populate the Q_0 and Q_1 distributions (respectively), upon which the t-test is conducted. We repeat the fixed-versus-random t-test using several distinct data sets, in order to prevent "false positives" or "missed negatives" that can occur during analysis of only one data set [17].

C. Threshold Implementations

Threshold implementations, or TI, are an algorithmic countermeasure against power-analysis side-channel attack. TI are based on secret sharing and multi-party communications, where the communications of a single party cannot be exploited to learn the secret content [19, 20].

TI improve upon traditional Boolean masking in that they provide security in the presence of glitches. Although Boolean masking provides mathematically-secure protection against DPA, it can fail in CMOS technology, since the power change that occurs in a CMOS gate during a transition due to a glitch is relatively large compared to normal operation of a device. Measuring the toggle rate of CMOS glitches has been used to successfully attack a masked version of AES [18].

A threshold implementation must have the following three properties, outlined in [7], to be provably secure against power analysis in the presence of glitches:

1. Non-completeness. Every function is independent of at least one share of each of the input variables. Defined formally, if $c = F(a, b)$ and a and b are divided into d shares, then $c_1 = f_1(a_2, a_3, \ldots, a_d, b_2, b_3, \ldots, b_d)$, $c_2 = f_2(a_1, a_3, \ldots, a_d, b_1, b_3, \ldots, b_d)$, $c_i = f_i(a_{\forall j:j \neq i}, b_{\forall k:k \neq i})$. In other words, If c_i does not depend on a_i and b_i, it cannot leak information about a_i or b_i.

2. Correctness. The sum of the output shares gives the desired output. Formally, $c = \oplus_{i=1}^{d}$, where $c_i = f_i(a, b)$.

3. Uniformity. A realization of sharing $c = F(a, b)$ is uniform if for all distributions of the inputs a and b, the output distribution preserves the input distribution. In other words, if the input function is a permutation, the output function should also be a permutation.

A non-linear function of algebraic degree 2, such as $c = ab$ (e.g., a 2-input and gate), can be shared using three TI shares, since $d + 1$ shares are required to share a function of degree d. However, as discussed in [21, 22], achieving the TI uniformity property is not trivial. This property can be achieved by supplying fresh random bits (e.g., "resharing" or "remasking" randomness), however, this requires the resourcing of sufficient randomness, which must either be imported into the device, or generated internally at run-time. Thus, the decision to use 3-share TI which require an increased number of random bits, or 4-share TI with more required resources but no additional randomness, is an engineering design tradeoff.

D. Our contribution

AES, PRESENT, LED, SIMON and SPECK have been previously protected against differential power analysis using threshold implementations, and the subsequent resistance has been evaluated in ASIC or FPGA [21 – 27]. However, these evaluations are by individual research groups, which implement only the targeted cipher and do not conduct direct measurements of other ciphers. Although these results can be compared to other results in literature, it is more desirable to perform a direct comparison of all ciphers, i.e., implemented by the same hardware designers and evaluated on the same test bench, to eliminate differences in implementer style or choice of hardware. We facilitate a relevant comparison by implementing six block ciphers, protecting each cipher with an identical level of protection to DPA, evaluating the unprotected and protected versions of all ciphers in an identical analysis suite, and comparing ciphers in terms of area, throughput, throughput-to-area (TP/A) ratio, power, and energy.

To our knowledge, we present the first documented, verified, and benchmarked results of 3-share TI-protected implementation of TWINE.

Additionally, whereas most studies focus solely on the increase in resources (e.g., LUTs, slices, gate equivalents, etc.), we select throughput-to-area (TP/A) ratio (i.e., Mbps/LUT) as an evaluation metric. This helps to emphasize the fact that 1) sufficient throughput is a valid but often under-prioritized metric in evaluation of lightweight ciphers, and 2) the maximum clock frequency of FPGAs is significantly affected by additional routing complexity of DPA-protected designs, as well as additional logic contributing to the critical path.

Furthermore, implementations in [21 – 25] utilize "anti-optimization" features to ensure the compiler does not remove protections during synthesis and implementation, but do not discuss the costs of anti-optimization constraints. Our data collected from six protected cipher implementations allow us to characterize the expected degradations in area, throughput, and TP/A ratio on both target FPGAs.

Finally, the t-test leakage detection methodology introduced in [5] and further explained in [17] is designed to provide a less-comprehensive, but far-less time consuming evaluation of side channel leakage. We validate this methodology by providing a large-scale comparison of multiple ciphers which would be enormously difficult using traditional methods of DPA evaluation.

III. Methodology

A. Overview

Our methodology for this research is as follows: 1) We develop implementations for the six ciphers using register transfer level (RTL) methodology in VHDL. In order to maximize throughput-to-area (TP/A) ratio, we use a full-width datapath, basic iterative architecture when possible; 2) We evaluate DPA resistance of the unprotected ciphers using the FOBOS architecture (see below description) and the Welch's t-test leakage detection methodology. Leakage is evaluated using a non-specific "fixed-versus-random" t-test consisting of 2000 high-fidelity (i.e., over 20,000 samples per block encryption) traces, on a custom-modified Spartan 3E FPGA clocked externally at 500 KHz to minimize inductive and capacitive leakage attenuation; 3) We modify victim ciphers to include a maximum of three shares of TI protection (3-share TI), and try to minimize additional required randomness for refreshing and resharing masks; 4) We verify improved DPA resistance of the protected ciphers on FOBOS using the methodology described above. Per the recommendations of [17], we verify the results of the fixed-versus-random t-test with at least two sets of fixed data; 5) We implement all versions on two FPGAs, the Spartan 3E (i.e., used in the FOBOS architecture) and in the Virtex-7 (i.e., a high-end FPGA). Implementations use Xilinx 14.7 ISE. We prevent Block RAM (BRAM) and DSP instantiation in order to ensure a fair comparison between ciphers. Ciphers are compared in terms of area (LUTs), throughput (Mbps), and throughput-to-area (TP/A) ratio; and 6) We measure actual power (mW) for each version on the Spartan 3E FPGA at a fixed frequency of 5 MHz and compute energy-per-bit (nJ/bit) by measuring an amplified voltage across a shunt resistor coupled to the FOBOS test bench.

B. Flexible Open-source workBench fOr Side-channel analysis (FOBOS)

FOBOS is a free and open tool which provides a single "acquisition to analysis" solution to measure resistance to power analysis side-channel attack (SCA) and evaluation of the effectiveness of countermeasures [6]. In this research, we leverage open-source, low-cost hardware, specifically, the Diligent Nexys 2 and Xilinx Spartan 3E FPGA Starter Board.

A complete description of FOBOS capabilities is available at [28]. We start with the baseline FOBOS software suite available at [28], and modify the analysis tool set to perform non-specific t-tests as described above.

C. Cipher-specific 3-share TI protection methodology

1. AES – We start with an implementation of an S-Box using combinational logic, as described in [29, 30]. As the AES polynomial is of degree 7 (with field inversion modulo an 8th-degree polynomial), a direct sharing would require a minimum of 8 shares. An 8-share TI is not feasible, even if such a sharing could be discovered that meets all TI properties (none has been discovered to date). However, using the method of Tower Fields, where inversions in GF(2^8) are represented as operations in GF(2^4), which are in turn represented in GF(2^2), field multiplications and inversions in low-degree non-linear representations become feasible.

We choose not to produce a full-width, basic iterative architecture TI-protected version of AES for the following reasons: 1) Each 8-bit S-Box using Tower Fields requires nine GF(2^2) regular multiplications and three GF(2^2) scaled multiplications, which is enormously costly when implementing multiple S-Boxes; 2) The Tower Fields approach results in multiple cascaded non-linear sharings which could cause long glitch-dependent circuit paths; and 3) Non-linear multiplications do not satisfy TI Property 3 (Uniformity) in that they are not permutations. Therefore, they require mask refreshing during or after every TI-shared calculation. The total fresh randomness required either increases I/O requirements, or increases area if generated on-chip. Therefore, it is better to distribute this requirement over multiple clock cycles.

Therefore, we leverage approaches in [21, 22] to develop a hybrid 8-bit and 32-bit datapath in a pipelined approach. We follow the method of [22] and instantiate only one complete 8-bit S-Box, which is separated into five stages. However, we adopt a method described in [21] to employ a hybrid 2-/3-share TI approach, where linear calculations (such as round key addition, column multiplications, basis conversions, affine transformations, etc.) are conducted on only two shares to save resources.

Our resulting protected design has a 5-stage pipeline, where one S-Box operation commences every clock cycle. A 128-bit round completes every 16 cycles, with one additional cycle occupied by a programmed stall. Therefore, a 128-bit block encryption executes in 175 clock cycles. The design uses 16 bits of fresh randomness for resharing from two to three shares, and two fresh remasking bits per GF(2^2) multiplier and multiplier-scalar instance, resulting in a total of 40 random bits required for each S-Box. The three shares are recombined into two shares at the end of the non-linear chain to reduce resources required for affine transformation, change of basis, 32-bit column multiplications, and round key addition.

2. SIMON – We adopt the SIMON 96/96 full-width implementation with basic iterative architecture available at [14] and modify as necessary for our test methodology. A 3-share threshold implementation (TI) of SIMON is easily achieved using the methodology described in [7] and [25]. SIMON, a member of the ARX (Addition, Rotation, XOR) family of ciphers, uses only a single 2-input 48-bit AND to achieve non-linearity. Therefore, a 3-share TI of this quadratic equation is achieved without requiring any cascading or composite functions.

TI properties 1 (non-completeness) and 2 (correctness) are satisfied, as each share lacks at least one of the component shares in its calculation. The non-linear nature of the round function suggests that the shares are not a permutation, and therefore do not automatically satisfy Property 3 (Uniformity). However, uniformity is satisfied in this case by considering the key shares, included in each TI-share calculation, as a source of randomness [25]. Therefore, no mask refreshing is required in SIMON 3-share TI, which leads to a very efficient TI-protected implementation.

3. SPECK – SPECK, like SIMON, is an ARX cipher with non-linearity provided by addition modulo 2^{48}. Masking additions against DPA is possible using formulas such as $x = x' + r_x \ mod \ 2^n$, where *x'* is a masked variable and r_x is an arithmetic mask. However, SPECK operations contain components that require Boolean masking (e.g., rotations and XORs) in addition to arithmetic masking. Techniques that employ both Boolean and arithmetic masking have been investigated for ciphers using modulo addition, e.g., [31].

While it is possible to apply the above conversion techniques to SPECK, the resulting protected design is likely to be resource intensive and highly complex. Accordingly, we have chosen an alternative approach using only Boolean masking. The authors in [32] describe a technique to achieve a three-share threshold implementation (TI) for a 32-bit adder using the Kogge-Stone adder. First published in [33], the Kogge-Stone adder produces recursive carry "generate" and "propagate" trees. The total number of stages required is $n = [log_2 k] + 1$, where k is number of adder bits (e.g. 48 bits required for SPECK). Therefore, $n = 7$ for SPECK, where the first stage (Stage 0) is a preprocessing stage.

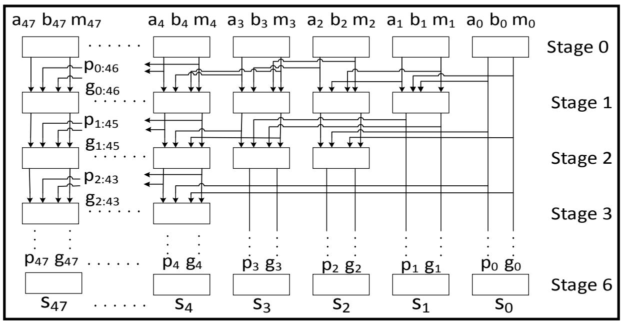

Fig. 1 48-bit 3-share TI-protected Kogge-Stone Adder used in SPECK, where a_i and b_i are operands, p_i are propagation bits, g_i are generation bits, m_i are random masking bits, and s_i are summation bits.

In the Kogge-Stone adder, the largest AND-gate is 2-input. Therefore, the maximum degree d of non-linearity is 2, and the adder can be shared using $d + 1 = 3$ shares. We adopt the TI method as outlined in [32], which requires $k = 48$ bits (denoted m_i) for mask refreshing in the preprocessing stage in order to satisfy the TI uniformity property. However, we also provide mask refresh bits for stages 1 through 6 in order to satisfy the TI uniformity property. The number of mask refresh bits required decreases logarithmically for each stage. The total number of bits required is $k + \sum_{i=1}^{[log_2 k]} k - 2^{i-1}$, or 273 bits for one complete 48-bit addition.

It is infeasible to provide 273 random bits in one clock cycle. Additionally, the performance of seven levels of cascaded non-linear TI operations in one clock cycle risks leaking information through glitch dependencies. Therefore, we adopt a multi-cycle architecture executing in eight clock cycles per round, in which carry chain results are registered after every non-linear computation on shared components. 273 bits amortized over eight clock cycles results in a requirement of 34 bits per clock cycle – a large requirement but at least feasible.

The Kogge-Stone modulo 2^{48} adder, as implemented in SPECK, is shown in Fig. 1. Registers are placed at the output of each stage (including preprocessing Stage 0), and at the end of the round.

4. PRESENT – A 3-share TI-protected version of PRESENT is efficiently achieved using the strategies described in [23, 24]. PRESENT uses a 4-bit S-Box of cubic degree. Therefore, a direct sharing using a minimum of four shares is possible. However, a 3-share version is achieved by defining composite functions F and G such that $F(x) \cdot G(x) = S(x)$, and where F and G are quadratic. Such a composition is $\mathrm{S}(x) = A(G(G(Bx \oplus c) \oplus d)$, where A, B, G, c and d are defined in [23].

We utilize the innovation described in [24] where one reusable function G is defined for all 3-share S-Box computations. However, in keeping with our strategy of full-width implementations using basic iterative architecture, we instantiate all six instances of the function *G*, vice the single instance described in [24].

As discussed in [24], the uniformity property is satisfied for the shared functions *G*, since the output is a permutation on the input. Therefore, no additional randomness is required.

5. LED – A 3-share TI implementation of LED is achieved using the methodology described above for PRESENT, since LED uses the PRESENT S-Box. The only additional consideration for LED is that the PRESENT permutation is essentially "no-cost" in hardware," whereas linear transformations conducted in LED (e.g., MixColumnsSerial) are costly. Therefore, there is a tradeoff to consider in using a

hybrid 2-/3-share structure as documented for AES in order to reduce the number of matrix multiplier instances. However, this would require addition of random bits for resharing, whereas LED would otherwise require no random bits. Therefore, we maintain a strict 3-share TI-protected LED and accept the cost of instantiating three matrix multipliers for our full-width basic iterative architecture.

6. TWINE – TWINE uses a 4-bit S-Box based on a cubic function (i.e., $d = 3$) and is designed using the same strategy as the AES S-Box, i.e., a field inversion followed by an affine transformation. The S-Box is defined as $S(x) = A(x \oplus b)^{-1} mod\ p$, where A, p, and b are defined in [12].

To achieve a three-share TI we employ a strategy previously used for AES in [34]. According to Fermat's Little Theorem (FLT), $a^p \equiv a\ mod\ p$, and $a^{p-2} \equiv a^{-1}\ mod\ p$. In this case, we can compute $x^{14} \equiv x^{-1}$ in GF(2^4). This conveniently decomposes into two cascaded multipliers of quadratic order, which enables our three-share TI. The FLT inverter also uses three squares per share, but the squares are nearly free (e.g. two XOR gates) and are linear operators.

In contrast to PRESENT and LED, the cascaded multipliers on GF(2^4) are not permutations – they do not satisfy the TI uniformity property. Refreshed masking is required at each of the two levels to ensure this property. Uniformity is achieved with one random bit per 4-bit multiplier, for a total of two bits per S-Box and 16 bits per clock cycle in a basic iterative architecture. The three-share FLT inverter as applied in TWINE is shown in Fig. 2.

D. *Assumptions and Simplifications*

We adopt several assumptions and simplifications: 1) Any required round keys are computed "on-the-fly;" 2) Only the encryption case is implemented, as use of the encryption mode is often sufficient to implement both encryption and decryption in an authenticated cipher based on a given block cipher; 3) Only round functions are masked; key scheduling is not masked (with the exception of SIMON, where key sharing is required to achieve uniformity and is relatively low cost). As discussed in [24], a relevant comparison of ciphers is achieved without key masking; 4) Randomness is simulated by ingesting a large number of random bits (e.g., 256 bits for AES) and reusing them after rotations by prime numbers (such as 43 or 61 bits), since an integrated PRNG would require significant additional resources. This assumption of randomness does not affect our tests for a short number of total clock cycles (i.e., 30 – 250 cycles) but is not secure for long-term cipher operation.

IV. Results

A. *Side channel resistance of unprotected versions*

The t-test graphical results for the unprotected implementations of AES, SIMON, SPECK, PRESENT, LED, and TWINE are shown in Figs 3a – 8a, respectively, where time-domain (samples 1 through 20,000) are on the horizontal axis, and t-values are on the vertical axis; $t = \pm 4.5$ are shown by the horizontal lines. All unprotected ciphers fail the t-test, since t-correlation values of $|t| > 4.5$ appear at multiple sample values in each case.

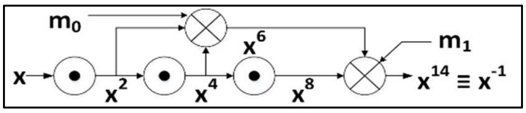

Fig. 2 3-share TI GF(2^4) inverter in TWINE, with 1-input squares and 2-input multipliers; All signals are triplicated for 3-share TI; m_0 and m_1 are random bits. Bus widths are 4 bits, except for m_0 and m_1, which are a single bit.

B. *Successful 3-share TI protected ciphers*

The ciphers, protected against 1st order DPA using 3-share threshold implementations (TI) as described above, were retested using the same t-test methodology. Randomness for initial masking is externally generated in software. SIMON, PRESENT, LED, and TWINE achieved satisfactory t-tests using full-width basic iterative architecture. Their results are shown in Figs. 4b and 6b – 8b, respectively. We do not achieve full-width basic iterative architecture protected versions of AES and SPECK. The results for the AES 5-stage pipelined version and the SPECK 8-cycle-per-round multi-cycle version are shown in Figs. 3b and 5b, respectively.

C. *Benchmarking of results*

Table II shows the results of benchmarking of the unprotected version of the ciphers in this study. Correct results are verified both in simulation (using Xilinx iSim) and by

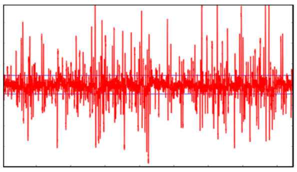
Fig. 3(a) Unprotected AES

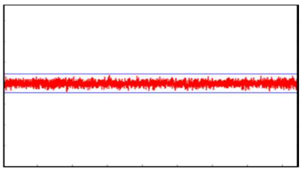
Fig. 3(b) 2-/3- share TI AES

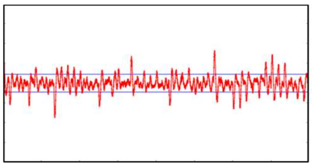
Fig. 4(a) Unprotected SIMON

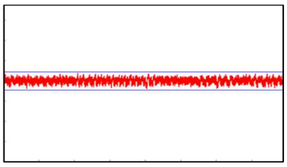
Fig. 4(b) 3-share TI SIMON

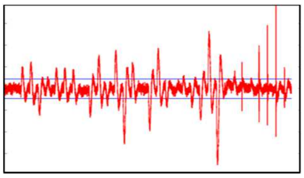
Fig. 5(a) Unprotected SPECK

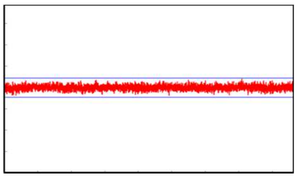
Fig. 5(b) 3-share TI SPECK

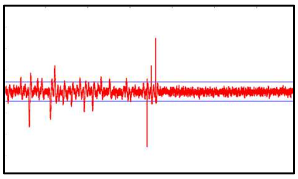
Fig. 6(a) Unprotected PRESENT

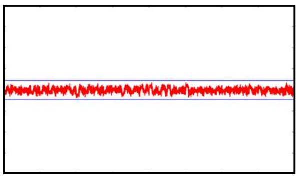
Fig. 6(b) 3-share TI PRESENT

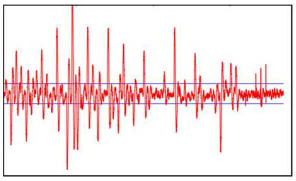
Fig. 7(a) Unprotected LED

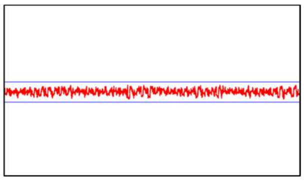
Fig. 7(b) 3-share TI LED

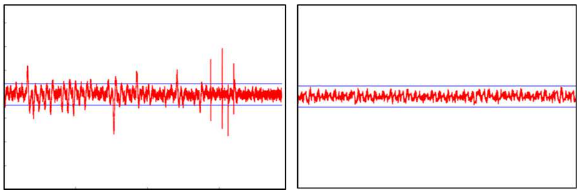

Fig. 8(a) Unprotected TWINE Fig. 8(b) 3-share TI TWINE

verifying the ciphertext output on actual hardware. The results are generated using Xilinx 14.7 for the Virtex-7 (shown as V7), and the Spartan 3E (shown as S3E). Frequency "Freq" is shown in MHz; Throughput (TP) is shown in Mbps, and throughput-to-area (TP/A) ratio is shown as Mbps/LUT. The rankings are based on results in the Virtex-7, and are in order of lowest area (in LUTs), highest throughput, and highest TP/A ratio.

Table III shows the results of benchmarking of the protected versions of the ciphers that successfully passed the t-test and did not show signs of leakage. Table IV shows average power (consisting of static and dynamic power, in mW) and energy/bit (nJ/bit) for unprotected and protected ciphers as measured by FOBOS across a shunt resistor on the Spartan 3E at 5 MHz.

D. *Cost of anti-optimization constraints*

KEEP constraints prevent nets from being absorbed into adjacent or higher-echelon logic blocks. KEEP HIERARCHY prevents Xilinx XST from attempting to "flatten" a hierarchy of netlists, which could result in optimized results for resource use and reduction of critical path. The Xilinx KEEP constraints are not designed to provide designs that preserve algorithmic countermeasures, but rather are designed to produce better implementation results by permitting modules and blocks to be optimized separately.

However, KEEP can ensure separation of signal paths that are intended by the designer to be logically separate, in order to reduce the possibility of correlation through DPA. We follow the recommendations of [21 – 25] and apply KEEP constraints for all protected versions in this research.

KEEP, however, imposes a cost in terms of area, throughput, and throughput-to-area (TP/A) ratio. In our implementations, the use of anti-optimization constraints causes an average of 22% increase in LUTs, 4% reduction in frequency, and 21% reduction in TP/A ratios in the Virtex-7; and an average of 5% increase in LUTs, 16% reduction in frequency, and 20% reduction in TP/A ratios in the Spartan 3E.

TABLE II
RESULTS OF IMPLEMENTATION OF UNPROTECTED CIPHERS ON VIRTEX-7 (V7) AND SPARTAN 3E (S3E) FPGAS

	Dev	AES	AES	SMN	SPK	PRT	LED	TWN
Arch		Full	Pipl	Full	Full	Full	Full	Full
Area	V7	2620	697	435	385	381	602	302
(LUT)	S3E	2845	1182	565	634	595	727	296
Area	V7	991	253	146	130	133	211	122
(Slice)	S3E	1691	806	403	462	408	486	229
Freq	V7	229	326	624	363	537	309	552
(MHz)	S3E	73	128	176	111	177	116	200
TP	V7	2937	238	1152	1245	1108	411	982
(Mbps)	S3E	934	94	329	380	366	134	355
TP/A	V7	1.12	0.34	2.65	3.24	2.91	0.68	3.25
ratio	S3E	0.33	0.08	0.57	0.61	0.62	0.19	1.2
Rank								
Area	V7	7	6	4	3	2	5	1
TP	V7	1	7	3	2	4	6	5
TP/A	V7	5	7	4	2	3	6	1

Ciphers abbreviated as "SMN" (SIMON), "SPK" (SPECK), "PRT" (PRESENT), "TWN" (TWINE). "Arch" refers to architecture; "Full" (full-width, basic-iterative), "Pipl" (pipelined), or "MC" (multi-cycle). Virtex-7 contain 4 6-input LUT per slice; Spartan 3E contain 2 4-input LUT per slice.

TABLE III
RESULTS OF PROTECTED CIPHERS AS IMPLEMENTED ON VIRTEX-7 (V7) OR SPARTAN 3E (S3E) FPGAS

	Dev	AES	SMN	SPK	PRT	LED	TWN
Arch		Pipl	Full	MC	Full	Full	Full
Area	V7	1791	1520	3328	1317	1691	2573
(LUT)	S3E	2387	2151	4792	1707	2175	2946
Area	V7	902	434	1714	429	928	1256
(Slice)	S3E	1736	1404	3958	1221	1290	1777
Freq	V7	106	456	334	189	145	207
(MHz)	S3E	86	176	108	70	55	67
TP	V7	77	841	143	390	193	367
(Mbps)	S3E	63	326	46	143	73	118
TP/A	V7	0.043	0.553	0.043	0.296	0.114	0.143
ratio	S3E	0.026	0.151	0.010	0.084	0.033	0.040
Rnd bits		40	0	34	0	0	16
Rank							
Area	V7	4	2	6	1	3	5
TP	V7	6	1	5	2	4	3
TP/A	V7	5	1	6	2	4	3

"Rnd bits" indicates number of required random bits per clock cycle for mask resharing, refreshing, and satisfaction of TI uniformity property.

V. ANALYSIS

A. *Analysis of results in this research*

SIMON has the highest throughput-to-area (TP/A) ratio of ciphers protected against 1st order DPA in this research, followed by PRESENT, TWINE, LED, SPECK, and AES. In terms of area, PRESENT is the smallest, followed by SIMON, LED, AES, TWINE, and SPECK; in terms of throughput, SIMON is the highest, followed by PRESENT, TWINE, LED, SPECK, and AES. It is important to consider TP/A ratio as a key performance metric, since 1) changes in cipher protection schemes affect frequency as well as area, and 2) the factors of throughput and area are needed to normalize ciphers that have different block sizes, different architectures, and varying clock cycles per block.

SIMON is particularly well-suited for threshold implementations, since its only non-linearity is the equivalent of a two-input 48-bit AND gate. Therefore, cascading of non-linearity and random mask refreshing bits are not required.

PRESENT has the lowest area and second best TP/A ratio. Additionally, PRESENT has the lowest energy-per-bit and second-lowest average power. The 3-share TI S-Box implementation used in [23] and [24] requires only two cascaded levels of functions, and did not leak information in our t-test, even in a full-width implementation with no random refresh bits. The linear permutation layer of PRESENT is essentially "no cost" in hardware, which saves area in 3-share threshold implementations.

TABLE IV
AVG POWER AND ENERGY-PER-BIT ON SPARTAN 3E FPGA @ 5 MHZ

	AES	SMN	SPK	PRT	LED	TWN
Unprotected						
Avg Pwr (mW)	15.0	13.4	14.0	12.9	15.2	12.9
Energy/bit (nJ/bit)	4.10	1.45	0.82	1.25	2.28	1.45
Protected						
Avg Pwr (mW)	19.2	17.7	19.4	18.4	30.0	27.9
Energy/bit (nJ/bit)	5.25	1.92	9.22	1.78	4.50	3.14

LED uses the same S-Box as PRESENT, and thus has a relatively low masking cost. However, it has a higher area and significantly lower throughput than PRESENT, due to a higher number of rounds required (48 versus 31 for 80-bit keys). Additionally, the linear transformations in LED are more costly than PRESENT. In fact, each instantiation of MixColumnsSerial takes 140 LUTs on the Virtex-7, and must be instantiated three times for a 3-share TI. A recommendation would be to attempt a hybrid 2-/3- sharing, like that used in AES, at the cost of random resharing bits required at runtime.

TWINE comes in third for throughput and TP/A ratios. While the S-Box non-linear GF(2^4) inverter strategy employed is simple, requires only two cascaded non-linear layers, and uses 16 bits of randomness per clock cycle for a validated 3-share TI, it has a relatively large growth in area, especially compared to an optimal unprotected TWINE using 4-bit LUT S-Boxes. A recommendation would be to consider a protected S-Box using the techniques discussed in [29] or [35].

AES trails most lightweight ciphers in terms of relative growth due to protection. This is a function of its 8-bit S-Box, which has a high algebraic degree, and requires four levels of cascaded non-linear functions (fewer levels are possible but require more complex non-linear functions).

Unfortunately, our protected version of SPECK finishes last or nearly last in all categories. This is due to the large cost of masking an adder defined in purely Boolean logic. A recommendation would be to investigate TI-constructions based on arithmetic masking techniques, and alternative adders with higher propagation delay (such as a multi-stage TI-protected carry propagate adder), but lower gate count.

B. *Comparison with previous results*

Techniques and strategies from previous TI-implementations have been considered for adaptation to ciphers in this research. However, direct comparison with previous results is still difficult, since authors adopt different technologies and different optimization strategies, e.g., serial, low-area, etc.

Table V below shows previously reported results in ASIC or FPGA. Growth factor ("ratio") is shown as the ratio of protected to unprotected implementations, in terms of Gate Equivalents (GE) or slices. In some cases authors produce only a protected version, and compare to a previously published unprotected version. All ASIC implementations are compiled at a fixed frequency of 100 KHz; therefore, frequency, throughput, and throughput-to-area ratio are not relevant.

In terms of AES, our 8-bit 5-stage pipelined design has less area growth (i.e., 2.56 times more LUTs when comparing protected 8-bit to unprotected 8-bit pipelined AES), takes fewer clock cycles (175 versus 266 or 256), and uses fewer random refresh bits (40 versus 48 or 44). However, we mask only the status word and not the secret key, which accounts for some of the above savings.

Regarding PRESENT, although we employ a similar S-Box TI-protection strategy (i.e., the 3-share six-function technique used in [23]), we build a full-width basic iterative architecture, in contrast to their serial architectures. Since we invoke 16 S-Boxes per clock cycle, this accounts for the area growth of our protected version of nearly twice that of [23] and [24]. A direct comparison of growth in TP/A ratio is not possible, since the ASIC versions are implemented at a fixed clock frequency which is not representative of the best performance achievable.

TABLE V
PREVIOUS RESULTS OF TI-PROTECTED CIPHERS ADDRESSED IN THIS WORK

	AES	AES	PRT	PRT	SMN	SPK	LED
Width	8	8	4	4	1	1	64
Arch	5	3	S	S	S	S	2
Tech	180	180	180	180	S3E	S3E	180
UnPr	2400	2400	1111	1111	36	43	-
Pr	10793	8171	2282	2105	96	99	20212
Ratio	4.50	3.40	2.05	1.89	2.67	2.30	-
Shares	3	2-/3-	3	3	3	3	3
Rnd	48	44	0	0	0	0	0
Cycl	266	256	547	2996	4835	2048	96
Ref	[22]	[21]	[23]	[24]	[25, 36]	[26]	[27]

"Width is datapath in bits; "Arch" is n-pipelined stages, or "S" (serial); "Tech" denotes FPGA or ASIC (nm); "UnPr" is unprotected, "Pr" is protected; "Ratio" is Pr/UnPr; "Rnd" is random bits; "Cycls" denotes clock cycles. "UnPr" and "Pr" express areas in GE (ASIC) or slices (FPGA).

Regarding SIMON and SPECK, a closer comparison with previous results is possible, since the authors of [25, 26, 36] use the Spartan 3E FPGA. However, the goal of these studies is low area using strictly serial implementations, whereas our goal is optimal TP/A ratio. This explains why their ratios of growth in terms of area for protected versus unprotected are 2.7 (SIMON) and 2.3 (SPECK), which are less than our relative costs of 3.8 (SIMON) and 7.6 (SPECK) on the Spartan 3E.

The implementation of LED at [27] is similar to our full-width 64-bit datapath with 3-share TI-protection, however, it contains additional features to present fault attacks and uses a 128-bit key, and is thus not directly comparable.

VI. CONCLUSION

In this research we performed a comparison of six secret-key ciphers – AES, SIMON, SPECK, PRESENT, LED, and TWINE – in terms of cost of protection against differential power analysis (DPA). We tested resistance to 1st order DPA of the unprotected versions of the above ciphers using the t-test leakage detection methodology on the FOBOS test bench. The results show that unprotected versions of all of the above ciphers failed the t-test and are vulnerable to DPA.

We then leveraged available published and theoretical techniques to produce 3-share threshold implementation (TI)-protected versions of the above ciphers. We verified improved resistance of the protected ciphers to DPA using the t-test leakage detection methodology and the FOBOS test bench.

We then compared the unprotected and protected versions in terms of throughput, area, throughput-to-area (TP/A) ratio, power, and energy. Given an identical level of protection (i.e., 3-share threshold implementation) against DPA, SIMON has the highest TP/A ratio, followed by PRESENT, TWINE, LED, AES, and SPECK. However, PRESENT uses the least energy-per-bit of the above protected ciphers.

All of the protected ciphers used anti-optimization constraints to ensure logical separation of shared signals and separation of non-linear modules to reduce potential leakage. An analysis of all ciphers in this research showed that anti-optimization techniques result in an average of 22% increase in LUTs, 4% reduction in frequency, and 21% reduction in TP/A ratios in the Virtex-7; and an average of 5% increase in LUTs, 16% reduction in frequency, and 20% reduction in TP/A ratios in the Spartan 3E FPGAs.

In comparison to previous results, our AES results are approximately on par with previous 3-share TI-protected ciphers, although the use of different assumptions, architectures, goals, and technologies makes a direct comparison difficult. Our protected implementations of SIMON, SPECK and PRESENT experience roughly twice the growth of previously reported results, due to our selection of full-width basic iterative architecture vice the area-optimized serialized approach adopted in previous research.

The difficulty in making comparisons with previous TI-protected results that use varying architectures, technologies, and power analysis techniques shows the value of our direct comparison of unprotected and protected versions of six ciphers. Our research validates one of the advantages of the t-test leakage detection methodology in providing a comparative analysis of a large number of ciphers, which would be very time-consuming using historical examples of differential power analysis and attack-based key-recovery techniques alone.

VII. AREAS FOR FUTURE RESEARCH

Future research should evaluate protected versions of these ciphers against higher-order DPA. Additionally, it is valuable to expand this study toward the comparison of side channel resistance of authenticated ciphers, particularly those competing in CAESAR Round Three and Final Round.

REFERENCES

[1] "CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness." Internet: http://competitions.cr.yp.to/caesar.html.

[2] D. Bernstein, 2016, Jul. 16, Google Groups, "Cryptographic Competitions," https://groups.google.com/forum/#!forum/crypto-competitions

[3] National Institute of Standards and Technology (NIST), Lightweight Cryptography, Internet: https://www.nist.gov/programs-projects/lightweight-cryptography [Oct. 24, 2017]

[4] K. McKay, L. Bassham, M. Turan and N. Mouha, "Report on Lightweight Cryptography (NISTIR 8114)," National Institute of Standards and Technology (NIST), 2017, Internet: http://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.

[5] G. Goodwill, B. Jun, J. Jaffe and P. Rohatgi, "A testing methodology for side channel resistance validation," NIST Non-invasive Attack Testing Workshop, 2011.

[6] R. Velegalati and J.P. Kaps, "Towards a Flexible Opensource Board for Side-channel analysis (FOBOS)," Cryptographic architectures embedded in reconfigurable devices, CRYPTARCHI 2013, Jun, 2013.

[7] S. Nikova, C. Rechberger and V. Rijmen, "Threshold Implementations Against Side-Channel Attacks and Glitches," Information and Communications Security, Volume 4307 of the series Lecture Notes in Computer Science pp. 529-545, 2006.

[8] Federal Information Processing Standards Publication 197, Advanced Encryption Standard (AES), Nov. 26, 2001

[9] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith and L. Wingers, "The SIMON and SPECK lightweight block ciphers," 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, 2015, pp. 1-6.

[10] A. Bogdanov, L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, C. Vikkelsoe, P. Paillier and I. Verbauwhede, "PRESENT: An Ultra-Lightweight Block Cipher" Cryptographic Hardware and Embedded Systems - CHES 2007: 9th International Workshop, Vienna, Austria, September 10-13, 2007. Proceedings, Springer Berlin, pp. 450-466

[11] J. Guo, T. Peyrin, A. Poschmann and M. Robshaw, "The LED Block Cipher,"in *Cryptographic Hardware and Embedded Systems (CHES 2011): 13th International Workshop*, Nara, Japan, Sep. 28 – Oct. 1, 2011, pp. 326-341.

[12] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, "TWINE: A Lightweight Block Cipher for Multiple Platforms," *SAC*, vol. 7707, 2012, pp. 339–354.

[13] T. Iwata, K. Minematsu, J. Guo, S. Morioka, and E. Kobayashi, "CLOC and SILC v3," Sep. 2016, https://competitions.cr.yp.to/round3/clocsilcv3.pdf

[14] H. Wu and T. Huang, "JAMBU Lightweight Authenticated Encryption Mode," 2016, http://www3.ntu.edu.sg/home/wuhj/research/caesar/caesar.html

[15] P. C. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Proceedings of CRYPTO '99 - 19th International Conference on Cryptology*, Aug. 15-19, 1999, Santa Barbara, CA.

[16] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," C. Kaya Koc, ed., *Journal of Cryptographic Engineering*, Apr. 2011, Vol. 1, Is. 1, pp 5-27.

[17] T. Schneider and A. Moradi, "Leakage assessment methodology", Journal of Cryptographic Engineering, Jun. 1, 2016, Vol. 6, No. 2, pp. 85-89

[18] A. Shamir. "How to Share a Secret." Communications of the ACM, Vol. 22 No. 11, 1979, pp. 612-613.

[19] A. Yao, "Protocols for Secure Computation," FOCS 1982, pp. 160-164

[20] S. Mangard, N. Pramstaller and E. Oswald, "Successfully attacking masked AES hardware implementations," CHES 2005, LNCS, vol. 3659, pp. 157–171.

[21] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov and V. Rijmen, "A More Efficient AES Threshold Implementation," 7th International Conference on Cryptology in Africa (AFRICACRYPT 2014), Marrakesh, Morocco, May 28-30, 2014, pp. 267-284.

[22] A. Moradi, A. Poschmann, S. Ling, C. Paar and H. Wang, "Pushing the Limits: A Very Compact and a Threshold Implementation of AES," 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2011), Tallinn, Estonia, May 15-19, 2011, pp. 69-98.

[23] A. Poschmann, A. Moradi, K. Khoo, C. Lim, H. Wang and S. Ling, "Side-Channel Resistant Crypto for Less than 2,300 GE," Journal of Cryptology, 2011, vol. 24, No. 2, pp. 322-345.

[24] S. Kutzner, P. Nguyen, A. Poschmann and H. Wang, "On 3-Share Threshold Implementations for 4-Bit S-boxes," Constructive Side-Channel Analysis and Secure Design: 4th International Workshop, COSADE 2013, Paris, France, Mar. 6-8, 2013, pp. 99-113

[25] A. Shahverdi, M. Taha and T. Eisenbarth, "Lightweight Side Channel Resistance: Threshold Implementations of Simon," in IEEE Transactions on Computers, vol. 66, no. 4, pp. 661-671, April 1 2017.

[26] C. Chen, M. S. Inci, M. Taha and T. Eisenbarth, "SpecTre: A Tiny Side-Channel Resistant Speck Core for FPGAs", Smart Card Research and Advanced Applications: 15th International Conference, CARDIS 2016, Cannes, France, Nov. 7-9, 2016, Revised Selected Papers, pp. 73-88.

[27] T. Schneider, A. Moradi and T. Güneysu, "ParTI - Towards Combined Hardware Countermeasures Against Side-Channel and Fault-Injection Attacks," 36th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO 2016), vol. 9815, 2016, pp. 302-332.

[28] J.P. Kaps, "Flexible Open-source workBench fOr Side-channel analysis (FOBOS)," Oct. 25, 2016, Internet: https://cryptography.gmu.edu/fobos/

[29] D. Canright and L. Batina, "A Very Compact 'Perfectly Masked' S-Box for AES," 6th International Conference on Applied Cryptography and Network Security, ANCS 2008, New York, NY., Jun. 3-6, 2008, vol. 5037 pp. 446-459.

[30] K. Gaj and P. Chodowiec, "FPGA and ASIC Implementations of AES," In Ç. K. Koç (ed.) *Cryptographic Engineering*, Springer Science & Business Media, 2009, pp. 235-294.

[31] J. Coron, J. Großschädl, and P. Vadnala, Secure Conversion between Boolean and Arithmetic Masking of Any Order, Cryptographic Hardware and Embedded Systems – CHES 2014, 16th International Workshop, Busan, South Korea, Sep. 23-26, 2014.

[32] T. Schneider, A. Moradi and T. Güneysu, "Arithmetic Addition over Boolean Masking," *Applied Cryptography and Network Security: 13th International Conference*, ACNS 2015, New York, Jun. 2-5, 2015, pp. 559-578.

[33] P. Kogge and H. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations". *IEEE Transactions on Computers*, 1973, *C-22*, pp. 783-791.

[34] M. Rivain and E. Prouff, "Provably Secure Higher-Order Masking of AES", Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, USA, Aug. 17-20, 2010, LNCS, vol. 6225, pp 413-427.

[35] B. Bilgin, S. Nikova, V. Nikov and V. Rijmen, "Threshold Implementation of all 3x3 and 4x4 S-Boxes," Cryptographic Hardware and Embedded Systems CHES 2012: 14th International Workshop, Leuven, Sep 9-12, 2012. pp. 76-91.

[36] A. Aysu, E. Gulcan and P. Schaumont, "SIMON Says: Break Area Records of Block Ciphers on FPGAs," in *IEEE Embedded Systems Letters*, vol. 6, no. 2, pp. 37-40, June 2014.