The ISC Int'l Journal of Information Security

July 2020, Volume 12, Number 2 (pp. 67–80)

http://www.isecure-journal.org

CPA on COLM Authenticated Cipher and the Protection Using Domain-Oriented Masking

Mohsen Jahanbani [1], Nasour Bagheri [2,3,*] and Zeinolabedin Norouzi [1]

[1] *Department of Electrical Engineering, Imam Hossein University, Tehran, Iran*

[2] *Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran 16788-15811, Iran*

[3] *School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran*

ARTICLE INFO.

Article history:
Received: 28 June 2019
Revised: 18 February 2020
Accepted: 9 July 2020
Published Online: 11 July 2020

Keywords:
Authenticated Cipher, COLM, Correlation Power Analysis , Domain-oriented Masking.

Abstract

Authenticated encryption schemes are important cryptographic primitives that received extensive attention recently. They can provide both confidentiality and authenticity services, simultaneously. Correlation power analysis (CPA) can be a thread for authenticated ciphers, similar to the any physical implementation of any other cryptographic scheme. In this paper, a three-step CPA attack against COLM, one of the winners of CAESAR competition, is presented to indicate its vulnerability. To validate this attack, COLM is implemented on the FPGA of the SAKURA-G board. A successful CPA attack with zero value power model is mounted by measuring and collecting 1,800 power traces. In addition, a protected hardware architecture for COLM is proposed to make this design secure against first-order CPA attacks, where a domain-oriented masking (DOM) scheme with two-input/output shares is used to protect it. To verify these countermeasures, we mount first and second-order CPA attacks and a non-specified t-test on the protected COLM.

1 Introduction

Authenticated encryption (AE) schemes provide confidentiality and authenticity of plaintext simultaneously. The traditional way to achieve such properties is the combination of several cryptographic primitives, which usually are encryption algorithms for confidentiality and message authentication codes (MACs) for message integrity. However, this approach is not optimal and may be accompanied by flaws 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 codebook mode (OCB) [2]. Moreover, 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, the Competition for Authenticated Encryption âĂŞ Security, Applicability, and Robustness (CAESAR) [6] was started to select a portfolio of AE that (1) provides advantages over AES-GCM and (2) is 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 were expected to be optimized and ultimately selected in the final round. In March 2019, six winners were selected for these applications, include COLM [7] as a winner of the competition.

* Corresponding author.
Email addresses: mjahanbani@ihu.ac.ir, nbagheri@srttu.edu, znorozi@ihu.ac.ir

ISSN: 2008-2045

The COLM authenticated cipher was selected for in-depth defense in nonce-misuse. It provides strong security features, such as security against nonce-misuse adversaries and security under release of unverified plaintext. In the latter scenario, the decrypted ciphertext is accessible by an adversary before that the related authentication tag is verified. In addition, 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, which is also announced by the competitionâĂŹs committee. Therefore, it is favorable to evaluate the resistance of CAESAR winners against the differential power analysis (DPA) attacks [8], and also to determine the protection costs. Nevertheless, no side-channel attack on COLM has been presented so far, and there is no protection scheme on COLM.

1.1 Related Work

Adomnicai *et al.* [9] investigated the resistance of both lightweight winners, ACORN and Ascon, CAESAR competition 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 the Ascon cipher could be performed at the initialization and finalization stages. In addition, ACORN is based on stream ciphers, and the keystream is calculated independently of the plaintext, that makes 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 introduced a system of boolean equations to solve this problem, that is an All-SAT problem. Their results justify the need for countermeasures at the software implementation level. Therefore, they presented two masking schemes to protect ACORN.

Samwel and Daemen [10] presented a successful power analysis attack on Keyak and Ascon. Both ciphers have sponge construction and use the same type of S-box. Then, they added a linear layer after the S-box to make this attack much harder. Gross *et al.* [11] proposed several hardware implementations for Ascon. They showed that this cipher 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 the third round and CAESAR winners include CLOC, JAMBU, Ascon, Ketje Jr, SILC, and ACORN to first-order DPA using the t-test leakage detection methodology. However, 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 regarding the area, frequency, and throughput.

Jahanbani *et al.* [14] used power leakage detection t-test to indicate the vulnerability of OCB and COLM. Then, they protect OCB and COLM with TI masking scheme. They measure the effectiveness of their countermeasures by the first and second-order t-test. However, they did not present a power analysis attack to recover the correct key and did not protect COLM with other masking schemes.

1.2 Contribution

Security against the side-channel attacks was one of the security goals in the CAESAR competition [6]. COLM, as a CAESAR winner, offers a strong security guarantee. However, side-channel key recovery attacks has not been investigated for this cipher. In this research, we present a power analysis attack scheme against COLM. As inputs of COLM are masked with a mask value Δ , the power analysis attack is more challenging compared to block ciphers. To solve this problem, a three-step attack approach is presented. The proposed method is implemented on FPGA. Then, the traces of power are recorded, and the key is recovered using correlation power analysis (CPA) [15] 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) [16]. In this method, the sensitive variable is shared and placed into separate domains. The proposed protected COLM uses two input/output shares. The resistance of our protected version against the first-order attack is verified by the CPA and t-test.

1.3 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 schemes in FPGA is compared. Also, the security of protected COLM is evaluated by first and second-order attacks using t-test. Finally, this paper is concluded in Section 6.

2 Background Information

In this section, the concept of power analysis attack, countermeasures, 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 a power analysis attack that reveals the secret key using the consumed power leakage. Power analysis attack has different kinds, including simple power analysis (SPA), DPA [8], CPA [15], mutual information analysis [17] and template-based attack [18]. 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 the 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. The better power model needs less trace for a successful 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 registers are updated. In addition, the zero value (ZV) power model is helpful when combinatorial circuit such as S-box consume the power. This model assumes that data value 0 has less power consumption than other values [19]. A higher-order power analysis 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 [20]:

- 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 software implementation, the intermediate values are processed in 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 intermediate values are processed simultaneously, the preprocessing function is applied to a single point in the trace. This case typically occurs in hardware implementation. For a univariate second-order attack, squaring the power traces is an appropriate preprocessing function [19].

2.2 Masking

The hardware countermeasures are classified as hiding and masking schemes 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 $(x^1, ..., x^s)$ such that $x = \bigoplus_{i=1}^{s} x^i$. Due to the boolean structure of masks, it is easy to apply a linear function $L(.)$ over shares because of $L(x) = \bigoplus_{i=1}^{s} L(x^i)$. However, the implementation of a non-linear function $F(.)$ by shares representation is very hard since $F(x) \neq \bigoplus_{i=1}^{s} F(x^i)$. Although this masking scheme is applied in the hardware implementation of AES with $s = 2$ [21], that was not successful due to glitch in hardware [22]. To solve this problem, two masking approaches have been proposed so far: threshold implementation (TI) [12] and DOM [16].

2.2.1 Threshold Implementation

In 2011, TI was introduced based on mathematical foundations including threshold Boolean secret sharing and 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 [23]:

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

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

TI scheme is secure if the three properties correctness, non-completeness, and uniformity are satisfied as follows:
Correctness: suppose TI representation of the function $\bar{y} = F(\bar{x})$ is intended, which $\bar{x} = (x^1, ..., x^s)$ is the shared representation of input x with s shares such that $x = \bigoplus_{i=1}^{s} x^i$, then for output shares $\bar{y} = (y^1, ..., y^n)$ we should have $y = \bigoplus_{i=1}^{n} y^i$ to ensure correctness property. For this purpose, we can use component functions $f^{i \in \{1,...,n\}}(\bar{x}) = y^i$ to calculate F. Although, it is not easy to find these component functions when function F is non-linear.
Non-completeness: The function F is d^{th}-order non-complete if each combination of d component functions f^i is independent of at least one input share x^i, where d is a security order.
Uniformity: the security of a Boolean masking scheme is based on the uniform distribution of the shares and masks. If the input of a TI function is shared

uniformly, the output should be a uniform sharing as well, because the output of a nonlinear function such as S-box is used as the input of the next function (for example the next round of cipher). Hence, given all possible input shares $\mathcal{X} = \{\bar{x}| \bigoplus_{i=1}^{s} \bar{x} = x\}$,all output shares set $\{f^1, ..., f^n|\bar{x} = \mathcal{X}\}$ should be selected uniformly from set $\mathcal{Y} = \{\bar{y}| \bigoplus_{i=1}^{s} \bar{y} = y\}$ as all possible shares of $y = F(x)$. Providing uniformity for nonlinear functions, especially functions with a high algebraic degree, is a challenge ahead of this structure. One solution is the re-masking using uniform fresh random in case of the non-uniform output of component functions [23, 24]. This value is produced by a pseudo-random number generators (PRNG). For example, if the TI version of multiplication $z = xy$ in $GF(2^m)$ is desirable then products $x_i y_i$, for $i, j \in \{1, 2, , d + 1\}$, are calculated, i.e. $(d + 1)^2$ components are computed. To calculate the output, the number of output shares should be compressed securely. The first-order TI for this multiplication with $t = 1,\ d = 1,\ s_{in},\ s_{out} = 3,\ r = 2$ is described as Equation 2 [8]:

$$\begin{aligned} z_1 &= x_1y_1 \oplus x_1y_2 \oplus x_2y_1 \oplus r_1 \oplus r_2 \\ z_2 &= x_2y_2 \oplus x_2y_3 \oplus x_3y_2 \oplus r_1 \\ z_3 &= x_3y_3 \oplus x_1y_3 \oplus x_3y_1 \oplus r_2 \end{aligned} \tag{2}$$

Note that three shares have been used to achieve the first-order security and uniformity was obtained by adding two fresh random r_1 and r_2.

2.2.2 Domain-Oriented Masking

Another masking scheme called DOM [25] 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 value is added to these terms to keep them independent. In addition, 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 Figure 1. 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

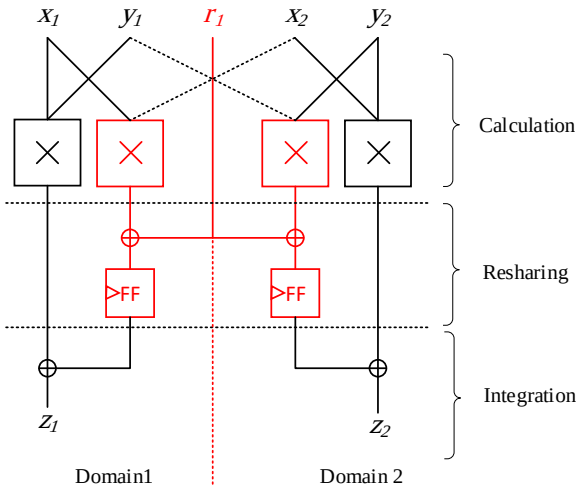


Figure 1. First-order secure AND gate [25]

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.

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 Figure 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). In COLM, E_K is AES cipher. 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 $C[l + 1] = C^{'}[l + 1]$.

3 CPA Attack against COLM

Mounting power analysis attack requires knowing the attack points on the scheme. According to Figure 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. In addition, the secret key should be combined with that part. Since the inputs of the E_K in the lower part of COLM is unknown and the key is not as an input in the function ρ, these points are not good choice. In addition, the input of E_K in AD and plaintext processing is combined with an unknown and variable value Δ in each input block that is not a proper choice. In the nonce processing, Npub is combined with unknown but constant value Δ and unknown

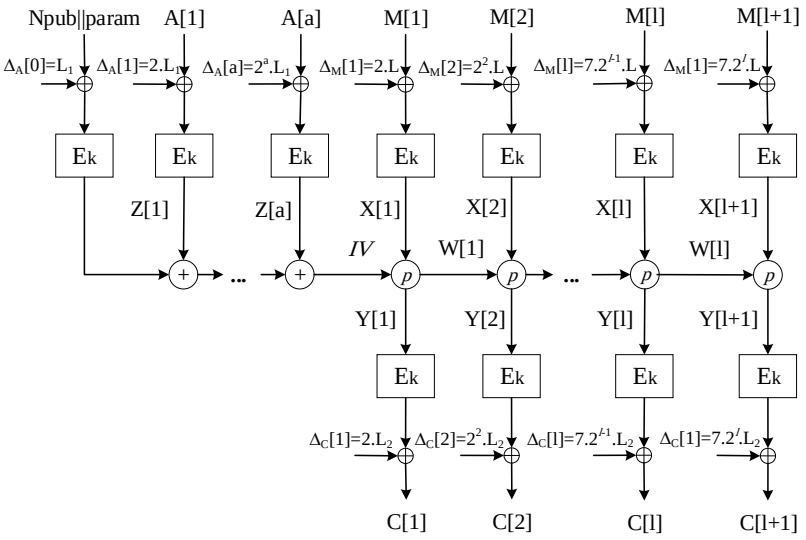


Figure 2. COLM authenticated encryption mode of operation [7]

key. In addition, Npub is concatenated with 8 bytes constant. These constant bytes cause the key bytes cannot be recovered with a usual power analysis attack. 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 modified first round key. Then input of the second round of AES is calculated and the third CPA attack can recover the second round key. Given this key, we can recover the actual key. In the next subsection, 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 nonce processing of COLM to recover the encryption key K. If Z_i denote *AddRoundkey* output state, W_{ij} denote S-box output bytes and X_{ij} denote unknown bytes in MixColumn output, then attack details is described as following steps:

Step 1: First CPA attack. Power traces are recorded for the first 3 rounds of AES for different Npub block. The S-box input of the first round of AES is $Z_0 = AddRoundKey(K_0, N \oplus \Delta_N) = K_0 \oplus N \oplus \Delta_N$, where N is the Npub. By rolling Δ_N to key, the S-box input can be rewritten as $Z_0 = AddRoundKey(\widetilde{K}_0, N) = \widetilde{K}_0 \oplus N$. When CPA attack targets the first round, first 8 bytes of the modified first round key $\widetilde{K}_0 = K_0 \oplus \Delta_N$ is recovered (attack point 1 in Figure 3).

Step 2: Second CPA attack. The first 8 bytes of $\widetilde{K}_0$ are known. Therefore, the S-box output of the first round for the first 8 bytes is known and the

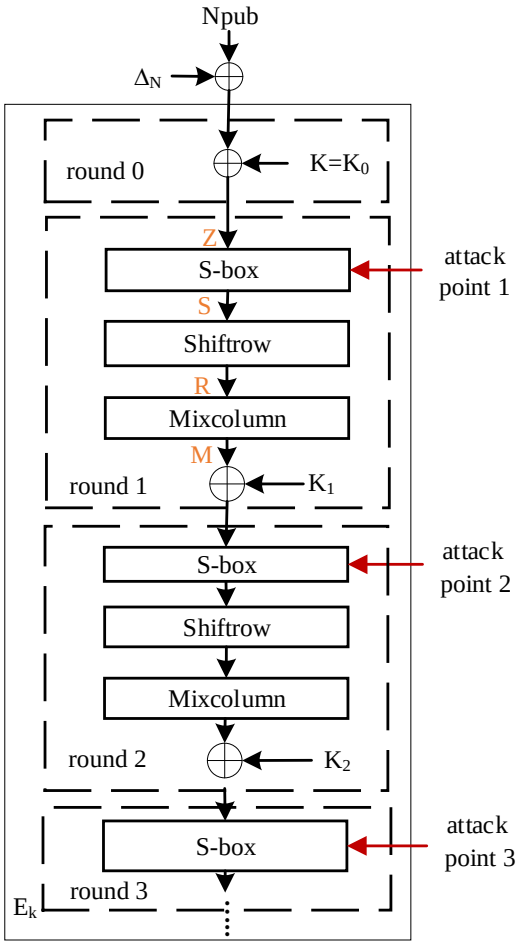


Figure 3. Attack procedure on the 3 round of AES in COLM

second 8 bytes is constant and unknown based on Equation 3a. The *ShiftRow* and *MixColumn* are calculated according to Equations (3b) and (3c). Finally, the first round *AddRoundkey* is calculated based on Equation 3d so that $X_{1,j}$ is rolled to part of the first

round key K_1. By the CPA attack on the second round S-box (attack point 2 in Figure 3), all bytes of this modified key $\widetilde{K}_1$ are recovered.

$$R_1 = \begin{bmatrix} W_{1,0} & W_{1,4} & ? & ? \\ W_{1,5} & ? & ? & W_{1,1} \\ ? & ? & W_{1,2} & W_{1,6} \\ ? & W_{1,3} & W_{1,7} & ? \end{bmatrix} \tag{3a}$$

$$S_1 = \begin{bmatrix} W_{1,0} & W_{1,4} & ? & ? \\ W_{1,1} & W_{1,5} & ? & ? \\ W_{1,2} & W_{1,6} & ? & ? \\ W_{1,3} & W_{1,7} & ? & ? \end{bmatrix} \tag{3b}$$

$$M_1 = \begin{bmatrix} 2W_{1,0} \oplus 3W_{1,5} \oplus X_{1,1} & W_{1,0} \oplus 2W_{1,9} \oplus X_{1,2} & \cdots \\ 2W_{1,4} \oplus W_{1,3} \oplus X_{1,5} & \cdot & \cdots \\ W_{1,2} \oplus W_{1,7} \oplus X_{1,9} & \cdot & \cdots \\ \cdot & \cdot & \cdots \end{bmatrix} \tag{3c}$$

$$Z_1 = AddRoundKey(W_{1,j} \oplus X_{1,j}, K_1) = (W_{1,j} \oplus X_{1,j}) \oplus K_1 = W_{1,j} \oplus (X_{1,j} \oplus K_1) = W_{1,j} \oplus \widetilde{K}_1 \tag{3d}$$

Step 3: Third CPA attack. By recovered $\widetilde{K}_1$ in step 2, the output of the first round is calculated based on Equation 3d that is as second round input. The output of S-box, *ShiftRow* and *MixColumn* of the second round are calculated Equation 4.

$$R_2 = \begin{bmatrix} W_{2,0} & W_{2,4} & W_{2,8} & W_{2,12} \\ W_{2,5} & W_{2,9} & W_{2,13} & W_{2,1} \\ W_{2,10} & W_{2,14} & W_{2,2} & W_{2,6} \\ W_{2,1} & W_{2,3} & W_{2,7} & W_{2,11} \end{bmatrix} \tag{4a}$$

$$S_2 = \begin{bmatrix} W_{2,0} & W_{2,4} & W_{2,8} & W_{2,12} \\ W_{2,1} & W_{2,5} & W_{2,9} & W_{2,13} \\ W_{2,2} & W_{2,6} & W_{2,10} & W_{2,14} \\ W_{2,3} & W_{2,7} & W_{2,11} & W_{2,1} \end{bmatrix} \tag{4b}$$

$$M_2 = \begin{bmatrix} 2W_{2,0} \oplus 3W_{2,5} \oplus W_{2,10} \oplus W_{2,1} & \cdots \\ W_{2,0} \oplus 2W_{2,5} \oplus 3W_{2,10} \oplus W_{2,1} & \cdots \\ \cdot & \cdots \\ \cdot & \cdots \end{bmatrix} \tag{4c}$$

$$Z_2 = AddRoundKey(W_{2,j}, K_2) = W_{2,j} \oplus K_2 \tag{4d}$$

The second round output (third round input) is calculated based on Equation 4. By the CPA attack on the third round S-box (attack point 3 in Figure 3), all bytes of K_2 are recovered. The master encryption key K is recovered from round key K_2 by running the AES key schedule in inverse mode.

3.2 Practical CPA Attack Results

To implement the CPA attack on COLM, the hardware implementation of AES is required. The hardware architecture proposed in [24] is used to implement AES as shown in Figure 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 a 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 same input will not be precisely same. Hence, the best choice could be the serial structure.

The required equipment for the CPA attack includes FPGA, digital oscilloscope, and PC. SAKURA-G board [26] is one of the most popular boards used in side-channel attacks. This board includes two FPGAs 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 an ultra-low-noise design, and an onboard amplifier that 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 Figure 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 [26]. Figure 5 shows the setup used for capturing the trace and CPA attack.

The crypto FPGA is clocked at 1 MHz by an 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

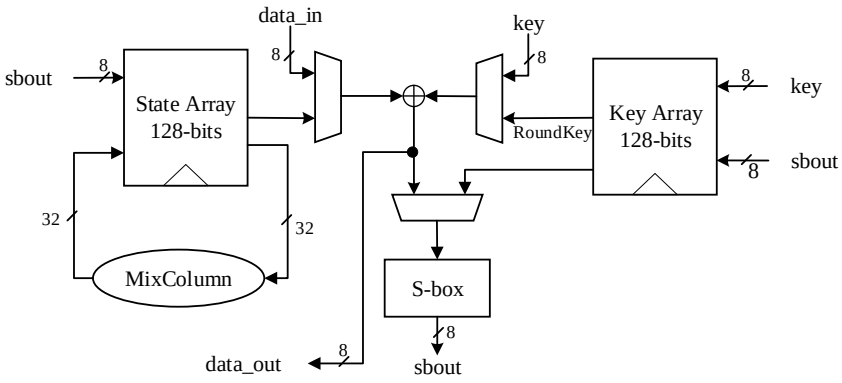


Figure 4. AES architecture for the attack against COLM [24]

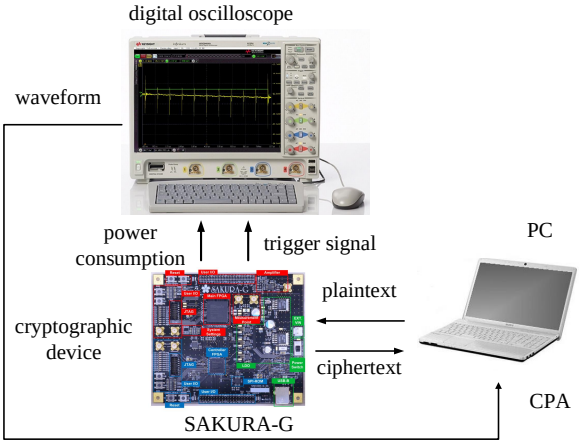


Figure 5. Setup for capturing trace with the SAKURA-G

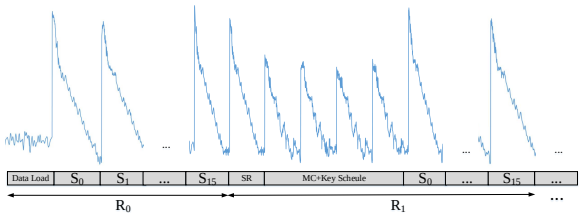


Figure 6. A measured power trace for the initial and first round of unprotected AES

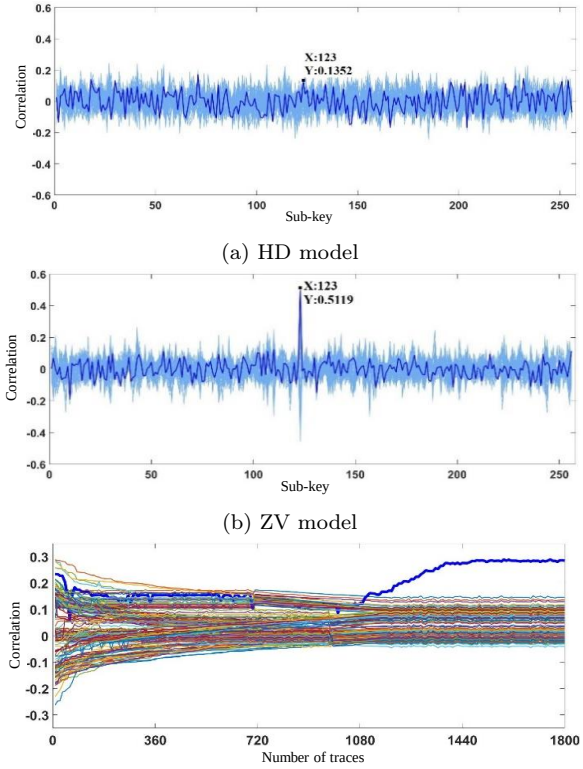
that consumes the power and disturbs the measurements. Thus all those LED switched off. To synchronize oscilloscope with FPGA, a trigger signal is generated by FPGA and sent to the oscilloscope. Additionally, to communicate between the SAKURA-G board, PC and oscilloscope, an interface software program has been developed in C# language. This program generate the required plaintext and key and send them to FPGA and receive 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. Figure 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 Npub and the S-box output is calculated. In the first round operation, *ShiftRow*, *MixColumn*, *AddRoundKey*, and S-box of the next round are performed, respectively.

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 as shown in Figure 7a was not successful, while the attack on S-box output by ZV model was successful with 1,800 traces (Figure 7b). Furthermore, 1,080 traces are sufficient for a successful CPA attack with ZV power model (Figure 7c). The correlation peak in Figure 7b 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), is recovered. As the chart axis starts from one, the value 123 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. The CPA attack was performed using the ZV power leakage model on the 1st round S-box. By repeating the attack, the bytes of 0 to 7 modified key are recovered as $\widetilde{K}_0=\widetilde{K}$=8005A1D8B9EF6B94 - - - - - - - - - - - - - - - -.

After recovering 8 bytes of modified key, the CPA attack on 2st round S-box (step 2) is repeated for all bytes and the modified 1st round key is equal to $\widetilde{K}_1$= 7C67050C25A39B405D28634FF8E4D13E. After recovering all modified key bytes of the 1st round, the third CPA attack on 3st round S-box (step 3) is repeated for all key bytes and the second key is recovered K_2=DDE9CB03F1ED6A3C46E5EF341EB06A3E. Finally, the encryption key is calculated by the key schedule computation software program [27], equal to K= 6E1DDBB60F7A0D569B0C2437EF5D0002.

4 A DOM Implementation of COLM

The previous works review shows that no protected architecture for COLM has been presented, so far.



(a) HD model

(b) ZV model

(c) Over the traces number

Figure 7. CPA attack results on unprotected COLM using 1,800 measurements (k'_0 is obtained) at time 1.67 μs

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 a 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. Figure 8 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.

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_1 and Δ_h are used to store the output.

Function ρ has two inputs and two outputs 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 the following is protected using the DOM method with two shares, s is equal to 2.

4.1 Protected AES Architecture

So far, several first-order protected architectures has been presented for AES [12], [23], [24], [28–31]. Most of these architectures are based on TI masking scheme. As already has been mentioned, an optimization building on TI introduced in [25] that decreases the area overhead as well as the required randomness called DOM.

Table 1 compares protected AES implementations in terms 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 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 decreases the throughput of a design.

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)
[23]	TI	4	3	3	3.7	20+22	2	246	9.1
[24]	TI	3	3	4	4.2	44	3	266	11.1
[29]	TI	2	2	6	1.9	54	2	276	6.7
[32]	TI	3	3	3	2.9	20	–	–	–
[30]	TI	3	3	3	2.8	16	2	246	8.1
[33]	TI	2	4	5	1.4	64	2	219	6.3
[31]	TI	4	4	2	4.2	0	2	2804	7.6
[25]	**DOM**	**2**	**2**	**7**	**2.2**	**18**	**2**	**246**	**6**

According to Table 1, the number of fresh ran-

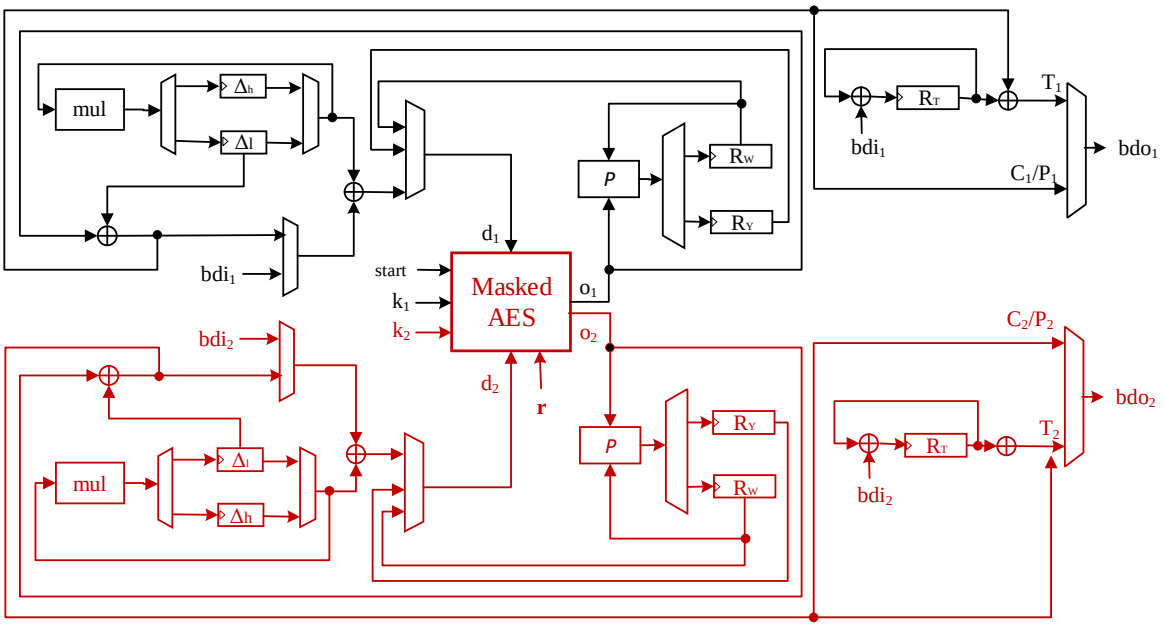


Figure 8. The proposed 8-bit hardware architecture for protected COLM

dom bits in [31] 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.* [25] 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 a nonlinear part of AES is more complex and is the sensitive part.

Figure 9 shows the first-order protected variant of CanrightâĂŹs AES S-box design architecture [34] with DOM masking approach [25], which 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 like 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 2.2.

The protected S-box presented in [25] has seven stages of the pipeline to increase the throughput. In Figure 9, the pipeline registers are shown by red and gray circles. The lines separate these registers. 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, the 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 their 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.

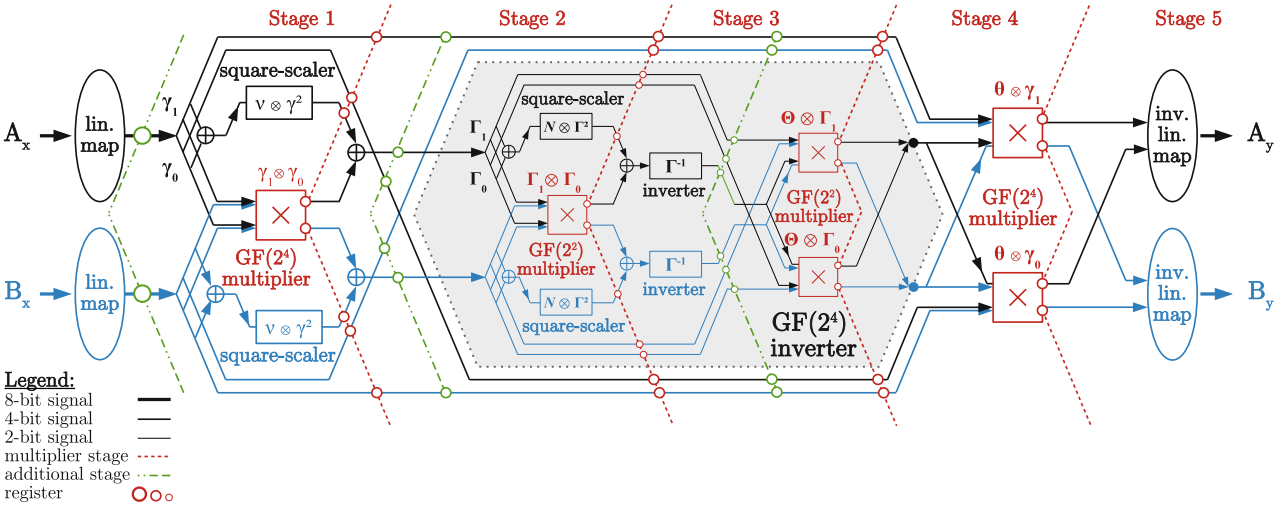


Figure 9. First-order masked AES S-box using DOM implementation [25]

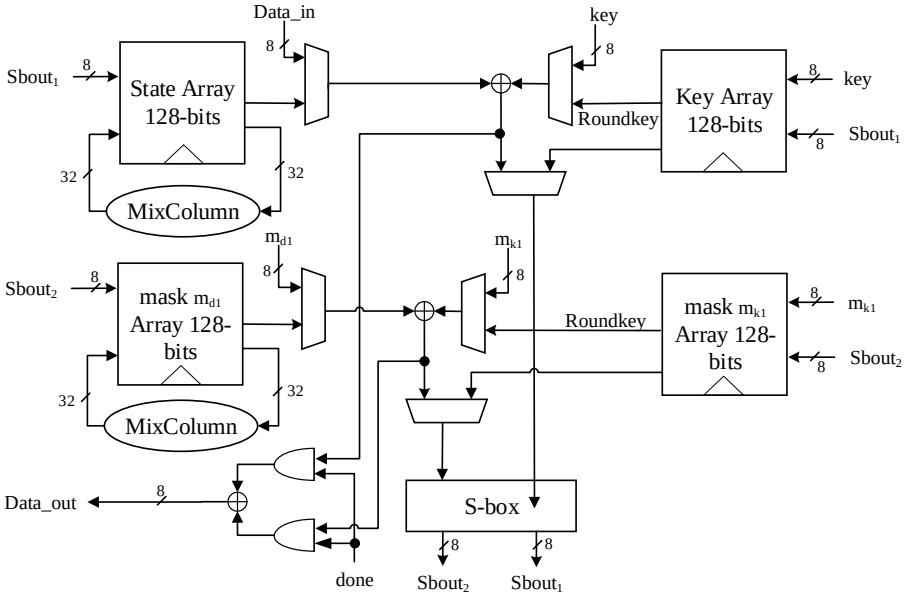


Figure 10. Proposed hardware architectures for DOM protected AES-128

- Increase required fresh random bits number.
- Increase the vulnerability against power analysis attacks due to a 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 Figure 10. Using the mask n_k, the key is shared into two key k_1 and k_2. The Npub is also divided into two shares of d_1 and d_2 using the mask n_p. Therefore, shares are stored in two state registers and two key registers. Additionally, two *MixColumns* are implemented. In the 10^{th}-round of AES, as done signal becomes 1, two output shares are XORed and the AES output is obtained.

5 Experimental Results of Protected COLM

To compare protected and unprotected COLM in terms of hardware performance, these schemes 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.

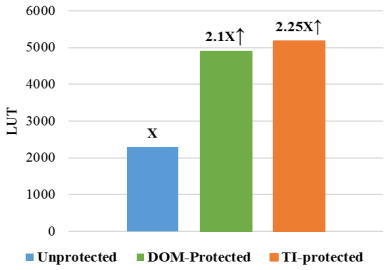
5.1 Performance Results

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

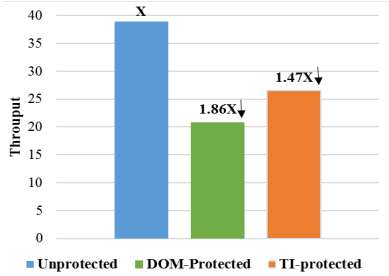
10.1.C tool and the test vectors presented in [35]. Table 2 and Figure 11 compare the results for the unprotected and protected version of COLM in terms of LUT, slice, frequency (F_{max}) and throughput (Tp). Given that no power analysis attack against COLM is presented so far; thus a comparison with previous work was not possible.

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	Area (LUT)	GR (Area)	Area (Slice)	GR (Area)	F_{max} (MHz)	DR (Fmax)	Tp (Mbps)	DR (Tp)
Unprotected	2296	–	991	–	149.6	–	38.94	–
DOM-Protected	**4894**	**2.1**	**2038**	**2**	80.2	1.87	20.8	1.86
TI-Protected [14]	5176	2.25	2379	2.4	87.3	1.71	26.5	1.47



(a) Area



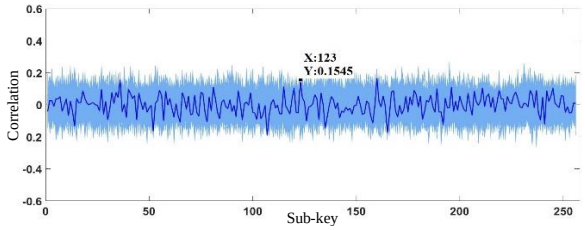
(b) Throughput

Figure 11. Area (in LUTs) and throughput comparison ratios of unprotected and protected COLM

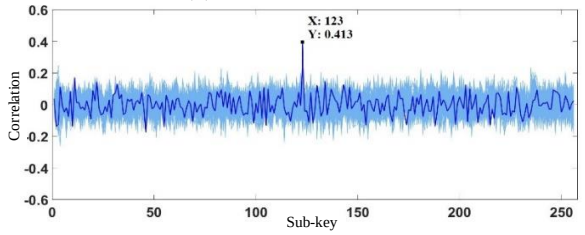
Results show the implementation area of DOM-protected COLM increases two times compared with the unprotected version. Also, throughput in protected implementation reduces by factor 1.89. The maximum frequency in the protected version decrease by factor 1.87 that is because of the increase in the critical path. In addition, comparison results with [14] show that DOM-protected COLM has the lower area and TI-protected has higher throughput. Therefore, in lightweight application that area is important, DOM-protected COLM is more suitable.

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 n 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 (Figure 12a), and on the correct key we do not have any correlation peak, while the second-order attack was successful (Figure 12b). Therefore, the proposed protected COLM is resistant to the first-order DPA attacks.



(a) First-order attack



(b) Second-order attack

Figure 12. CPA attack results on protected COLM using ZV model with 1,080 traces

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. A leakage evaluation method has been presented in [36] and described in more detail in [37]. 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 provide 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 5.

$$t = (\mu_0 - \mu_1)/\sqrt{(s_0^2)/n_0 + (s_1^2)/n_1} \qquad (5)$$

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 ρ is computed by $\rho = 2\int_{|t|}^{\infty} f(t)dt$ 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 a key recovery attack or showing the severity of the attack, then the non-specific t-test can be helpful. 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 $(INPUT^1, INPUT^2)$ as suggested in [37]. Then the next inputs are calculated as Equation 6:

$$in_{i+1} = f(INPUT, out, random) = \begin{cases} (INPUT \oplus r^1, r^1) & \text{if } random_{bit} \text{ is 0,} \\ (r^1, r^2) & \text{if } random_{bit} \text{ is 1.} \end{cases} \qquad (6)$$

Where $random_{bit}$ is equivalent to coin flipping and r is the random value of the mask. To perform t-test, 1,800 traces are collected and analyzed using the ttest2 command in MATLAB and t values are calculated. In Figure 13, 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.

6 Conclusions and Further Research

In this research, for the time, we mounted 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 attack to the cipher directly. Therefore, a three-step 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 illustrate 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 1.86 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 are reconfirmed.

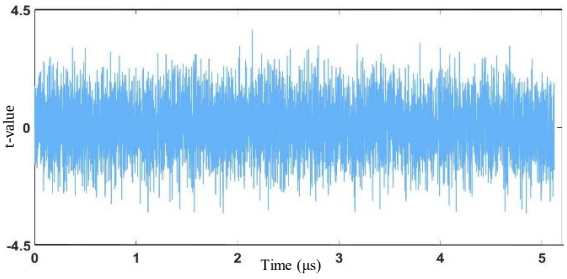


Figure 13. Attack results on protected COLM by t-test with 1,080 traces

Some security analyses for the final winners of CAESAR competition are presented; however, the security of winners against the side-channel attacks less studied yet. In addition, proposing an optimized protection scheme against CPA attack and comparing the cost of protection with final winners could be good directions for future researches.

References

[1] Doug Whiting, Russ Housley, and Niels Ferguson. Counter with CBC-MAC (CCM). RFC3610, 2003.

[2] Ted Krovetz and Phillip Rogaway. The OCB authenticated-encryption algorithm. internet engineering task force (IETF) RFC 7253. https://tools.ietf.org/html/rfc7253, 2014.

[3] David McGrew and John Viega. The galois/counter mode of operation (GCM). *submission to NIST Modes of Operation Process*, 20, 2004.

[4] Niels Ferguson. Authentication weaknesses in GCM. *Comments submitted to NIST Modes of Operation Process*, pages 1–19, 2005.

[5] Hanno BÃűck, Aaron Zauner, Sean Devlin, Ju-

raj Somorovsky, and Philipp Jovanovic. Nonce-disrespecting adversaries: Practical forgery attacks on gcm in tls. Cryptology ePrint Archive, Report 2016/475, 2016. https://eprint.iacr.org/2016/475.

[6] CAESAR: Competition for authenticated encryption: Security, applicability, and robustnes. http://competitions.cr.yp.to/caesar.html .

[7] Elena Andreeva, Andrey Bogdanov, Nilanjan Datta, Atul Luykx, Bart Mennink, Mridul Nandi, Elmar Tischhauser, and Kan Yasuda. COLM v1. CAESAR competition proposal, 2016. http://competitions.cr.yp.to/round3/colmv1.pdf .

[8] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Annual International Cryptology Conference*, pages 388–397. Springer, 1999.

[9] Alexandre Adomnicai, Jacques J.A. Fournier, and Laurent Masson. Masking the lightweight authenticated ciphers ACORN and Ascon in software. Cryptology ePrint Archive, Report 2018/708, 2018. https://eprint.iacr.org/2018/708.

[10] Niels Samwel and Joan Daemen. DPA on hardware implementations of Ascon and Keyak. In *Proceedings of the Computing Frontiers Conference*, pages 415–424. ACM, 2017.

[11] Hannes Gross, Erich Wenger, Christoph Dobraunig, and Christoph Ehrenhöfer. Ascon hardware implementations and side-channel evaluation. *Microprocessors and Microsystems*, 52:470–479, 2017.

[12] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of nonlinear functions in the presence of glitches. *Journal of Cryptology*, 24(2):292–321, 2011.

[13] William Diehl, Abubakr Abdulgadir, Farnoud Farahmand, Jens-Peter Kaps, and Kris Gaj. Comparison of cost of protection against differential power analysis of selected authenticated ciphers. *Cryptography*, 2(3):26, 2018.

[14] Mohsen Jahanbani, Zeinolabedin Norozi, and Nasour Bagheri. DPA protected implementation of OCB and COLM authenticated ciphers. *IEEE Access*, 7:139815–139826, 2019.

[15] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *International workshop on cryptographic hardware and embedded systems*, pages 16–29. Springer, 2004.

[16] Hannes Gross, Stefan Mangard, and Thomas Korak. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. Cryptology ePrint Archive, Report 2016/486, 2016. https://eprint.iacr.org/2016/486 .

[17] A Generic Side-Channel Distinguisher, Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In *Cryptographic Hardware and Embedded Systems–CHES 2008: 10th International Workshop, Washington, DC, USA, August 10-13, 2008, Proceedings*, page 426. Springer Science & Business Media, 2008.

[18] Dakshi Agrawal, Josyula R Rao, and Pankaj Rohatgi. Multi-channel attacks. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 2–16. Springer, 2003.

[19] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer Science & Business Media, 2008.

[20] David Canright and Lejla Batina. A very compact âĂIJperfectly maskedâĂİ S-box for AES. In *International Conference on Applied Cryptography and Network Security*, pages 446–459. Springer, 2008.

[21] Amir Moradi, Oliver Mischke, and Thomas Eisenbarth. Correlation-enhanced power analysis collision attack. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 125–139. Springer, 2010.

[22] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. A more efficient AES threshold implementation. In *International Conference on Cryptology in Africa*, pages 267–284. Springer, 2014.

[23] Hannes Groß, Stefan Mangard, and Thomas Korak. An efficient side-channel protected AES implementation with arbitrary protection order. In *CryptographersâĂŹ Track at the RSA Conference*, pages 95–112. Springer, 2017.

[24] Amir Moradi. *Advances in side-channel security*. PhD thesis, Habilitation thesis, Ruhr-Universität Bochum, 2016.

[25] Amir Moradi, Axel Poschmann, San Ling, Christof Paar, and Huaxiong 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*, pages 69–88. Springer, 2011.

[26] Side-channel attack user reference architecture. http://satoh.cs.uec.ac.jp/SAKURA/hardware.html .

[27] https://github.com/newaetech/chipwhisperer .

[28] Josh Jaffe. A first-order DPA attack against AES in counter mode with unknown initial counter. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 1–13. Springer, 2007.

[29] Thomas De Cnudde, Oscar Reparaz, Begül Bilgin, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Masking AES with $d + 1$ shares in hardware. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 194–212. Springer, 2016.

[30] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Trade-offs for threshold implementations illustrated on AES. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(7):1188–1200, 2015.

[31] Felix Wegener and Amir Moradi. A first-order SCA resistant AES without fresh randomness. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*, pages 245–262. Springer, 2018.

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

[33] Rei Ueno, Naofumi Homma, and Takafumi Aoki. Toward more efficient DPA-resistant AES hardware architecture based on threshold implementation. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*, pages 50–64. Springer, 2017.

[34] David Canright. A very compact S-box for AES. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 441–455. Springer, 2005.

[35] GMU implementations of authenticated ciphers. george mason university. https://cryptography.gmu.edu/athena/index.php?id=CAESAR. .

[36] George Becker, J Cooper, Elke DeMulder, Gilbert Goodwill, Joshua Jaffe, G Kenworthy, T Kouzminov, A Leiserson, M Marson, Pankaj Rohatgi, et al. Test vector leakage assessment (TVLA) methodology in practice. In *International Cryptographic Module Conference*, volume 1001, page 13, 2013.

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

Mohsen Jahanbani Ardakani received the BS and MS degree in electrical engineering in 2004, and 2009, respectively, and Ph.D. degree in mathematics, in 2020, all from Imam Hossein University, Tehran, Iran. His research interest includes hardware implementation, lightweight implementation, side-channel attacks and countermeasures. He has published a few technical papers and participated in journals of information security and cryptology.

Nasour Bagheri received the M.S. and Ph.D. degrees in Electrical Engineering from Iran University of Science and Technology (IUST), Tehran, Iran, in 2002 and 2010 respectively. He is currently an associate professor at the electrical engineering department, Shahid Rajaee Teacher Training University, Tehran, Iran. He is the author of over 100 articles in information security and cryptology. His research interests include cryptology, more precisely, designing and analysis of symmetric schemes such as lightweight ciphers, e.g., block ciphers, hash functions and authenticated encryption schemes, cryptographic protocols for constrained environment, such as RFID tags and IoT edge devices and hardware security, e.g., the security of symmetric schemes against side-channel attacks such as fault injection and power analysis. A record of his publication is available at google scholar: https://scholar.google.com/citations?user=32llx44AAAAJ&hl=en

Zeinolabedin Norozi received his BS and MS degrees in applied mathematics in 2004 and 2007 from University of Tehran, respectively. He has Ph.D. degree in applied mathematics - cryptography in 2012 from Kharazmi University. In 2007, he joined the Electrical Engineering Department as an assistant professor at Imam Hossein University, Tehran, Iran. His research interest includes symmetric cryptology, with an emphasis on block cipher and authenticated encryption, steganography algorithms and lightweight implementation and side-channel attacks. He has published several technical papers and participated in many scientific conferences in information security and cryptology.