Comparison of Cost of Protection Against Differential Power Analysis of Selected Authenticated Ciphers

William Diehl, Abubakr Abdulgadir, Farnoud Farahmand, Jens-Peter Kaps and Kris Gaj
Department of Electrical and Computer Engineering, George Mason University
Fairfax, VA, U.S.A.
Email: {wdiehl, aabdulga, ffarahma, jkaps, kgaj}@gmu.edu

***Abstract*—Authenticated ciphers are vulnerable to side-channel attacks, including differential power analysis (DPA). Test Vector Leakage Assessment (TVLA) using Welch's t-test has been used to verify improved resistance of block ciphers to DPA after application of countermeasures. However, extension of this methodology to authenticated ciphers is non-trivial, since this requires additional input and output conditions, complex interfaces, and long test vectors interlaced with protocol necessary to describe authenticated cipher operations. In this research we augment an existing side-channel analysis architecture (FOBOS) with TVLA for authenticated ciphers. We use this capability to show that implementations in the Spartan-6 FPGA of the CAESAR Round 3 candidates ACORN, ASCON, CLOC (AES and TWINE), SILC (AES, PRESENT, and LED), JAMBU (AES and SIMON), and Ketje Jr., as well as AES-GCM, are potentially vulnerable to 1st order DPA. We then implement versions of the above ciphers, protected against 1st order DPA, using threshold implementations. TVLA is used to verify improved resistance to 1st order DPA of the protected cipher implementations. Finally, we benchmark unprotected and protected cipher implementations in the Spartan-6 FPGA, and compare the costs of 1st order DPA protection in terms of area, frequency, throughput, throughput-to-area (TP/A) ratio, power, and energy per bit. Our results show that ACORN is the most energy efficient, has the lowest area (in LUTs), and has the highest TP/A ratio of DPA-resistant implementations. However, Ketje Jr. has the highest throughput.**

***Index Terms*—Authenticated Cipher, field programmable gate array, side channel attack, countermeasure, t-test, FOBOS**

I. Introduction

The Internet of Things (IoT) consists of billions of devices that are often constrained by size, weight, and power (SWaP) considerations, but are particularly vulnerable to cyber-security threats, since they often reside physically apart from secure data facilities. Authenticated ciphers, such as AES-GCM, are well-suited for lightweight devices in the IoT, since they combine the functionality of confidentiality, integrity, and authentication services, and can potentially provide the same security as a conventional cipher combined with message authentication code at reduced cost.

Cryptographic algorithms which have been subjected to public scrutiny are generally secure against cryptanalysis given the capabilities of current computing, in that the best-known cryptanalytic attacks are no easier than a brute-force attack. However, actual ciphers exist in the physical world and are implemented in imperfect devices, which can be exploited by analyzing physical phenomena through side channel attacks such as differential power analysis (DPA), to recover all or part of sensitive variables.

The Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR), seeks to identify a portfolio of authenticated ciphers that offer advantages over AES-GCM, and are suitable for widespread adoption [1]. The CAESAR committee specified use-cases for which candidates would be optimized and ultimately selected for final rounds. One of these use cases is for lightweight applications (resource constrained environments), for which desired characteristics include natural ability to protect against side-channel attacks [2]. Accordingly, it is desirable to examine implementations of CAESAR candidates intended for lightweight applications to 1) determine resistance of unprotected and protected implementations to DPA, and 2) determine the relative costs of protection when required. However, to date, there has been no study of the side-channel resistance of a large group of authenticated ciphers, implemented using the same methodology and same test equipment, and no study of the comparative costs of protection against DPA.

In this work, we demonstrate a methodology for analyzing a large group of authenticated ciphers for vulnerabilities to power analysis side-channel attack, and evaluation of the effectiveness of countermeasures. We use the Test Vector Leakage Assessment (TVLA) methodology using Welch's t-test [3], and upgrade the Flexible Open-source workBench fOr Side-channel analysis (FOBOS) [4], to perform t-tests on authenticated ciphers. The FOBOS interface with the victim cipher implementation is standardized by leveraging the CAESAR Hardware Applications Programming Interface (API) for Authenticated Ciphers, which was adopted by the CAESAR committee in May 2016 [5], [6]. Additionally, the use of the Development Package for the CAESAR Hardware API, available at [7], facilitates a repeatable and exportable test methodology for all CAESAR candidates.

Using the augmented FOBOS, we demonstrate t-tests on

11 unprotected authenticated ciphers, implemented on the test device (Spartan 6 FPGA), including AES-GCM, ACORN, ASCON, CLOC (AES, TWINE), SILC (AES, PRESENT, LED), JAMBU (AES, SIMON), and Ketje Jr. [8]–[13]. After demonstrating potential vulnerabilities to DPA, we upgrade the cipher implementations using threshold implementation (TI)-protection, and verify improved resistance to DPA. Finally, we use the augmented FOBOS architecture to perform power analysis of the ciphers during operation on the Spartan 6 FPGA, using representative test vectors. The resulting unprotected and protected ciphers are compared in terms of FPGA resources (LUTs), maximum frequency (MHz), throughput (Mbps), throughput-to-area (TP/A) ratio (Mbps/LUT), power (mW), and energy per bit (E/bit) (nJ/bit), in order to determine costs of protection.

II. Background and Previous Work

A. Authenticated Ciphers

Authenticated Ciphers incorporate the functionality of confidentiality, integrity, and authentication. Input to authenticated ciphers consists of such fields as $Message$, associated data AD (which may include, for example, a header or trailer of a packet used in communication protocols), a secret Key, and a public message number $Npub$. In authenticated encryption, $Ciphertext$ is computed as a function of the inputs, ensuring the confidentiality of the transaction. A Tag, which is a function of all blocks of AD, $Message$, $Npub$, and Key, is produced at the conclusion of message encryption, and assures integrity and authenticity of the transaction. In authenticated decryption, $Ciphertext$ is decrypted to $Message$, and $Tag\prime$ is typically computed as a function of $Ciphertext$, AD, $Npub$, and Key. If $Tag = Tag\prime$ then authentication and integrity of the transaction are assured; otherwise the decrypted $Ciphertext$ is not released. If authenticity and integrity are verified, the outputs are AD and $Message$ [14].

In this research, we analyze Register Transfer Level (RTL) VHDL implementations of AES-GCM, ASCON, CLOC-AES, JAMBU-AES, and SILC-AES available at [15], ACORN at [16], CLOC-TWINE, SILC-PRESENT, and SILC-LED at [17], JAMBU-SIMON available at [18], and Ketje Jr. at [19]. However, we modify both the unprotected and protected implementations of the above ciphers as necessary to 1) achieve implementations protected against 1st order DPA, and 2) facilitate fair benchmarking comparisons.

B. Leakage Detection Methodology: Test Vector Leakage Assessment (TVLA)

Differential Power Analysis (DPA) is used to analyze differences between observed power measurements, and hypothetical power (based on presumed contents of a sensitive variable) according to a power model. However, coming up with a power model is difficult, time consuming, and requires expert knowledge of the underlying architecture [20], [21].

The TVLA methodology described in [3], [22], [23] uses the Welch's t-test to determine whether two distributions are different from one another. Some of the advantages in using the t-test for an assessment of leakage are that it 1) finds leakage of information without mounting an attack, 2) does not rely on knowledge of the underlying architecture, and 3) can quickly reveal when the information leaks and when a countermeasure has failed. However, it is not a complete substitution for DPA. For example, 1) There is no recovery of a key, message, sensitive intermediate values, or the correct power model, and 2) No information is gained about the difficulty of mounting an attack.

In TVLA, a confidence factor 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 (to be subsequently defined), 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. Given a normally distributed probability density function (pdf) $f(t)$, a probability of accepting a null hypothesis p is calculated as $p = 2\int_{|t|}^{\infty} f(t)dt$.

To use the t-test, we start with two distributions, and assume a null hypothesis – namely, that samples are drawn from the same distribution, and that samples are not distinguishable. We designate a threshold, e.g.,$|t| > 4.5$, beyond which we reject the null hypothesis. If this occurs during analysis of the two distributions, we reject the null hypothesis that the samples are from the same distribution and reason that the device is leaking information.

If our goal is to plausibly show that a device is leaking information (without a specific need to recover a sensitive variable or demonstrate the difficulty of an attack), we can use the so-called non-specific t-test. In the non-specific t-test, we preselect some fixed input data D (e.g., $Message$, AD, $Npub$). Then we randomly interleave the feeding of D, or random data, to the algorithm. We call this characterization a fixed versus random test [21]–[23]. This method has been used to show vulnerabilities in block ciphers, and to confirm the effectiveness of countermeasures to DPA (e.g., [21], [24]).

C. Threshold Implementations (TI)

One countermeasure against power analysis side channel attack is called threshold implementation [25]. In order to be provably secure against power analysis in the presence of glitches, a threshold implementation must have the following three properties: 1) Every function is independent of at least one share of each of the input variables (non-completeness); 2) The sum of the output shares gives the desired output (correctness); and 3) The output distribution should match the input distribution (uniformity).

Producing TI which are both uniform and non-complete is challenging, and often increases the cost of threshold implementations [26]. Uniformity can be achieved by supplying fresh masks inside pipelined stages. This method, called resharing or remasking is applied in threshold implementations such as in [26], [27]. However, there is a cost in terms of increased pipelining stages, increased number of clock cycles, increased hardware, and increased requirement to provide sources of fresh randomness.

D. Our contribution

This work expands on previous research to compare costs of DPA protection of several lightweight block ciphers (SIMON, SPECK, PRESENT, LED, TWINE and AES), and extends assessment methodology to authenticated ciphers [21]. Our methodology uses a free and open-source SCA test bench (FOBOS), published specification for the CAESAR Hardware API for Authenticated Ciphers, associated Development Package, and publicly-available source codes for the unprotected cipher implementations in this research.

Additionally, our implementation of these 11 authenticated ciphers in actual hardware exposed bugs in cipher implementations that were not detected through simulation alone, and contributed to the improvement of the CAESAR HW Development Package v2.0, published in Dec. 2017 [7].

Finally, it is well-known that the implementation of countermeasures against DPA is costly, in terms of resources and performance. However, comparison between multiple ciphers often occurs using ambiguous metrics, performed by diverse research groups, and operating on different hardware and test architectures. We illustrate a methodology for the comparison of the costs of protection against 1st order DPA which is suitable for adaptation across all authenticated ciphers, and could assist the CAESAR committee in selection of final round and final portfolio candidates.

III. Methodology

A. Leakage detection methodology for authenticated ciphers

In order to conduct fixed versus random t-tests on a large number of authenticated ciphers, we desire a flexible and repeatable methodology, with a standardized interface and protocol, facilitating the use of long test vectors that adequately test authenticated cipher functionality.

Our solution is facilitated by the CAESAR committee's adoption of the CAESAR HW API for Authenticated Ciphers, which defines a protocol for all necessary authenticated cipher operations, as summarized in [5], [6]. The API also specifies an AXI-compatible external interface, shown in Fig. 1, and further described in [28]. Additionally, the CAESAR HW API Development Package contains a test vector generator, `aeadtvgen.py`, which generates predictable and comprehensive test vectors adequate for power analysis testing [7].

We adapt the Flexible Open-source workBench fOr Side-channel analysis (FOBOS) to perform TVLA on authenticated ciphers. FOBOS uses a separate control board and victim board, where the Device Under Test (DUT), or victim, is instantiated in the victim board. The baseline FOBOS software suite, including acquisition and off-line side-channel analysis packages, is available for download at [4].

The FOBOS architecture, updated for authenticated ciphers, is shown in Fig. 2. The FOBOS DUT victim wrapper is configured with separate FIFOs corresponding to the data ports prescribed in [5], including public data interface (`pdi`), secret data interface (`sdi`), and data output (`do`). A fourth FIFO is aligned to the random data interface (`rdi`), which augments [5] to provide random data necessary for initial masking of public and secret data in protected ciphers.

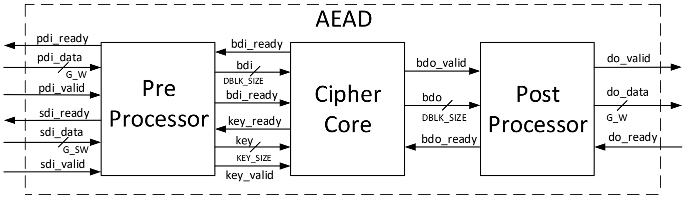

Fig. 1: External interface of the authenticated cipher module (AEAD), compliant with the CAESAR Hardware API [5], [6], and the internal top-level block diagram of AEAD supported by the Development Package [7].

To perform TVLA on an authenticated cipher, we create test vectors with randomly-interleaved fixed or random data, where random data is substituted for instances of cipher input fields, such as $Npub$, AD, and $Message$.

We store thousands of collected traces for post-acquisition off-line analysis. A utility routine splits the collected power traces into two distributions Q_0 and Q_1, according to a fixed-versus-random metafile created during test vector generation.

The tester then runs the t-test utility on distributions Q_0 and Q_1, which generates a two-dimensional display of samples (corresponding to the time domain on the x-axis), and t-values, where sustained and repeatable results of $|t| > 4.5$ are considered as possible vulnerability to DPA leakage.

B. TI-protected ACORN

ACORN can be implemented serially, or in n-bits of output generated in parallel. We choose the very lightweight 8-bit architecture (ACORN-8) available at [16]. We execute the state update in two clock cycles instead of one, in order to distribute the non-linearity across two clock cycles. We instantiate ten 8-bit hybrid 2- / 3- share TI-protected `and` functions, each of which consumes 16 random reshare, and 8 random refresh bits, to maintain the TI uniformity during each call. Amortized over two clock cycles, this results in an average of 120 random bits per clock cycle, which are provided by a PRNG.

C. Hybrid 2- / 3- share TI-protected ASCON

The ASCON-128 implementation at [15], with 64-bit block size and internal datapath, and basic-iterative architecture, is not ideal for protection against DPA. In order to minimize resources required for a 3-share 64-bit TI-protected `and` module, reduce required random refreshing and resharing bits, and reduce vulnerability due to energy and information leakage, we implement a hybrid 2- / 3- share TI-protected ASCON which executes one round in seven clock cycles. We use the bitslice S-Box discussed in [10], and instantiate only one hybrid 2- / 3- share 64-bit TI-protected `and` function, which uses 192 random bits per clock cycle – 128 bits for resharing (from 2 to 3 shares), and 64 bits to satisfy the TI uniformity property.

The randomness is provided by a 192-bit PRNG, which also performs pre-whitening during state initialization to begin round computations with an average Hamming Weight (HW)

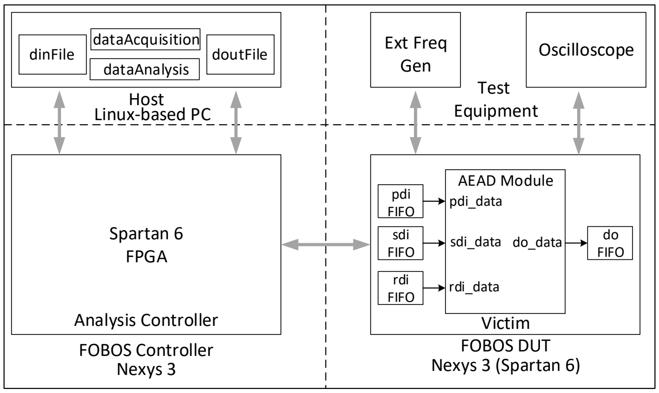

Fig. 2: FOBOS architecture modified for TVLA on authenticated ciphers.

of 0.5 per bit. We modify the unprotected version of ASCON at [15] to use the same seven-cycle architecture to facilitate fair benchmarking.

D. Protection of AES-GCM, CLOC-AES, SILC-AES, and JAMBU-AES

TI-protected versions of AES are documented in [21], [26], [27]. We improve upon on the hybrid 2- / 3-share 5-stage pipelined version in [21] by upgrading the pipeline with TI-protection for round keys generated on the fly. Our TI-protected AES uses an S-Box implemented using combinational logic, as described in [29]. 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.

Our resulting protected design has a 5-stage pipeline, where one S-Box operation commences every clock cycle. A 128-bit round completes every 20 cycles, and a 128-bit block encryption executes in 205 clock cycles. The design uses 16 bits of fresh randomness for resharing from 2 to 3 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 (i.e., per clock cycle). Required refresh randomness is supplied by a PRNG integrated in the AES core.

The authenticated ciphers at [15] using AES as a cryptographic primitive (i.e., AES-GCM, CLOC, SILC, and JAMBU) are optimized for high-speed operations, and use a full-width AES core which executes a 128-bit block encryption in 10 clock cycles. However, it is not feasible to build a full-width TI-protected AES with basic iterative architecture, due to 1) quadratic increase in resources for TI-protection, 2) large number of random refresh bits required, and 3) probability of increased vulnerability to SCA due to long paths of combinational logic along which glitches can occur. Therefore, in order to facilitate a relevant benchmarking comparison between unprotected and protected ciphers, we replace the full-width AES with an unprotected version of our 8-bit, 5-stage pipelined AES in the unprotected implementations of AES-GCM, CLOC-AES, SILC-AES, and JAMBU-AES.

Protection of AES-GCM additionally requires a 3-share TI-protected multiplier in $GF(2^{128})$. Each multiplication completes in 128 clock cycles, but does not further limit the overall throughput of AES-GCM.

E. Protection of Cipher Implementations Using SIMON, PRESENT, LED, and TWINE Primitives

Strategies to protect authenticated ciphers using SIMON, PRESENT, LED, and TWINE primitives are discussed in [21].

Protected implementations of JAMBU-SIMON, SILC-PRESENT and SILC-LED use 3-share TI that do not require increased randomness for uniformity, as discussed in [24] (SIMON) and [30], [31] (PRESENT and LED). However, the protected implementation of CLOC-TWINE leverages Fermat's Little Theorem, as described in [21] to compute $x^{14}{\equiv}x^{-1}$ in $GF(2^4)$, which decomposes into two non-linear multipliers, with several low-cost linear squares. The outputs of the multipliers, however, are not permutations on the input; they do not satisfy the TI uniformity property. Therefore, we use two bits of refresh randomness per S-Box per clock cycle, for a total of 20 random bits per clock cycle, including four bits for the S-Boxes for round key updates.

F. Protection of Ketje Jr.

Ketje uses the Keccak-p^* transformation (adapted from the Keccak-f in SHA-3). Only one transformation (χ) is non-linear, and protection is provided by a 3-share TI-protected `and` module. We implement a hybrid 2- / 3-share TI-protection on the implementation at [19], using two shares outside the χ transformation, resharing to three shares in χ, and recombining to two shares for the remainder of the round. We use 200 bits of resharing randomness per clock cycle, which is provided by an integrated PRNG.

G. TI protection of AEAD and CipherCore Modules

Our authenticated cipher implementations use the PreProcessor and PostProcessor modules, located in the AEAD module (shown in Fig. 1), and available as part of the CAESAR HW API Development Package [7]. The authenticated cipher, including computation layers above the primitive level, are located in the CipherCore module (shown in Fig. 1).

TI protection of cipher functionality within CipherCore is relatively straightforward, except that one must take care to account for occasional non-linear operations, such as padding, and ensure that derived control functions do not leak information. We use a 2-share TI for the AES-based ciphers (i.e., AES-GCM, CLOC-AES, SILC-AES, and JAMBU-AES), ACORN, ASCON, and Ketje Jr., and a 3-share TI for JAMBU-SIMON, CLOC-TWINE, SILC-PRESENT, and SILC-LED. The PreProcessor and PostProcessor modules, located at the top level of the AEAD module, are also capable of leaking information, and require TI protection. In order to comply with the CAESAR HW API, unmasked data and secret key enter AEAD through `pdi_data` and `sdi_data` (respectively), and are separated into two or three shares in PreProcessor. For protected ciphers, we add an additional external port to AEAD called `rdi_data`, in which randomness for initial masking of sensitive data enters the cipher. We also augment AEAD with an `rdi` PreProcessor, which provides the correct amount of random data to the PreProcessor at the proper time. In this approach, the amount of randomness required in `rdi` is $\#rnd\ bits = (\#bits\ public\ data + \#bits\ key\ data) \times (d-1)$, where d is the number of TI shares.

IV. Results

A. *Power analysis of unprotected authenticated ciphers*

We use the above FOBOS with TVLA to measure the resistance of the 11 cipher implementations to DPA. We perform 2000 fixed-versus-random traces, with approximately 20,000 samples per trace, with test vectors consisting of between four and eight combinations of authenticated encryption and decryption. The t-tests are performed on the Nexys 3 victim board, and instantiated in the Spartan-6 FPGA (xc6slx16csg324-3). For t-tests, the ciphers are clocked at 781 KHz, in order to minimize capacitive and inductive effects and present a cleaner power signature.

The results, shown in top half of Fig. 3, indicate significant leakage in all cipher implementations. The results are as expected for unprotected implementations. As a reminder, a failing t-test does not prove DPA vulnerability; it only shows that one can statistically distinguish between sets of power traces consisting of fixed and random test vectors.

B. *Power analysis of protected authenticated ciphers*

We next apply TI protection techniques to the above ciphers. TVLA using 2000 traces shows that the protected cipher implementations pass the t-test, and have improved resistance to 1st order DPA. The results of protected cipher implementations are shown in the bottom half of Fig. 3.

C. *Benchmarking of unprotected and protected ciphers*

Unprotected and protected versions of all ciphers are implemented using Xilinx ISE on the Spartan 6 FPGA. The results are compared in terms of area (LUTs), frequency, throughput (TP) assuming maximum frequency, and throughput-to-area (TP/A) ratio. Using FOBOS, we measure power consumed by the 1.2V Vcc_{INT} FPGA power supply during the application of test vectors, by measuring amplified voltage across a 1Ω shunt resistor. Mean power (P_{mean}) is measured for the 11 ciphers at 10 MHz, where the victim board is supplied by an external frequency generator. Energy per bit (E/bit) (nJ/bit) is computed as $P_{mean}(mJps)/TP_{Freq=10MHz}(Mbps)$.

Results are shown in Table I. Best results (highest for frequency, TP, and TP/A; lowest for area, power, and E/bit) for each metric are shown in boldface. For both the unprotected and protected implementations, ACORN is the smallest in terms of LUTs, followed by JAMBU-AES and JAMBU-SIMON. In terms of throughput, Ketje Jr. is highest among both unprotected and protected versions, followed by ACORN and JAMBU-SIMON, while ACORN has the highest TP/A ratio, followed by Ketje Jr. and JAMBU-SIMON.

TABLE I: Benchmarking of ciphers in Spartan-6 FPGA (Power and E/bit Measured at Fixed Frequency of 10MHz)

Cipher	Area	Freq	TP	TP/A	Pwr	E/bit
	LUT	MHz	Mbps	Mbps/ LUT	mW	nJ/bit
Unprotected						
AES-GCM	1947	176.0	103.4	0.053	10.3	1.754
ACORN	**549**	226.6	906.2	**1.651**	**7.8**	0.195
ASCON	2048	195.5	255.4	0.125	10.5	0.805
CLOC-AES	2496	150.0	93.2	0.037	12.4	1.996
CLOC-TWINE	1536	171.2	156.5	0.102	10.3	1.129
SILC-AES	1975	163.0	101.7	0.052	10.6	1.698
SILC-PRESENT	2057	**238.8**	238.8	0.116	9.7	0.972
SILC-LED	1990	203.4	132.8	0.067	10.9	1.666
JAMBU-AES	1073	163.1	50.9	0.048	9.4	3.001
JAMBU-SIMON	1105	137.9	509.3	0.461	19.7	0.534
Ketje Jr.	1242	96.9	**1550.4**	1.248	22.0	**0.138**
Protected						
AES-GCM	4828	116.8	68.57	0.014	23.9	4.070
ACORN	**2732**	**142.7**	570.6	**0.209**	**16.8**	**0.419**
ASCON	6364	103.1	134.6	0.021	34.8	2.664
CLOC-AES	5900	104.2	64.7	0.011	33.1	5.327
CLOC-TWINE	6467	70.7	64.7	0.010	71.6	7.848
SILC-AES	4865	102.8	64.2	0.013	23.7	3.796
SILC-PRESENT	4624	116.6	116.6	0.025	25.3	2.526
SILC-LED	4780	92.0	60.1	0.013	40.2	6.162
JAMBU-AES	2869	122.4	38.2	0.013	17.8	5.702
JAMBU-SIMON	3140	58.7	216.7	0.069	96.5	2.614
Ketje Jr.	4800	59.6	**954**	0.199	105.3	0.658

On average, the number of LUTs increases by a factor of 3.1, the throughput decreases by a factor of 1.8, and the TP/A ratio decreases by a factor of 5.6 when comparing protected to unprotected implementations.

ACORN is the most energy efficient of protected cipher implementations in terms of E/bit, followed by Ketje Jr. and SILC-PRESENT. Additionally, ACORN has the lowest mean power consumption, followed by JAMBU-AES and SILC-AES. The average power and E/bit of protected implementations increases by a factor of 3.4 compared to unprotected implementations, when measured at a common frequency of 10 MHz on the Spartan-6 FPGA.

V. Conclusions

In this research we introduced a methodology for conducting Test Vector Leakage Assessment (TVLA) on a large number of authenticated ciphers, in order to determine resistance to DPA side-channel attack, and to verify effectiveness of countermeasures against DPA. Our methodology, which leverages the open-source FOBOS test bench, CAESAR Hardware API standards, and related Development Package, shows that unprotected implementations of AES-GCM, ACORN, ASCON, CLOC (AES and TWINE), SILC (AES, PRESENT, and LED), JAMBU (AES and SIMON), and Ketje Jr. in the Spartan-6 FPGA, are potentially vulnerable to DPA.

We implement protected versions of all 11 ciphers, verify their improved resistance to 1st order DPA using TVLA, benchmark unprotected and protected cipher versions, and compare the resulting costs of protection against DPA.

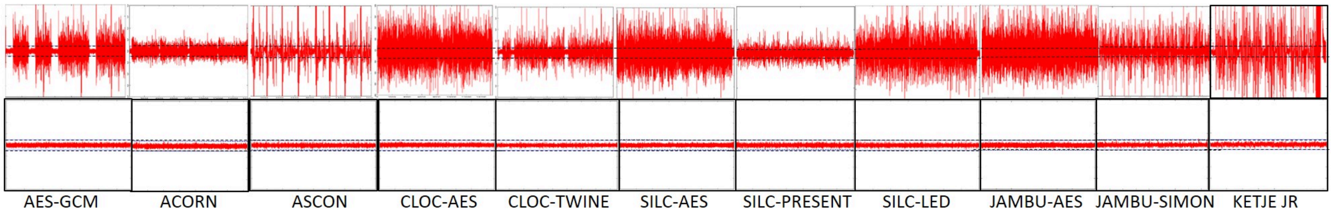

Fig. 3: Results of t-tests, with unprotected implementations on top, and corresponding protected implementations on bottom. Time domain (samples) are on the x-axis, t-values are on the y-axis. Horizontal dotted lines denote $t = \pm 4.5$.

ACORN has the lowest area (in terms of LUTs) of protected ciphers, followed by JAMBU-AES and JAMBU-SIMON. Likewise, ACORN has the highest throughput-to-area (TP/A) ratio, followed by Ketje Jr. and JAMBU-SIMON. However, Ketje Jr. has the highest throughput of protected implementations, followed by ACORN and JAMBU-SIMON.

At a fixed frequency of 10 MHz, ACORN is the most energy efficient of protected implementations (i.e., uses the lowest energy per bit), followed by Ketje Jr. and SILC-PRESENT. Additionally, ACORN has the lowest mean power consumption, followed by JAMBU-AES and SILC-AES.

In terms of costs of protection against 1st order DPA, the area increases by an average factor of 3.1, the throughput decreases by a factor of 1.8, and the TP/A ratio decreases by a factor of 5.6, when comparing protected to unprotected implementations. The mean power consumption and energy per bit of protected implementations increase by an average factor of 3.4 compared to unprotected implementations.

References

[1] "CAESAR Competition for Authenticated Encryption: Security, Applicability, and Robustness," 2012, http://competitions.cr.yp.to/caesar.html.

[2] D. Bernstein. (2016, Jul) Cryptographic Competitions. [Online]. Available: http://groups.google.com/forum/#!forum/crypto-competitions

[3] J. Cooper, E. DeMulder, G. Goodwill, J. Jaffe, G. Kenworthy, and P. Rohatgi, "Test Vector Leakage Assessment (TVLA) Methodology in Practice," International Cryptographic Module Conference, 2013.

[4] George Mason University. (2016, Oct) Flexible Open-source workBench fOr Side-channel analysis (FOBOS). [Online]. Available: https://cryptography.gmu.edu/fobos/

[5] E. Homsirikamol, W. Diehl, A. Ferozpuri, F. Farahmand, P. Yalla, J. Kaps, and K. Gaj, "CAESAR Hardware API," Cryptology ePrint Archive, Report 2016/626, 2016, http://eprint.iacr.org/2016/626.pdf.

[6] ——. (2016, Jun) Addendum to the CAESAR Hardware API v1.0. [Online]. Available: https://cryptography.gmu.edu/athena/CAESAR_HW_API/ CAESAR_HW_API_v1.0_Addendum.pdf

[7] George Mason University. (2017, Dec) Development Package for the CAESAR Hardware API, v2.0. [Online]. Available: https://cryptography.gmu.edu/athena/index.php?id=CAESAR

[8] M. Dworkin, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC," NIST SP800-38D, 2007.

[9] H. Wu. (2016, Sep) ACORN: A Lightweight Authenticated Cipher. Accessed February 16, 2018. [Online]. Available: https://competitions.cr.yp.to/round3/acornv3.pdf

[10] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer. (2016, Sep) Ascon v1.2. Accessed February 16, 2018. [Online]. Available: https://competitions.cr.yp.to/round3/asconv12.pdf

[11] T. Iwata, K. Minematsu, J. Guo, S. Morioka, and E. Kobayashi. (2016, Sep) CLOC and SILC. Accessed February 16, 2018. [Online]. Available: https://competitions.cr.yp.to/round3/clocsilcv3.pdf

[12] H. Wu and T. Huang. (2016, Sep) The JAMBU Lightweight Authentication Encryption Mode (v2.1). Accessed February 16, 2018. [Online]. Available: https://competitions.cr.yp.to/round3/jambuv21.pdf

[13] G. Bertoni, J. Daemen, M. Peeters, G. V. Assche, and R. V. Keer. (2016, Sep) CAESAR submission: Ketje v2. Accessed February 16, 2018. [Online]. Available: https://competitions.cr.yp.to/round3/ketjev2.pdf

[14] W. Diehl and K. Gaj, "RTL Implementations and FPGA Benchmarking of Selected CAESAR Round Two Authenticated Ciphers," *Microprocess. Microsyst.*, vol. 52, no. C, pp. 202–218, Jul. 2017.

[15] (2017, Dec) GMU Source Code of Round 3 & Round 2 CAESAR Candidates, AES-GCM, AES, AES-HLS, and Keccak Permutation F. Accessed Feb. 16, 2018. [Online]. Available: https://cryptography.gmu.edu/athena/index.php?id=CAESAR_source_codes

[16] T. Huang. (2017, Jul) Round 3 Hardware Submission: ACORN. [Online]. Available: https://groups.google.com/forum/#!topic/crypto-competitions

[17] T. Iwata. (2017, Jul) HW for CLOC and SILC 64-bit BC. [Online]. Available: https://groups.google.com/forum/#!topic/crypto-competitions

[18] T. Huang. (2017, Aug) SIMON-JAMBU. [Online]. Available: https://groups.google.com/forum/#!forum/crypto-competitions

[19] G. Bertoni. (2017, Dec) Ketje-Keyak Team. [Online]. Available: https://github.com/guidobertoni/caesar_gmu_vhdl

[20] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology — CRYPTO' 99*, 1999, pp. 388–397.

[21] W. Diehl, A. Abdulgadir, J.-P. Kaps, and K. Gaj, "Comparing the Cost of Protecting Selected Lightweight Block Ciphers Against Differential Power Analysis in Low-Cost FPGAs," in *Intl. Conf. on Field Programmable Technologies, Melbourne, Australia*, 2017, pp. 128–135.

[22] G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi, "A Testing Methodology for Side Channel Resistance Validation," NIST Non-invasive Attack Testing Workshop, 2011.

[23] T. Schneider and A. Moradi, "Leakage Assessment Methodology," *Journal of Cryptographic Engineering*, vol. 6, no. 2, pp. 85–89, Jun 2016.

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

[25] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold Implementations Against Side-Channel Attacks and Glitches," in *Information and Communications Security*, 2006, pp. 529–545.

[26] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "A More Efficient AES Threshold Implementation," in *Progress in Cryptology – AFRICACRYPT 2014*, 2014, pp. 267–284.

[27] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the Limits: A Very Compact and a Threshold Implementation of AES," in *Advances in Cryptology – EUROCRYPT 2011*, 2011, pp. 69–88.

[28] E. Homsirikamol, W. Diehl, A. Ferozpuri, F. Farahmand, and K. Gaj. (2017, Dec) Implementers Guide to the CAESAR Hardware API v2.0. [Online]. Available: https://cryptography.gmu.edu/athena/CAESAR_HW_API/ CAESAR_HW_Implementers_Guide_v2.0.pdf

[29] D. Canright and L. Batina, "A Very Compact 'Perfectly Masked' S-Box for AES," in *ACNS*, 2008.

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

[31] S. Kutzner, P. H. Nguyen, A. Poschmann, and H. Wang, "On 3-Share Threshold Implementations for 4-Bit S-boxes," in *Constructive Side-Channel Analysis and Secure Design*, 2013, pp. 99–113.