CPA on Hardware Implementation of COLM Authenticated Cipher and Protect it with DOM Masking Scheme

Mohsen Jahanbani[1], Zeinolabedin Norouzi[2] , Nasour Bagheri[3]
[1]Imam Hossein Comprehensive University- Tehran- Iran- mjahanbani@ihu.ac.ir
[2]Imam Hossein Comprehensive University- Tehran- Iran- znorozi@ihu.ac.ir
[3]Shahid Rajaee Teacher Training University- Tehran- Iran- nbagheri@srttu.edu

Abstract: Authenticated encryption schemes provide both confidentiality and integrity services, simultaneously. Correlation power analysis (CPA) can be a thread for authenticated ciphers, like all physical implementations of any cryptographic system. In this paper, for the first time, a three-steps CPA attack against COLM, one of the winners of CAESAR, is presented to indicate its vulnerability. For this purpose, in this research paper, this authenticated encryption scheme is implemented on the FPGA of the SAKURA-G board and, by measuring and collecting 1,800 power traces, a successful CPA attack with zero value power model has been mounted on it. In addition, a protected hardware architecture for the COLM is proposed to make this design secure against first-order CPA attacks. To this end, a domain-oriented masking (DOM) scheme with two inputs/outputs share is used to protect the COLM. To verify the security of these countermeasures, we mounted a first and second-order CPA attack and a non-specified t-test on the protected COLM.

Keyword*:* Authenticated Cipher, COLM, CPA, DOM, Masking.

1. Introduction

Authenticated encryption (AE) schemes provide confidentiality, integrity, and authenticity of plaintext simultaneously. The traditional way to achieve such properties is combination of several cryptographic primitives, which usually encryption algorithms for confidentiality and message authentication codes (MACs) for integrity and authenticity are used. This method is not optimal and may also be accompanied flaw in design or implementation. Therefore, many modes of operation are designed to provide an efficient and secure AE structure such as counter-with-CBC-MAC (CCM) [1] and offset code book mode (OCB) [2]. Also, there are new AE designing methods such as stream cipher-based, sponge-based and dedicated designs. Currently, AES-GCM [3] is widely used, but unfortunately, it is not efficient for many applications. In addition, several vulnerabilities are known in this design [4], [5]. In January 2013, a competition for authenticated encryption: security, application, and robustness (CAESAR) competition [6] was started to select a portfolio of AE that (1) provide advantages over AES-GCM and (2) appropriate for widespread adoption. In total, 57 schemes were submitted to this competition. In July 2016, the CAESAR committee suggested three categories of use cases for which candidates would be optimized and ultimately selected during the final round. In March 2019, six winners were selected for these applications.

The COLM authenticated cipher [7] is a winner of CAESAR competition, that was selected for in-depth defense in misuse scenarios. It provide strong security feature, such as security against nonce-misuse adversaries and security under release of unverified plaintext. In this scenario the decrypted ciphertext is accessible by an adversary before the authentication tag is verified. Also, COLM provides a trade-off between good efficiency and strong security. Besides, this scheme has a simple design mode based on AES block cipher that makes it extremely easy to use. One of the desired characteristics of the CAESAR winners is the capability to protect against side-channel attacks, that is also announced by the competition's committee. Therefore, it is favorable to evaluate the resistance of CAESAR winners against differential power analysis (DPA) attack [8] and also determine the protection costs. Nevertheless, no side-channel attack on COLM is presented so far and there is no protection scheme for COLM.

1.1. Related Works

Adomnicai et al. [9], investigate the resistance of lightweight winners of CAESAR competition, i.e., ACORN and Ascon, against power analysis attack. This evaluation was carried out on the software implementation on ARM Cortex-M3 microprocessor. Their results showed that power analysis attacks on Ascon scheme could be performed at the initialization and finalization stages. Also, ACORN is based on stream ciphers, and the keystream is calculated independently of the plaintext that makes the side-channel attacks more challenging compared to block ciphers. Thus, an attacker should focus on the initialization stage or re-synchronization mechanism. Their attack does not return the key, but they introduce a system of boolean equations to solve this problem, that was an All-SAT problem. Their results justify the need for countermeasures at the software implementation level. Therefore, they presented a two masking schemes for these.

Samwel and Daemen [10] presented a successful power analysis attack on Keyak and Ascon. Both schemes have sponge construction and use the same type of S-box. Therefore, the attack on both schemes was almost the same. Then, they added a linear layer after the S-box to makes this attack much harder. Gross et al. [11] proposed several hardware implementations for Ascon. They showed that this scheme could be easily protected against power analysis attacks through threshold implementation (TI) [12] masking scheme.

Recently, Diehl et al. [13] showed the vulnerability of some of third round and CAESAR winners include CLOC, JAMBU, Ascon, Ketje Jr, SILC, and ACORN to first-order DPA using t-test leakage detection methodology. But they did not present any attack scenario. Moreover, they proposed a protected version of these ciphers against first-order DPA, using TI. The t-test leakage detection methodology is used to verify improved resistance. Furthermore, they compared the performance of protected and unprotected schemes in regarding area, frequency, and throughput.

1.2. The Paper Contribution

COLM, as a CAESAR winner, offers strong security guarantee. However, the vulnerability of this scheme to power analysis attack has not been investigated, so far. To the best of our knowledge, this research is the first study on power analysis attack against COLM. Given that inputs of COLM are masked with a mask value (Δ), the power analysis attack is more challenging compared to the block ciphers. In this research, to solve this problem, a three-steps attack approach is presented. These steps implemented on FPGA. Then the traces of power is recorded, and the key is recovered using correlation power analysis (CPA) [14] attack. The results confirm the need for countermeasures. Therefore, a protected scheme for COLM is presented. This scheme is based on domain-oriented masking (DOM) [15]. In this method, the sensitive variable is shared and placed into separate domains. The proposed protected COLM uses two input/output shares. The comparison of hardware performance of protected and unprotected COLM showed that area and throughput of protected version have been doubled and halved, respectively. The resistance of the protected version to the first-order CPA is verified using CPA attack and by the t-test method.

1.3. The Paper Organization

This paper is organized as follows: In Section 2, background information including power analysis attack, masking schemes, and COLM is described briefly. In Section 3, the power analysis attack scenario on unprotected COLM is presented and a CPA attack based on this scenario is mounted. Proposed architectures to protect COLM by DOM masking approach is explained in section 4. In Section 5, the performance of the protected and unprotected scheme in FPGA is compared. Also, the security of protected COLM is evaluated by first and second-order attack using the t-test. Finally, the paper is concluded in section 6.

2. Background Information

In this section, the concept of power analysis attack, countermeasures method against this attack and COLM authenticated cipher specification are described.

2.1. Power Analysis Attack

One of the most well known and most effective practical attacks on cryptographic hardware is power analysis attack that reveals the secret key using the consumed power leakage. Power analysis attack has different kinds, including simple power analysis (SPA), differential power analysis (DPA) [8], correlation power analysis (CPA) [14], mutual information analysis [16] and template-based attack [17]. Each of these attacks has advantages in a particular aspect and is appropriate under certain circumstances. CPA is a general form of DPA that has received

more attention due to its higher capability in revealing the secret value.

In a CPA attack, the measured values are compared with estimated values from the theoretical model of power, and their correlation value is calculated. This model (leakage model) is selected based on the effect of intermediate values on power consumption. To estimate power consumption, a hypothetical model is used. A better power model needs less trace to attack. Typically, hamming weight (HW) model describe the power consumption of microcontroller and hamming distance (HD) model is suitable for CMOS circuit. HD model in hardware implementation is used at the moment of time that the registers are updated. Also, the zero value (ZV) power model is helpful when combinatorial circuit such as S-box consume the power. This model assumes that 0 data value has less power consumption that other values [18].

2.2. Masking

The hardware countermeasures are classified as hiding and masking scheme that can be performed at the logic (cell) or architecture (algorithm) level. In the masking method, the intermediate values are randomized which can be implemented at the algorithmic level. Boolean masking scheme is based on secret sharing concept in which a sensitive intermediate (key-dependent) value x is divided into s shares $(x^1, \dots, x^s)$ such that $x = \bigoplus_{i=1}^{s} x^i$. Due to the boolean structure of masks, it is easy to apply a linear function $\mathcal{L}(.)$ over shares because of $\mathcal{L}(x) = \bigoplus_{i=1}^{s} \mathcal{L}(x^i)$. But the implementation of a non-linear function $F(.)$ by shares representation is very hard since $F(x) \neq \bigoplus_{i=1}^{s} F(x^i)$. However, this masking scheme has been applied in the hardware implementation of AES with $s = 2$ [19], but was not successful due to the glitch in hardware [20]. To solve this problem two masking approaches is proposed so far: threshold implementation (TI) [12] and DOM [15].

In 2011, TI was introduced based on mathematical foundations include threshold Boolean secret sharing and the secure multi-party computations. Even with the existence of glitch, TI provides provable security. The number of shares s defines the order of scheme security. The lower bound of the number of required input and output shares is calculated based on Equation (1) [21]:

$$s_{in} \geq td + 1, \;\; s_{out} \geq \binom{s_{in}}{t} \tag{1}$$

Where d is security order and t is the algebraic degree of the function.

Recently another masking scheme that called DOM [22] has been presented, which has reduced the number of required shares from $td + 1$ to $d + 1$ for d^{th}-order security. DOM is based on the concept of shares distribution in d+1 domain such that all domains shares are independent of the others. For the implementation of nonlinear functions, the parts whose inputs come from several domains are critical parts. For cross-domain computations, a fresh random values is added to these terms to keep them independent. Also, to prevent glitch propagating, the registers are added between domains. For example, the first-order secure AND gate calculations need two domains. The first (x_1, y_1) and the second (x_2, y_2) input shares should be random and independent. The implementation of secure AND gate is performed in three stages of calculation, resharing, and integration. These stages are shown in Fig. 1.

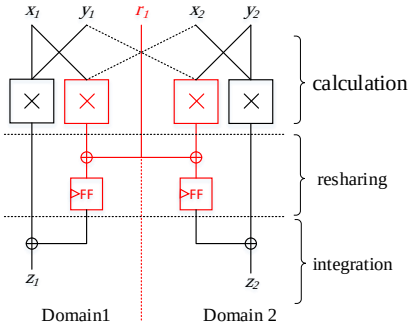


Fig. 1. First-order secure AND gate [22]

The underlying security model for both TI and DOM masking scheme is the same and the power consumption of component functions in both of them is independent of each other. Compared to TI, the number of shares has reduced from $td + 1$ to $d + 1$ for a t^{th}-order secure nonlinear function, and the number of required fresh random bits has reduced from $(d + 1)^2$ to $d(d + 1)/2$. In contrast to this improvement, the number of clocks increased and input

shares should be independent.

To attack the masked implementation, a higher-order attack is needed. This attack exploits the joint leakage of several intermediate values. Thus power traces require to preprocess. The order of an attack has defined in two ways in literature [23]:

- The attack combines v point in different clocks is called v-variant attack.
- The order of the statistical moments that are used in the attack defines the order of attack.

Typically in the masked software implementation, the shares are processed in the different clocks. For example, if a CPA attack combines two points of each trace by summing them up is called a bivariate first-order attack. If shares are processed simultaneously, the preprocessing function is applied to a single point in the trace. This case typically occurs in masked hardware implementation. For a univariate second-order attack, squaring the power traces is appropriate preprocessing function [18].

2.3. COLM Authenticated Cipher

COLM [7] is a block cipher mode based on the Encrypt-Linear mix-Encrypt mode. COLM has a 128-bit key, 64-bit tag and 64-bit security level for confidentially and integrity. COLM uses the linear mixing functions ρ and ρ^{-1}, which ρ has two inputs $x,\ st \in \{0,1\}^{128}$ and two outputs $y, st' \in \{0,1\}^{128}$ that $y = x \oplus 3.st$ and $st' = x \oplus 2.st$. COLM is depicted in Fig. 2. The stages of COLM are subkey L generation, IV generation, tagged ciphertext generation, decryption, and verification. The subkeys are calculated by $L = E_k(0)$, $L_1 = 3.L$ and $L_2 = 3^2.L$. IV is computed from the associated data (AD). The tagged ciphertext is computed from the padded plaintext and IV. Decryption is the same as encryption. The verification will be successful if we have $C[l+1] = C'[l+1]$.

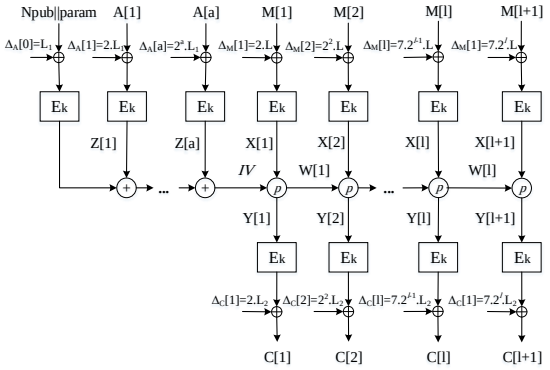


Fig. 2. COLM authenticated encryption mode of operation [7]

3. CPA attack against COLM

Mounting power analysis attack requires knowing the attack points on the scheme. According to Fig. 2, E_K and function ρ are the parts that can be used for the CPA attack on COLM. In general, the inputs of the attack point should be variable and known. Also, the secret key should be combined with that part. Since the inputs of the lower part's E_K are unknown and the key is not in the function ρ, these points are not propera choice. CPA attack on E_K of the upper part is more difficult compared to an attack on a block cipher since the inputs of E_K cannot be controlled directly. The plaintext is XORed with an unknown and variable value Δ in each block that is used as E_K input. Therefore, the key cannot be recovered with a usual power analysis attack. If the attack performs on the same block, then the unknown value Δ is constant [11]. On the other hand, in the upper stage of the Fig. 2, each block of AD or the plaintext are independently processing; thus the attack can be mounted on each type of input blocks. Here the plaintext is selected for our attack.

In COLM, E_K is AES cipher. Every encryption of AES has an initial process followed by ten rounds. In the initial process, the *AddRoundKey* is performed such that key is XORed with plaintext. The order of operations in other round are SubBytes (S-box), ShiftRows (SR), MixColumns (MC) and *AddRoundKey*, exclude the last round where do not has MC. We can shift the calculation of the S-box of the 1st round to initial round. Thus in the initial round, the *AddRoundKey* and S-box of 1st round are calculated. Order of operations in the 1st round are changed to SR, MC, *AddRoundKey* and S-box of the 2st round. Inputs of AES in COLM are the combination of the constant unknown value Δ and a known variable value (plaintext). The CPA attack can recover Δ as a part of the key. The recovered key is a combination of the actual key and Δ. By this key, we can calculate the input of the first round of AES. Then the CPA attack is repeated on the first round that gives us the first round key. Given this key, we can

recover the actual key. On the next section, we will describe the detail of the attack against COLM scheme as the first contribution of this paper.

3.1. Attack Details

A CPA attack is implemented on E_K of the upper part of COLM to recover the encryption key K. The attack is shown in Fig. 3 and is described as the following steps:

Step 1: Performing the CPA attack on S-box of the initial round of AES and recovering the modified key K_0' (the attack point 1 in Fig. 3). If we consider AES with the modified sequence of operation as described above, the S-box input of the initial round is $AddRoundKey(K_0,\ M \oplus \Delta_m) = K_0 \oplus M \oplus \Delta_m$, where M is the plaintext. By rolling Δ_m to key, the S-box input can be rewritten as $AddRoundKey(K_0', M) = K_0' \oplus M$. When CPA attack targets the initial round, $K_0' = K_0 \oplus \Delta_m$ is recovered.

Step 2: Recovering the first round key K_1. After recovering K_0', we can compute the output of the initial round that is the input of the first round, i.e. $Output_0 = Sbox(K_0' \oplus M) = Input_1$. By mounting the CPA attack on the S-box of the 1st round, the K_1 is recovered (attack point 2 in Fig. 3).

Step 3: Recovering K_0 from K_1. By running the AES key schedule function inversely, the encryption key K of the algorithm $K_0 = K$ is recovered from K_1.

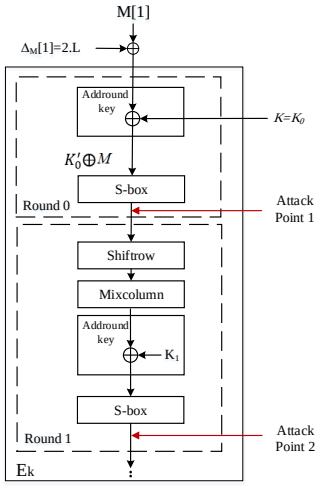


Fig. 3. Attack procedure on the initial and the first round of AES in unprotected COLM

3.2. Practical CPA Attack Results

To implement the CPA attack on COLM, the hardware implementation of AES is required. The hardware architecture in [24] is used for the implementation of AES (Fig. 4). This architecture is serial, with 8-bit data-path, and the order of operations is modified based on the above description. Also, one S-box was used for two parts of SubBytes and key schedule in a serial manner, and each S-box is calculated in one clock. The serial architecture has less switching noise compare to parallel architecture. Additionally, in the parallel architecture, the measured power consumption is the superposition of power consumption of several S-boxes, which makes the power analysis very hard. Moreover, in a non-serial architecture, that several S-boxes are implemented, the traces generated by different S-boxes with the same input will not be precisely the same. Hence, the best choice could be the serial structure.

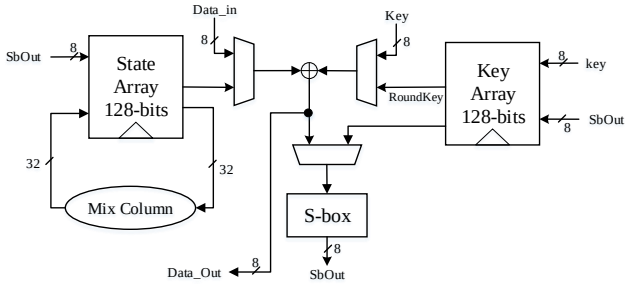


Fig. 4. AES architecture for the attack against COLM [24]

The required equipment for the CPA attack includes FPGA, digital oscilloscope, and PC. SAKURA-G board [25]

is one of the most popular boards used in side-channel attacks. This board includes two FPGA of Xilinx SPARTAN-6 series include control FPGA and cryptographic FPGA. The control FPGA manages the communications between the cryptographic FPGA and PC. The PC is connected to the FPGA through USB using the FTDI interface. This board has the ultra-low-noise design, and an onboard amplifier makes power analysis easier. The power is measured by voltage drop over a 1Ω resistor at the Vdd of cryptographic FPGA after amplification. The measurement was performed using an Infinium Keysight DS090604A digital oscilloscope with sampling rates of 20 Gs/s and 6 GHz bandwidth. To implement AES, the architecture of Fig. 4 is described in RTL-level using VHDL code and then synthesized using Xilinx ISE V14.7 software. The functional verification is done using Mentor Graphics Modelsim v10.1c by test vectors. SAKURA-G board is configured using Dip-Switch according to guide [25]. Fig. 5 shows the setup used for capturing the trace and CPA attack.

The crypto FPGA is clocked in 1 MHz by onboard clock oscillator. Faster or unstable clock cause overlap power peak of the adjacent clock cycle. Several users LED is there connected to FPGAs on board that consumes the power and disturbs the measurements. Thus all those LED switched off.

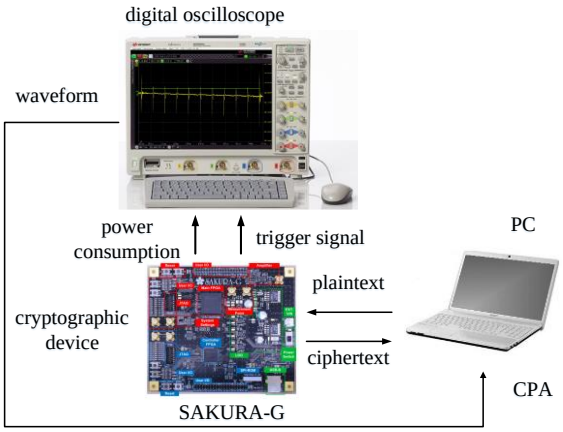


Fig. 5. Setup for capturing trace with the SAKURA-G

To synchronize oscilloscope with FPGA, a trigger signal is generated by FPGA and sent to the oscilloscope. Additionally, to communicate between the SAKURA-G board, PC and oscilloscope, an interface software program has been developed in C# language. This program is responsible for generating the required plaintext and key and sending them to FPGA and receiving the outputs through the USB port. This program also sends the command to the oscilloscope to store the traces in the memory. Then, traces are transferred to the PC for analysis. To reduce the noise, every input repeated 1000 times and then averaged using MATLAB R2017. Fig. 6 shows the measured power consumption waveform for the initial and first round of AES. Before the start of the initial round, the key is XORed with plaintext and S-box output is calculated. In the first round operation, SR, MC, *AddRoundKey*, and S-box of next round are performed, respectively.

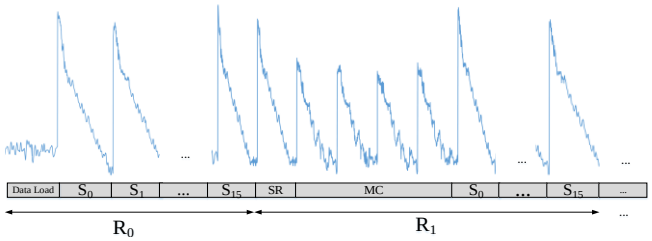


Fig. 6. A measured power trace for the initial and first round of unprotected AES

The first-order CPA attack is mounted with HD and ZV model on unprotected COLM. The attack on S-box input/output using the HD model was not successful (Fig. 7a), while the attack on S-box output using the ZV model was successful with 1,800 traces (Fig. 7b). Furthermore, 1,080 traces are sufficient for a successful CPA attack with ZV power model (Fig. 7c). The correlation peak in Fig. 7(b) is related to the correct modified key guess, and the bold line is associated with the specific time that it has occurred. Here the first modified key byte, i.e. 122 (Dec) or 7A (Hex), are recovered. As the chart axis starts from one, the value 123 on this chart is equal to 122. By repeating the attack for next S-boxes, the next bytes of the modified key is recovered. This step of the attack corresponds to step 1

of the attack procedure, described in section 3.1.

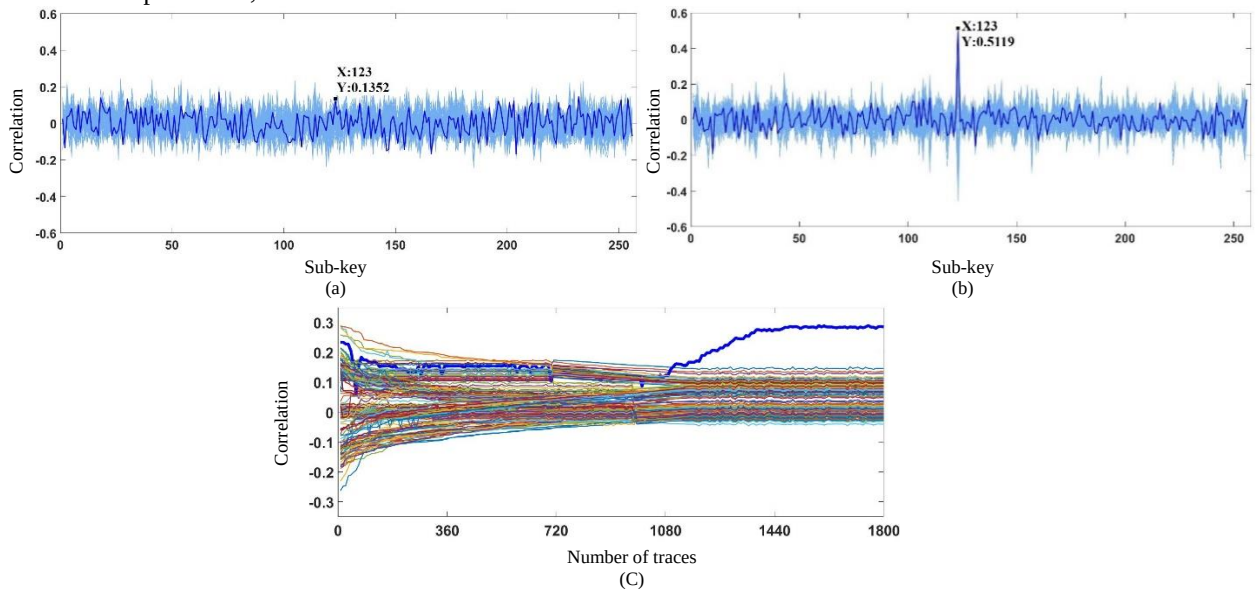
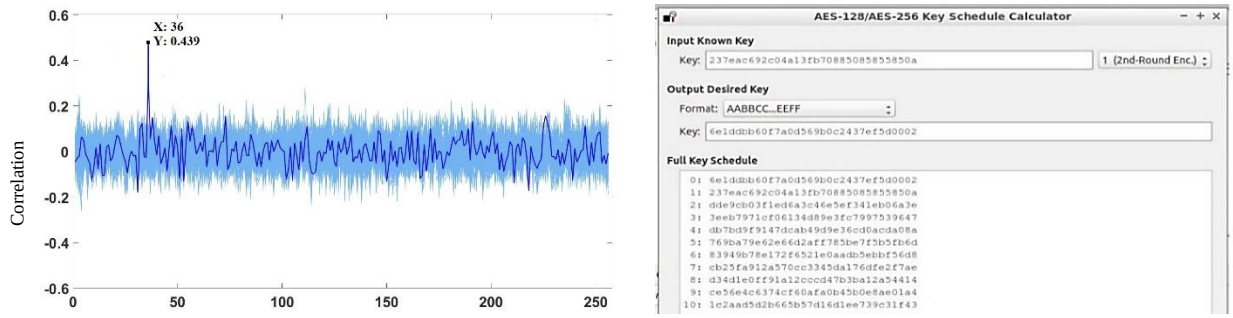


Fig. 7. CPA attack results on unprotected COLM using 1,800 measurements (K_0' is obtained) at time 1.67 μs (a) HD, (b) ZV model (c) over the traces number

When the 16 bytes of the initial round modified key is recovered, the attack is repeated at the first round of AES, based on step 2 of the attack procedure. According to Fig. 8(a), the first byte of the 1st round key is recovered, which is 35 (Dec) or 23 (Hex). As described above, 36 on the chart is 35. As the attack continues, the first round key is obtained that it is equal to K1=237EAC692C04A13FB70885085855850A. Finally, based on step 3 of the attack procedure, the encryption key is calculated by the key schedule computation software program as shown in Fig. 8(b) that is equal to K=6E1DDBB60F7A0D569B0C2437EF5D0002.



(b) key schedule calculator

Fig. 8. CPA attack results on unprotected COLM (a) first round key and (b) encryption key

4. A DOM Implementation of COLM

The previous works review shows that no protected architecture for COLM has been presented, so far. Therefore, in this section, using DOM masking scheme, a hardware architecture with two input and output shares is presented for COLM that makes it secure against first-order power analysis attacks. As mentioned in section 2.2, to protect the linear parts of the scheme, these parts are implemented in parallel s times, where s is equal to the number of input shares, but sharing the nonlinear section is not that simple.

The COLM units include AES, linear function ρ, Δ calculations and multiplication over $GF(2^{128})$. AES cipher has linear parts and the S-box as nonlinear part. The computation of ρ and Δ requires the multiplication with constant in the field that is linear [4]. As mentioned in section 2.3, COLM has three stages: generation of L, IV and tagged ciphertext. AES processes the AD and the plaintext/ciphertext and output enter to the second AES after passing through the function ρ. Finally, ciphertext/plaintext and the tag are generated. As each protected AES unit occupies a

large area, one protected AES is implemented and used serially. Fig. 9 shows the proposed first-order protected 8-bit hardware architecture for COLM. The masked AES unit is protected with DOM that is described in the following.

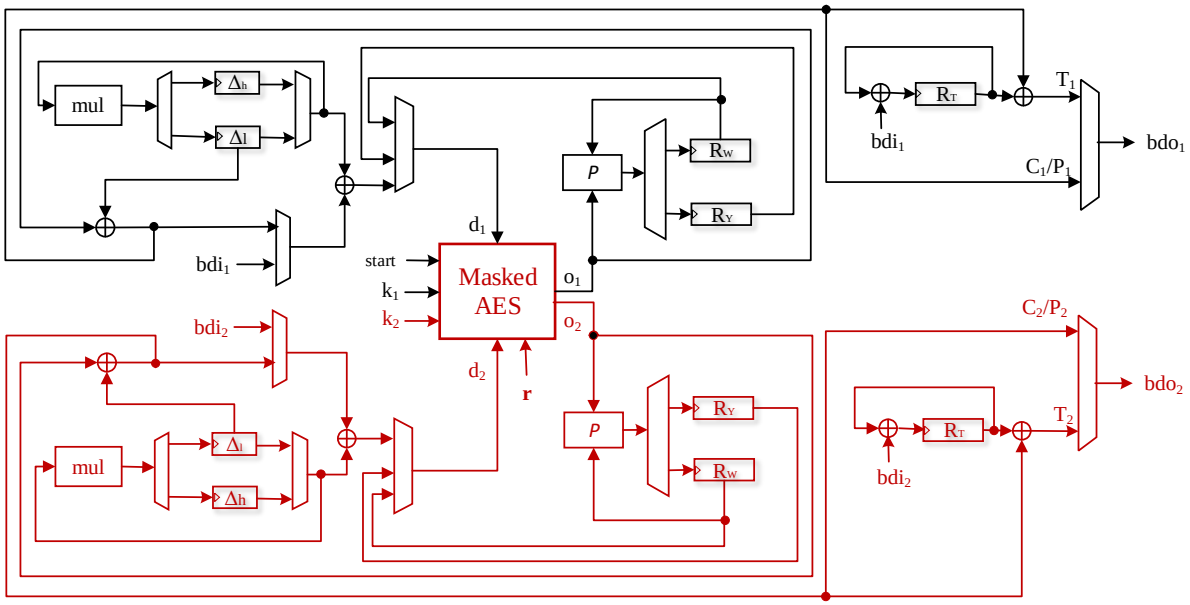


Fig. 9. The proposed 8-bit hardware architecture for protected COLM

Generating Δ requires multiplication by constants 2, 3, 7 and 49 on $GF(2^{128})$. Filed multiplication has the heavy computation, but multiplication by 2 (doubling) on this field can be reduced to a shift and a few XORs. Other multiplication can be calculated as $mul3(x) = mul2(x) \oplus x$; $mul7(x) = mul3(mul2(x)) \oplus x$; $mul49(x) = mul7(mul7(x))$. Two 128-bit registers Δ_l and Δ_h are used to store the output.

Function ρ has two input and two output as $Y = x \oplus 3.st$ and $W = x \oplus 2.st$. Calculation of this function requires multiplication by constant 2 and 3 that is done in the ρ unit. The outputs are stored in two 128-bit registers of R_W and R_Y, respectively. Because the function ρ and Δ should be computed simultaneously, $mul2(x)$ is implemented two times. The 128-bit registers R_T are responsible for keeping the sum of inputs to calculate the tag.

All units of COLM architecture, except AES, are linear. Thus these units should be implemented s times in protected architecture, where s is the number of shares. Because the AES that is described in following is protected using Dthe OM method with two shares, s is equal to 2.

4.1. Protected AES Architecture

So far, several first-order protected architecture has been presented for AES [12], [21], [24], [26]–[29]. Most of these architectures are based on TI masking scheme. As already has been mentioned, recently an optimizations building on TI introduced in [22] that decrease the area overhead as well as the required randomness that called DOM.

Table 1. Comparison of state-of-art first-order protected AES

Ref.	Masking scheme	S-box					AES		
		Input shares	Output shares	Latency (clk)	Area (KGE)	Fresh random	Share	Latency (clk)	Area (KGE)
[21]	TI	4	3	3	3.7	20+22 state	2	246	9.1
[24]	TI	3	3	4	4.2	44	3	266	11.1
[27]	TI	2	2	6	1.9	54	2	276	6.7
[30]	TI	3	3	3	2.9	20	-	-	-
[28]	TI	3	3	3	2.8	16	2	246	8.1
[31]	TI	2	4	5	1.4	64	2	219	6.3
[29]	TI	4	4	2	4.2	0	2	2804	7.6
[22]	**DOM**	**2**	**2**	**7**	**2.2**	**18**	**2**	**246**	**6**

Table 1 compares protected AES implementations in term of latency (the number of clocks), the area of S-box

and AES and the number of fresh random bits required per S-box. Reducing the number of required clocks will increase the throughput. Additionally, the smaller area of AES reduces the implementation cost of COLM. Producing random numbers in hardware increases the chip area and energy consumption, and also decreases the throughput of a design.

According to Table 1, in [29] the number of fresh random bits has reduced to zero, but the AES latency is 2804 clocks, that is very inefficient in term of throughput. The architecture proposed by Gross et al. [22] has the smallest area that is more efficient than other schemes. This architecture is similar to architecture proposed by Moradi et al. [24], but its S-box is protected with DOM masking scheme that has a smaller area and the less random bits. Masking the S-box as nonlinear part of AES is more complex and is the sensitive part.

Fig. 10 shows the first-order protected variant of Canright's AES S-box design architecture [32] using DOM masking approach [22], that is used to protect AES of COLM in this paper also. The AES S-box includes many linear operations such as mapping, square-scale, and non-linear operations such as multiplication in the field. In this architecture, multiplication operation over $GF(2^n)$ is converted to operation over sub-fields down to eight elements in $GF(2)$ using the tower field concept. These $GF(2^n)$ multipliers are replaced by some two-share masked AND gates, that was described in section 3-1.

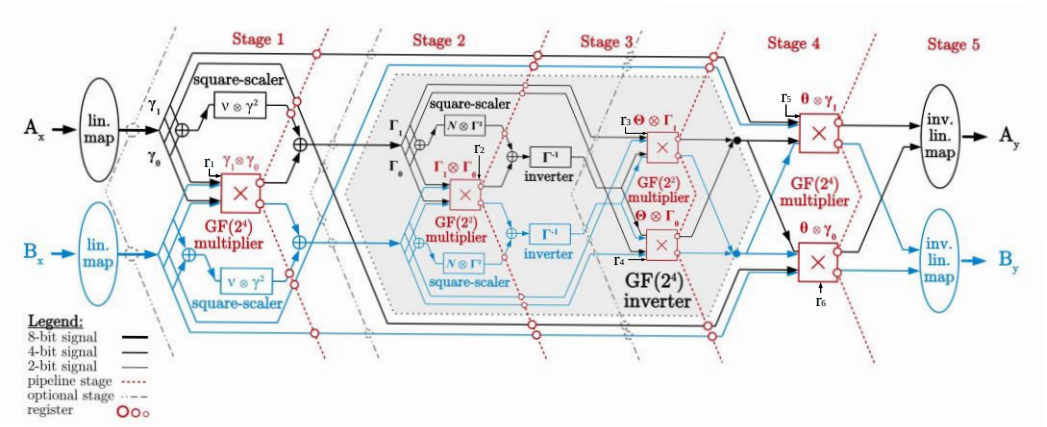


Fig. 10. First-order masked AES S-box using DOM implementation [22]

The protected S-box presented in [22] has seven stages of the pipeline to increase throughput. Pipeline registers are shown by red and gray circles, and lines in Fig. 10 separate theme. The red line is related to the pipeline stages of multiplication, which are numbered from 1 to 5. To establish the independence between inputs of adjacent multiplication gates in the presence of glitch, registers are added with gray color. The stages of masked S-box are as follows:

Stage 1: multipliers $GF(2^4)$ receive their inputs from the linear mapping. The linear mapping gets 8-bit inputs shares A_x and B_x and combines them in the own domain. Two registers are added after linear transformation to avoid glitch propagation.

Stages 2, 3: similar to the previous stage, the glitch can occur due to the combination of square-scalar outputs and multiplication gates. Therefore, some registers in gray color are added.

Stage 4: the inputs of this stage are the outputs of the GF gates and the S-box inputs that are independent and so do not require any register.

Stage 5: linear mapping in this stage is not critical, because the S-box outputs are stored in the state or key registers or fed into the next S-box that is prepared for processing the related sharing.

In the described scheme, every S-box running requires 18 fresh random bits. For each $GF(2^4)$ the multiplier, four fresh random bits are required and in total 12 bits are needed for three multipliers. The $GF(2^4)$ inverter has three $GF(2^2)$ multipliers, that each of them needs two fresh random bits, which in total, six bits are required for the inverter. Random bits are generated by a PRNG that is embedded within the AES core.

In the hardware architectures, usually, the AES core is implemented with 128-bit data-path and calculated in 10 clocks. However, it is not the case for the protected AES with the described architecture, for the following reasons:

- High area growth.
- Increase the number of required fresh random bits.

- Increase the vulnerability against power analysis attacks due to long combination path that leads to an increase in the glitches.

Therefore, to protect COLM, an AES with 8-bit architecture is implemented. This architecture is shown in Fig. 11. Using the mask m_k, the key is shared into two key k_1 and k_2. The plaintext is also divided into two shares of d_1 and d_2 using the mask m_p. Therefore, shares are stored in two state registers and two key registers. Additionally, two MCs is implemented. In the 10th-round of AES, as done signal becomes 1, two output shares are XORed and the AES output is obtained.

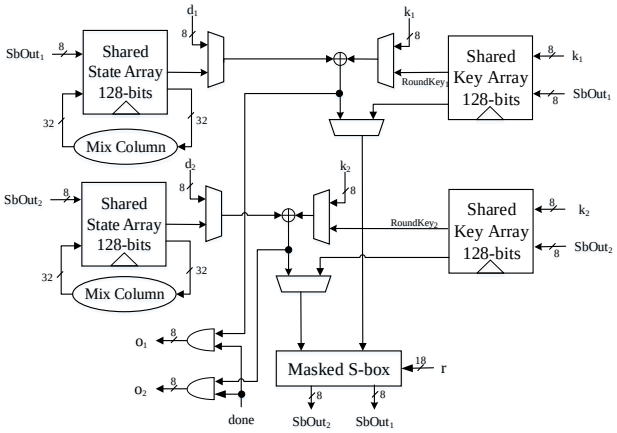


Fig. 11. Proposed Hardware architectures for DOM protected AES-128

5. Experimental Results of Secure COLM

To compare protected and unprotected COLM in term of hardware performance, this scheme is synthesized and implemented on FPGA platform. In addition, to show the resistance of the proposed architecture for COLM against power analysis attack, CPA attack and t-test are used on the protected scheme.

5.1. Performance Results

The protected and unprotected COLM are described in RTL level with VHDL language and synthesized and implemented on SAKURA-G board FPGA using Xilinx ISE v14.7 tool. The correctness of implementation is verified using the Mentor Graphics ModelSim 10.1.C tool and the test vectors presented in [33].

Table 2 and Fig. 12 compare the results for the unprotected and protected version of COLM in terms of LUT, slice, frequency (F_{max}) and throughput (Tp.). Given that, to the best of our knowledge, no power analysis attack against COLM is presented so far; thus a comparison with previous work was not possible. The implementation area of protected COLM increases two times compared with unprotected version. This factor for AES is 3.2. Also, throughput in protected implementation reduces by factor 1.89. The reduction for AES is 2.02. The maximum frequency in the protected version decrease by factor 1.87 that is because of the increase in the critical path. This factor for AES is 1.71.

Table 2. Hardware Implementation results of the proposed design for unprotected and protected COLM ('GR' and 'DR' means growth ratio And decrease ratio, respectively)

Design	Scheme	Area (LUT)	GR (Area)	Area (Slice)	GR (Area)	F_{max} (MHz)	DR (F_{max})	Tp. (Mbps)	DR (Tp.)
Unprotected	AES	228	-	92	-	278.7	1.71	170	2.02
	COLM	2296	-	991	-	149.6	1.87	38.94	1.89
Protected DOM	DOM- AES	738	3.2	219	2.4	163	-	84.3	-
	DOM-COLM	4894	2.1	2038	2	80.2	-	20.8	-

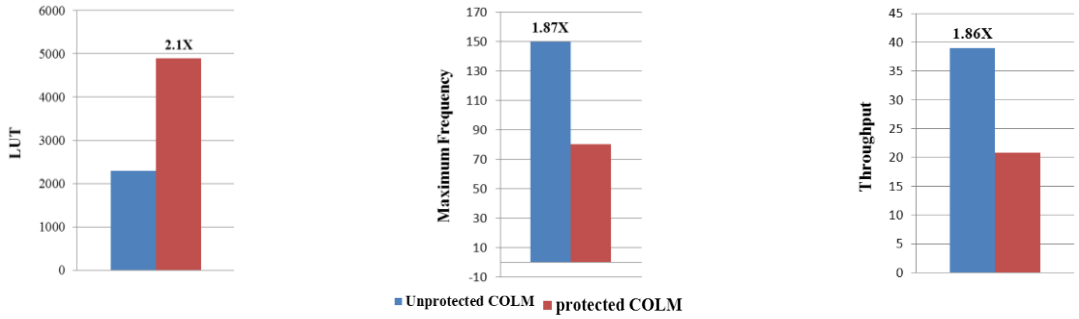


Fig. 12. Area (in LUTs), maximum frequency and throughput comparison ratios of unprotected and protected COLM

5.2. CPA attack and t-Test Results

In the protected COLM, the PRNG generates the new fresh random values in every running of the algorithm. Also, to share the plaintext, AD and Npub, the mask m is generated and renewed in every run. To minimize noise, the PRNG and COLM do not operate in parallel, and random bits is produced and stored before running the algorithm. Parallel operation increases the noise level and required more traces for a successful attack.

Similar to section 3-1, to assess the security of the protected COLM against power analysis attack, the nonlinear part of the scheme is attacked. As before, the ZV model for protected S-box output is selected, 1,080 traces are recorded. Then the first and second-order CPA attack is performed. The results of the attack on protected implementation indicate the first-order attack was failed (Fig. 13a), and on the correct key we do not have any correlation peak, while the second-order attack was successful (Fig. 13b). Therefore, proposed protected COLM is resistance against the first-order DPA attacks.

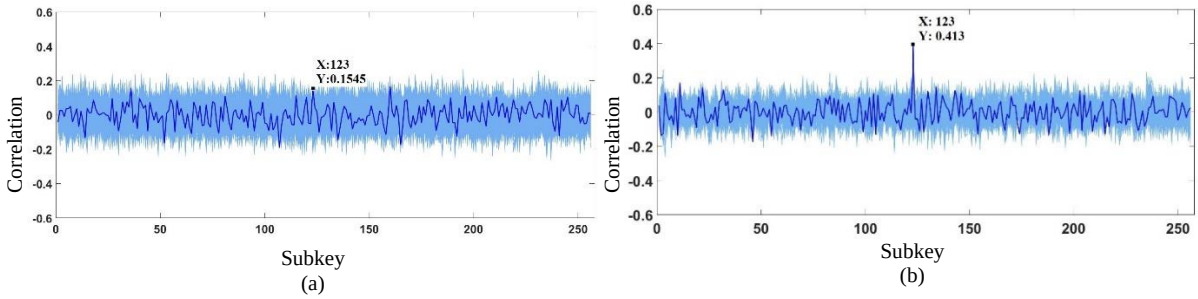


Fig. 13. CPA attack results on protected COLM using ZV model with 1,080 traces (a) the first-order attack and (b) the second-order attack

CPA attack is limited by the number of measurements and it requires determining the power model, which is a time-consuming task and it requires the knowledge of the underlying architecture. Recently, a leakage evaluation method has been presented in [34] and described in more detail in [35]. Also, this method is offered as a proposal to NIST to be a fast, robust and reliable evaluation of side-channel threads. This method uses Welch's t-test to identify the difference between the two distributions. This test can quickly find information leakage without launching the attack and knowing the underlying architecture, when device leaks information because of a mistake in countermeasures or a flaw in design engineering. However, it cannot be a complete substitution for the power analysis attack. For example, it cannot recover the key plaintext or intermediate values recovery. In addition, it does not provides any information on the correct power model or the severity of raising an attack. In Welch's t-test, the confidence factor t is calculated based on Equation (4).

$$t = (\mu_0 - \mu_1)/\sqrt{s_0^2/n_0 + s_1^2/n_1} \tag{4}$$

Where μ_0 and μ_1 are the mean of two distributions Q_0 and Q_1, s_0 and s_1 are the standard deviations, and n_0 and n_1 are the number of distribution samples. The probability of accepting the null hypothesis p is computed by $p = 2\int_{|t|}^{\infty} f(t)dt$ where $f(t)$ is the distribution function of the normal probability. To use t-test, we consider a null hypothesis and two distributions. For the null hypothesis with two distributions, the distributions will be

indistinguishable if it would not be possible to distinguish which distribution is given sample belongs to. If the threshold $|t| > 4.5$, the null hypothesis is rejected.

If the goal is only to show the information leakage without key recovery attack or showing the severity of the attack, then the non-specific t-test can be used. In this type of test, some fixed input data D (such as the plaintext, AD, or Npub) is preselected. Before each measurement a coin is flipped, and correspondingly fresh-randomly chosen data or D is given to the algorithm. This type is famous with a fixed vs. random test. This method is used to demonstrate the vulnerability of cipher algorithms as well as assess the effectiveness of power analysis countermeasures.

A non-specific t-test is used in this research to validate the countermeasure on COLM using DOM method. As inputs are in the form of two-shares, to collect the power traces, the fixed input (INPUT) is broken into two shares $(\text{INPUT}^1, \text{INPUT}^2)$ as suggested in [35]. Then the next inputs are calculated as Equation (5):

$$in_{i+1} = f(\text{INPUT}, \text{out}, random) = \begin{cases} (\text{INPUT} \oplus r^1, r^1) & if\ random_{bit}\ is\ 0 \\ (r^1, r^2) & if\ random_{bit}\ is\ 1 \end{cases} \quad (5)$$

Where $random_{bit}$ is equivalent to coin flipping and r is the random value of the mask. To perform t-test, 18,000 traces are collected, and analyzed using the ttest2 command in MATLAB and t values is calculated. In Fig. 14, the t values is depicted, where $|t| < 4.5$. This shows the effectiveness of the countermeasures on COLM against the first-order power analysis attack.

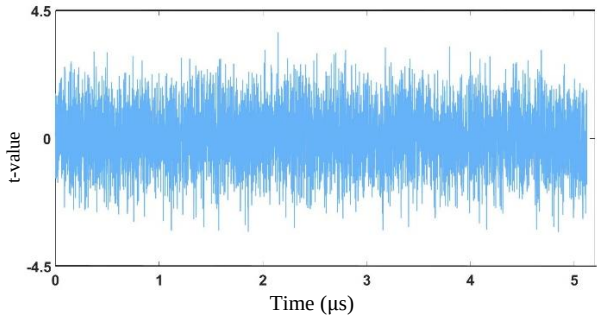


Fig. 14. Attack results on protected COLM t-test using 1,080 traces

6. Conclusions and Further Research

In this research, for the first time, we mount a CPA attack on the hardware implementation of the authenticated cipher COLM to determine the resistance against power analysis attack. As CLOM inputs are masked with Δ, it was not possible to attack the scheme directly. Therefore a three-steps attack procedure is proposed and implemented. The results of attack using laboratory equipment, SAKURA-G board and recording 1,800 traces, lead to successful key recovery.

A hardware architecture is proposed for COLM using DOM masking scheme to cope with this vulnerability. Results of FPGA implementation shows that the unprotected COLM has a usage area of 991 slices compared to 2038 slices for protected COLM. The area increases almost twice. In addition, the throughput decreases by a factor of 1.89 and the maximum frequency decreases by the factor 1.87. According to the results of CPA attack, the protected COLM with two shares using the DOM method provides resistance against the first-order CPA attack, but is not resist against the second-order CPA attack. Also, the non-specific t-test is performed that $|t| < 4.5$; thus the countermeasures is reconfirmed.

Some security analysis for the final winners of CAESAR competition are presented; however, the security of winners against the side-channel attacks heve been less studied yet. In addition, proposing an optimized protection scheme against CPA attack and comparing the cost of protection for the final winners could be a good direction for future research.

References

[1] D. Whiting, R. Housley, and N. Ferguson, “Counter with cbc-mac (ccm), RFC3610,” 2003.

[2] T. Krovetz and P. Rogaway, “The OCB authenticated-encryption algorithm, RFC 7253,” 2014. [Online]. Available: https://tools.ietf.org/html/rfc7253.

[3] D. McGrew and J. Viega, "The Galois/counter mode of operation (GCM)," *Submiss. to NIST Modes Oper. Process*, vol. 20, 2004.
[4] N. Ferguson, "Authentication weaknesses in GCM," *Comments submitted to NIST Modes of Operation Process*, 2005. [Online]. Available: https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/comments/cwc-gcm/ferguson2.pdf.
[5] H. Böck, A. Zauner, S. Devlin, J. Somorovsky, and P. Jovanovic, "Nonce-Disrespecting Adversaries: Practical Forgery Attacks on GCM in TLS.," *IACR Cryptology ePrint Archive, Report 2016/475*, 2016. [Online]. Available: https://eprint.iacr.org/2016/475.
[6] "CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustnes." [Online]. Available: http://competitions.cr.yp.to/caesar.html.
[7] E. T. and K. Y. A. Elena, A. Bogdanov, N. Datta, A. Luykx, B. Mennink, M. Nandi, "COLM v1.," *CAESAR competition proposal*, 2016. [Online]. Available: https://competitions.cr.yp.to/round3/colmv1.pdf.
[8] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Cryptology Conference*, 1999, pp. 388–397.
[9] A. Adomnicai, J. J. Fournier, and L. Masson, "Masking the Lightweight Authenticated Ciphers ACORN and Ascon in Software," *Cryptogr. Inf. Secur. Balk. Springer Int. Publ. Cham*, 2018.
[10] N. Samwel and J. Daemen, "DPA on hardware implementations of Ascon and Keyak," in *Proceedings of the Computing Frontiers Conference*, 2017, pp. 415–424.
[11] H. Gross, E. Wenger, C. Dobraunig, and C. Ehrenhöfer, "Ascon hardware implementations and side-channel evaluation," *Microprocess. Microsyst.*, vol. 52, pp. 470–479, 2017.
[12] S. Nikova, V. Rijmen, and M. Schläffer, "Secure hardware implementation of nonlinear functions in the presence of glitches," *J. Cryptol.*, vol. 24, no. 2, pp. 292–321, 2011.
[13] W. Diehl, A. Abdulgadir, F. Farahmand, J.-P. Kaps, and K. Gaj, "Comparison of cost of protection against differential power analysis of selected authenticated ciphers," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2018, pp. 147–152.
[14] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *International workshop on cryptographic hardware and embedded systems*, 2004, pp. 16–29.
[15] H. Gross, S. Mangard, and T. Korak, "Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order.," *IACR Cryptology ePrint Archive, Report 2016/486*, 2016. [Online]. Available: https://eprint.iacr.org/2016/486.
[16] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual information analysis," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2008, pp. 426–442.
[17] D. Agrawal, J. R. Rao, and P. Rohatgi, "Multi-channel attacks," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2003, pp. 2–16.
[18] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks: Revealing the secrets of smart cards*, vol. 31. Springer Science & Business Media, 2008.
[19] D. Canright and L. Batina, "A very compact 'perfectly masked' S-box for AES," in *International Conference on Applied Cryptography and Network Security*, 2008, pp. 446–459.
[20] A. Moradi, O. Mischke, and T. Eisenbarth, "Correlation-enhanced power analysis collision attack," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2010, pp. 125–139.
[21] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "A more efficient AES threshold implementation," in *International Conference on Cryptology in Africa*, 2014, pp. 267–284.
[22] H. Gross, S. Mangard, and T. Korak, "An efficient side-channel protected AES implementation with arbitrary protection order," in *Cryptographers' Track at the RSA Conference*, 2017, pp. 95–112.
[23] A. Moradi, "Advances in side-channel security." Habilitation thesis, Ruhr-Universität Bochum, 2016.
[24] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the limits: a very compact and a threshold implementation of AES," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2011, pp. 69–88.
[25] "Side-channel AttacK User Reference Architecture." [Online]. Available: http://satoh.cs.uec.ac.jp/SAKURA/hardware.html.
[26] J. Jaffe, "A first-order DPA attack against AES in counter mode with unknown initial counter," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2007, pp. 1–13.
[27] T. De Cnudde, O. Reparaz, B. Bilgin, S. Nikova, V. Nikov, and V. Rijmen, "Masking AES with d+ 1 Shares in Hardware," in *International Conference on Cryptographic Hardware and Embedded Systems*, 2016, pp. 194–212.
[28] B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, and V. Rijmen, "Trade-offs for threshold implementations

illustrated on AES," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 34, no. 7, pp. 1188–1200, 2015.
[29] F. Wegener and A. Moradi, "A First-Order SCA Resistant AES Without Fresh Randomness," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*, 2018, pp. 245–262.
[30] A. Ghoshal and T. De Cnudde, "Several masked implementations of the Boyar-Peralta AES s-box," in *International Conference in Cryptology in India*, 2017, pp. 384–402.
[31] R. Ueno, N. Homma, and T. Aoki, "Toward more efficient DPA-resistant AES hardware architecture based on threshold implementation," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*, 2017, pp. 50–64.
[32] D. Canright, "A very compact S-box for AES," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2005, pp. 441–455.
[33] "GMU Implementations of Authenticated Ciphers, George Mason University, U.S.A." [Online]. Available: https://cryptography.gmu.edu/athena/index.php?id=CAESAR.
[34] G. Becker *et al.*, "Test vector leakage assessment (TVLA) methodology in practice," in *International Cryptographic Module Conference*, 2013, vol. 1001, p. 13.
[35] T. Schneider and A. Moradi, "Leakage assessment methodology," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2015, pp. 495–513.