Power-based Side Channel Analysis and Fault Injection: Hacking Techniques and Combined Countermeasure

Noura Benhadjyoussef[1], Mouna Karmani[2], Mohsen Machhout[3]
Faculty of Sciences of Monastir, Electronics and MicroElectronics Laboratory (LEME)
Monastir 5019, Tunisia

***Abstract*—Over the last years, physical attacks have been massively researched due to their capability to extract secret information from cryptographic engine. These hacking techniques are based on exploiting information from physical implementations instead of cryptographic algorithm flaws. Fault-injection attacks (FA) and Side-channel analysis (SCA) are the most popular techniques of implementation attacks. Aiming to secure cryptographic devices against such attacks, many studies have proposed a variety of developed and sophisticated countermeasures. Hence, the majority of these secured approaches are used for precise and single attack and it is difficult to thwart hybrid attack, such as combined power and fault attacks. In this work, the Advanced Encryption Standard is used as a case study in order to analyse the most well-known physical-based Hacking techniques: Differential Fault Analysis (DFA) and Correlation Power Analysis (CPA). Consequently, with the knowledge of such contemporary hacking technique, we proposed a low overhead countermeasure for the AES implementation that combines the concept of correlated power noise generating with a combined-approach based fault detection scheme.**

Keywords—Advanced encryption standard; fault attack; power attacks; combined countermeasure; hardware implementation

I. Introduction

From a data security viewpoint, securing secret information requires using algorithms that resist theoretical hacking techniques. Though, treating an algorithm in a purely mathematical way or, in other words, shying away from its physical implementation opens the door to numerous threats in the real-world security. In the modern age of electronics, cryptanalysis attempts to reveal sensitive data based on physical property of a cryptographic device rather than making use of the theoretical flaws in the implemented cryptographic algorithm. Indeed, as the cryptographic algorithms are implemented on a physical platform, they are susceptible to well-known physical attacks, namely Fault Attacks (FA) and Side-Channel Analysis (SCA). These two classes of attacks exploit the physical interactions with cryptographic systems to break their security and extract secret information. These attacks are practical due to their methods to reveal the secret information from most cryptosystems which supposed to be cryptanalytically secure.

In SCA hacking technique, a passive adversary observes platform side-channel information to reveal the sensitive information. Indeed, these devices leak sensitive correlated information in the form of electromagnetic emissions (EM), time execution, power consumption, allowing a hacker to reveal the secret key from the cryptographic device [1]–[4].

On the other hand, FA hacking technique is an active cryptanalysis based on perturbing the cryptographic device processing in order to obtain an abnormal behaviour. The hacker then exploits the erroneous of the cryptographic device result to retrieve the secret key [5], [6]. Many studies combine SCA and FA in order to form even more sophisticated attacks [7], [8].

Aiming to secure cryptographic devices against such real-world attacks, countermeasures must be thus designed to harden cryptographic implementations before they are used in the wild. Many studies have proposed a variety of developed and sophisticated countermeasures. In particular, countermeasure for the AES crypto-core has been massively researched for many years against both fault injection Attack and power analysis attacks.

For SCA, most implemented countermeasures aim to decreasing the signal-to-noise ratio (SNR) by using two key approaches; the noise insertion or the leaked sensitive correlated information destruction. These countermeasures are categorized as logical [9], architectural [10], [11], and circuit-level countermeasures [12]–[14]. For the logical and architectural countermeasures, the used approach is specific to the crypto-core and design. On the other hand, physical countermeasures are nonspecific and can be used to protect any crypto-core by providing cover around it [2].

Error detection schemes against FAs are, generally, based on some redundancy approaches. Either using the temporal redundancy, where a given operation is computed twice, or using hardware redundancy by executing the same transformation at the same time to compare the obtained results and check whether a fault was induced. Adding correction and error detection codes to intermediate values is another option to protect the considered cryptographic system named information redundancy [15], [24].

Although the secured approaches have been widely studied for each kind of physical attack, the study of combined countermeasures has not been well explored in the existing literature. In this paper, we perform an in-depth study of Differential fault attack (DFA) and Correlation power analysis

(CPA) and we propose a dual complementary AES cryptographic circuit to defend against both fault and power SCA attacks.

The main contributions of this paper are as follows:

1) We firstly present a fault based Hacking technique to indicate how a fault injection can be useful to extract the sensitive data of the AES crypt core. In the suggested case study, we clarify the main procedures that can threat the security of the considered AES design basing on DFA attack.

2) We study the power based Hacking technique and we perform a successful CPA on the FPGA based AES implementation using the Side-channel Attack Standard Evaluation Board (SASEBO).

3) We develop a combined fault detection scheme to secure the AES cryptocore based on an error detection code for linear AES transformation and temporal redundancy for the nonlinear SubByte transformation. This proposed scheme can be applied for both LUT and GF SubByte transformation implementations.

4) To avoid information leakage, the proposed fault detection scheme is enhanced using a correlated power noise generator. This enhanced countermeasure eliminates the AES cryptocore power correlation with the secret key by adding an interfering power signal which depends on the manipulated data and a nosey key.

5) Finally, we implement our AES cryptocore with the proposed countermeasure on a Virtex V Xilinx field-programmable gate array (FPGA) device. Moreover, to investigate the practicality and effectiveness of the proposed architecture, we compare our implementation results to similar secured AES implementations presented in literature.

The paper is organized as follows. Section 2 introduces the proposed AES Faults-based Hacking technique using the DFA attack. Section 3 studies the AES power Side-Channel analysis using the CPA attack. Section 4 presents the proposed fault-resilient AES implementation. The proposed power based SCA-Countermeasure for AES implementation is presented in Section 5. The experimental results and comparison with previous works reported in the literature is given in Section 6. Section 7 concludes the paper.

II. The Faults-based Hacking Technique using the DFA Analysis against AES Implementation

A. *Related Works*

Fault attacks exploit the possibility to inject a fault into cryptographic devices in order to reveal the secret key. The fault injection is done by means of Electromagnetic field, supply voltage variation, laser beam, or temperature control. In particular, Differential Fault Attacks on AES has become a widely research topic using different fault-models; single-bit, single-byte fault and multibyte fault.

In [16], two DFA attacks on AES are proposed, the first attack inject a theoretical single-bit fault into an intermediate result allowing hacker to extract the AES-128 secret key with 50 faulty ciphertexts. While the second attack inject a byte-fault and reveal the key with less than 250 faulty ciphertexts.

Reference [17] presents an improved DFA attack approach on AES using unknown and random multi-byte faults. The authors focused on combined fault model that inject single-byte and multi-byte faults. Their attack showed that about 97.3% of the attacks can be completed within 3 pairs of correct and faulty ciphertexts.

In [18], authors presented a DFA combined fault model combining a single-byte random faults model in encryption process with a single-byte faults model in the key schedule process. Their experimental results showed that 6 pairs of correct and faulty ciphertexts could reveal the AES-128 secret key. Reference [19] shows that a FA can break the advanced encryption standard (AES) by exploiting the existing target devices Input/Output signals instead of the artificial triggering implementation. Indeed, authors identify fault injection time by employing target devices electromagnetic emission. Using one-byte fault model, the attack was successfully executed 55 times out of the 1000 conducted fault injection attacks overall.

In this paper, we aim to propose a low-cost fault-resilient AES architecture that resists side-channel attack. So, as a first step, we must study how a fault injection can be used to extract the cryptographic key. In this step, the Giraud's Single-Bit Differential Fault Analysis [16] is adopted as a case study to give details of the main procedure that can menace the security of the considered AES implementation. Consequently, with the knowledge of the considered Fault-hacking technique, we will propose the adequate DFA countermeasure for AES cryptocore.

B. *The Fault Injection Step*

In this step, we have adapted the single-bit fault model to simulate the physical real defects. This model assumed that only one bit in the considered circuit is faulty and supposed that one line in the circuit behaved as if it is at logic 0 or logic 1. For the multiple-bit fault model it is assumed that the same basic assumptions as the single-bit fault models, except allowing two or more lines in the circuit to be faulty at the same time.

For our considered DFA attack, the Single-Bit model is adopted, where only one bit was injected at the beginning of the final round (see Fig. 1).

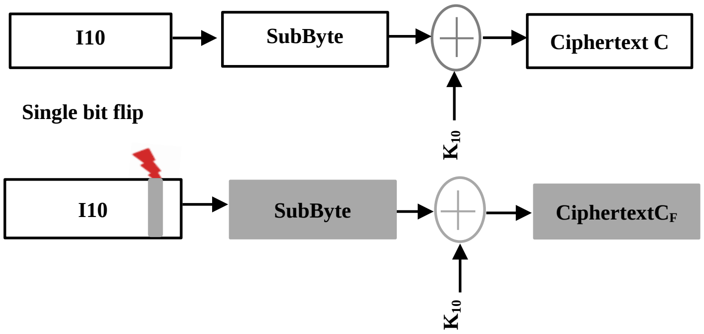

Fig. 1. The Single-Bit Fault Injection into the Input of the 10th Round.

C. The Fault Propagation Step

Using the DFA Single-Bit model, only a faulty one-bit *'e'* was injected in the output of the 9th AES round (I_{10}). When flipping a single bit between the MixColumns of the ninth round and the SubBytes of the tenth round, the changed bit spreads in the last round and generates a faulty Ciphertext (CF) with single faulty byte.

As presented in Fig. 2, the injected fault into the AES-128 bloc modifies 8-bit through the SubBytes operation. Indeed, this non-linear operation is a byte substitution and executes each 8-bit input separately. The affected byte is Xor-ed with another round key byte of the tenth round and produces one differential fault.

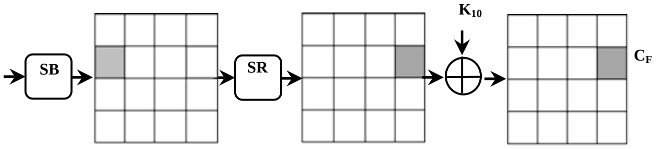

Fig. 2. Propagation of Injected Fault in $I_{10.}$

D. The Fault Exploitation Step

In order to reveal the AES secret key, the hacker exploits the observable fault by exploiting some relation between the two obtained ciphertexts. Indeed the hacker must repeat the experiment with the same plaintext and same key but without inducing fault. As a result, two ciphertexts derived from the same plaintext and key are obtained, where one of the ciphertext is fault-free (C) and the other is faulty (CF).

As a first step, the hacker tries to solve these equations.

$$C = SB\,(I_{10}\,) \oplus k_{10} \tag{1}$$

$$C_F = SB\,(I_{10} \oplus e) \oplus k_{10} \tag{2}$$

$$\Delta = C \oplus C_F \tag{3}$$

where $SB(I_{10})$ is the result of the SubByte transformation applied on one byte of the 9th round inputs (I_{10}) and k_{10} is the 10th round key corresponding byte (k_{10}). While Δ is the injected fault differential. As the single bit flip on I_{10} is the adapted fault model, the Δ Hamming weight (HW) must equal 1. In order to reveal the *k10* value, the hacker must ensure a full exploration of all possible key values. Therefore, the hacker first computes for each key-assumption ($\tilde{k}$) of the real-key byte (k), the corresponding hypothesized fault differential $\tilde{\Delta}$ as follows:

$$\widetilde{I_{10}} = SB^{-1}(C \oplus k\widetilde{10}) \tag{4}$$

$$I\widetilde{_{10F}} = SB^{-1}(C_F \oplus k\widetilde{10}) \tag{5}$$

$$\tilde{\Delta} = \widetilde{I_{10}} \oplus I\widetilde{_{10F}} \tag{6}$$

where $\widetilde{k_{10}}$ denotes the hypothetical key and $\widetilde{\mathrm{I}_{10}}$ is the input of the corresponding tenth round. Finally, the hacker must verify if the calculated $\tilde{\Delta}$ is identical to Δ. Indeed, the hypothetical key $\widetilde{k_{10}}$ may be a correct assumption for the real key k_{10} if the $\tilde{\Delta}$ Hamming weight is equal to 1. Otherwise, the hypothetical key $\tilde{k}$ is rejected. This process is recomputed for each k_{10} byte to reveal the overall round key k_{10}.

Fig. 3 present the number of injected fault needed to retrieve the whole 16-byte last-round key of AES-128. As shown in Fig. 2 the considered hacking technique needs only 32 fault injections to extract the whole 128-bit tenth round key. Finally, since the AES key expansion is invertible, the hacker can compute the original key (k_0) going backwards.

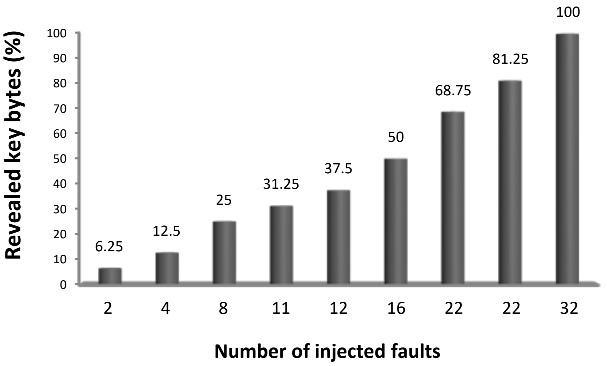

Fig. 3. The DFA Attack Results.

III. The Power Analysis-based Hacking Technique using the CPA Attack Against AES Implementation

Power-based side-channel attacks assume that there is a correlation between the level of power consumption and cryptographic operations manipulated by the cryptocore.

Simple power analysis (SPA) [20], differential power analysis (DPA) [21], and correlation power analysis (CPA) [22] are three fundamental techniques of power-based SCA attacks. The CPA hacking technique requires the least power traces to extract the secret-key and it has been considered as the most powerful power-based SCA. In this paper, the CPA-based Hacking technique was adopted as a case study in order to indicate the main procedure that can threat the AES cryptocore security.

A. CPA based Hacking Theory

The goal of CPA-based hacking technique is to accurately model the power consumption of the cryptographic circuit under attack in order to find correlation between characteristics of real power consumption traces and a predicted power trace. Therefore, choosing the accurately power model enable hackers to predict correctly the secret key by obtaining highest level of correlation.

1) The CMOS device power consumption model: For the cryptographic platforms based on the SOC design, the CMOS technology still the principal hardware solution due to its various advantages. The total power consumption of a CMOS device (P_{total}) is composed of two components: the static power (P_{static}) and the dynamic power ($P_{dynamic}$) [21]. P_{static} is the result of the transistors leakage current and depends on the circuit design. Hence, $P_{dynamic}$ is consumed when switching occurs. Indeed, if a CMOS cell changes from 0 to 1 or from 1 to 0, switching happens in transistors and Pdynamic is consumed. Therefore it depends on the manipulated data and the operation being done.

$$P_{dynamic} = P_{0\rightarrow 1}.C_{L.}\ f.\ V_{dd} \tag{7}$$

where C_L denotes the gate load capacitance, f denotes the clock frequency, VDD is the supply voltage and P0→1 the probability of a 0→1 output transition. As shown in (7), at a given time, the dynamic power dissipation depends upon the number of bit switching from one position to another [23][22]. Power consumption-based SCA uses a leakage model in order to define a relationship between the device power consumption and the secret information employed.

Various power models are proposed to estimate the power consumption of device under attack when processing the target data. The most well-known models are the Hamming distance and Hamming weight models.

a) The Hamming Weight Model

The Hamming weight model (HW) is the most basic power consumption model [21]. This model computes, in a data word, the number of bits set to 1. Considering multiple bits at a time, it is important to understand that the power consumption is, exclusively, based on the numbers of bits that are at logical 1 and not the number those bits are meant to represent. [25]. So, the predicted power consumption PW in an n-bit microprocessor is computed simply by:

$$PW= a*HW(D)+b= a * \sum_{j=0}^{n-1} d_j +b \tag{8}$$

where dj = 0 or 1 is the bit values of the binary data D (D=∑) handled by the cryptographic device under attack. is a scaling factor between the power consumed and the Hamming weight. and b is a term for everything like static power dissipation, the variation from one clock cycle to another, and time dependent components.

b) The Hamming Distance Model

The Hamming Distance model (HD) was proposed by Brier et al in [22], where the leakage is assumed to depend on switching activity in CMOS device. This model supposes that the power consumption in the target circuit correlates to the bits number changing from one state to another. Indeed, to estimate the device power consumption, the hacker uses the HD model and count number of 1→0 and 0→1 transitions that occur in a register or bus of a cryptographic device when it changes from one state to the next state. The consumptions for a bit switching from 0 to 1 or from 1 to 0 are further assumed to be same. Let R the reference state for a data word from which the bits are switched and D the current state manipulated by the target device. The power consumed PW is described by the mathematical equation for the hamming distance model as follows:

$$PW=a*HD(D)+b = a*HW(R \oplus D) + b \tag{9}$$

where $HW(R \oplus D)$ is the number of flipping bits from binary data R to D.

2) Pearson correlation coefficient: To evaluate the correlation between the estimated power consumption and the real power trace, the Pearson coefficient, $\rho_{W,PW}$ is considered as an efficient way. This correlation coefficient calculates the correlation between estimated power consumption PW of target data D and the equivalent real power traces measured W during processing the target cryptographic operation. $\rho_{W,PW}$ is described as follows:

$$\rho_{W,PW} = \frac{Cov(PW,W)}{\sigma_{PW}*\sigma_{W}} \tag{10}$$

where Cov denotes the covariance between PW and W, and and are standard deviation for PW and W respectively. When manipulated, the selected data D must depend on the desired secret key and the correlation coefficient is adopted as a distinguisher. Therefore the hacker predicts the unknown key and calculates the correlation coefficient $\rho_{W,HD}$ for every key candidate. The values will respect the inequality $0 \leq |\rho_{W,HD}| \leq 1$ and the right key assumption is supposed to indicate the biggest value.

B. CPA based Hacking Practice

In this section, we demonstrate the efficiency of the power analysis-based side-channel attacks on AES-128 engine implemented on FPGA (see Fig. 4). The steps involved in a successful CPA-hacking technique are:

1) Attack point selection step: In this step, a hacker chooses the attack point which can be a register or some function manipulating an intermediate result of the algorithm. This point must depend on both the known variable (e.g. the output of S-box) and secret keys K. In this case, the 10th round encryption is attack because the latter has been isolated from the other rounds and have relatively clear power signals [26]. Then, we calculate the original secret key, K0, going backwards since the AES Key Schedule is invertible. Fig. 5 shows the selected intermediate node D denoted as the output of Subbytes transformation and the reference point R defined as the corresponding Ciphertext. The AES-128 is used as a case study but this power-hacking technique can be applied to AES-192 and AES-256.

2) Power Assumption step: This step consists of predicting the target device power consumption with certain leakage model to estimate the dynamic power consumption reflecting secret-data moving and manipulated operations. As explained in previous section, power models present the correlation between the power consumed of the cryptographic CMOS device and data processed by this device at the same time. Indeed, bus value switching or registers value switching from 0 to1 or from 1 to 0 consumes some energy amount to achieve the transition. Hence, by counting the bits transition number at a given time, the hacker may predict the device under attack power consumption.

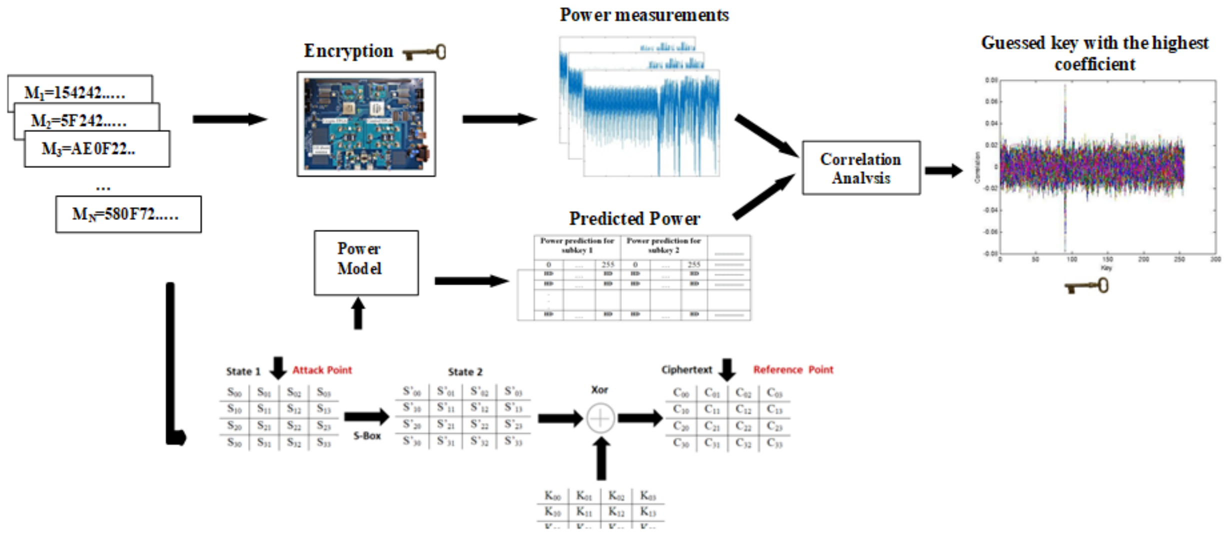
Power measurements
Encryption
Guessed key with the highest coefficient
M_1=154242.....
M_2=5F242.....
M_3=AE0F22..
...
M_N=580F72.....
Correlation Analysis
Predicted Power
Power prediction for subkey 1
Power prediction for subkey 2
Power Model
State 1
Attack Point
State 2
Ciphertext
Reference Point
S-Box
Xor
S_{00} S_{01} S_{02} S_{03}
S'_{00} S'_{01} S'_{02} S'_{03}
C_{00} C_{01} C_{02} C_{03}
K_{00} K_{01} K_{02} K_{03}

Fig. 4. Power Side-Channel Attack on 128-AES.

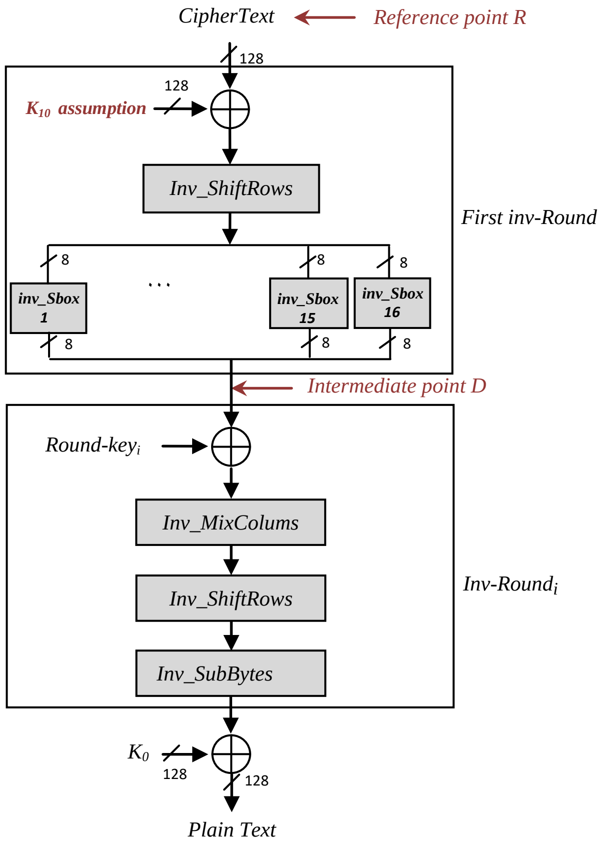

Fig. 5. The Selected Node for CPA Attack.

In this step, we have adapted the hamming distance model to predict the power consumption of the last AES-128 round encryption. The AES-128 decryption is an inv_round based encryption algorithm that process 128-bits data blocks as 16 bytes using 128-bits cipher keys. Each inv_rounds manipulates 128-bits round keys (K1 to K10) calculated from the original AES key, K_0 [27]. Indeed, this secret round-key is Xor-ed with the previous inv_round output, followed by an Inverse-ShiftRows transformation and Inverse-SubBytes transformation. The Inverse SubBytes operation divides the resulting 128-bits into 16 bytes and passing each through a Substitution S-Box. The S-box block takes 8-bit as input and produces 8 bite as output. Therefore, predicting one byte of the considered key is simple to calculate. For N Ciphertexts (N=20 000 in our case study) we predicts a subkey (The number of sub-key guess is limited to 28 assumptions: 256) and we calculate *HD(D)* predicted power consumption of the selected point *D* by the hamming distance model.

This step is repeated for 16 S-box outputs. So, we obtain a predicted power matrix P of size N x 256 x 16 as shown in Table I. *HD* value can be 1,2,3 or 8.

3) Measuring Power consumption step: The common setup for all Power-based side-channel attacks uses a PC in order to send known plaintexts to the target cryptographic circuit, trigger a device and save its power measurements traces. Therefore the hacker must obtain a matrix with a data pair of same Ciphertext used in the Power assumption step and their corresponding power measurements. The power measurements traces are normalized using pre-amplifier and gathered by oscilloscope during the AES encryption process. In this work, power measurements were performed by the "DPA contest v2" competition from the COMELEC Telecom department. The platform used to perform the power measurements acquisition is the Xilinx FPGA based Side-channel Attack Standard Evaluation Board (SASEBO)[28].

4) Correlation analysis step: This step evaluates the correlation between the predicted power and the power measurements using the Pearson coefficient. In this CPA statistical step, the measured power traces, denoted W, are compared to the predicted power consumption, denoted PW, for each 256 sub-key guesses. The correlation coefficient ρ_{WPw} is applied as follows:

TABLE I. PREDICTED POWER MATRIX P

Power prediction for subkey 1			**Power prediction for subkey 2**			**……………**	**Power prediction for subkey 16**		
0	…	255	0	…	255	……………	0	…	255
HD	…	**HD**	**HD**	…	**HD**	……………	**HD**	…	**HD**
HD	…	**HD**	**HD**	…	**HD**	……………	**HD**	…	**HD**
. . .									
HD	…	**HD**	**HD**	…	**HD**	……………	**HD**	…	**HD**

$$\rho_{W,Pw} = \frac{E(W,Pw)-E(W)*E(Pw)}{\sqrt{V(W)V(Pw)}}, \qquad (11)$$

If the sub-key assumption is correct, we expect that only one value, corresponding to the correct sub-key prediction, leads to a high correlation coefficient. The experimental result with only 3000 power measurements is shown in Fig. 6. As illustrated, the correlation power traces do not reveal the correct secret-key. Indeed there is no high correlation value in the obtained trace. The same correlation coefficient is calculated for the 256 sub-key assumption using 20000 power measurements. As indicated in Fig. 7, a unique correlation value, corresponding to the correct sub-key assumption, have high correlation value. Besides, the correct key assumption regularly stands out with a notable difference leading to a sure verdict of a successful attack. Fig. 9 corresponds to the correlation coefficient of correct sub-key assumption of. Hence, Fig. 8 presents the correlation value when the key guess is incorrect. This trace proves that there is no correlation between the predicted trace P and the corresponding measured trace W.

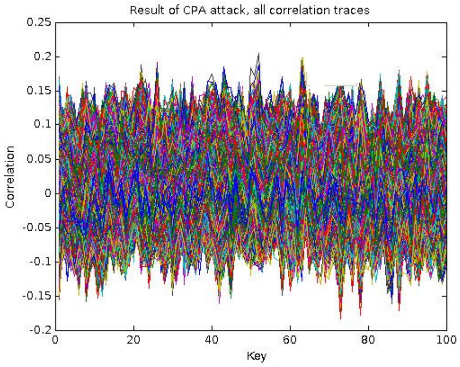

Fig. 6. Failed CPA using 3000 Inputs.

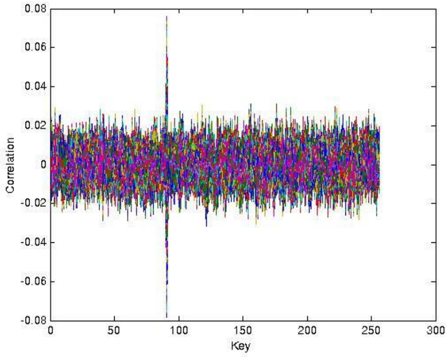

Fig. 7. Successful CPA using 20000 Inputs.

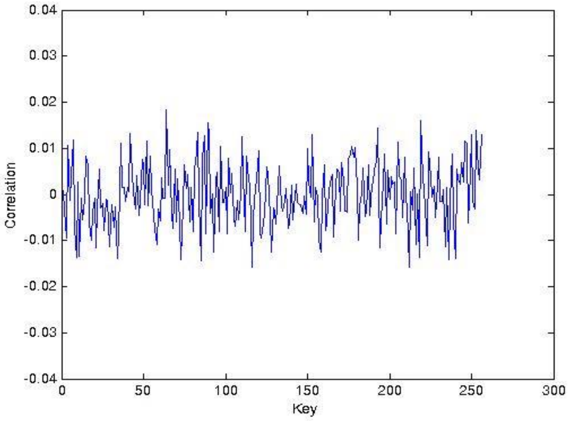

Fig. 8. Correlation Coefficient of an in Correct Sub-Key Assumption.

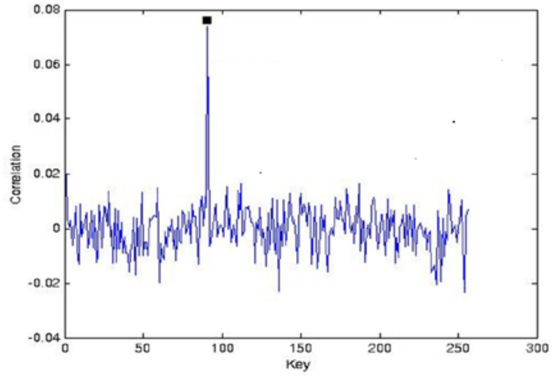

Fig. 9. Correlation Coefficient of a Correct Sub-Key Assumption.

In Fig. 10 the correlation coefficient for all the sub-key assumption, in terms of the number of power measurement was presented. This correlation trace presents the correlation coefficient between the measured power consumption and the predicted power consumption for various numbers of traces.

In fact, we remark that the correct sub-key assumption (plotted in black) can be distinguished by approximately 6000 power traces. This obtained result proves that CPA attack is an efficient power side-channel attack technique to extract the secret key.

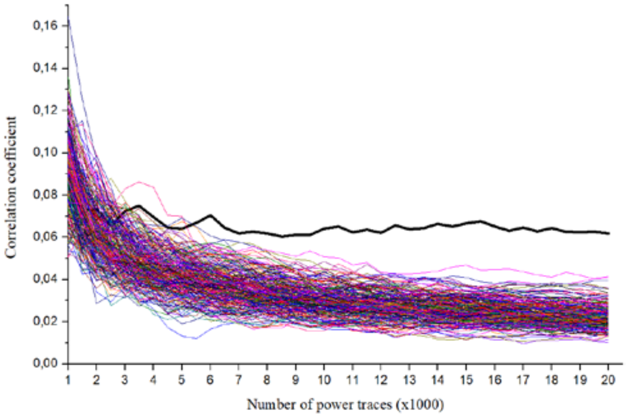

Fig. 10. Correlation Coefficient for all the Sub-Key Assumption.

IV. The Proposed Fault and Side-Channel Resistant 32-Bit AES

Power-Channel Analysis and Fault Attacks have seen a rise in popularity these last years, due to their practical methods to reveal the sensitive data from most cryptosystems which supposed to be cryptanalytically secure.

Our goal is to protect the cryptographic application against such attacks by detecting injected faults in the cryptocore and artificially introducing a noise in order to enhance the attack difficulty and reduce the probability of successful attacks. This section presents the proposed AES hardwarebased countermeasure to resist both Fault injection and Power sidechannel attacks. Fig. 11 shows the 32-bits AES block used as a case study.

The proposed design takes four 32-bit columns for the input data, one by one, executes them independently, and at the end, produces four 32-bit output columns.

A. Proposed Fault Detection Scheme for the AES

To secure the AES implementation against fault injection attacks, we incorporate fault-resilient techniques into the considered cryptographic hardware. Various DFA countermeasure schemes have been proposed to secure AES implementations, which are based on some sort of redundancy: information redundancy, hardware redundancy, or time redundancy in order to detect injected faults.

In this section we present an efficient combined fault detection based countermeasure that applies time and information redundancy to secure the considered AES cryptocore.

In this section we present an efficient combined fault detection based countermeasure that applies time and information redundancy to secure the considered AES cryptocore. The proposed Fault resilient AES implementation uses error detection code based on the (5, 4) CRC [29] to protect linear transformations and temporal redundancy approach for the Subbyte nonlinear transformation.

In fact, the information and hardware redundancy techniques induce more hardware overhead which can degrade cryptographic devices performances [30]. On the other hand The SubBytes operation is the most important non-linear operation in the AES, it occupies 70% of the AES Round area and 60% of the Key_Generator area.

Fig. 12 shows the temporal redundancy-based countermeasure to protect the Subbyte implementation.

As shown in Fig. 11, using same inputs, the Subbytes transformation is calculated twice with the same S-box hardware block and at the end, the result of the main calculation is compared with the result of the recalculation.

The fault detection scheme checks whether the indicator Sbox_FLAG equals zero or not. This Sbox_flag is calculated using a parity tree of exclusive-or gates.

Fig. 13 shows the information redundancy based countermeasure for linear transformations (ShiftRow, Mixcolum and AddRoundKey) with concurrent error detection. The ShiftRow shifts the bytes in each state row by a certain position without changing the parity from its input to its output. Correspondingly, there is no parity modification from the inputs of MixColumn to its outputs.

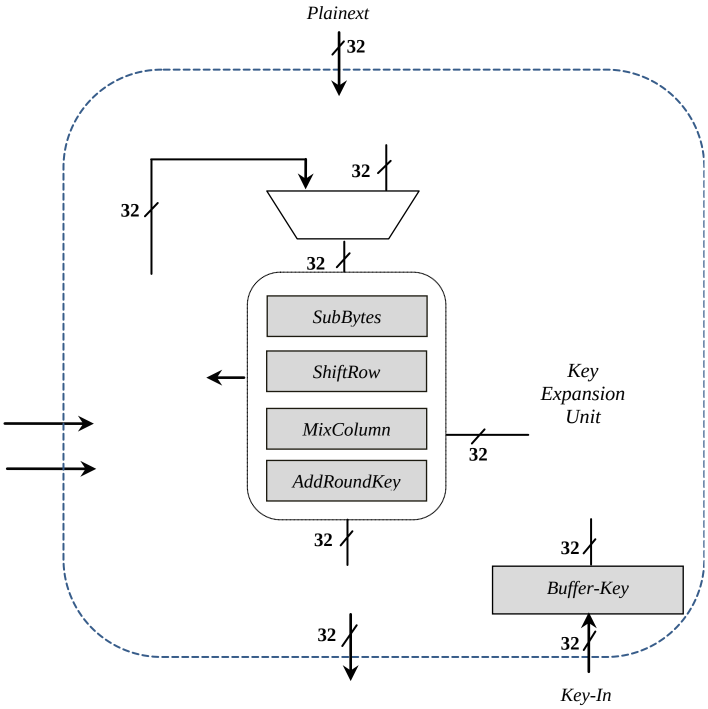

Fig. 11. Proposed 32-Bit Data-Path.

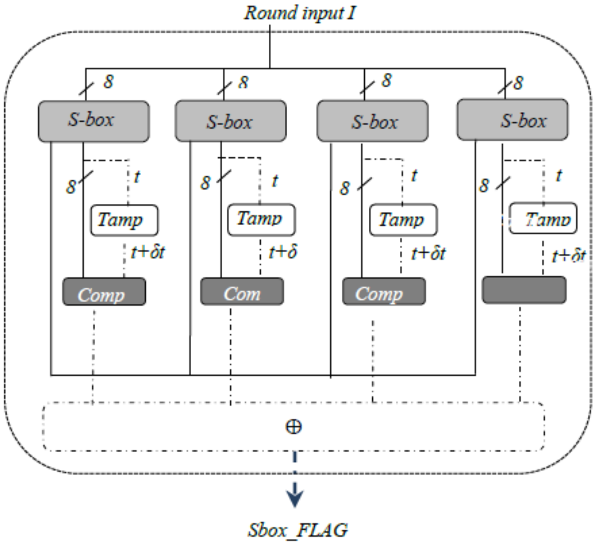

Fig. 12. SubByte Block with Error Check.

To detect injected fault in the Shiftrow and the Mixcolumn operations, we apply the information redundancy technique by using redundant information to protect these transformation.

Let SB(x) the Subbytes output and SR(x) the Shiftrow output as shown in Fig.13, where SB(x) = $sb_0+sb_1x + sb_2 x^2 + sb_3 x^3$ and SR(x) = $sr_0 + sr_1 x + sr_2 x^2 + sr_3 x^3$, $\{sb\mathit{i}, sr_i\} \in GF(2^8)$. P_{SB} is the Shiftrow's input parity obtained by (12).

$$P_{SB}=P(SB)= \sum_{i=0}^{3} sb_i \qquad (12)$$

where $sb_i \in GF(2^8)$ is fault detection approach checks whether the flag Shiftrow_fg , obtained by (13), equals zero or not.

$$Shiftrow\_fg = P_{SB} \oplus \sum_{i=0}^{3} sr_i \qquad (13)$$

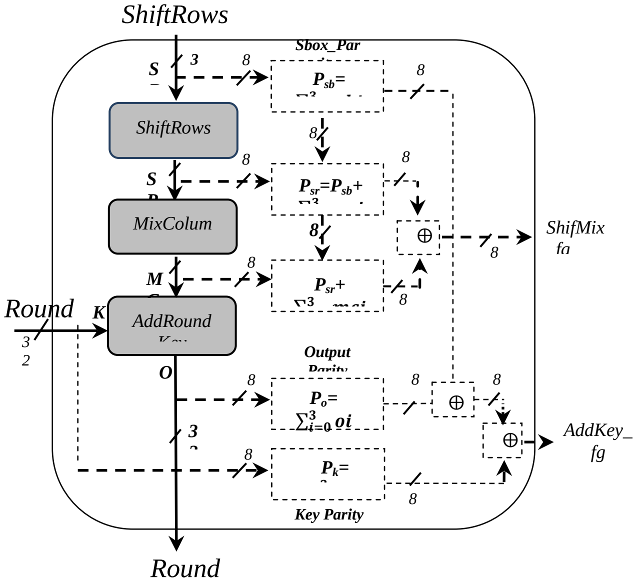

Fig. 13. Concurrent Error Detection Bloc for Linear Operations.

To protect Mixcolum operation, we use the same information redundancy based technique and we compute the flag mixcolumn_fg. The produced Flag will be XOR-red with the ShiftRow_fg in order to produce (ShifMix_fg).

At the Round output, the Shiftrow's input parity Psb will not be modified by the Mixcolum and the ShiftRow operations. But it will be changed by the AddRoundkey operation. So, to secure the AddRoundKey operation against fault injection, the key's parity PK must be calculated and xor-ed with the parity Psb. The obtained result will be XOR-ed with PO, the output's parity round, to produce the *AddKey_fg* flag. This DFA-countermeasure can be used to detect injected faults during the encryption process and produces Flags in order to interrupt the AES process. (see Fig.13).

B. Proposed SCA-Countermeasure for AES Cryptocore

The threats from Power attacks and Fault attacks challenge the integrity and security of cryptosystems. Various countermeasures for these attacks have been extensively studied in the existing literatures.

The author in [31] shows the impact of the countermeasure for one type of attack on the efficiency of another type of attack has been well explored.

Their experimental results show that the parity check code based fault detection technique makes CPA attack more difficult to retrieve the key than the original AES implementation. Based on this study, our Fault attack-resistant AES will affect the key retrieve speed of the CPA attack.

To more improve our fault resilient AES implementation, the hiding technique aims at lowering the Signal Noise Ratio (SNR) during the cryptographic operation by either adding more sources of noise or lessening the strength of the signal power trace that relates to the cryptocore operations. This makes CPA attack much harder as the data leaked has also to be correlated with external interfered key used inside the secured core.

Fig. 14 shows the Power analysis countermeasure for the AES cryptocore. This technique will be applied to the combined Fault resistant architecture presented in previous section. As shown in this figure, a parallel noise was incorporated into the AES cryptocore. The noise generating circuit obfuscates the AES cryptocore power traces by a power trace signal correlated with the plaintext and an interfering random key K_{interf}.

The AES cryptocore execute the AddRoundKey operation using the plaintext and the secret key K. Simultaneously, the noise generating circuit performs the same operation with the same plaintext but with the interfering keyK_{interf}. Two similar Subbytes transformation will takes the two AddRoundKey outputs as input and produces two signals *S* and S_{interf}.

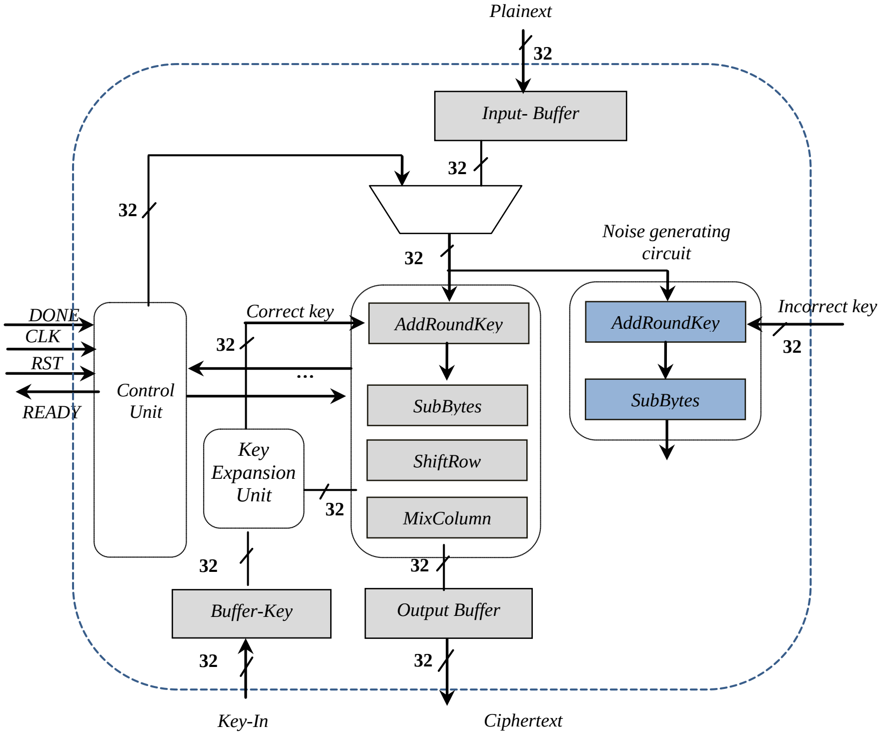

Fig. 14. Power Countermeasure for 32-Bits AES.

This proposed noise injection technique obfuscates the global AES cryptocore power trace by decreasing the correlation and bond between the secret values and the leaked information. In fact, the AES cryptocore power consumption correlates to plaintext and the secret keys couple (K, Kinterf). Furthermore, this added noise cannot be removed by statistical differential method; therefore, even if the power consumption curves was moved precisely and the Sboxes corresponding key successfully recovered, the hacker will still end up in failure because of the interference of the generated noise. This noise injection based countermeasure technique was experimentally proved in [32].

V. Implementation Results and Comparison

The proposed SCA/DFA Countermeasures for the AES design is practically examined using a Xilinx FPGA device, while the FI resistance is evaluated using the extensive fault simulations.We synthesized our implementation with the Xilinx ISE using the XC5VFX70t FPGA platform. The results and comparison with similar reported works are presented in Table II.

The AES-encryption implementation without the proposed countermeasure takes 445 slices for 296.43 MHz. The FPGA implementation result shows that our secured AES-encryption design occupies 14 % more area and 13% less throughputs compared to the original AES-cryptocore.

It can be seen from Table II, our proposed design has the minimum area overhead compared to [33], [34], [35] and [37].

TABLE II. FPGA Implementation of the Fault Resilient AES: Result and Comparison

	Area Overhead (%)	Time Overhead (%)	FC (%) Single-bit	FC (%) Random-bit
Our secured AES	**14**	**-13.5**	**100**	**100**
[33]	**43.33**	**-7.91**	**67.70**	**-**
[34]	**26.9**	**≈ 0**	**100**	**99.996**
[35]	**81**	**≈ - 0.17**	**100**	**93.75**
[36]	**2.3**	**-50**	**100**	**75**
[37]	**14.45**	**-18.71**	**85.958**	**98.54**

The temporal redundancy countermeasure presented in [36] add 2.3% overhead in terms of added hardware overhead and presents, approximately, four times degradation in terms of throughput overhead compared to our proposed countermeasures. Compared to [37] and [33], our secured AES design has the minimum area overhead and time overhead. These results prove that our proposed circuit is relevant to be arranged in many security domains such as embedded services routers, smartcards and emerging technologies using IoTs.

In order to evaluate the fault coverage of our protected AES cryptocore, fault-simulations are performed using the VHDL language. Two type of fault are used in the considered fault coverage simulation: Single-bit faults and Random-bit fault. For the single bit-fault type, we consider that a single-bit fault is inserted into 1-bit in random locations at random clock cycles of random rounds. On the other hand, random-bit fault type considers that faults are injected with random faulty bit number at random locations of random rounds. Fault simulations are performed over 1 000 000 times. As shown in the Table II, our fault simulations show that for single bit and random-bit faults, our protected AES-cryptocore have error coverage of 100 %.

Comparing our design to other similar countermeasures, our obtained fault coverage has the highest protection. Although our secured AES needs more resource overhead compared to some designs, it allows an excellent trade-off between fault-attack coverage, implementation area and throughput, which are relevant to secure embedded systems with resource constraint. In our future work, our power-baser cannel attack must be enhanced using new approaches based on deep-learning models.

VI. Conclusion

Recently, cryptographic embedded platforms used for trusted execution environment have been proven to be vulnerable to the power SCA and fault attacks. In this paper, we present a detailed fault and power based Hacking techniques to demonstrate how the fault injection or power analysis can be exploited to reveal the AES secret key. In the proposed case study, we explain the principal techniques that can threat the security of the AES design by using DFA and CPA attacks. This study was conducted in order to propose an adequate low-overhead hardware countermeasure that secures critical 32-bit AES cypto-core against both fault injection and power-based side-channel attacks. The proposed countermeasure gathers a combined fault resistance approach using parity testing for linear operations and time redundancy for the non-linear SubBytes operation with an artificially introduced noise provided by a correlated power noise block. The proposed countermeasure can be used for the encryption and decryption designs in order to enhance attack difficulty and reduce the probability of successful attacks. The proposed combined countermeasure has low overhead and achieves a 100% fault coverage during the considered AES process.

References

[1] A. Singh, M. Kar, S. K. Mathew, A. Rajan, V. De, and S. Mukhopadhyay, “Improved Power/EM Side-Channel Attack Resistance of 128-Bit AES Engines With Random Fast Voltage Dithering,” IEEE J. Solid-State Circuits, vol. 54, no. 2, pp. 569–583, Feb. 2019, doi: 10.1109/JSSC.2018.2875112.

[2] D. Das and S. Sen, “Electromagnetic and Power Side-Channel Analysis: Advanced Attacks and Low-Overhead Generic Countermeasures through White-Box Approach,” Cryptography, vol. 4, no. 4, p. 30, Oct. 2020, doi: 10.3390/cryptography4040030.

[3] C. Reinbrecht, A. Susin, L. Bossuet, G. Sigl, and J. Sepúlveda, “Timing attack on NoC-based systems: Prime+Probe attack and NoC-based protection,” Microprocess. Microsyst., vol. 52, pp. 556–565, 2017, doi: https://doi.org/10.1016/j.micpro.2016.12.010.

[4] Eng. Mustafa M. Shiple, Prof. Dr. Iman S. Ashour and Prof. Dr. Abdelhady A. Ammar, “Attacking Misaligned Power Tracks Using Fourth-Order Cumulant” International Journal of Advanced Computer Science and Applications (IJACSA), 4(12), 2013. http://dx.doi.org/10.14569/IJACSA.2013.041202

[5] H. S. Lim, J. H. Lee, and D. G. Han, “Novel fault injection attack without artificial trigger,” Appl. Sci., vol. 10, no. 11, 2020, doi: 10.3390/app10113849.

[6] R. Wang, X. Meng, Y. Li, and J. Wang, “Towards Optimized DFA Attacks on AES under Multibyte Random Fault Model,” Secur. Commun. Networks, vol. 2018, pp. 1–9, Aug. 2018, doi: 10.1155/2018/2870475.

[7] S. Patranabis, J. Breier, D. Mukhopadhyay, and S. Bhasin, “Side-Channel Assisted Fault Analysis,” 2018, pp. 59–77.

[8] S. Patranabis, J. Breier, D. Mukhopadhyay, and S. Bhasin, “One Plus One is More than Two: A Practical Combination of Power and Fault Analysis Attacks on PRESENT and PRESENT-Like Block Ciphers,” in 2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), Sep. 2017, pp. 25–32, doi: 10.1109/FDTC.2017.11.

[9] D. D. Hwang et al., “AES-Based Security Coprocessor IC in 0.18um CMOS With Resistance to Differential Power Analysis Side-Channel Attacks,” IEEE J. Solid-State Circuits, vol. 41, no. 4, pp. 781–792, Apr. 2006, doi: 10.1109/JSSC.2006.870913.

[10] M.-L. Akkar, R. Bevan, P. Dischamp, and D. Moyart, “Power Analysis, What Is Now Possible...,” 2000, pp. 489–502.

[11] A. Poschmann, A. Moradi, K. Khoo, C.-W. Lim, H. Wang, and S. Ling, “Side-Channel Resistant Crypto for Less than 2,300 GE,” J. Cryptol., vol. 24, no. 2, pp. 322–345, Apr. 2011, doi: 10.1007/s00145-010-9086-6.

[12] C. Tokunaga and D. Blaauw, “Securing Encryption Systems With a Switched Capacitor Current Equalizer,” IEEE J. Solid-State Circuits, vol. 45, no. 1, pp. 23–31, 2010, doi: 10.1109/JSSC.2009.2034081.

[13] M. Kar, A. Singh, S. K. Mathew, A. Rajan, V. De, and S. Mukhopadhyay, “Reducing Power Side-Channel Information Leakage of AES Engines Using Fully Integrated Inductive Voltage Regulator,” IEEE J. Solid-State Circuits, vol. 53, no. 8, pp. 2399–2414, Aug. 2018, doi: 10.1109/JSSC.2018.2822691.

[14] A. Singh et al., “Enhanced Power and Electromagnetic SCA Resistance of Encryption Engines via a Security-Aware Integrated All-Digital LDO,” IEEE J. Solid-State Circuits, vol. 55, no. 2, pp. 478–493, Feb. 2020, doi: 10.1109/JSSC.2019.2945944.

[15] N. Benhadjyoussef, M. Karmani, and M. MacHhout, “The Secured AES designs against Fault Injection Attacks: A comparative Study,” 2020, doi: 10.1109/ATSIP49331.2020.9231942.

[16] C. Giraud, “DFA on AES BT - Advanced Encryption Standard – AES,” 2005, pp. 27–41.

[17] N. Liao, X. Cui, K. Liao, T. Wang, D. Yu, and X. Cui, “Improving DFA attacks on AES with unknown and random faults,” Sci. China Inf. Sci., vol. 60, no. 4, p. 42401, 2016, doi: 10.1007/s11432-016-0071-7.

[18] Y. Liu, X. Cui, J. Cao, and X. Zhang, “A hybrid fault model for differential fault attack on AES,” in 2017 IEEE 12th International Conference on ASIC (ASICON), 2017, pp. 784–787, doi: 10.1109/ASICON.2017.8252593.

[19] H. Lim, J. Lee, and D.-G. Han, “Novel Fault Injection Attack without Artificial Trigger,” Appl. Sci., vol. 10, no. 11, p. 3849, Jun. 2020, doi: 10.3390/app10113849.

[20] N. Benhadjyoussef, M. Karmani, and H. Mestiri, “Power Analysis for Smartcard’s Authentication-Protocol,” 2019, doi: 10.1109/ASET.2019.8870994.

[21] P. Kocher, J. Jaffe, and B. Jun, “Differential Power Analysis BT - Advances in Cryptology — CRYPTO’ 99,” 1999, pp. 388–397.

[22] E. Brier, C. Clavier, and F. Olivier, “Correlation Power Analysis with a Leakage Model BT - Cryptographic Hardware and Embedded Systems - CHES 2004,” 2004, pp. 16–29.

[23] O.-X. Standaert, E. Peeters, G. Rouvroy, and J.-J. Quisquater, “An Overview of Power Analysis Attacks Against Field Programmable Gate Arrays,” Proc. IEEE, vol. 94, no. 2, pp. 383–394, Feb. 2006, doi: 10.1109/JPROC.2005.862437.

[24] H. Mestiri, N. Benhadjyoussef, M. MacHhout, and R. Tourki, “An FPGA implementation of the AES with fault detection countermeasure,” 2013, doi: 10.1109/CoDIT.2013.6689555.

[25] M. Randolph and W. Diehl, “Power Side-Channel Attack Analysis: A Review of 20 Years of Study for the Layman,” Cryptography, vol. 4, no. 2, p. 15, May 2020, doi: 10.3390/cryptography4020015.

[26] H. Liu, G. Qian, S. Goto, and Y. Tsunoo, “Correlation Power Analysis Based on Switching Glitch Model BT - Information Security Applications,” 2011, pp. 191–205.

[27] 2001. Advanced encryption standard (AES). Natl. Inst. Stand. Technol. 8– Fips-197, “Fips-197, 2001. Advanced encryption standard (AES). Natl. Inst. Stand. Technol. 8– 12,” 2011, [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.

[28] “DPA Contest v2,” 2009/2010. dpacontest.org /v2.

[29] H. Yen and B.-F. Wu, “Simple error detection methods for hardware implementation of Advanced Encryption Standard,” IEEE Trans. Comput., vol. 55, no. 6, pp. 720–731, 2006, doi: 10.1109/TC.2006.90.

[30] N. Benhadjyoussef, M. Karmani, M. Machhout, and B. Hamdi, “A Hybrid-Countermeasure based Fault-Resistant AES Implementation,” J. Circuits, Syst. Comput., 2019, doi: 10.1142/S0218126620500449.

[31] H. Pahlevanzadeh, J. Dofe, and Q. Yu, “Assessing CPA resistance of AES with different fault tolerance mechanisms,” in 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), 2016, pp. 661–666, doi: 10.1109/ASPDAC.2016.7428087.

[32] N. Kamoun, L. Bossuet, and A. Ghazel, “Correlated power noise generator as a low cost DPA countermeasures to secure hardware AES cipher,” in 2009 3rd International Conference on Signals, Circuits and Systems (SCS), 2009, pp. 1–6, doi: 10.1109/ICSCS.2009.5412604.

[33] H. Mestiri, N. Benhadjyoussef, and M. Machhout, “Fault attacks resistant AES hardware implementation,” 2019, doi: 10.1109/DTSS.2019.8914979.

[34] M. Mozaffari-Kermani and A. Reyhani-Masoleh, “Concurrent Structure-Independent Fault Detection Schemes for the Advanced Encryption Standard,” IEEE Trans. Comput., vol. 59, no. 5, pp. 608–622, 2010, doi: 10.1109/TC.2010.33.

[35] J. Chu and M. Benaissa, “Error detecting AES using polynomial residue number systems,” Microprocess. Microsyst., vol. 37, no. 2, pp. 228–234, 2013, doi: https://doi.org/10.1016/j.micpro.2012.05.010.

[36] J. Rajendran, H. Borad, S. Mantravadi, and R. Karri, “SLICED: Slide-based concurrent error detection technique for symmetric block ciphers,” in 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2010, pp. 70–75, doi: 10.1109/HST.2010.5513109.

[37] H. Mestiri, F. Kahri, B. Bouallegue, and M. Machhout, “A high-speed AES design resistant to fault injection attacks,” Microprocess. Microsyst., vol. 41, pp. 47–55, 2016, doi: https://doi.org/10.1016/j.micpro.2015.12.002.