7-1-2023

Systematic Characterization of Power Side Channel Attacks for Residual and Added Vulnerabilities

Aurelien Tchoupou Mozipo
Portland State University

Recommended Citation

Mozipo, Aurelien Tchoupou, "Systematic Characterization of Power Side Channel Attacks for Residual and Added Vulnerabilities" (2023). *Dissertations and Theses.* Paper 6515.
https://doi.org/10.15760/etd.3651

Systematic Characterization of Power Side Channel Attacks for Residual and Added Vulnerabilities

by

Aurelien Tchoupou Mozipo

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Dissertation Committee:
John M Acken, Chair
David Burnett
Dan Hammerstrom
James McNames

Portland State University
2023

Abstract

Power Side Channel Attacks have continued to be a major threat to cryptographic devices. Hence, it will be useful for designers of cryptographic systems to systematically identify which type of power Side Channel Attacks their designs remain vulnerable to after implementation. It's also useful to determine which additional vulnerabilities they have exposed their devices to, after the implementation of a countermeasure or a feature. The goal of this research is to develop a characterization of power side channel attacks on different encryption algorithms' implementations to create metrics and methods to evaluate their residual vulnerabilities and added vulnerabilities. This research studies the characteristics that influence the power side leakage, classifies them, and identifies both the residual vulnerabilities and the added vulnerabilities. Residual vulnerabilities are defined as the traits that leave the implementation of the algorithm still vulnerable to power Side Channel Attacks (SCA), sometimes despite the attempt at implementing countermeasures by the designers. Added vulnerabilities to power SCA are defined as vulnerabilities created or enhanced by the algorithm implementations and/or modifications.

The three buckets in which we categorize the encryption algorithm implementations are:

i. Countermeasures against power side channel attacks,
ii. IC power delivery network impact to power leakage (including voltage regulators),

iii. Lightweight ciphers and applications for the Internet of Things (IoT)

From the characterization of masking countermeasures, an example outcome developed is that masking schemes, when uniformly distributed random masks are used, are still vulnerable to collision power attacks. Another example outcome derived is that masked AES, when glitches occur, is still vulnerable to Differential Power Analysis (DPA).

We have developed a characterization of power side-channel attacks on the hardware implementations of different symmetric encryption algorithms to provide a detailed analysis of the effectiveness of state-of-the-art countermeasures against local and remote power side-channel attacks. The characterization is accomplished by studying the attributes that influence power side-channel leaks, classifying them, and identifying both residual vulnerabilities and added vulnerabilities. The evaluated countermeasures include masking, hiding, and power delivery network scrambling. But, vulnerability to DPA depends largely on the quality of the leaked power, which is impacted by the characteristics of the device power delivery network.

Countermeasures and deterrents to power side-channel attacks targeting the alteration or scrambling of the power delivery network have been shown to be effective against local attacks where the malicious agent has physical access to the target system. However, remote attacks that capture the leaked information from within the IC power grid are shown herein to be nonetheless effective at uncovering the secret key in the presence of these countermeasures/deterrents. Theoretical studies and experimental analysis are carried out to define and quantify the impact of integrated voltage regulators, voltage noise injection, and integration of on-package decoupling capacitors for both remote and

local attacks. An outcome yielded by the studies is that the use of an integrated voltage regulator as a countermeasure is effective for a local attack. However, remote attacks are still effective and hence break the integrated voltage regulator countermeasure. From experimental analysis, it is observed that within the range of designs' practical values, the adoption of on-package decoupling capacitors provides only a 1.3x increase in the minimum number of traces required to discover the secret key. However, the injection of noise in the IC power delivery network yields a 37x increase in the minimum number of traces to discover. Thus, increasing the number of on-package decoupling capacitors or the impedance between the local probing site and the IC power grid should not be relied on as countermeasures to power side-channel attacks, for remote attack schemes. Noise injection should be considered as it is more effective at scrambling the leaked signal to eliminate sensitive identifying information. However, the analysis and experiments carried out herein are applied to regular symmetric ciphers which are not suitable for protecting Internet of Things (IoT) devices.

The protection of communications between IoT devices is of great concern because the information exchanged contains vital sensitive data. Malicious agents seek to exploit those data to extract secret information about the owners or the system. Power side channel attacks are of great concern on these devices because their power consumption unintentionally leaks information correlatable to the device's secret data. Several studies have demonstrated the effectiveness of authenticated encryption with advanced data (AEAD), in protecting communications with these devices. In this research, we have proposed a comprehensive evaluation of the ten algorithm finalists of the National

Institute of Standards and Technology (NIST) IoT lightweight cipher competition. The study shows that, nonetheless, some still present some residual vulnerabilities to power side channel attacks (SCA). For five ciphers, we propose an attack methodology as well as the leakage function needed to perform correlation power analysis (CPA). We assert that Ascon, Sparkle, and PHOTON-Beetle security vulnerability can generally be assessed with the security assumptions “Chosen ciphertext attack and leakage in encryption only, with nonce-misuse resilience adversary (CCAmL1)” and “Chosen ciphertext attack and leakage in encryption only with nonce-respecting adversary (CCAL1)”, respectively. However, the security vulnerability of GIFT-COFB, Grain, Romulus, and TinyJambu can be evaluated more straightforwardly with publicly available leakage models and solvers. They can also be assessed simply by increasing the number of traces collected to launch the attack.

Dedication

I would like to dedicate this work to my family for supporting and encouraging me throughout this research:

To Horthense for your unconditional love, support, and encouragement;

To Gaby, for your constant words of encouragement, and for reminding me that I'm an old man;

To Lily for your constant push to tenacity.

Acknowledgments

I would like to start by thanking my advisor Dr. John M Acken for his technical advice, his words of wisdom, and the supervision of this research. I'm truly grateful for him continuously enlightening and coaching me.

I also extend my gratitude to my dissertation committee members: Dr. David Burnett, Dr. Dan Hammerstrom, and Dr. James McNames. Their oversight and guidance helped me immensely.

Finally, I want to thank my wife, Horthense, without whom this research wouldn't have been possible. Her everlasting support was the driving force behind my perseverance.

Table of Contents

List of Tables

List of Figures

1 INTRODUCTION

Information transmitted over a network or processed or stored within an integrated circuit is susceptible to theft or tampering, depending on the intentions of the malicious agent interfering with the information. Devices in cloud data centers that process sensitive information are vulnerable to attacks by agents with the intent to either reveal the secret information itself or prevent the device from performing its task. Devices in IoT are susceptible to attacks to steal the private information they process, store and transmit. They are also vulnerable to destructive attacks such as Denial of Service (DoS). In the era of the Internet of Things (IoT), lightweight ciphers will play a huge role in providing security for the connection of small devices with the internet or to one another. The security of the connection and communication of systems and devices to the network (public or private) cannot be complete if these three areas are not addressed: confidentiality, message integrity, and authentication. Various types of encryption algorithms provide means to protect data before transmission or storage. They are mainly classified as symmetric encryption algorithms such as Advanced Encryption Standards (AES), Data Encryption Standards (DES), all 10 finalists of the National Institute of Standards and Technology (NIST) Lightweight Cipher Competition, and asymmetric encryption algorithms such as Rivest-Shamir-Adleman (RSA), Diffie Hellman, Elliptic-Curve Cryptography. Malicious agents have attempted to uncover the secret hidden behind the above mentioned encryption algorithms with direct attacks (for example brute force attacks) or by exploiting side channel information.

The implementations of the encryption algorithms can unintentionally leak information about their operations which malicious agents exploit to uncover confidential information like the encryption key. Examples of leaks include the IC power consumption and the IC Electromagnetic radiations. Such information is called side channel information and the act of recovering secrets by exploiting side channel information is called a side channel attack (SCA). Once the malicious agent has measured the side channel information, they can compare it to an estimated quantity that is chosen based on the algorithm being attacked. The closer the match, the higher the probability that the estimated quantity is representative of the secret. Side channel attacks have become an important field of research for designers who want to decrease the vulnerability of their systems to malicious attacks. Our research will focus on analyzing power side channel leakage to evaluate the vulnerability to power side channel attacks of multiple encryption algorithms implementations.

1.1 Objective of the research

Power SCA has continued to be a major threat to cryptographic devices. It will be useful for designers of cryptographic systems to systematically identify which type of power SCA their designs remain vulnerable to after implementation. It's also useful to determine which additional vulnerabilities they have exposed their devices to, after the implementation of a countermeasure or a feature. The goal of this research is to develop a characterization of power side channel attacks on different symmetric encryption algorithms' implementations to create metrics and methods to evaluate their residual vulnerabilities and added vulnerabilities. This research studies the characteristics that

influence the power side leakage, classifies them, and identifies both the residual vulnerabilities and the added vulnerabilities. Residual vulnerabilities are defined as the traits that leave the implementation of the algorithm still vulnerable to power SCA, sometimes despite the attempt at implementing countermeasures by the authors. Added vulnerabilities to power SCA are defined as vulnerabilities created or enhanced by the algorithm implementations and/or modifications.

The three buckets in which we categorize the encryption algorithm implementations are:

i. Countermeasures against power side channel attacks,
ii. IC power delivery network impact to power leakage (including voltage regulators),
iii. Lightweight ciphers and applications for the Internet of Things (IoT)

Countermeasures are architectural changes to the algorithm implementation and hardware modifications to the IC environment intended to render power side attacks less successful or impossible. Countermeasures studied in this research are masking, hiding, shuffling, and key rotations. This research studies multiple implementations of these countermeasures and derives patterns that leave them vulnerable to specific kinds of power side channel analysis and patterns that show new vulnerabilities opened by the implementation of the countermeasure.

Power Delivery Network (PDN) plays a critical role in power side channel attacks, as the measured voltage representing the IC power consumption can be filtered or distorted

by the IC PDN. Theft of secret information such as the encryption key of a cryptographic algorithm may use a sophisticated analysis like differential power analysis (DPA), correlation power analysis (CPA), or differential fault analysis (DFA). DPA is a form of SCA that works on the premise of exploiting a piece of information that a system leaks unintentionally to the attacker while completing its task of interest [9]. In our case, the leaked information is the local voltage fluctuation that the attacker measures locally (with an oscilloscope) or remotely (with a trojan RTL implemented in the FPGA). The secret information of interest is the encryption key of the cryptographic algorithm running by an innocent victim on the same package or inside the same FPGA fabric.

The countermeasures used in the industry to scramble the power leaked and their signature are on-chip integrated voltage regulators, system noise generation, and clock noise generation. This research will evaluate the impact of these PDN countermeasures and develop a pattern that generalizes the implementation of residual vulnerabilities and potentially added vulnerabilities to power SCA.

With the emergence of lightweight, interconnected devices in the Internet of Things, sensor networks, healthcare, distributed control systems, and cyber-physical systems, there has been a growing need to develop encryption algorithms suitable to secure them and protect their data privacy. Typical encryption algorithms are geared toward computers, ASIC, and FPGA implementations that consume a power level higher than what a lightweight device can tolerate. For this reason, numerous lightweight ciphers have been developed to fill the gap. PRINCE, SIMON, and PRESENT are examples of lightweight ciphers described in the literature. Moreover, the National Institute of

Standards and Technology (NIST) has launched a competition to develop standardized lightweight ciphers based on Advanced Encryption with Associated Data (AEAD) with hashing. This research studies the 10 finalists to characterize their vulnerability to power SCA.

Most of our research practical experiments use the AES encryption algorithm because it's the most common symmetric algorithm used in the industry. Though AES attacks with byte-wise key enumeration are common in the literature [9][21], this research exposes the weakness consisting of attacking AES implementations with full key enumeration. Specifically, attackers who do not have access to the ciphertexts are forced to use the plaintexts as known variables. A drawback of full key enumeration is the very large size of the key search space. Hence, another goal of this research is to demonstrate a way of reducing the key search space via key ranking.

1.2 Thesis organization

The rest of this thesis is organized as follows. Chapter 2 is dedicated to the introduction of the basic concepts used through this research: encryption and hashing, side channel attacks principles, the applicability of power side channel attacks, the difference between remote and local side channel attacks, countermeasures, brute force vs key enumeration and concept of estimating the success rate of an attack. Chapter 3 introduces the background and prior art that constitute the basis for our research. It presents the concept of correlation power analysis, then delves into countermeasures and what they mean in power side channel attacks. It also shows the type of power side channel attacks depending on application sectors and shows the impact a power delivery network can

have on the measured power and hence on the success of side channel cryptanalysis. Chapter 4 marks the beginning of our contribution to the art. Here, we introduce two new cryptanalysis concepts: a reformulation of the correlation power analysis and the concept of Kullback-Leibler Rank (KLR). Chapter 5 presents and elaborates on the experimental results of attacking AES128, AES256, and RSA algorithms. In this chapter, the attack principle is first defined, then we define how this research estimates the device power consumption used in correlation coefficients computations and in the Kullback-Leibler estimation. Chapter 6 is dedicated to the characterization of power side channel attack countermeasures on different encryption algorithms implementation schemes, with goals to create metrics to evaluate their residual vulnerabilities. The main countermeasures studied are power delivery network scrambling, masking, and hiding. Chapter 7 studies power delivery network based countermeasures. It demonstrates through theoretical analysis, simulations, and lab experiments, the impact of these countermeasures on residual vulnerabilities to remote and local attacks. Chapter 8 evaluates the implementation of lightweight ciphers countermeasures in IoT devices, against residual and added vulnerabilities. It mostly focuses on identifying vulnerabilities to power SCA in seven (out of ten) LWC finalists and proposes methodologies for attacking five of them. It also proposes the leakage functions needed to perform CPA on those lightweight ciphers. Chapter 9 summarizes our contribution to the research and proposes future work to further this research.

2 BACKGROUND ON CRYPTOLOGY AND GENERAL SIDE CHANNEL ATTACKS

The broad concept of cryptology encompasses cryptography, the science of hiding the meaning of a message, and cryptanalysis, the science or art of studying cryptographic systems to uncover the secret message or to render the cryptographic system unusable. Cryptography is a very important field that ensures electronic devices, and communications between them, are secure and trusted. Modern cryptography focuses on encryption, symmetric or asymmetric, and secure hashing. Though cryptography is important, this study focuses more on cryptanalysis, precisely on Side Channel Attacks (SCA) in general and power side channel attacks in particular. Side channel cryptanalysis exploits a piece of information unintentionally leaked by the cryptographic system to infer a secret message, usually the secret encryption key. The devices of interest in this research are FPGA, though some of the findings made here equally apply to cryptographic algorithms implemented in ASICs. Though early cryptanalysis has been relevant to local attacks scenario, i.e. the attacker has physical access to the device, recent research has demonstrated that remote side channel attacks against implementations in FPGAs are also possible [10][33].

In this chapter, a brief overview of symmetric, asymmetric encryptions, and secure hashing is provided. Then, this is followed by an introduction of the concepts behind side channel attacks (power, electromagnetic radiations, scanning electron microscope). Then we show the applicability of power side channel attacks, including the corresponding thread models and their usage as metrics to evaluate the security level of countermeasures. This chapter also introduces the notion of countermeasures and how

they are used to thwart malicious attacks. Local attacks and remote attacks are introduced here as well, and so is the notion of brute force attack (or exhaustive key search) vs key enumeration attacks. The chapter concludes with a section covering how the success rates of power SCA are computed in the prior art.

2.1 Encryption and hashing

As mentioned previously, in other to protect devices and communication channels against malicious attacks, the following features are desirable properties of secure communications:

- Confidentiality: The data exchanged between the sender and the receiver needs to be encrypted to prevent a third unauthorized party from eavesdropping on the information.
- Message integrity: The sender and the receiver need assurance the message is not altered either intentionally by a malicious agent or accidentally by a system failure. Hashing is one of the common methodologies to ensure data integrity.
- Authentication: The sender and the receiver need to confirm that the other is genuine and are indeed who they claim to be. Authentication, mostly performed by sending Message Authentication Code (MAC) along with a message, is necessary to ensure that an imposter is not sending rogue messages by pretending to be an authorized sender.

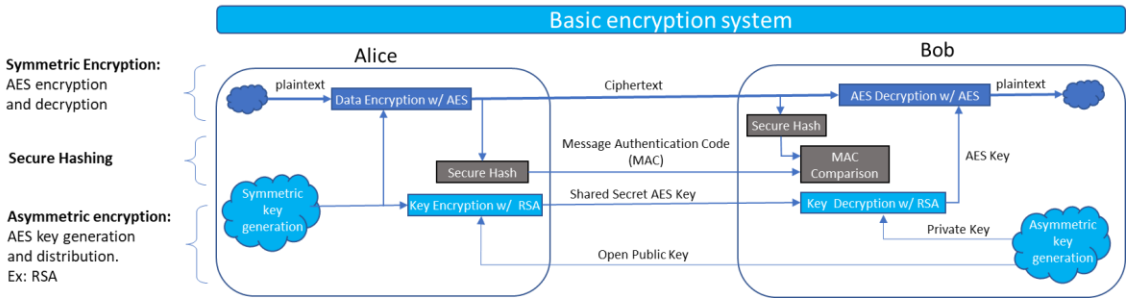


Fig. 1. Overview of a typical encryption system over a communication channel

Encryption algorithms can be categorized as symmetric encryption or asymmetric encryption. Let's illustrate the two concepts by assuming two users, Alice and Bob, want to exchange information secretly, as shown in Fig. 1. Alice generates a secret symmetric key and encrypts its plaintext with a symmetric encryption algorithm such as AES. The encrypted plaintext, called ciphertext, can now be transmitted in an open channel to Bob. In other to decrypt the ciphertext, Bob needs the same encryption key used by Alice. Bob then generates an open/public key which he sends to Alice to encrypt the secret symmetric key. Bob then decrypts the key with another private key known only to him. The pair of private/public keys is known as an asymmetric key and the algorithm using this key is an asymmetric encryption algorithm. It's shown in Fig. 1 that for symmetric encryptions, the same secret key is used for both encryption and decryption. Conversely, in asymmetric encryption algorithms, the encryption key and decryption key are different.

2.1.1 Symmetric encryption

Recently, AES has been the most used symmetric algorithm in cryptography. For symmetric encryption algorithms, the same secret key is used for both encryption and

decryption. Encryption and decryption layers are generally very similar. For an algorithm like Data Encryption Standard (DES), those layers are identical.

Fig. 2 represents an overview of symmetric encryption, showing the same key *k* being used by both the sender and the receiver. Symmetric encryption does not use a compact mathematical description throughout the algorithm between the input and output like asymmetric encryption. Symmetric encryption algorithms have shortcomings like key distribution schemes and cheating by either sender or receiver.

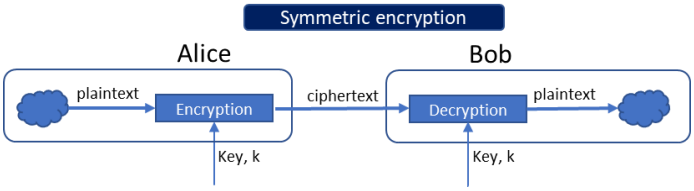


Fig. 2. Overview of a typical symmetric encryption scheme

2.1.1.1 AES

AES was introduced by NIST in 2001 and is based on block cipher Rijndael (from Belgian authors Rijmen and Daemen). It's outlined in FIPS PUB 97 [34]. It's a symmetric block cipher encryption algorithm that uses a key length of 128, 192, or 256 bits. The algorithm is mathematically hard/impractical to attack with brute force. Because of the byte operation nature of AES, software implementation is very efficient in 8-bit microprocessors (like smart cards), however, modern CPUs use lookup tables (T-Box). Hardware implementations in ASIC/FPGA are very efficient and can reach 25Gbits/s with pipelining [35]. AES is the dominant symmetric encryption algorithm for commercial systems. AES is allowed by NSA at the TOP SECRET level with AES256 [36].

As shown in Fig. 3 and Fig. 4, the AES algorithm consists of 10, 12, or 14 rounds for key size of 128bits, 192bits, or 256bits respectively. AES is Rijndael with a block size of 128 [34]. Each round operates with a separate key generated by an algorithm called key schedule (or key expansion). Each round has the following 4 transformation layers, except for the last round that does not have the mix-column layer:

- Key addition layer: add (XOR) the round key to the state
- Byte substitution layer (S-Box): Each byte of the state is replaced, from a lookup table
- Shift row layer: byte permutations
- Mix-column layer: matrix operation that mixes columns (set of 4 bytes)

Power side channel attacks have exploited the absence of mix-column in the last round to launch successful attacks. But, in chapter 6, a scheme to attack the first round even in the presence of the mix-column transformation is introduced.

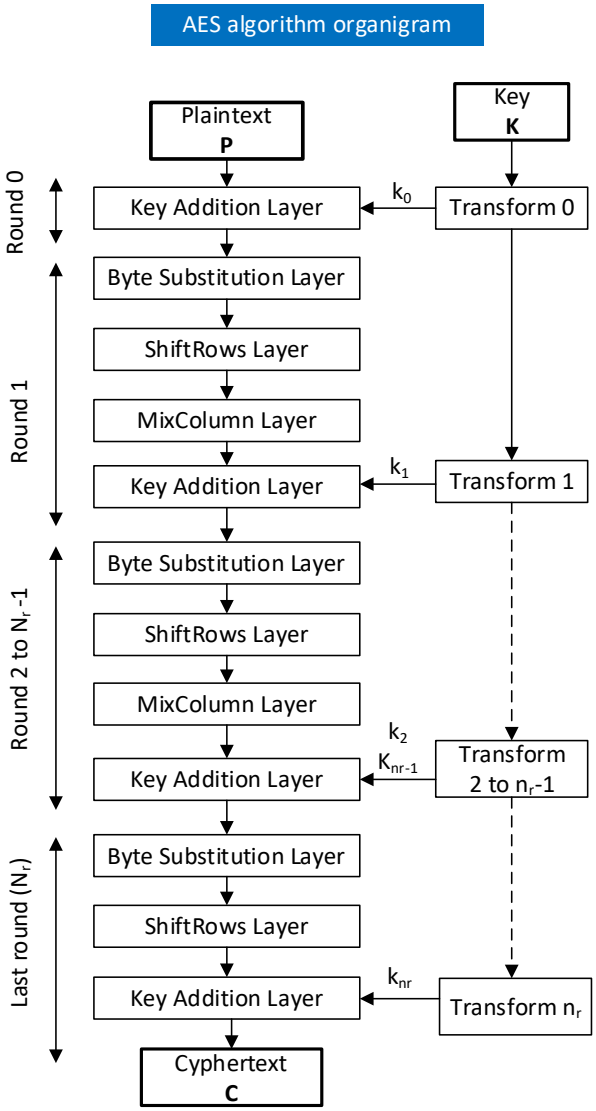


Fig. 3. Overview of AES algorithm [34]

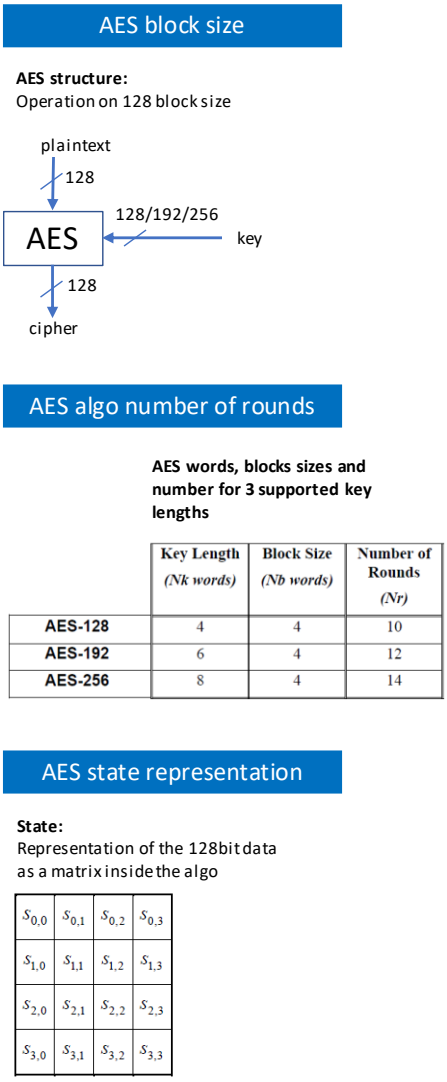


Fig. 4. AES algorithm features and characteristics [34]

Below, an expansion on the AES transformations that are used during the side channel attacks orchestrated in this research is shown. The decryption and the key schedule aspect of AES are not relevant to this research and thus are not addressed.

2.1.1.1.1 S-box layer

The byte substitution layer (S-BOX) is a non-linear transformation that replaces each byte in the state with a unique byte. But for two bytes A and B,

S-Box(A) + S-Box(B) ≠ S-Box(A+B).

The S-Box is a bijective mapping so that one can uniquely reverse byte substitution during decryption. There's no fixed point in the transformation, meaning there is no byte A so that S-Box(A) = A.

2.1.1.1.2 Shiftrows layer

ShiftRows layer and MixColumns layer together are called the diffusion layer. The diffusion layer is linear. ShiftRows cyclically shift to the right of each row of the state matrix by a given number of bytes as shown in Fig. 5.

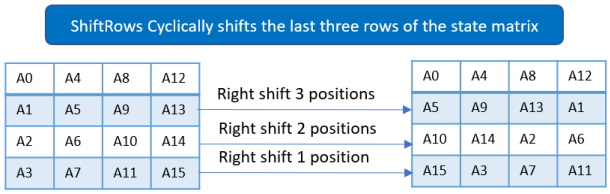


Fig. 5. ShiftRows layer

2.1.1.1.3 Mix-Column Layer

MixColumn is the second part of the diffusion layer. It's linear and mixes each column of the state matrix. Mix-column and ShiftRows transform the state so that after 3 rounds, each byte of the state matrix depends on all 16 bytes of the plaintext. Each output column is obtained by matrix multiplication of a constant matrix with the input column, performed in Gallois Fields GF(2^8), as illustrated in Fig. 6.

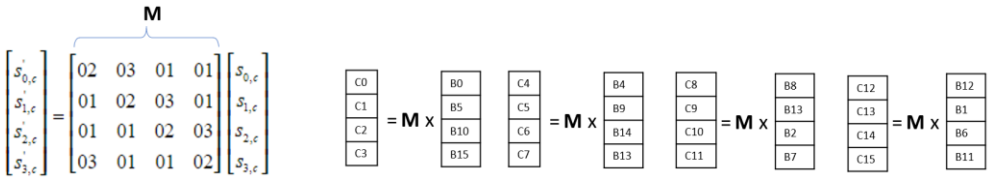


Fig. 6. Mix-Column layer

2.1.1.1.4 Key addition layer

This transformation adds the state to the round key. It's a bitwise XOR operation. This property makes it easy for attacks to be orchestrated on smart card implementations of AES because the key search space size can be reduced to 16x256=4096keys for AES128 [21]. Roundkeys are generated from the key schedule operations.

AES algorithm has shown over time to have a high power consumption for some emerging domains. For those devices in those areas, a lower power consumption and lower area overhead are preferred. Consequently, lightweight ciphers were introduced to fill the gap.

2.1.1.2 Lightweight ciphers

Application domains such as embedded and IoT are very compact and have very low power consumption. They, therefore, require ciphers with very low hardware implementation complexity algorithms. Because IoT devices are typically low resource devices, there's the need to use lightweight crypto algorithms to encrypt data before transmission to the cloud.

The most common lightweight ciphers studied in the literature for side channel attack susceptibility are PRESENT [29], SIMON-128 [32][39][40], and PRINCE [37][38]. It is worth noting that lightweight ciphers standardization is currently an ongoing project at the National Institute of Standards (NIST). The final round of candidates and the winner were announced and can be found on the NIST website [27].

2.1.1.2.1 PRESENT

PRESENT is a substitution-permutation-based network block cipher that contains 31 rounds. As opposed to AES which has block sizes of 128 bits, PRESENT block length is 64 bits. It supports a key length of 80 and 128 bits, compared to 128, 196, and 256 for AES. Each round has 3 layers/transformations as follows:

- addRoundKey: a round key is added to the state at the beginning of each round, just like in AES
- sBoxLayer: In AES S-box replaces a byte with another byte from a lookup table (or Galois Field computation). In PRESENT, the S-box is done on 4-bit and thus allows a much more compact implementation.
- pLayer: This is the mixing layer performing bit permutation. The operation can be expressed in the following way: bit *i* is moved in bit position *P(i):*

$$P(i) = \begin{cases} i.16 \bmod 16, & i \in \{0, \dots, 62\} \\ 63, & i = 63 \end{cases} \quad (1)$$

where *i* is a bit position in the block.

2.1.1.2.2 SIMON

SIMON was designed by the National Security Agency (NSA) as a lightweight cipher to provide security for resource-challenged devices [41]. Much like AES and PRESENT, SIMON is a block cipher. Its block size is 2*n* bits, where n is the word size. The key length is *m.n*, where *m* is the number of words. SIMON64/96, which is addressed in this research, processes a 64-bit plaintext block with a 96-bit key and has 42 rounds of encryption.

As shown in Fig. 7, SIMON does not rely on S-Box like in AES and PRESENT for non-linear layers, rather it has bitwise AND between selected bits following the left circular shift S(i), with i being the shift value.

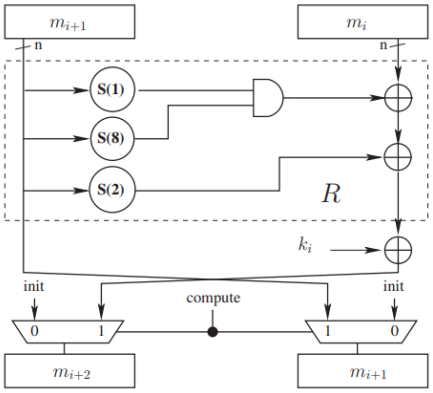


Fig. 7. Structure of a SIMON algorithm [39].

2.1.1.2.3 PRINCE

PRINCE is another low-cost, low latency cipher suitable for devices constrained in computing resources like in IoT for example. The block size is 64 bits and the encryption key length is 128 bits. It has 5 round functions, one intermediate processing, and 6 inverse round functions [37].

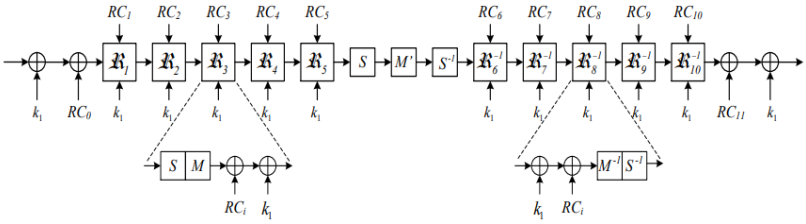


Fig. 8. Structure of a PRINCE algorithm [42].

Each PRINCE round and inverse-round has 4 layers:

- k_i-add: the subkey k_i is XORed to the state

- S-Layer: A 4-bit substitution is performed, similar to SIMON.
- The Matrices: The 64-bit state matrix is multiplied with 64x64 matrix M or M' (for the inverse round).
- RC_i-add: A predefined 64-bit round constant RC_i is XORed with the state

2.1.2 Asymmetric encryption and RSA

A feature of an asymmetric encryption algorithm is the fact that it needs two different keys: one public key is used for encryption and a private key is used for decryption. The asymmetric key encryption algorithms are mostly based on number-theoretic functions, making the encryption/decryption mathematically complex. They use popular one-way functions like integer factorization, the case of RSA, or Discrete Logarithm Problem (DLP) used in the Diffie-Hellman Key Exchange (DHKE). A common use of asymmetric encryption is digital signatures.

Fig. 9 is an example of asymmetric encryption showing the sender Alice transmits a message to the receiver Bob. Bob generates two keys when communication is initiated between the pair, then sends the public to Alice for encrypting messages. Bob then uses the private secret to decrypt the ciphertext received from Alice.

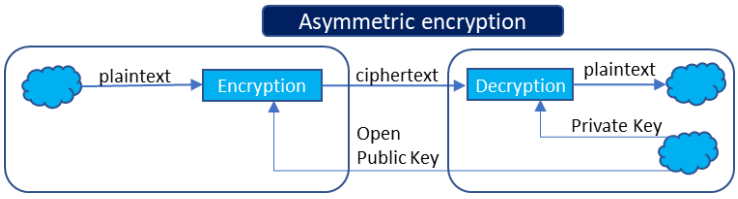


Fig. 9. Overview of a typical asymmetric encryption scheme

The asymmetric encryption algorithm of interest in this research is RSA. RSA scheme is most used in practice for the encryption of small pieces of information like the symmetric encryption key, in key transport, as shown in Fig. 1, and digital signatures, for digital certificates on the internet.

Given an RSA public key $k_{pub} = (e,n)$ and private key $k_{pri} = d$, the encryption and decryption algorithms are given as follows:

$$Encrpytion: c = m^e \ mod \ n \qquad 2$$
$$Decryption: m = c^d \ mod \ n \qquad 3$$

Where: *m* is the plaintext message, *c* is the cyphertext, with $m, c \in \{0, 1, \dots, n-1\}$. *e* is referred to as the public exponent and *d* is the private exponent. The integrity of the algorithm is based on its computational complexity because the numbers *e*, *d*, and *n* are generally very large, usually 1024 bits or more. The modulus *n* is the product of two large prime numbers. The exponents *e* and *d* are generated with a key generation algorithm that will not be discussed here because it's beyond the scope of this research.

Various methods have been shared in the literature to simplify or accelerate the computation of the cyphertexts or plaintexts in both hardware and software implementations of RSA [43]: fast exponentiation, Chinese Remainder Theorem (CRT), and successive modular multiplications. The latter technique will be used to study power side channel attacks on RSA implementation in an FPGA.

2.2 Side channel attack principle

A side-channel attack works on the premise of exploiting a piece of information that a system leaks unintentionally to the attacker while completing its task of interest. As in any

attack scenario, the malicious agent has more chances of success when they can either read the system output information or drive known information into the input in addition to observing leaked information. Side channel attacks are the opposite of direct attacks, which involve brute force attack by enumerating all possible keys, of the cryptographic device's algorithm implementation.

The input or output information garnered by the malicious agent, combined with the measured leaked information, i.e. power consumption in our case, can be analyzed to guess the system encryption key. This type of side channel attack, classified as passive side channel attacks [1], is the most practically implementable type of side channel attack. Active side channel attacks require the malicious agent to tamper with the target, which not only brings more complexity to the attack but most often requires physical access to the target device.

The leaked information measured by the malicious agent can be the power consumption (power side channel attack), or the Electromagnetic radiation (EM side channel attack), the visual image presented to a microscope or photographic equipment (example: Scanning Electronic Microscope Side Channel Analysis), the delay the circuit takes to process a reference signal (delay analysis), or the response to a glitch injection into the input of the circuit (fault attack). The sections below describe in a little more detail the first 3 types of side channel analysis and then our research will focus primarily on power side attacks.

2.2.1 *Power side channel attacks*

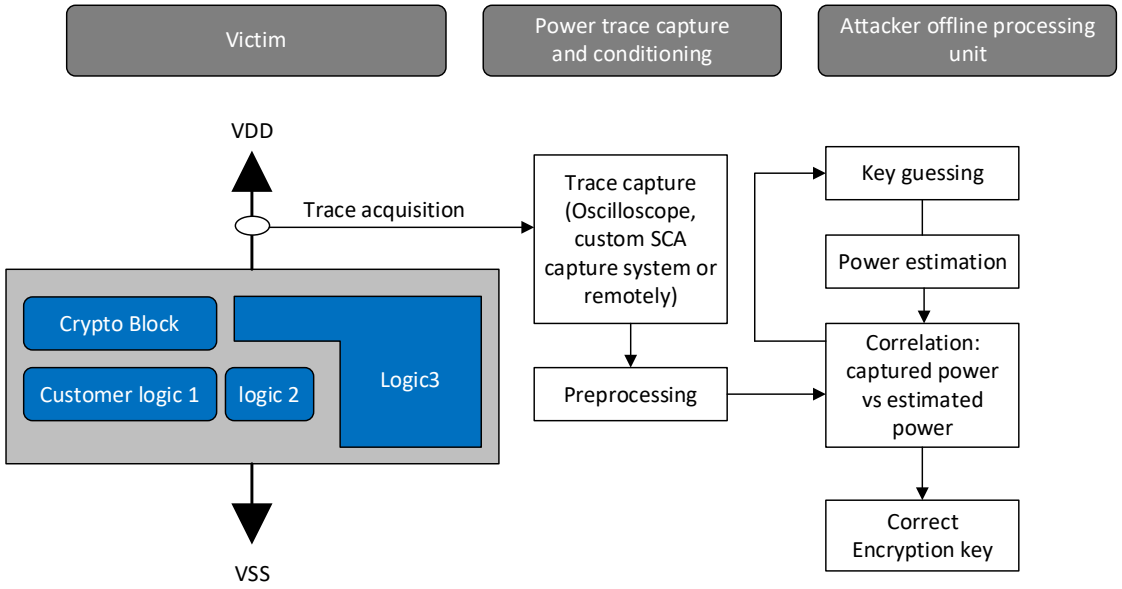


Fig. 10. Overview of a typical power side channel attack system

In passive side channel attacks, measuring the system leaked power is not considered tampering, as it's merely observing the system behavior from outside (Fig. 10). The power leaked by our FPGA in one of our experiments is shown in Fig. 11. The last 10 current spikes correspond to the 10 rounds of AES128 encryption. Knowing this information, the goal here is to focus on the last round of encryption and correlate that power spike against potential guesses, with correlated power analysis (CPA) methodology or Kullback-Leibler rank. While there's a focus on these two distinguishers in our research, other works have used distinguishers such as Simple Power Analysis (SPA) and Differential Power Analysis (DPA).

The structure of the cryptographic algorithms allows for multiple scenarios for attacking the encryption key. A malicious agent can either drive the inputs and measure the power or read the output and measure the power. Deriving a set of known ciphertexts and using

the hamming distance model [2] as the estimated power can lead to uncovering the original key. For AES attacks, the malicious agent considers that they have knowledge of the output ciphertexts or knowledge of the input plaintext, then attempt to decode the last round encryption key, based on the measured power. After successfully guessing the AES128 last round key, the key expansion is then rolled back to get the encryption key input to the system. The process of rolling back the expansion key is beyond the scope of this research and will not be addressed here. For an attack on the first round, the successfully guessed key in the algorithm encryption key.

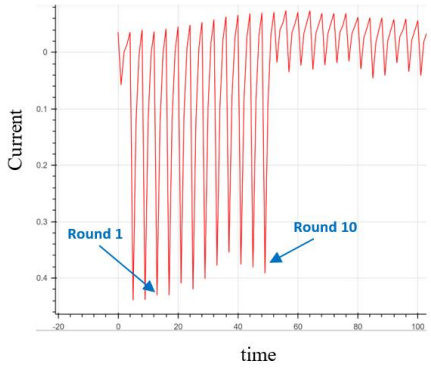


Fig. 11. Power measurement of the FPGA while running an AES128 encryption

2.2.2 *EM side channel attacks*

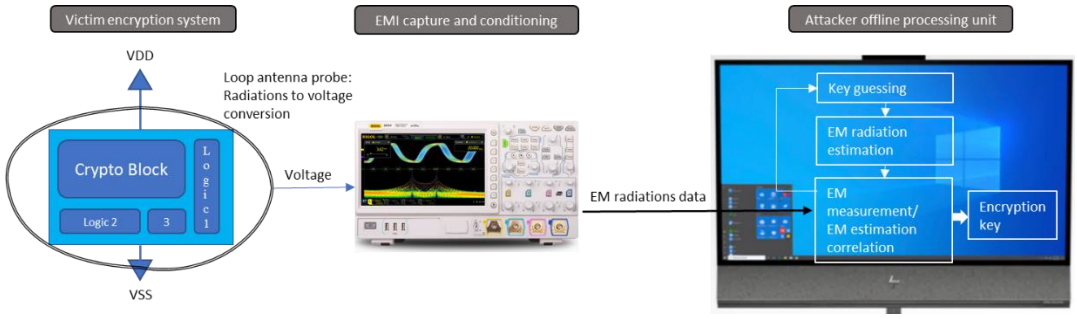


Fig. 12. Overview of a typical EM side channel attack system

In the previous section, we shared that power side channel attacks exploit the power consumption of the encryption to extract the secret encryption key. The IC power dissipation has three sources: leakage current, short circuit current, and switching current. The leakage current is a DC component and typically does not change with the IC workload. The short circuit and the switching currents are AC components. However, the switching current which represents the parasitic capacitors charging/discharging currents is more dominant and is representative of the logic running in the IC. This AC component, while flowing through the IC inductive power grid, generates an electromagnetic field that radiates outside the IC package.

EM side channel attacks consist of capturing the EM radiation of the IC with a probe and running analysis similar analysis to the ones in power side channel attacks.

The output voltage of the antenna probe can be represented as follows [3][4]:

$$V = -\frac{d\Phi}{dt} \qquad 4$$

Where Φ, which is the magnetic flux around the probe, is given by:

$$d\Phi = \mu\vec{H}.\overrightarrow{dS} \qquad 5$$

$$\overrightarrow{dH} = \frac{(I\vec{dl})\wedge\vec{u}}{4\pi x^2} \qquad 6$$

μ is the permeability, $\vec{H}$ is the magnetic field strength, $\overrightarrow{dS}$ is a unit surface, $\vec{u}$ is the unit vector normal to the surface, $\overrightarrow{dl}$ is the elementary conductor of the circuit and *x* is the between the probe and the elementary conductor $\overrightarrow{dl}$.

Similar to SPA and DPA, Simple EM Analysis (SEMA) and Differential EM Analysis (DEMA) are two methodologies used for side channel EM analysis.

2.3 *Relevant types of side channel attacks*

There are numerous types of side channel attacks published in the literature. The ones relevant to IoT devices and our study are local vs remote attacks, and passive vs active attacks. With local attacks, the malicious agent has physical access to the target device to capture the measurements needed to perform side channel analysis. In the case of remote attacks, the agent can capture the leaked information remotely with no physical access to the device. Passive attacks occur when the device naturally and unintentionally leaks side channel information to the outside world, in the course of its normal operation. However, with active attacks, the malicious agent has to modify the device's intended behavior to forcibly produce or alter the side channel information.

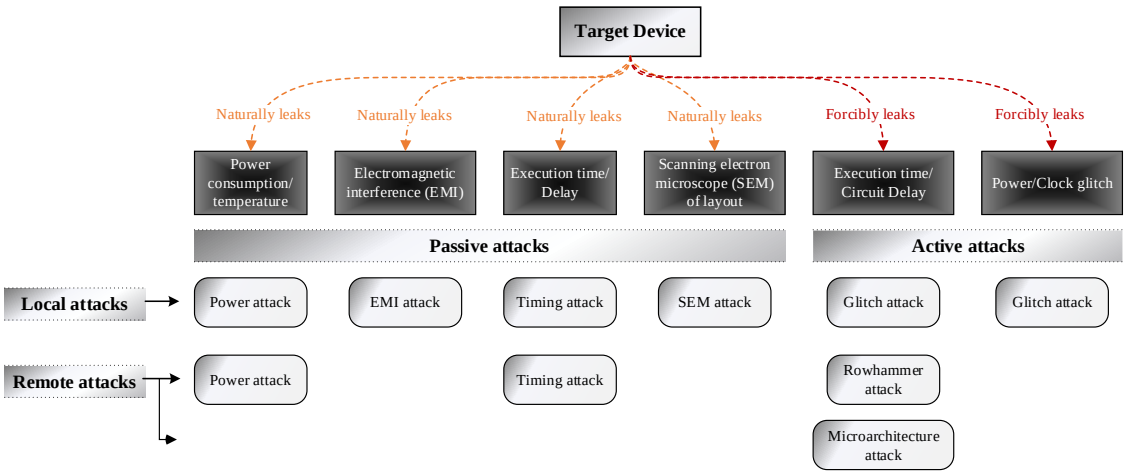


Fig. 13. Classification of relevant types of side channel attacks

The following passive side channel information is most likely to be leaked by IoT devices implementing cryptographic algorithms, leading to the undermentioned types of attacks (Fig. 13):

- Power consumption or temperature rise: local and remote power side channel attacks [37][10]
- Electromagnetic emanation: local electromagnetic interference attacks [4]
- Program execution time of circuit delay: remote and local timing attacks
- Scanning electron microscope (SEM) of device layout: local SEM attacks [68]

Active attacks force the alteration or leakage of the following side channel information, which leads to the undermentioned types of attacks (Fig. 13s):

- Execution time or circuit delay: glitch attacks, rowhammer [143] attacks, and microarchitecture attacks [144]
- Power and clock glitch: local power and clock glitch attacks [145]

2.4 Applicability of power side channel attacks

2.4.1 Power side channel attack targets

Systems that are potentially at risk of being targeted by attackers are most cryptographic devices that can reasonably provide secret info to a malicious agent. They include smart cards [5][6], cryptographic tokens, microcontrollers, CPU, mobile devices, FPGA, ASIC, and others that implement cryptographic algorithms.

Power side channel attacks were pioneered in 1999 and first demonstrated on 8-bit smart cards with differential power analysis (DPA) [9]. They have since garnered interest as improvements on that first differential power analysis methodology have been made and more effective power analysis methodologies have been developed. Side channel attacks

on mobile devices have also gained attention because it has become easier with an open operating system, for a malicious agent to install a piece of software inside the devices remotely to gather secret information [1]. The remote accessibility of FPGA in modern datacenter applications is one of the reasons they have increasingly become of interest by attackers and researchers.

Attack on FPGA is a well-studied area of side channel attacks because these devices present exceptional flexibility to implement and update algorithms in the field without requiring a costly chip re-spin. Because of this flexibility, multiple companies, especially Cloud Service Providers (CSP) are increasingly adopting FPGA to implement their encryption applications. Other areas that present security threats where researchers have shown interest are:

- FPGA bitstream protection against IP theft,
- Cloud infrastructure acceleration with the advancement of Infrastructure Processor Units (IPU)
- FPGA in multitenancy models, aka FPGAaaS
- IoT devices, especially with lightweight ciphers because of their low resource requirements

2.4.2 Threat models of FPGA and ASIC

FPGAs that have been adopted in various fields have been subject to various security threats where a malicious agent either tries to stealthily uncover secret information or just causes harm by taking down the component. ASICs have also been the target of various malicious agents for the same reason. The types of attacks that these devices are

subjected to depend on the nefarious objectives of the agent. Some agents target denial of service by rendering the component useable, others want to uncover the secret key on the encryption algorithm running inside the component.

The soft logic programmed inside an FPGA can be the target of individuals who are trying to steal the bitstream. To protect those intellectual properties (IP) against theft, most FPGAs encrypt the bitstream during the configuration by using standard encryption algorithms like AES. However, just like any mathematical encryption implementation, it has been demonstrated that the bitstream encryption scheme can be recovered with power side channel attack techniques developed in the literature [30].

In datacenter applications, cloud acceleration has been an emerging area where FPGAs are used to accelerate functions like storage, packet processing, and network function virtualization. In typical IPU applications, encryption functions are implemented either as soft logic or as hardened IP. Encryption needs from the host computer are then offloaded to the attached FPGA accelerator resulting in a substantial throughput increase because the encryption operations in the FPGA are performed in a fraction of the time it will have taken to perform in software. Also, offloading to these accelerators means expensive CPU compute cycles are reserved for important operations and thus increasing the server's overall performance per watt per dollar. Unfortunately, these encryption implementations are susceptible to power or EMI side channel attacks. Malicious agents exploit the power or EMI involuntarily leaked by the FPGA to attend to recover the secret encryption keys of the cryptographic algorithms. These power or EMI side channel attack techniques are as effective on ASIC as they are on FPGA because the power

consumption or EMI signatures of the devices are similar when running the same algorithm.

With cloud workloads scaling and growing, FPGAs are also been used in datacenters by where Cloud Service Providers (CSP) rent out a fractional portion of the fabric inside an FPGA to multiple clients to install and run their cloud acceleration functions. This multitenancy model called FPGA as a Service (FPGAaaS) has opened the door to a threat model where a malicious agent can rent the space adjacent to another customer cryptographic engine, install a trojan logic that monitors the power consumption leaked by the neighbor, then perform power side channel attack to uncover its secret encryption key. Timing attack threat models have also been demonstrated in these multitenant computing environments [31].

Besides cloud acceleration, FPGAs are also used in IoT devices sitting at the edge that transmit sensor data to cloud devices for processing are also the target of attacks by malicious agents trying to steal secret data. Because of the ubiquitous nature of these edge devices, the data transmitted to the cloud must be encrypted because more often than none, they contain sensitive confidential information [32]. IoT devices are typically power constraints and therefore are more suitable for lightweight ciphers (like SIMON, PRESENT, etc.). These symmetric ciphers share the same vulnerabilities to SCA attacks as AES and DES. Hence, DPA and CPA are also used by malicious agents to attempt to uncover the secret key on the cryptographic algorithms.

2.5 *Local attack vs remote attack on FPGAs*

FPGA side channel attacks can be classified as local vs remote. Local or direct side channel attacks are implemented when the agent has physical access to the targeted device (FPGA in our case), by observing the current consumed by the device [1]. Such current constitutes leaked information that can be exploited by the malicious agent to guess the algorithm encryption key. Fig. 14 depicts an example of a local attack vs a remote attack.

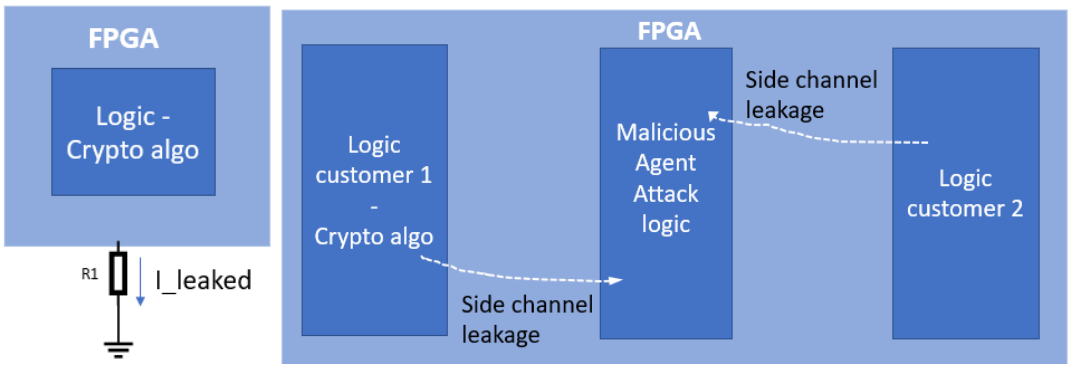


Fig. 14. Side channel leaked information of an FPGA running a cryptographic algorithm. Left: power leaked for a local attack. Right: malicious IP running in an FPGA to monitor local information and sending back to the malicious agent.

With the emergence of the cloud computing field, like FPGA as a Service (FPGAaaS), more cryptographic algorithms are implemented in an FPGA located in a data center, where the compute fabric is shared with unknown workloads from unknown customers. Such co-implementation of multiple algorithms on the same FPGA fabric allows a malicious agent to attack against a cryptographic algorithm implemented nearby to decipher the encryption key. The malicious agent who is renting partial sectors in the FPGA will be running a snooping IP that monitors vital information of the FPGA, like the local voltage and temperature. The feasibility of such an attack has been demonstrated

by [10] where a malicious program consisting of a ring oscillator (RO) delivers a clock frequency depending on the IC local voltage. Time-to-digital converters (TDC) have also been demonstrated as an effective way to monitor nanosecond scale transient voltage fluctuations in an FPGA [33]. For both the RO and the TDC voltage monitoring scheme, the digital information returned is a representation of the local voltage, i.e where the malicious agent logic is implemented within the fabric. The voltage information is then used to run a power analysis to attempt to guess the encryption key of a cryptographic algorithm implemented nearby.

2.6 Power side channel attack usage as a security metric

An interesting aspect of side channel attacks is that they can not only attack crypto algorithms but can also be used to evaluate the strength of a security feature implemented in a device. An example application is the use of a power side channel attack to evaluate the strength of the AES encryption of gate key-bits in a logic locking technique.

With the prevalence of IC fabrication outsourcing, security, and protection of intellectual property (IP) against theft and maintenance of the manufactured IC have become paramount. A promising method involves the implementation of logic locking, i.e. inserting several key gates at strategic locations inside the original code, then storing the keys in a tamper resistant memory. Since logic locking inserts several key gates in the netlist, those keys are required by future integrators to unlock the logic, or else the code will yield unusable results [15]. In [19] and [20] logic locking implemented with polymorphic gates using CMOS technology is used to protect an IC from piracy. Then an

AES cryptographic engine with a 32-bit key is used to protect the logic locking scheme against the reverse engineering based attack Boolean Satisfiability (SAT).

The effectiveness of the logic locking implementation in [15] is verified by performing DPA and using mutual information analysis techniques to estimate the success rate.

2.7 Brute force vs key enumeration

Since uncovering cryptographic keys with power analysis consists of guessing keys first and then comparing the estimated power with the measured power, a problem arises regarding how many keys the malicious agent can guess to successfully uncover the key in a prompt manner.

Brute force attacks, where all possibilities of the key are tested, require the agent to test all 2^{256} keys for AES256. This is not practical, as it takes a long time for power analysis to be completed. However, due to the nature of the AES algorithm, the secret key can be attacked one byte at a time when targeting the last round of an FPGA/ASIC implementation [21]. This reduces the number of guessed keys required per operation to 2^8=256, for a total of 16x2^8=4096 keys for 128-bit AES or 32x2^8=8192 for 256-bit AES. This is far more practical for an attack [21].

The authors of [22] demonstrated an attack on the first round of AES128 with a key search space size of 16x256; however, this was carried out on an 8-bit MCU. This naturally resulted in high correlations between the 8-bit key guesses and the power leakage. However, when AES is implemented in hardware/FPGA, and the attacker does not have access to the output of the last round (cyphertext), this weakness is not present

and thus requires key enumerations and/or ranking techniques to avoid having to brute force all possibilities for the keyspace.

Side channel analysis techniques have been used for key enumeration techniques to reduce the number of keys to guess in software implementations of AES algorithms [23][22]. Various methods for attacking software-based encryption have been proposed that reduce the key set needed to attack them. In [24], the authors enumerated likely candidate keys using time- and memory-efficient algorithms. They could reduce the complexity to 2^{48} for AES128 and run an attack in 30 hours. [24][23] and [26] generated only reduced sets of 2^{40} and 2^{50} key candidates, respectively, and then rated the guesses according to their respective probabilities based on a Bayesian extension.

2.8 Estimating the success rate of an attack

Various methods for estimating the success rate of an attack have been published in the literature. The success rate of an attack is defined as the probability that a secret key is recovered with a given number of measurements. In some attack scheme, recovering a partial key constitute a success, assuming that the key bits recovered can help lead to the information of interest. [24] defines the 1^{st} order success rate as the probability that the defined key guessing entropy ranks the correct key first. However, [25] defines a success rate in a more straightforward method: the Euclidean distance fluctuation devia. A correlation power analysis is based on the premise of calculating the Pearson correlation coefficients of multiple guess keys, based on measurements with a given number of plaintexts. The maximum correlation coefficient corresponds to that of the candidate key. The Euclidean distance fluctuation is a characterization of the separation between the

maximum correlation coefficient and the second-highest, with the maximum being one. So, the closer the Euclidean distance fluctuation is to 1, the highest the confidence is of having recovered the correct key.

As mentioned before, the encryption algorithms described in the literature may leak side channel information that malicious agents exploit to try to uncover secrets about the information been transmitted or stored. Such information is the device power consumption. The following chapter describes prior art on Power Side Channel Attacks.

3 PREVIOUS WORK ON POWER SIDE CHANNEL ATTACKS

Power SCA exploits information leaked by the system about the secret key (private key for asymmetric algorithms or encryption for symmetric algorithms), through the power delivery network physical channel. The attacker must measure a piece of information directly or indirectly related to the device's power consumption, then use a distinguisher to extract enough information that closely matches the behavior of the system during encryption with the secret key. The main distinguishers used in the literature are Simple Power Analysis (SPA), differential Power Analysis (DPA), and Correlation Power Analysis (CPA). Other statistical methods such as Test Vector Leakage Assessment (TVLA) and Kulback-Leibler divergence are also used for discriminating the mean information between the measured leaked power and an estimated power, which the attacker theorizes represents the system power.

Just as power side channel attacks have proven to be effective in cryptanalysis, countermeasures have also shown promise in thwarting those side channel attacks. Countermeasures are needed to protect FPGA bitstreams' encryption mechanism from being broken for example or to protect any other encryption algorithm from leaking pertinent information to a malicious agent. Multiple countermeasures have been proposed to thwart side channel attack attempts.

3.1 Correlation power analysis and information theory for side channel attacks

3.1.1 Simple Power Analysis

The authors of [9] are widely known to have pioneered power side channel attacks with the publication of the principles of Simple Power Analysis (SPA) and Differential Power

Analysis (DPA) in side channel crypto-analysis (also referred to as “cryptanalysis” in other works of literature). The paper has been cited over 7000 times since its publication at Crypto’99 in 1999. SPA directly measures system power to reveal the system's internal operations. They measure the current of a smart card while it’s running a DES encryption. On a single trace containing 5000 points, they show that one can observe the 16 rounds of DES. Furthermore, zooming in at each processor clock cycle reveals differences in the power consumption of different microprocessor instructions. The DES key schedule has a rotation of a 28-bit register. Because a conditional branch is used to implement the rotation of the last bit, i.e the wrap-around of the last bit if it’s a ‘1’, the power consumption signature shows a spike if the branch is taken. By observing these spikes at each consecutive clock cycle, the bits of the encryption are uncovered one by one.

The paper that claims to be the first to ever demonstrate an experimental power analysis attack on an FPGA is [57]. The setup is a daughterboard with Xilinx XCV800 FPGA, plugged into the motherboard of a PC. The ceramic bypass capacitors of the power delivery networks are not removed. They provide strong evidence that Elliptic Curve (EC) cryptosystems without countermeasures are vulnerable to SPA attacks. The authors can extract the key (001100) of a 160-bit EC point multiplication by analyzing the power consumption trace.

3.1.2 Differential Power Analysis

As mentioned in the section above, [9] is the first to use DPA for SCA cryptanalysis. For DPA, an attacker first observes the power traces of several given encryptions for given

cyphertexts. Then it guesses keys and correlates the estimated power consumption to samples. The higher the number of samples the lower the correlation error. In the paper, the authors propose a selection function, that depends on the ciphertext and the guessed key, which computes the value of the bit key *b* entering the DES S-Box at the beginning of the 16th round, with probability ½. The differential power is then computed with the equation given by the authors. In the paper, they show illustrations of power consumption of the smartcard for DES encryption of a given plaintext and the differential power of three different encryption keys, taken with 1000 samples: one for the correct encryption key, and two for two incorrect keys. For the correct key differential power trace, spikes appear in regions where the DPA selection function correlates to values been processed. Such spikes do not appear on the differential power trace of the incorrect keys.

3.1.3 Correlation Power Analysis

3.1.3.1 Principle

Correlation Power Analysis (CPA) is a distinguisher that followed DPA described above. For correlation power analysis, one calculates the correlation coefficient, which is the covariance between the leaked power and the estimated power, normalized by the product of the standard deviations, to keep the number between -1 and +1.

Let us assume that the attacker has M time-dependent observations (power measurements), and s/he theorizes that they will be correlated with estimated power H. The correlation coefficients between these two variables (also called Pearson correlation coefficients) are defined as [59][60]:

$$\rho_{MH}(k,t) = \frac{P\sum_1^P M_i(t)H_{i,K} - \sum_1^P M_i(t)\sum_1^P H_{i,K}}{\sqrt{P\sum_1^P M_i^2(t) - \left(\sum_1^P M_i(t)\right)^2}\sqrt{P\sum_1^P H_{i,K}^2 - \left(\sum_1^P H_{i,K}\right)^2}} \quad 7$$

With: P: number of measurements; $H_{i,k}$: estimated power of the system if running encryption of plaintext *i* with guess key *k*; $M_i(t)$: power measurement while running encryption *i* with the secret key.

If the estimated power consumption model accurately reflects the measured power, then the deviation from the mean of the correct key guess deviates from the mean in the same magnitude and direction as the measured power.

3.1.3.2 Other relevant literature on Correlation Power Analysis

Besides the work presented in [59][60], the authors of [58] have also investigated the application of CPA for SCA, in particular, they have studied FPGA implementations of block ciphers DES and AES. The basic hypothesis made in the paper is that an estimation of the power consumed by an FPGA at time *t* is given by the number of bits that change values in the registers. They perform analysis with simulated data and measured data on Xilinx Virtex and Spartan FPGAs. With simulated data, they conclude that the minimum numbers of measurements needed to uncover the secret key are between 300-600. But they also find that the results vary depending on the attacker's knowledge of the design details. With measured data, the correlation coefficient distinguisher requires about 1200 traces to yield the secret key. This paper further expands into simulating a system with additive noise on the power traces and estimates the success rate of the attack in terms of the signal-to-noise ratio. The success rate defined depends on the probability to distinguish the correlation coefficient of the candidate key from that of the incorrect

guess key. The theoretical success rate reaches 1 with ~2000 plaintexts when the correlation coefficient is 0.09. It is worth noting that the success rate of a power SCA depends on the metric, so it's hard to compare this work with others who have defined different metrics.

While most authors have defined the leak model for CPA as simply the hamming distance of the portion of the algorithm of interest, the authors of [59] define a broader linear model as:

$$W = aH(D \oplus R) + b \qquad 8$$

Where a and b are two independent scalars and H() is the hamming weight (Hamming distance of D and R), W is the estimated power consumed, R is the secret key, and D is the observed register value.

They defined the correlation coefficients between the measured power W and the hamming distance by:

$$\rho_{WH} = \frac{cov(W,H)}{\sigma_W \sigma_H} \qquad 9$$

The inference of the secret key is based on the following reasoning.

Assuming that R' is another candidate instead of the true reference R. If R' has *k* bits that differ from the reference R, then the correlation coefficients of R' is:

$$\rho_{WH'} = \frac{cov(W,H')}{\sigma_W \sigma_{H'}} = \rho_{WH}\rho_{HH'} = \rho_{WH}\frac{m-2k}{m} \qquad 10$$

With:

$$H = H(D \oplus R)$$

$$H' = H(D \oplus R')$$

$$k = H(R \oplus R'),$$

m is the number of bits in R and R'

The authors demonstrate the following statements:

- The solution is unique: the correlation coefficient of R is always higher than any R' candidate
- The correlation factor is capable of rejecting the wrong candidates for R
- In an 8-bit system, if a single bit is wrong, then the correlation factor is reduced by ¼.

3.1.4 Kullback and Leibler divergence

Let's first introduce the concept of information theory: information and sufficiency theory is a generalization of information theory [60]. It determines the amount of information to extract from a system that can comprehensively summarize the system for a particular estimation. So, information is sufficient if estimates computed from it could have statistically come from the full system directly. In general, information sufficiency is concerned with the criterion of sufficiency [62] based on a reduced set of statistical observations of a system's parameters.

Now to define the Kullback-Leibler theory: let's consider a system $\boldsymbol{S}$ with 2 random observations $\boldsymbol{X_i}$, i=1,2. In the probability space $(\boldsymbol{S}, \boldsymbol{X})$, it exists functions $f_1(x)$ and $f_2(x)$ positive finite that characterize the observations $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ on $\boldsymbol{S}$.

$$0 < f_i(x) < \infty, i = 1,2$$

The information in random variable *x* for discriminating between information on $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ is defined as:

$$log\frac{f_1(x)}{f_2(x)} \qquad 11$$

With this definition, the mean information for discriminating between $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$, based on observations $\boldsymbol{X_1}$ is derived as [61]:

$$\int f_1(x) log\frac{f_1(x)}{f_2(x)} dx \qquad 12$$

The above equation defined by Kulback and Leibler is commonly known as the Kulback-Leibler divergence theorem.

For the particular case where $f_1(x)$ and $f_2(x)$ are defined as the probability density functions of observations $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$, one can apply this formula to two given distributions to compute the expected value of the divergence between the two distributions.

3.1.5 Test Vector Leakage Assessment (TVLA)

Similarly to the Kulback-Leibler divergence, TVLA is used to evaluate whether two populations have the same distribution. It is also used to evaluate how much information about the secret key an encryption algorithm implementation leaks. It uses Welch's T-test as defined in the following equation:

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{N_1} + \frac{\sigma_2^2}{N_2}}}$$

The series index 1 and 2 represent two sets of power traces, one fixed and the other chosen randomly. μ_i, σ_i^2 and N_i represent the mean, the variance, and the number of traces for the population *i* (= 1,2) [97].

If the value of *t* exceeds a predetermined threshold of 4.5, as defined by NIST sponsored NIAT workshop 2011, the measured traces are said to carry sensitive distinguishing information which could be exploited by a malicious agent to try and uncover the secret key. Note that TVLA method itself does not reveal that secret key.

Welch's T-test determines whether two distributions are different from each other. T-test uncovers leakage of information without mounting an attack. But it does not provide info on how hard an attack is and cannot recover the secret key or other sensitive info.

3.2 Power side channel attack countermeasures

A few examples of power SCA countermeasures recently published involved protection against power SCA by exposing a fake key to the potential attacker [11], protection of elliptic curve cryptosystems [12], scrambling the power monitored by the attacker using on-chip regulators [70] or dynamic voltage scaling (DVS) [14].

Also, masking techniques have been proposed as very effective against side channel attacks in the literature. Masking consists of modifying the execution of the algorithm implementation so that the power or EMI leaked is modified in a way that will not correlate to the power consumption of the unmasked algorithm implementation [16]. It has been proven that masking of AES by fake key addition reveals the fake key instead of the candidate's secret key. To some extent, this type of countermeasure is resistant to power SCA methods like CPA, the difference of means (DPA by Kocher), and the t-test. Masking is so far thought to be the most efficient countermeasure against power SCA [17][18].

Finally, the last countermeasure that is introduced here is key rotation. The premises of the power side channel attack it to run encryption with several plaintexts and the same key each time measuring the power consumption needed to perform offline DPA. By rotating the encryption key, one can thus prevent the attack from having the leaked information of a single key. So, the implementation needs to avoid key exhaustion by randomly rotating the encryption key.

3.2.1 Masking and Hiding

Unlike the majority of papers published on countermeasures of SCA that focused on protecting the leaked information (Electromagnetic or power consumption), the authors of [17] have modified an AES implementation to return a fake key when a malicious agent attempts to attack the system with power SCA. They demonstrate that the new countermeasure is resistant to SCA methods like CPA, the difference of means (another appellation for DPA by Kocher), and the t-test. The experiments were implemented on a Xilinx Virtex 5 FPGA where an AES-128 implementation not only leaked the incorrect key but also did not show any sign that the system was protected against SCA. The masking scheme presented by the authors consists of adding a mask key *Kmask* to the real key *Kreal* and feedback the resulting fake key *Kfalse* into round 0 of the AES algorithm. In the last round, the mask is removed by adding the mask to the output to generate the unmasked ciphertext.

$$K_{false} = K_{real} \oplus K_{mask} \qquad 13$$

The authors implement the masking scheme with either one extra clock per round or two clocks per round. With two clocks per round, the layers AddRoundKey, ShiftRows, and

SubBytes are executed first in the pipeline, then the MixCloumn layer is executed in the second clock. With only one clock per round, the buffer after the MixColumn layer is eliminated. The implementation of these two techniques generates the same extra resources in the FPGA versus an unprotected system: 13.8% and 13.2% of the FPGA lookup tables for one register and two registers, versus 9.2% and 9.1% of lookup tables without the protection by faking the key. In the same experiment, they demonstrate that a CPA analysis on the AES-128 implementations each reveals the same fake round key of 211 while the real key is 134.

Three-share threshold implementations (TI) are also used as a masking countermeasure in cryptography. The authors in [64] apply TI to lightweight ciphers and AES in Internet of Things (IoT) applications and use DPA to verify improved protection against SCA. TI is a change in the cryptographic algorithm implementation to protect against SCA. With TI, transactions from a single party in the communication cannot be used to uncover secret information. They use T-test to compare multiple lightweight ciphers' resistance to DPA. Then they apply TI protection and verify improvement in resistance against DPA. TI is an improvement on Boolean masking because they provide security in the presence of glitches. Power in a CMOS gate during a transition due to a glitch is a lot higher than the gate power in normal operation. Hence, measuring the toggle rate in CMOS during glitches is used to mount an attack against masked AES.

3.2.2 Integrated Voltage Regulators and Power Delivery Network signature masking

Since power side channel attacks rely upon the leaked power from the internal operation of the circuit, some techniques published in the literature have attempted to change the

signature of the leaked information to cancel its correlation with the algorithm. One way this has been done is to use an integrated voltage regulator within the IC, that behaves as scrambling or filtering for the input current of the regulator, which is the current that the malicious measures. With this, the leaked current is no longer correlated to the regulator output current, which is the circuit operating current. Another countermeasure is scrambling the power delivery network by injecting noise with either the integrated regulator or a clocking circuitry.

The authors of [48] study the concept of using an internal on-die voltage regulator to mask the IC power consumption signature, as a SCA countermeasure. They propose COnverter REshuffling (CoRe) as a way to scramble in the signature of the current of an AES algorithm. CoRe is reshuffling the converter phases. For example, if an 8-phase switched capacitor converter needs 4 phases to be on at a given current, the 4 on phases are reshuffled every 10 cycles to provide different current spikes at the IC package input.

They compared this CoRe method to prior arts that are COnverter GAting (CoGa), Low Drop Out (LDO) voltage regulator and switched capacitor voltage converter. CoGa is auto-phase shedding: the voltage regulator controller autonomously drops/increases the number of phases based on the output current. The metrics used for comparison are PTE (Power Trace Entropy) and TTE (Time Trace Entropy). The higher the PTE, the more robust the system is against SCA.

The paper also compares the PTE of the above VR schemes with or without Dynamic Voltage and Frequency Switching (DVFS). They conclude that DVFS in general

increases robustness against SCA. Specifically, PTE increases for CoRe with DVFS, but it decreases drastically for the CoGa scheme with DVFS.

PTE and TTE metrics are also used in [50] to evaluate the effectiveness of their proposed integrated voltage regulator DFVS techniques against power SCA. These authors propose three different dynamic frequency variations techniques and evaluate their effectiveness. They demonstrate that the most advanced of the three designs can block all SCA attempts while delivering 27% energy reduction with a 16% encryption time overhead for a DES algorithm.

In [49], they propose a quantitative formulation of the output parameters of a voltage regulator in the presence of Random Dynamic Voltage and Frequency Scaling (RDFVS), Random Dynamic Voltage Scaling, and Aggressive Voltage and Frequency Scaling (AVFS). These formulations are used in the simulation of switched capacitor (SC) converter input current to assess the impact of these countermeasures on the IC susceptibility to power SCA. To evaluate the effectiveness of the proposed countermeasures, they compute the correlation coefficient between the circuit input current and the converter input current. They demonstrate that when RDFVS is implemented in a circuit, the correlation coefficient is reduced by more than 80% against differential power analysis (DPA) attack and more than 92% against leakage power analysis attack. By masking the converter clock frequency, the supply voltage, and the current, from the malicious agent, they can increase the measurement-to-disclosure value by over 1 million.

The research team in [51][52][53] has published various Integrated Voltage Regulators IVR topologies as countermeasures against power SCA. In [51], they propose an IVR with an on-die all-digital controller: digital proportional Integral Defiierentiator (PID), a Discontinuous Mode (DCM) controller, and a loop randomizer. They can demonstrate that CPA attacks with 100 000 traces are not successful in recovering the encryption key of an AES-128 algorithm. They also converted the IVR input voltage in the frequency domain with an FFT and ran CPA on it, which was unsuccessful.

The study in [52] depicts another IVR, this time geared toward improving the resistance of the device against Electromagnetic side channel attacks. They propose a Random Fast Voltage Dithering (RFVD) and an All-Digital Clock Modulation (ADCM) to reduce the noise signature of an AES encryption block by up to 37x. Also, the number of Minimum Traces to Disclose (MTD) increases by a factor of 692x for CPA. Similarly, the MTD for Correlation Electromagnetic Analysis (CEMA) increased by 37x.

The research in [53] has a lot of similarities with that in [52], in that they both use an IVR control loop randomizer to change the signature of the information leaked by the circuit to the outside world. Also, similar to [52], the IVR in this paper is a buck converter topology, i.e with an inductive output stage. However, here they implemented the new countermeasure in a 130-nm test chip, with an output inductor of 11.6nH, an output capacitor of 3.2nF, and a converter base switching frequency of 125MHz. With this loop randomizer topology, they eliminate the leakage information while incurring only a 3% performance reduction and a 5% power increase compared to the IVR-AES implementation without the countermeasures. Furthermore, the number of plaintexts

required to successfully extract the secret encryption key of the device with the proposed countermeasures increases to 100 000, up from 1000 for the device with IVR and no loop randomizer.

3.3 Attacks depending on application sectors

3.3.1 Internet of things and lightweight ciphers

The pervasiveness of edge computing has given rise to security vulnerabilities in the data being processed and transferred by edge devices. Thus it becomes necessary to encrypt the data on the device. The authors of [32] have demonstrated the use of the lightweight cipher SIMON-128 for this purpose. Algorithmic implementation and bit-serial datapath, are two countermeasures against SCA demonstrated here. They implement the algorithm in a 15nm CMOS ASIC and a Spartan-6 45nm Xilinx FPGA and perform power, performance, and area (PPA) analysis. From the implementations, they conclude that a 6-round unrolled datapath provides 143x higher performance and is 80x more energy-efficient than the baseline bit-serial design. The 6-round unrolled implementation also increases the minimum traces to disclosure by 384x and does not allow a secret key recovery with a CPA attack using 500 000 traces. It's then insinuated that with more than 500 000 traces, one can successfully uncover the secret key with CPA. But the authors of [40] show that SIMON 64/96 can be made completely resilient to CPA with a round-unrolling countermeasure, albeit at the cost of 66.6 area and 13.4% performance penalties.

Another related lightweight cipher proposed in the literature is PRINCE. The authors of [37] demonstrate that a PRINCE implementation in an FPGA with a glitch canceller is

vulnerable to SCA. Unrolled architecture generates glitches (in the combinatorial circuit) that increase power consumption. Glitch cancellers are then introduced to reduce power consumption. The proposed glitch canceller for PRINCE was implemented in an FPGA on a SASEBO-G board, and the vulnerability to power SCA was evaluated with the T-test method.

3.3.2 Remote side channel attacks in cloud and edge

The research in [10] demonstrates the ability to implement SCA against an FPGA or the CPU of an FPGA+CPU SoC, remotely without physical access to the target system. They show the implementation of a Ring Oscillator (RO) based power monitor system that can reside in the same FPGA as the victim logic. With the RO power monitor, they can estimate the power consumption of a victim logic inside the same FPGA (FPGA-FPGA attack) or that of the CPU (FPGA-CPU attack). Because an FPGA is programmed by loading bitstream into the FPGA, an attacker can perform SCA remotely. She/he just needs to have permission to program it. The attacks scenarios demonstrated in the paper are:

- Side Channel Attack
- Covert-channel attack: an attacker can implement an FPGA circuit or CPU program that monitors the level of switching activities. With the capability to both monitor and control the power consumption, the attacker can create covert channels between modules of the FPGA to bypass traditional control access mechanisms to intentionally leak secrets.

- Timing attacks using the power SCA: power consumption is dependent on operations like cache hit and cache hit. The FPGA-based power monitor is used to monitor timing-based info and perform SCA.
- Program identification:
    - different operations have different power consumptions: for example, floating-point operations consume more power than integer operations;
    - cache miss vs active computations cycles: power monitor can be used to determine which program or hardware accelerator is running.
- Attack detection: if someone else programs a malicious power virus to ramp up power in an attempt to attack the system, the power monitor detects it.

Another type of circuit used in remote attacks is demonstrated in [54] and [55], where the authors use Time to Digital Converter (TDC) to measure the nano-seconds fluctuations of the voltage within the power grid of an FPGA. The digital measurement technique measures the small fluctuations in delay in a delay line containing LUTs and latches. The TDC features a carry chain with an input signal that propagates through the logic, racing against a clock that propagates through the parallel series of latches. The clock sets the latches as it ripples through if it gets there earlier than the input signal. The implementation of this TDC in a 28nm FPGA, which runs at 500x faster than the FPGA internal Analog to Digital Converter (ADC) [54], shows that the voltage has a spike of ~300mV, 10x more than the allowable part specifications.

A TDC-based remote SCA is demonstrated in [56] with the implementation in Xilinx ZynQ 7000 heterogeneous SoC. The SoC is a monolithic design featuring a Xilinx Artix

7 FPGA and an Arm Cortex-A9 CPU core. They have integrated 8 TDC-based delay line sensors with a sampling rate of 200MS/s to attack two implementations of AES encryption algorithms in software: 8-bit Tiny AES (AES layers are implemented byte-wise and computed sequentially) and 32-bit OpenSSL AES. For the attack on Tiny AES, the successful recovery of a byte of the secret key requires 111 000 traces with CPA. Between 87 000 to 130 000 traces are needed to attack the 32-bit OpenSSL AES software implementation.

For CPU/FPGA co-packaged chiplets architectures, a cryptographic die is placed as a multi-chip package with the CPU/FPGA and other dies [66]. The interconnection to other dies is via a physical bus and through the package substrate [67]. Security of the cryptographic die is a concern because a malicious agent has access to the IC balls for power measurement. Though [10] has demonstrated remote attacks against FPGA+CPU implementations, the co-packaged chiplets are vulnerable to insider SCA when an agent has physical access and can directly probe the power consumption of the cryptographic chiplets on the package.

3.4 Power delivery network modeling for side channel cryptanalysis

Several prior research has focused on modeling the Power Delivery Network (PDN) to study their susceptibility to side channel attacks. Most papers aim at evaluating an IC before fabrication and implementing appropriate countermeasures to eliminate or lessen the exposure to malicious attacks.

In [45], an equivalent circuit model of the IC power circuit is generated to evaluate the vulnerability of the systems before fabrication. So, this paper is about analyzing the PDN

of a circuit, beforehand to estimate vulnerability to power SCA. The PDN model is subjected to SCA and results are analyzed. The paper shows an equation for the IC voltage in terms of the IC current and PDN transmittance.

They are two linear equivalent circuit models developed for predicting IC current: Linear Equivalent Circuit and Current Source (LECCS) model and ICEM model. These are developed by [46] referenced in the paper. This research center has developed a circuit for evaluating encryption devices PDN against SCA. They run an AES algorithm with a 128-bit key, with 1000 plain texts, as two sets of 500 plaintexts. One 500-plaintext set gave a Hamming Distance (HD) in the tenth round of 2 and the other gave it as 124. The AES-128 encryption process was a 10-round operation with pre-operation, including the preparation of subkeys.

The CPA method was used here because the authors believe it's the most powerful method on AES. For CPA, hackers used the variation of the current magnitude to infer the changes in plaintext. Changes in the plaintext change the Hamming Distance (HD). HD is the number of registered gates that shift the states. So a large current will imply a large HD. Because the variation of HD for all possible keys is modeled in advance by the hackers, they can guess the crypto algorithm key.

The authors measure the PDN Z-parameters and S-parameters of the Device Under Test (DUT) with various lab equipment (VNA, Scopes, probes). Then they measure the voltage at the device, which shows 11 sharp peaks corresponding to the 10 rounds plus 1 key schedule operation. Using a simulator and equations, the IC current is deducted. They tested three decoupling schemes and found that adding a large equivalent series

inductance (ESL) yielded a low CPA. No decoupling and some decoupling cap yielded the same large CPA. Meaning keeping the current signature from being measured by hackers is the best countermeasure.

In the end, the paper can correctly estimate CPA utilizing circuit simulations with the equivalent circuit models they built. Other publications do this by measurement exclusively.

In [47] a fast PDN simulation method for an IC is proposed. The method is used on an IC running a cryptographic algorithm (AES) and then CPA is used as SCA to attack the IC. This research proposes a simulation technique to reduce the time it takes to explore the vulnerability of the crypto IC to SCA. The method is based on end-to-end system-level modeling that includes the chip power grid model, the substrate PDN model, and the active elements representing the Si dynamic current. When the digital operation dynamic model of the IC current consumption is added to these two elements, the resulting model is a 3-D representation of R-C elements stringed in the network.

They built an AES test chip and ran side channel cryptanalysis on it. The analysis consisted of measuring the power multiple times, running CPA, and noting the rank of the correct key amongst key guesses. Then they compare it to simulated data. They show a close correlation between only two sample measurements and the simulated results when all elements of the end-to-end system model are considered. On the other side, if only the Si dynamic circuit is modeled, then the attacker needs more than 10 waveforms to be successful.

Chapter 4

Developing new side channel power analysis formulations

A. T. Mozipo and J. M. Acken, "Power Side Channel Attack of AES FPGA Implementation with Experimental Results using Full Keys," 2021 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), 2021, pp. 1-6, doi: 10.1109/DTS52014.2021.9497976.

Authors: Aurelien. T. Mozipo, John. M. Acken

CRediT Taxonomy:

Aurelien Mozipo:

Conceptualization, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing

John Acken:

Conceptualization, Methodology, Supervision, Visualization, Writing – review & editing

doi: 10.1109/DTS52014.2021.9497976.

The success of an enumeration algorithm depends largely on the quality of the measured information. In this chapter, this research proposes a way to reduce the keyspace by ranking enumerable keys, where such ranked keys are necessary for successfully attacking a victim. First, this study shows that with CPA, one can successfully attack the first round of an AES implementation by defining a leakage function based on the full key rather than just a single byte of the key. The concept is also applied in the last round to demonstrate that it can uncover the encryption key, similar to when the key is guessed one byte at a time. Then, the Kullback-Leibler rank is defined, which aims at reducing the search space, and this, in turn, can facilitate a key enumeration technique. The two concepts introduced herein are most useful when used in a multistage attack strategy as the initial step to reduce the keyspace, paving the way for another attack to be run on the full space of the reduced key set. These next steps will be left to the follow-up of this work.

4.1 Objectives

The goals of this chapter are threefold: (i) demonstrate an attack on the 1st round of AES encryption, that is practically useful once an attacker has reduced the key search space to a computationally feasible size; (ii) demonstrate a way of reducing the key search space via key ranking, with particular applications for power side-channel attacks on the implementation of cryptographic algorithms in an FPGA; and (iii) develop a qualitative analysis to yield the optimal key search space and plaintext size. The ability to attack AES algorithms with correlation power analysis (CPA) has been demonstrated and implemented in prior works [21]. Those methods mostly take advantage of the fact that each S-Box output depends only on a single byte of the round key, and in the last round,

they have no mix-column operation. Hence, the key search space size is reduced to 4096 or 8192. In some system implementations of AES, the attacker does not have a practical means to access the ciphertext and thus cannot easily attack the last round.

4.2 Reformulation of the correlation power analysis hypothesis

This section proposes a reformulation of the correlation power analysis principle, which is a generalization of the theory presented in prior literature.

Published studies have demonstrated that correlation power analysis yields the maximum coefficient for the candidate key with a certain number of samples [58][59]. This research introduces a reformulation of the correlation power analysis problem that takes the sufficiency of the information collected about the victim system into consideration. Therefore, information and sufficiency theory is applied to generalize the correlated power analysis and extend it to key enumerations or nonexhaustive key guessing techniques. Given that correlation power analysis is purely a guess regarding the behavior of one key candidate among a random population, one cannot detect the candidate key guess with certainty unless all the possible keys have been analyzed, but this is practically impossible.

Therefore, the following lemma is deducted, which is demonstrated empirically during our simulations:

$$\lim_{\substack{n\to\infty \\ k_n\neq k^*}} \left[\max_t \left(\rho_{MH}(k_n, t)\right)\right] = \max_t (\rho_{MH}(k^*, t)) \qquad 14$$

k_n are random key guesses, $k^*$is the encryption key, $\rho_{MH}(k^*, t)$ are the corresponding correlation coefficients and *t* is the time.

In side-channel attack models with nonexhaustive key guesses, the actual target correlation factor is unknown to the attacker; hence, one can only observe a trend in their analysis to deduct (with some probability not equal to 1) that they have guessed the correct key. Therefore, our new formulation of the problem states that as the number of key guesses increases, the maximum correlation coefficient of all the guesses converges toward the correlation coefficient of the encryption key. This reformulation of correlation power analysis theory makes sense considering Fisher's criterion of sufficiency [62], which stipulates that the statistical population should summarize all of the relevant information.

4.3 Modified Kullback-Leibler Theory for Power Side-Channel Analysis

Given the background and theory developed in the previous section, this research modifies the information theory concept and applies it to power side-channel attacks on an implementation of a crypto algorithm.

Let's assume the side-channel model of Fig. 15, a cryptographic engine performing encryption on input texts **I,** yielding ciphers **C,** and involuntarily leaking power information **M** that the malicious agent can measure remotely or with a physical presence. Note that the victim here can be an entire encryption system or just a portion of it. To develop the attack model, the power model necessary to run the power analysis is first defined.

Fig. 15. Encryption model showing the variables of interest to a malicious agent.

4.3.1 Power Estimation Modeling

Let's define the attack variable as the Hamming distance (*hd*) between the input and output of the first encryption round of AES. Once the system power is measured, the adversary uses a distinguisher or power analysis method of choice to segregate the candidate key *k** from guessed keys.

$$\boldsymbol{H} = hd(\boldsymbol{C}, \boldsymbol{I}) \qquad 15$$

4.3.2 Defining Probabilities

This section introduces and defines probabilities used in the Kullback-Leibler divergence computations.

Let's define the following probabilities:

P(**H**): where **H** is the hypothesis that the estimated power has a certain value, "**H** = *h*".

$P[H = h]$ is the probability that the estimated power takes the value *h.*

P(**M**): where **M** is the hypothesis that the leaked measurement has a certain value, "**M** = *m*"

$P[M = m]$ is the probability that the measured power takes the value *m.*

As opposed to making a hypothesis directly on the key values as in [26], this study defines the probability series on the leaked measurement and the estimated power, and the Kullback-Leibler divergence is used as a distinguisher, a function that differentiates the power of key guesses versus that of the key candidate. Hence, an estimation of the

likelihood that the leaked data have the same distribution as the estimated power of the candidate key ***k**** is calculated.

Given that the space for variables **M** and **H** are different, one must define a common space to allow us to use the Kullback-Leibler theory as a distinguisher. One way to accomplish this is to define a transformation that unifies the operating spaces into a chosen one, let's say space **H**. Function *f* defined in equation (17c) below implements that transformation.

Let us define the following conditional probabilities:

$P[M = x/H, T = t]$ is the probability that the measured power takes the value *m* for a given plaintext *t* when the range of values of the estimated power H is known.

$P[H = h/T = t]$ is the probability that the estimated power takes the value *h* for a given plaintext *t.*

Finally, let's define the target functions $m_t(x)$ and $h_t(x)$ as follows:

$$m'_t(\tilde{x}) = P[M = \tilde{x}/T = t] \qquad 16a$$

$$m_t(x) = (m'_t \circ f)(x) \qquad b$$

$$F(X): \boldsymbol{M} \rightarrow \boldsymbol{H},\ f(x) = \tilde{x} = ax + b \qquad c$$

$$h_t(x) = P[H = x/T = t] \qquad d$$

The linear transformation *f(x)* maps the values in the leaked space **M** to the same level (magnitude) as the values in the estimation space **H**. The coefficients (*a*,*b*) are computed based on the maximum and minimum values in the data spaces.

4.3.3 Discriminating the power measurements from the estimations

The target functions $m_t(x)$and $h_t(x)$ are viewed as representing the conditional probability mass functions of the distributions of **M** and **H,** respectively. Therefore, using the Kullback-Leibler divergence measure defined in equation (12), the mean information required for discriminating the measurement distribution from a power estimation distribution in discrete form is written as

$$\hat{d} = \sum_T m_t(x) log\left(\frac{m_t(x)}{h_t(x)}\right) \qquad 17$$

4.4 Kullback-Leibler Rank

Exploiting the concept of mutual information in cryptoanalysis [60][85], which measures the extent to which the leaked information allows for the discrimination of different keys, this research defines a parallel, albeit straightforward, metric for evaluating a side-channel leakage model. Let us introduce the Kullback-Leibler rank (KLR), which is a combination of the smallest distinguisher [26], conditional entropy [86], and Kullback-Leibler divergence [87]. In [26], the concept of the smallest distinguisher was used to define the success rate of an attack, i.e., the probability that the correct key was ranked first by the distinguisher. The concept is defined in a more practical, less computationally intense way. The KLR estimates the likelihood that the leaked data have the same distribution as that of the estimated power of the candidate key *k**.

Define *N* as the number of keys whose Kullback-Leibler divergences are lower than that of the key candidate:

$$N = \#\{k \in \boldsymbol{K_e}, \hat{d}(k) \leq \hat{d}(k^*)\} \qquad 18$$

The Kullback-Leibler rank is thus defined as the expected number of keys with Kullback-Leibler divergences lower than that of the key candidate; for uniformity, it's normalized to the size of the key space in the trial, $\boldsymbol{K_e}$:

$$\tilde{E} = \frac{\boldsymbol{E}_k(N)}{|\boldsymbol{K_e}|} \qquad 19$$

Where $\boldsymbol{E}_k$ is the sample expected (averaged) value, over multiple trials.

Experimental results showing the outcomes of the Kullback-Leibler rank applied to AES256 attack are shown in 5.5.3, i.e., the percentage of the average number of keys with Kullback-Leibler divergences lower than that of the known key.

Chapter 5

Experiments on attacking AES and RSA Algorithms

A. T. Mozipo and J. M. Acken, "Power Side Channel Attack of AES FPGA Implementation with Experimental Results using Full Keys," 2021 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), 2021, pp. 1-6, doi: 10.1109/DTS52014.2021.9497976.

Authors: Aurelien. T. Mozipo, John. M. Acken

CRediT Taxonomy:

Aurelien Mozipo:

Conceptualization, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing

John Acken:

Conceptualization, Methodology, Supervision, Visualization, Writing – review & editing

doi: 10.1109/DTS52014.2021.9497976.

Setting up side channel attacks of cryptographic algorithm require the attack to have access to system input plaintexts or output cyphertexts in addition to the knowledge of the inner working of the cipher. This chapter describes the attack strategy and defines our power consumption modeling that yields the leak functions used to emulate the system power consumption.

The experimental application is performed on an AES256 algorithm and RSA, implemented in a Xilinx Artix 7. The platform used is the ChipWhisperer side-channel attack ecosystem with CW1170 Lite. The choice of AES256 vs AES128 was done to provide diversity in the research, and to show the extra complexity of the 256bit AES256 does not provide extra protection against side channel attacks. An AES 256-bit implementation has 14 rounds of encryption vs 10 for AES128. However, since the first and last round layers are the same in both algorithms, the attack principle applies equally to the two implementations.

5.1 Overview of AES implementation

Fig. 17 (a) shows an AES algorithm consisting of N_r rounds and the corresponding round key generation process [43]. The numbers of rounds for 128-, 192-, and 256-bit key lengths are 10, 12, and 14, respectively, per NIST standards.

When this algorithm is implemented in hardware, such as an FPGA or an ASIC, all rounds can theoretically be performed in one clock cycle, assuming that the IC has enough resources and that the clock is slow enough. However, one of the most common and realistic implementations of AES in an FPGA consists of running each round in one

clock cycle. This study uses the AES256 algorithm in [21] implemented in a Xilinx Artix 7 FPGA.

5.2 Attack Principle

Like with any attack scenario, the malicious agent has more chances when they can either read the system output information or drive known information into the input in addition to observing a piece of leaked information.

The input or output information garnered by the malicious agent, combined with the measured leaked information, i.e. power consumption in our case, can be analyzed to guess the system encryption key. This type of side channel attack, classified as passive side channel attacks [1], is the most practically implementable type of side channel attack. Active side channel attacks require the malicious agent to tamper with the target, which not only brings more complexity to the attack but most often requires physical access to the target device.

In this classification, measuring the system's leaked power is not considered tampering, as it’s merely observing the system's behavior from outside. The power leaked by our FPGA is shown in Figure 5 below. The last 10 current spikes correspond to the 10 rounds of AES18 encryption. Knowing this information, the goal here is to focus on the last round of encryption and correlate that power spike against potential guesses, with correlated power analysis (CPA) methodology or Kullback-Leibler divergence (more on these later).

The structure of the AES algorithm allows for multiple scenarios for attacking the encryption key. A malicious can either drive the inputs and measure the power or read the

output and measure the power. Deriving a set of known ciphertexts and using the hamming distance model [2] as the estimated power can lead to uncovering the original key. But this method works better for software-based implementations of AES. Given the knowledge of the output ciphertexts, the attempt to decode the last round encryption key will be based on the measured power. After successfully guessing the AES128 last round key, the key expansion is then rolled back to get the encryption key input to the system. The process of rolling back the expansion key is beyond the scope of this paper and will not be addressed here.

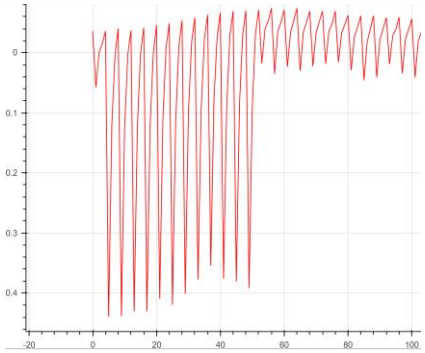


Fig. 16. Power measurement of the FPGA while running AES128 encryption

Because the assumption is made that the attacker has knowledge of the input plaintext when performing the attack, Correlation Power Analysis (CPA) can be used to decode the round 0 encryption key based on the measured power.

The Kullback-Leibler rank is also computed for AES, and it provides a way to perform key ranking and reduce the key space for an attack.

5.3 Power Consumption Modeling

The main analytical step in a power side-channel attack is processing the leaked power traces while running known plaintexts or reading known ciphertexts. As shown in Fig.

17, attacking the first round requires knowledge of the input plaintext, and attacking the last round requires knowledge of the ciphertext.

To extract the encryption key, the traces are compared against an estimated power consumption model built to emulate the device's power consumption during the portion of the encryption algorithm that one wants to attack.

5.3.1 Last Round Attack Model

Using the Hamming distance model, the power consumption of the device during the last round is modeled by the Hamming distance between the input of the round and the output, which is the cyphertext. The input of the round and the Hamming distance (*hd*) are calculated as follows:

$$ST_{rL} = inv\_Sbox\big(inv\_shiftRows(C \oplus K_{rL})\big) \qquad 20a$$

$$W_L = hd(ST_{rL}, C) \qquad b$$

ST_{rL} is the state of the input during the last round of the encryption algorithm. K_{rL} is the last round key that one wants to uncover, W_L is the estimated device power during the last round and C is the ciphertext.

5.3.2 First round AES attack model

Instead of using the last round, the leakage model is encrypted starting with the plaintext as the input and using the round zero and round one expansion keys.

Let's define the leakage function for attacking the last round as follows:

$$ST_{r2} = mix\_columns\left(shift\_rows\big(Sbox(P \oplus K_{r0})\big)\right) \oplus K_{r1}$$
$$W_1 = hd(P, ST_{r2}) \qquad 21$$

ST_{r2} is the second-round input state, and K_{r0} and K_{r1} are the round keys of rounds 0 and 1, respectively. Round 0 is the initial key addition at the beginning of round 1. W_1 is the estimated device power for rounds 0 and 1.

5.4 RSA Attack

A Kullback-Leibler side channel analysis is performed on an RSA algorithm implementation in a Xilinx Artix 7 FPGA. The algorithm implemented is based on successive modular multiplications as shown in Fig. 18. As in the AES case, both the correlation power analysis and the Kullback-Leibler Rank are used to attack this Implementation.

5.5 Experimental Results

5.5.1 Setup

The tests are carried out on the ChipWhisperer side-channel attack platform with CW1170 Lite as the capture module and CW305 Artix 7 FPGA as the target [21]. The AES algorithm of the platform is modified to implement a 256-bit version. The encryption core runs at 10MHz, with one round of AES executed per clock cycle. The sampling rate is 4x the core frequency. The attack setup is shown in Fig. 17 (b). Computations are carried out on a custom-built computer with the following characteristics: an i9-7900K processor, 128 GB of DDR4 RAM, a 512 GB NVMe SSD, a 1 TB SATA SSD, and a GPU with 8 GB of memory.

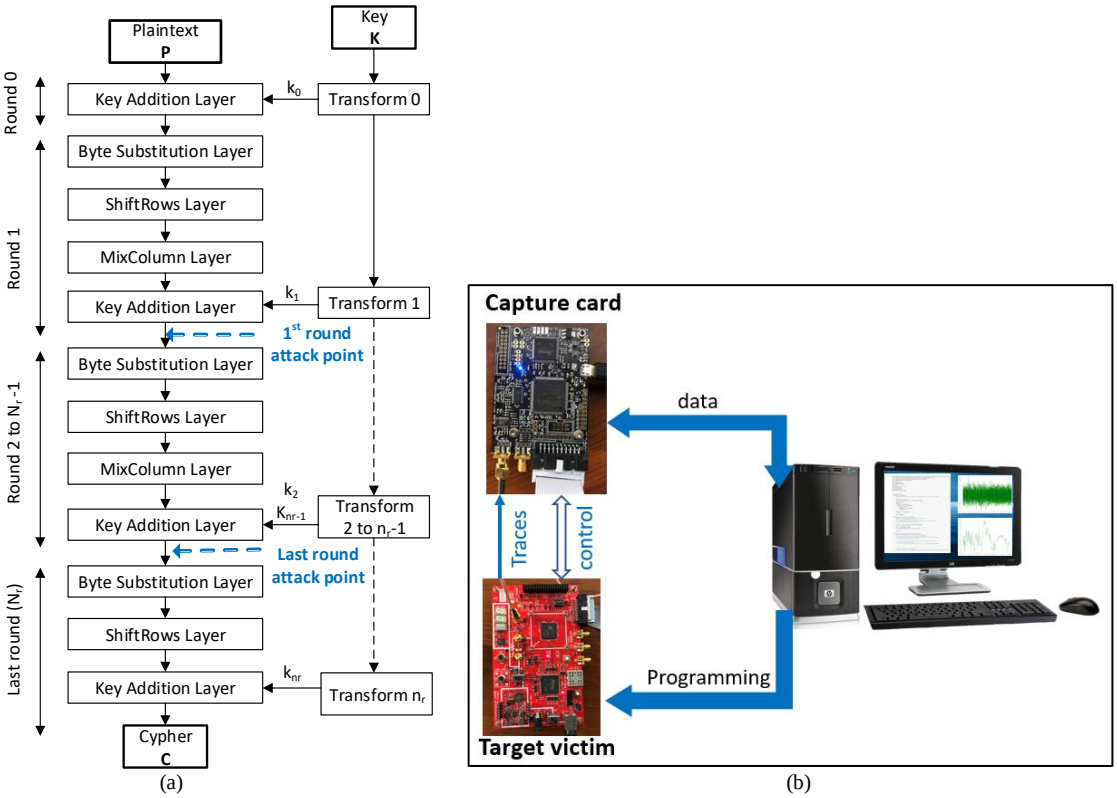


Fig. 17. (a) AES block diagram showing the side-channel attack points. (b) Side Channel attach test bench setup.

```
modular_exp(M,e,N)
{
      P1 = M*M
      X = P1 mod N
      Y(0) = X
      If (e mod 2 == 0)
            iter = e/2 – 1
      else
            iter = e/2 – 1
      for (i=1 to i=iter)
      {
            P2 = Y(i+1)*X
            Y(i) = P2 mod N
      }
      if (e mod 2 ==0)
            C = Y(i) mod N
      else
            C = Y(i)*M mod N
```

Fig. 18. Pseudo-code for the RSA algorithm

5.5.2 *Key Recovery results on AES256 with CPA*

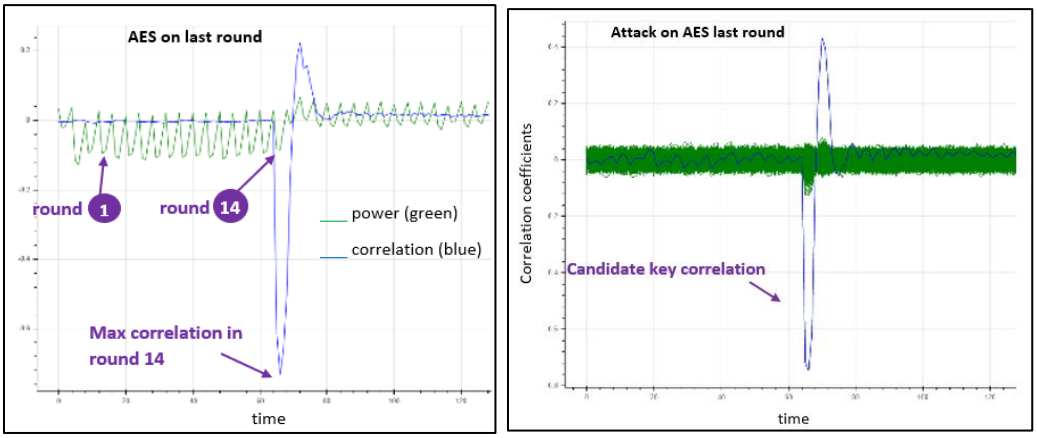
To confirm that the leakage models of Equation (27) are successful for AES first round attacks, AES256 is implemented in the FPGA, and the power consumption traces are measured while it is running. The sample traces collected for attacks on the last round and on the first round are shown in Fig. 19 and Fig. 20, respectively. The power traces display 17 power spikes (negative in the picture due to measurement polarity): the last 14 clock cycles with high power consumption (4^{th} to 17^{th} spikes) represent the AES256 rounds, and the initial three (1^{st} to 3^{rd} spikes) are related to the circuit implementation.

5.5.2.1 *First round attack results*

Our attack model, as described in Fig. 17a, attempts to guess the AES encryption in round 1 and then the round 14 key by calculating the correlation coefficients of a set of guessed keys with the known key included in the set. Fig. 20b shows one of the traces (green) overlaid on the maximum (in magnitude) correlation coefficient (blue) computed with 1000 key guesses, including the known key. For the attack on round 14, as shown in Fig. 19b, one sees that at time sample 66, corresponding to the execution of the last round, the correlation coefficient of the known key surges and exhibits a peak. The appearance of a peak indicates that the Hamming weight/distance leakage model of (9a-b) accurately differentiates between round 14 and other parts of the algorithm. This finding concurs with prior research regarding attacks in the last round with an exhaustive key space search and with the key divided into bytes [25].

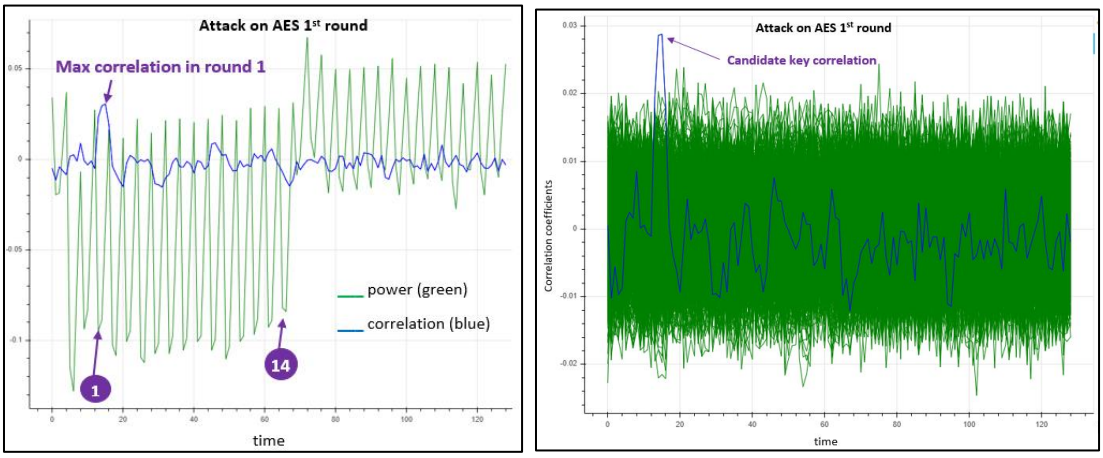
However, even with the presence of the mix-column layer in the first round (which prevents the key from being divided into bytes for an exhaustive CPA analysis), attacking

the 1st round with the full 256-bit key leads to successfully uncovering the candidate key with our defined power model, as shown in Fig. 20. The attack by the CPA distinguisher on the first round yields the max correlation coefficient for the candidate key.



(a) (b)

Fig. 19. AES attack on the last round. (a): Correlation coefficient of the candidate key overlaid on one measurement. (b): Correlation coefficients of guessed keys (green) and the candidate key (blue).



(a) (b)

Fig. 20. AES attack on the first round. (a): Correlation coefficient of the candidate key overlaid on one measurement. (b): Correlation coefficients of guessed keys (green) and the candidate key (blue).

5.5.2.2 Correlation power analysis based distinguisher

The key discrimination results of the attack on the 1st round are plotted in Fig. 21a, which shows the correlation coefficients of 1000 random key guesses plus that of the known key. The prominent blue waveform of the known key converges similarly to those of other keys (green), but it is ultimately the highest with approximately 10000 plaintexts.

The same experiment is repeated 1000 times in a loop, i.e., a total of 10^6 random 256-bit key guesses, and the same result is yielded on 100% of the passes. The convergence observed herein combined, with the work in [58][59], allows us to assert that these correlation coefficients indeed remain lower (in absolute value) than that of the candidate key. This is thus enough for us to assert that attacking the 1st round of AES with the full key still leads to information that facilitates the recovery of the secret key.

5.5.2.3 Success rate estimation: Euclidean distance fluctuation

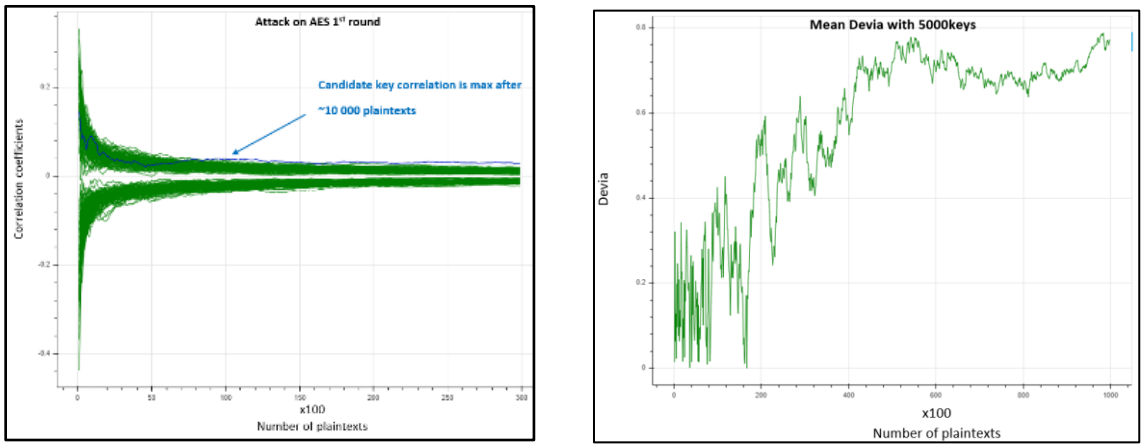
Practically speaking, the predictability of the attack model laid out for the first round is very important. So, to estimate the success rate of the attack, the Euclidian distance fluctuation Devia [25] is used, which is the representation of the distance between the maximum correlation coefficient value, the 2nd maximum value, and the average of values.

$$\rho_{devia} = 1 - \frac{E(\rho) - \rho_{submax}}{E(\rho) - \rho_{max}} \qquad 22$$

Where $E(\rho)$, ρ_{max}, ρ_{submax}, are the average, maximum, and second maximum values of the Euclidian distance of all keys, respectively.

From Equation (11), one sees that the farther the max correlation coefficient is from the second-highest, the closer the devia is to 1.

20 independent experiments with 5000 random keys each were run and the average devia of 0.788, was obtained, Fig. 21(b). This is higher than the value of 0.71 obtained in [25] for similar experiments.



(a) (b)

Fig. 21. (a) Maximum correlation coefficients versus the number of plaintexts. (b) Average Devia with 20 sets of 5000 keys (100000 keys) and 100000 plaintexts.

5.5.3 Kullback-Leibler Rank distinguisher on AES256 and RSA

As with the correlation power analysis performed above, this divergence information in the side-channel attack model aims to estimate the mean information of a guessed key and compare it to the mean information of the known key that is gathered through measurements. Here, Kullback-Leibler rank is used as a distinguisher to narrow down the key space.

First, let us generate the space unifying function *f*, which transforms the space of measured values **M** to the space of estimated values **H.** The coefficients (a,b) of the linear transformation defined in equation (17c) are computed with the two following formulas:

$$h_{max} = f(m_{max}) \; and \; h_{min} = f(m_{min}) \qquad 23$$

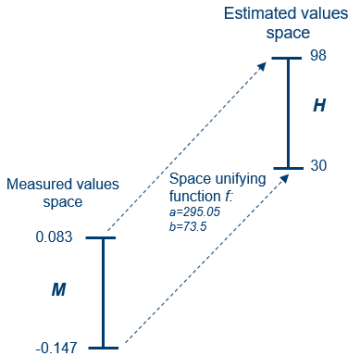


Fig. 22. Transformation function f: translates the values of observed leakage info into the space of the estimated values.

With the above transformation, the conditional probability of the measurement vector is computed by counting the occurrence of each value in the vectors and dividing by the number of traces. In other words, the probability of a measurement taking a value *m* is equal to the number of times the value *m* appears in the measurements divided by the number of traces. This is performed at each time sample, so the probability vector is time-dependent.

$$m_t(m) = P[M(t) = m] = \frac{\#\; occurences\; of\; value\; m\; in\; all\; traces, at\; time\; sample\; s}{number\; of\; traces}$$

The probabilities for the estimated power are computed similarly, except they are not time-dependent.

5.5.3.1 Attacking AES256

Fig. 23 shows the outcomes of the Kullback-Leibler rank experiments regarding the AES256 attack, i.e., the percentage of the average number of keys with Kullback-Leibler divergences lower than that of the known key. The experiment runs 20 trials for each set of keys/traces shown and averages the outcomes. The entropy decreases as the number of keys increases, but in general, it hovers around low single-digit percentages, 0.3% - 4%, and converges to a steady-state value of 1.2% for 100000 keys (20 sets of 5000). This result is an improvement over the 2% success rate achieved by [23] with their template attacks. Considering that one starts with the full key space for AES, an 88.8% reduction might still leave us with a rather larger key space for a practical full key set attack. However, this analysis is geared at facilitating another key enumeration technique or being used in distinguisher-combining attack techniques. Thus, achieving this result is quite significant for this use case.

For practical purposes, when the encryption key is unknown by the attacker, they can rank their guessed keys from the lowest Kullback-Leibler rank to the highest and select the top 1.2% of candidates.

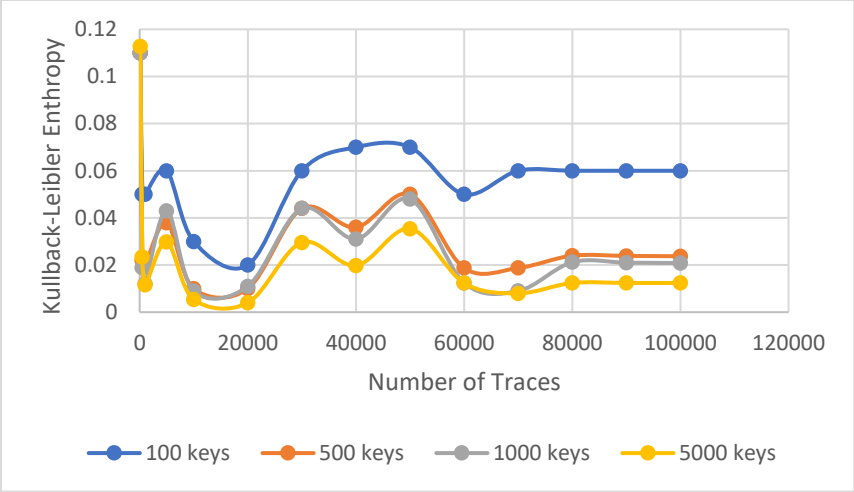


Fig. 23. Kullback-Leibler rank when attacking the last encryption round in AES256 with various key space sizes.

5.5.4 Attacking an RSA implementation

The RSA algorithm of Fig. 18 was implemented in the Xilinx FPGA on our platform. The leak function was set as the hamming distance of the input plaintext and the output cipher.

With a trace set of 100000 traces and a key space of 400000 keys (10 sets of 40000), the Kullback-Leibler rank converges toward 10^{-4}. This means 0.01% of traces have a divergence lower than the candidate key. Consider that one starts with the full key space for RSA1024, a 99.99% reduction might still leave a rather larger key space for a practical full key set attack. However, this analysis is geared at facilitating another key enumeration technique or being used in distinguisher-combining attack techniques. Thus, achieving a 99.99% reduction is quite significant for this use case.

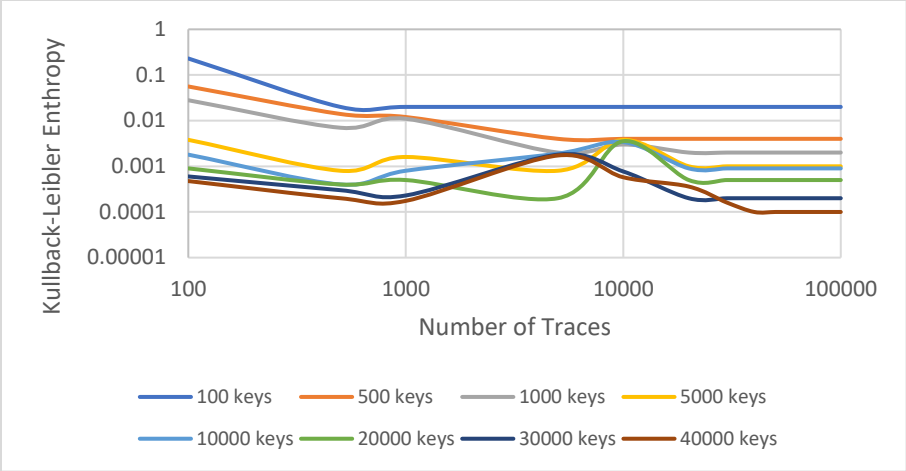


Fig. 24. Outcomes of Kullback-Leibler Rank on RSA attack for 100000 traces and various key space sizes.

5.5.5 *Comparative results*

An attack model that achieves a 73% correlation between the key candidate and the next guess key, was demonstrated. This is better than the prior art [21], and it's especially significant because of the 20-point separation with the next guess key CPA, which is 53% only. Based on the data summarized in Table 1, the CPA distinguisher is more effective on AES but KLR is the better choice for RSA attacks.

Table 1 - Comparative results of CPA and KLR distinguishers outcomes on AES and RSA attacks.

	AES256	RSA
Candidate key CPA	**73%**	3.90%
Key guesses 2nd max CPA	53%	**11%**
KLR	1.20%	**0.01%**

5.6 Conclusion

The power side channel of implementations of AES presents a serious threat to the security of devices in high-performance computing and the IoT. For applications that do not have access to ciphertexts for mounting an attack on the last round, this research has demonstrated the applicability of uncovering the full key with a CPA distinguisher using plaintexts. The Euclidian distance fluctuation (success rate) is 0.788. This research has also shown how keys can be enumerated and ranked with the new concept of Kullback-Leibler rank in applications where the key set is large and trying all possibilities of the encryption key is not practically feasible. It can then be asserted that based on the indications of our experiments, the key field can be reduced to approximately 1.2% for AES.

6 CHARACTERIZATIONS OF POWER SIDE CHANNEL ATTACKS COUNTERMEASURES

Implementations of countermeasures against power SCA on cryptographic algorithms have gained significant traction since SCA was first demonstrated in 1999 [9]. More importantly, with the adoption of AES as the main symmetric encryption standard by NIST [34], and the advancement of the area of cloud computing, power SCA methods have proliferated significantly. It has thus become necessary to develop innovative countermeasures to thwart those attacks. However, some countermeasure implementations fall short of their goal of fully protecting an implementation from power SCA because of their incompleteness or their lack of thoroughness. Moreover, though these SCA countermeasures have a long history and have been well studied, some of those implementations open the door to other types of SCA. This section presents a systematic characterization of countermeasures on implementations of cryptographic algorithms, to develop metrics that identify the residual vulnerabilities and added vulnerabilities of those countermeasures.

6.1 Characterization of countermeasure implementations

Although power SCA countermeasures are well studied, some of them have been weakened by enablers tied to technological advancements, such as cloud service (FPGA as a service), Si process node shrinkage (impacting static power consumption), or the advancement in computer processing power (used in offline processing of captured traces). Table 2 shows an in-depth analysis of various published countermeasures on implementations of symmetric encryption and hashing with key authentication algorithms.

The table shows the countermeasure implementations, targeted cipher/hash, and types of countermeasures studied, including masking, shuffling, 3-share threshold implementations (3-share TI), “dual-rail logic and delayed completion tree”, hiding, voltage noise generation, clock noise generation, on-chip voltage regulators, and power delivery network (PDN) scrambling. For each countermeasure, in the right five columns, we propose the types of attacks (or distinguishers) that remain capable of uncovering secret information if applied to the mentioned cryptographic algorithm. These distinguishers can highlight residual vulnerabilities, i.e., weakly addressed or not covered at all by the countermeasure (marked with the symbol ‘○’), and added vulnerabilities, i.e., created or amplified (marked with symbol ‘◊’) by the implementation of the countermeasures on the cryptographic algorithm.

High-order DPA: Traditional DPA (also referred to as the first-order DPA) has been proven to be ineffective on masked implementations of the AES and lightweight ciphers. However, the high-order DPA (HO DPA) has been proven to be effective in uncovering the secret key [64]. For HO DPA attacks, traces and information are captured from multiple sources (power, EM, plaintext, ciphertexts) and combined in offline analysis. Because the SCA revolves around the hypothesis that there is a dependency between secret key and leaked data, the correlation coefficients computed with a combination of multiple sources of data yield higher values than with the first-order DPA.

Collision power attacks: Masking schemes with uniformly distributed random masks are vulnerable to collision power attacks. [18] has demonstrated that collision attacks are effective on masked AES S-boxes and masked linear layers.

DPA when a glitch occurs: For some masked implementations of AES [69] [91] [92], glitches occur in the masked S-box. Glitches stem from logic gate switching and are caused by gate timing properties and interconnect delays. These glitches now provide algorithm-dependent data that lead to leakage (of masked gates) that malicious agents can exploit.

Profiled attacks: The distinguishers such as SPA, DPA, and CPA are classified as non-profiled attacks. As opposed to a non-profiled attack, a profiled attack emulates the behavior of the target victim on a similar device/environment to create a leaking template (profiling phase), then compared the correlated power traces of the victim with the template to uncover the secret key (extraction phase). The most popular profiled attacks in literature are template-based attacks [95], machine learning side channel attacks [93], and deep learning side channel attacks (DL-SCA) [96].

Template-based attacks, which are based on the Gaussian assumption (i.e., observed traces are well described by a Gaussian distribution) use the multivariate normal distribution to create a profile, which consists of the traces' specific covariance matrices and mean vectors [84][95].

Another example of profiled attack is a machine learning (ML) based attack. In ML-based attacks, an ML technique replaces the multivariate normal distribution used in template-based attacks [84] [93]. The binary classifier Support Vector Machine (SVM) can be used to first reduce the length of the power trace (feature selection), and then to learn the features of the power traces (classifier phase). SVM has been demonstrated as being effective to attack symmetric algorithms [94].

DL-SCA are very effective against not only single countermeasures such as masking, jitter, and random delay insertion but also on multiple countermeasures combined in implementation [84] [96].

T-test: Welch's T-test, which is used in the Test Vector Leakage Assessment (TVLA) methodology, determines whether two distributions are different from each other. T-test uncovers leakage of information without mounting an attack. But it does not provide info on how hard an attack is and cannot recover the secret key or other sensitive info. Therefore, a study of countermeasure implementation relying on TVLA and Welch's T-test is proven to be not too accurate [97]. It's less accurate than the standard CPA distinguisher because it's just a conformance assessment method, and the latter can recover the secret key.

Table 2 - Proposed residual and induced vulnerabilities of various types of countermeasures

Legend:
●*: Type of attack protected against by the countermeasure*
○^{n}*: residual vulnerability*
◊^{n}*: added vulnerability*
n (= 1, 2, 3…): Specific vulnerability uncovered. See

Table 3

Countermeasures	Cipher/ Hash	Countermeasure types	Types of Attacks or Distinguishers						
			HO DPA	Collision Power Attack	DPA	DPA when a glitch occurs	Profiled Attacks	T-test	Static Power Analysis
An Efficient Collision Power Attack on AES Encryption in Edge Computing [18]	AES	Masking Shuffling		○[1]	●				
Successfully Attacking Masked AES Implementations [69] [91] [92]	AES	Masking			●	◊[2]			○[11]
Hardware Architecture	AES	Masking			●	◊[2]			○[11]

Countermeasures	Cipher/ Hash	Countermeasure types	Types of Attacks or Distinguishers						
			HO DPA	Collision Power Attack	DPA	DPA when a glitch occurs	Profiled Attacks	T-test	Static Power Analysis
Implemented on FPGA for Protecting Cryptographic Keys against Side-Channel Attacks [11]									
2DDifferential Power Analysis of HMAC Based on SHA-2, and Countermeasures [73]	SHA2-HMAC	Masking	○[3]		●	◊[2]			○[11]
Power SCA countermeasures classifications [84]	AES, RSA, ECC, Various				● (SPA)		○[4]		
Multiplicative Masking for AES in Hardware [74]	AES	Masking			●	●	○[4]		○[11]
Low-Latency Hardware Masking with Application to AES [75]	AES	"Masking 3-share TI"	○[10]		●	◊[9]		●	○[11]
Static Power SCA of Sub-100 nm CMOS ASICs and the Insecurity of Masking Schemes in Low-Noise Environments [76]	AES, PRESENT, SKINNY	Masking						●	○[11]
Comparing the Cost of Protecting Selected Lightweight Block Ciphers Against Differential Power Analysis in Low-Cost FPGAs [64]	AES, SIMON, SPECK, PRESENT, LED, TWINE	3-share TI	○[6]		● ○[7]			●	
A First-Order DPA Attack Against AES in Counter mode with Unknown Initial Counter [77]	AES		○[6]		●				
Introduction to Differential Power Analysis and Related Attacks [78]	Various		○[13]		●				
Side-Channel-Attack Resistant Dual-Rail Asynchronous-	AES S-Box	Hiding			● ○[8]				

Countermeasures	Cipher/ Hash	Countermeasure types	Types of Attacks or Distinguishers						
			HO DPA	Collision Power Attack	DPA	DPA when a glitch occurs	Profiled Attacks	T-test	Static Power Analysis
Logic AES Accelerator Based on Standard Library Cells [79] **Counteracting differential power analysis: hiding from circuit cells [63]**									
Leveraging On-Chip Voltage Regulators as a Countermeasure Against Side-Channel Attack [48]	AES	On chip VR			●○[9]				
Improved Power-Side-Channel-Attack Resistance of an AES-128 Core via a Security-Aware Integrated Buck Voltage Regulator [51]	AES	On chip VR			●○[9]				
Reducing Power Side-Channel Information Leakage of AES Engines Using Fully Integrated Inductive Voltage Regulator [53]	AES	On chip VR			●○[9]				
Improved Power/EM Side-Channel Attack Resistance of 128-Bit AES Engines with Random Fast Voltage Dithering [52]	AES	On chip VR			●○[9]				
High Efficiency Power Side-Channel Attack Immunity using Noise Injection in Attenuated Signature Domain [71]	AES	On chip VR			●○[9]				
Combining Clock and Voltage Noise Countermeasures against Power Side-Channel Analysis [72]	AES	Voltage/Clock noise generation			●○[9]				

Countermeasures	Cipher/ Hash	Countermeasure types	Types of Attacks or Distinguishers						
			HO DPA	Collision Power Attack	DPA	DPA when a glitch occurs	Profiled Attacks	T-test	Static Power Analysis
2.5 Root of trust: physical separation of untrusted chiplets from trusted chiplets. [80]	Various	PDN Scrambling			●○[9]				
An Interposer-Based Root of Trust: Seize the Opportunity for Secure System-Level Integration of Untrusted Chiplets [81]	Various	PDN Scrambling			●○[9]				
Guest Editors' Introduction to the Special Issue on Hardware Security [82]	Various	PDN Scrambling			●○[9]				

Table 3 – Explanation of the proposed residual and induced vulnerabilities of countermeasures

#	Residual and added vulnerabilities to power SCA of above countermeasures
1	Masking schemes (when uniformly distributed random masks) are vulnerable to collision power attacks
2	Masked AES is still vulnerable to DPA, when glitches occur in the circuit. Masking of AES S-Boxes is still vulnerable against DPA, when glitches occur.
3	This paper also mentioned that they are protecting against 1st order DPA only. So, it's asserted that they are vulnerable to higher order DPA [73]
4	Asymmetric algorithms are more vulnerable to non-profiled attacks like SPA [84].
5	Masking is insecure in low noise environments [76]
6	All the ciphers presented here could still be vulnerable against higher order DPA [64]
7	AES protected with full-width basic iterative architecture is vulnerable to 1st order DPA [64]
8	Careless async can ruin the SCA resistance [79]
9	With integrated VR, the IC remains vulnerable to SCA by remote power measurement because power is measured at the local power grid
10	Vulnerable to higher-order DPA [75]
11	Masking schemes in some implementations are insecure against static power analysis [76]

6.2 Masking

Masking constitutes undoubtedly the most effective countermeasure against SCA under practical and rational leakage assumptions. Masking provides security against power and electromagnetic SCA under first-order DPA distinguisher. Masking attempts to mask the relationship between the encryption algorithm and its leaked power. It consists of the technique of splitting sensitive information and variable inside a cryptographic algorithm

into parts called shares so that each share analyzed on its own does contain the sensitive information. Therefore, only the combinations of all shares will contain information needed to uncover the sensitive information of interest. The sensitive variables that masking techniques are concerned with protecting are obviously the ones that manipulate the secret encryption key. The cryptographic system is thus protected against SCA at the algorithm level because masking randomizes the key-dependent computations.

Various masking schemes have been proposed in the literature, each of which recommends a unique method to protect the variables containing information about the secret key. Table 4 shows a cross-selection representative of the masking techniques available in the literature. All but one (multiplicative masking) are Boolean masking techniques implemented in hardware. Multiplicative masking has been historically implemented in software, but the authors of [74] propose a hardware implementation that protects against SCAs with DPA even in the presence of a glitch. The second column in Table 4 represents the SCA attack that the masking scheme is designed to protect against, and the third column represents the SCA techniques (distinguishers or attack types) for which the masking scheme still shows vulnerabilities. The last column shows the vulnerabilities introduced by the masking scheme, which are otherwise not weaknesses of the cryptographic algorithms without masking.

Table 4: A selection of masking countermeasures and their strengths, plus their residual and added vulnerabilities

Masking Schemes	SCA protected Against	Residual vulnerabilities	Added vulnerabilities
3-share Threshold Implementations (TI) [64][75]	• 1st order DPA • 1st order DPA even in the presence of a glitch	• Higher order DPA on AES, TWINE, SIMON, SPECK, PRESENT, LED • 1st order DPA on AES with full-width basic iterative architecture • Template based attacks	

Masking Schemes	SCA protected Against	Residual vulnerabilities	Added vulnerabilities
Fake key addition with masking [11]	• 1st order DPA: reveals a fake key	• Collision power attacks • Insecure in low noise environments • Template based attacks	• 1st order DPA in the presence of a glitch
Fake key addition without masking [11][69][76]	• 1st order DPA: reveals fake key • DPA with the difference of means distinguisher	• Higher order DPA • Template based attacks	• Higher order DPA (CPA)
Uniformly distributed random masked in linear layers [18]	• 1st order DPA	• Higher order DPA • Template based attacks	• Collision power attacks
Multiplicative masking [74]	• 1st order DPA even in the presence of a glitch • DPA with univariate and bivariate attacks	• Template attacks	
Masks on SHA2-HMAC [73]	• 1st order DPA	• Higher order DPA • Template attacks	• 1st order DPA in the presence of a glitch
Modified LUT-based Masked Dual Rail with Precharge Logic	• 1st order DPA when one glitch occurs • 1-glitch extended probing technique	• Higher order DPA • Template attacks	

An example outcome developed from Table 4 is that masking schemes when uniformly distributed random masks are used, are still vulnerable to collision power attacks. Another example outcome derived is that masked AES implementations when glitches occur are still vulnerable to Differential Power Analysis (DPA).

6.3 Hiding

In contrast to masking, which acts at the algorithm level, hiding acts at the logic level to normalize the device power consumption to make it data-independent and reduce the power SCA success rate. Thus, hiding reduces the SNR to yield a current consumption independent of the data being processed. Multiple state-of-the-art circuit styles have been proposed to protect cryptographic implementations against power SCA. This section of our research focuses on an assortment of circuit styles that have been shown to reduce the vulnerabilities of the cryptographic algorithm to power SCAs.

The logic styles studied here include the balanced precharged static logic (PCSL) with and without noise injection [63], dual-rail asynchronous logic with delayed completion tree (DRALDCT) [79], dual-rail random switching logic (DRSL) [98], single rail masked logic (SRML) [98], masked dual-rail precharged logic (MDPL) [99], wave dynamic differential logic (WDDL) [100] and NULL convention logic (NCL) [101].

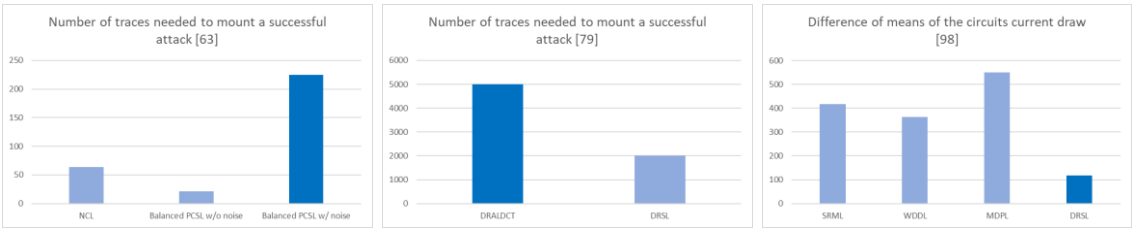
WDDL continues to be vulnerable to power SCAs because it still leaks side-channel information [98]. For masked implementations of SRML and MPDL, the output logic depends on the input data when glitches occur [91]; thus, it is susceptible to leaking secret information with the CPA and DPA distinguishers. Therefore, masked implementations are considered to have this added vulnerability, which only exists if the masks have been implemented. To lessen the disadvantage introduced by masked (and complementary) circuits, MDPL was proposed by [99], where no glitches occur. However, [98]found that the power dissipation remained predictable and dependent on input signal timing with or without glitches in the logic styles; thus, MPDL (and WDDL) remain vulnerable to power SCA with the DPA and CPA distinguishers.

Another mitigation to the power SCA by hiding is the introduction of delay variation by adding a delayed completion tree to an asynchronous dual-rail logic [79]. Although three other countermeasure features are added to their proposed AES processor, the SCA experiments merely increased the minimum traces to discover (MTD), i.e., the number of traces required to mount a successful attack, from 2000 to 5000, as illustrated in Fig. 25a. With state-of-the-art equipment currently available to malicious agents, this is hardly a deterrent, as complex experiments with more than 100k traces have been published in the

literature [102]. In addition to dual-rail logic styles, precharged logic styles hide countermeasures proposed against power SCA via the CPA and DPA. [63] proposed PCSL, a precharged logic that attempts to reduce the vulnerability to power SCA by balancing the charging and discharging paths, which reduces the data dependency of the power consumption. However, as illustrated in Fig. 25b for the SCA experiments performed on an AES implementation, the MTD is increased from 64 for the baseline NCL to 225 for the proposed PCSL with noise injection. Without the noise injection feature, the PCSL performance is far lower than that of the NCL, and the MTD is only 21 in this case.

The dual rail random switching logic style (DRSL) proposed in [98] is built to mitigate the added power SCA vulnerability (when glitches appear in the masked implementations) and synchronize the inputs to reduce the power consumption data dependency from the input signal arrival timing. However, the advantage of this new style is a 68% reduction of the current difference compared to the baselines, SRML, WDDL, and MDPL (Fig. 25c). The practical importance of this reduction, which was nonetheless not mentioned by the authors, is that the required MTD is increased by the same order of magnitude as the current reduction.

This analysis shows that in general, hiding circuits do not fully protect against power SCAs. They only increase the MTD, but they fail to increase it to a level that makes an attack impracticable to implement. Thus, a simple residual vulnerability consists of increasing the number of captured traces.



(a) (b) (c)

Fig. 25. Performance of implementation of hiding circuit styles to counter SCA. (a) – Number of traces needed to mount a SCA for circuit styles NCL, balanced PCSL w/o noise, and balanced PCSL w/ noise; (b) – Number of traces needed to mount a SCA for circuit styles DRALDCT and DRSL; (c) mean current draw for circuits SRML, WDDL, MDPL, and DRSL.

6.4 Thoughts on characterizations of countermeasures

The main problem that we solve in this research is the lack of formal methods by IC designers to evaluate the security vulnerability of their implementations of cryptographic algorithm countermeasures against power side-channel attacks. Our contribution includes a general methodology and process to evaluate the implementation of related security countermeasures, which can help guide IC designers with mechanisms to evaluate the residual vulnerabilities and added vulnerabilities (inherited from the countermeasure implementation) against power SCA. IC designers should follow the mechanisms provided herein for the masking and hiding countermeasures during the design phase to make decisions on whether the IC meets the security vulnerability, considering the IC power, performance, and area (PPA).

As concluded in the masking subsection, IC designers must evaluate masking schemes for vulnerabilities to collision power attacks. For the AES implementations of masked logic styles, the occurrence of glitches provides unintended added vulnerabilities to the DPA and CPA, since the glitches depend on the input data and can leak signatures of the

secret key. Furthermore, decisions on the effectiveness of a hiding logic style must involve pre-silicon simulations with an increased number of traces because published logic styles merely reduce the dependency of the leaked power on the secret data by making the cell power consumption less dependent on the value of its inputs. Increasing the number of traces usually uncovers the AES secret key with the CPA and DPA distinguishers. It is up to the designer to balance between the PPA impacts of the countermeasure circuitry and the level of vulnerabilities that the IC must withstand. The trade-off can be a countermeasure combining scheme that implements light versions of masking and hiding techniques to improve the SCA vulnerabilities. Indeed, the authors of [122] shared a low-cost true random noise generator countermeasure with a WDDL hiding circuit style, which yielded a trace with the SNR reduction from $5x10^{-3}$ to $5x10^{-5}$ and a corresponding 100x reduction of the mutual information metric compared to the standard CMOS synthesis. Additionally, the authors of [123] proposed the use of a WDDL-based XOR gate with false key masking to reduce the dependence between the secret key and power leaked during the SBox operation of a masked AES implementation. This gate increases the MTD to over 150 million.

6.5 Security analysis of implementations of cryptographic algorithms acceleration

Throughput improvements are implemented on symmetric ciphers with acceleration techniques such as round or loop unrolling, pipelining, and parallel datapath. Round unrolling, which has been previously demonstrated on DES, is a recommended algorithmic implementation for reducing vulnerability to CPA. For example, SIMON128 implemented with a 6-round unrolled datapath has an MTD (Minimum Traces to

Detection) 384 times the baseline bit-serial design. Also, with 500 000 traces, there's no successful CPA attack [172]. Even though MTD is higher for SIMON128-bit parallel and 6-round unrolled datapath, these algorithm implementations could still be vulnerable to CPA if the number of traces is high enough. However, high-degree of round unrolling (i.e.>6) makes CPA infeasible.

Table 5 summarizes a few relevant studies of vulnerabilities to power SCA of AES and SIMON implemented with pipelining and round unrolling architectures.

Table 5: AES/SIMON Pipelined and round unrolling SCA vulnerabilities

Algorithm	**Architectural implementation**	**Vulnerabilities**
AES	Full pipeline (outer-round, inner-round)	Vulnerable to CPA with power model based on Hamming distance intermediate register values [171]
AES	Loop unrolling	Vulnerable to CPA with a higher number of traces However, resilient to a high degree of unrolling (>6) [172]
SIMON	Round unrolling	Vulnerable to CPA with low order round unrolling (i.e. 6). Resilient to CPA with higher order unrolling, >6 [172][173][174]

7 POWER DELIVERY NETWORK BASED COUNTERMEASURES

Since power side-channel attacks target the device power consumption signature, various prior studies have focused on altering the power delivery network (PDN) to scramble the device power signature to eliminate or reduce device vulnerabilities. Various PDN-based techniques have been proposed as countermeasures. Table 6 summarizes the categories and their strengths (the SCA techniques they are protecting). A deduction of the residual countermeasure vulnerabilities is also proposed in the 3rd column. Most techniques involve the use of an IVR to scramble the current leaked to the external world. However, voltage noise injection and/or clock noise injection are also shown to be effective in reducing the vulnerability of the implementation against power SCA in local attack scenarios. The on-chip voltage regulator topologies used are multiphase, interleaved, buck converters, and multiphase switched capacitor converters. Conventional interleaved buck converters or switched capacitor converters have limited effectiveness in reducing the correlation factors used in CPAs. However, introducing random phase ordering or loop randomizing further reduces the correlation coefficients and thus renders the device less vulnerable [51][52]. However, those techniques are demonstrated only with local attacks. The side channel information is the power measured outside of the device power grid, with physical access to the victim. In the next subsections, the study attempts to show that remote attacks are still possible in the presence of PDN-based countermeasures. Remote attacks are the opposite of local attacks, as they require the attacker to be neither physically in proximity nor the vicinity of the target device. Remote

power measurement with a DPA distinguisher and with a higher number of traces is theorized to be a common weakness that these new techniques exhibit

Table 6: PDN based countermeasures strengths and residual vulnerabilities

PDN Countermeasure Schemes	Type of Power SCA analysis or distinguisher protected Against	Residual vulnerabilities
On-Chip Voltage Regulators as a Countermeasure [48]	• DPA • CPA	• SCA by remote power measurement • DPA with higher number of traces
Security-Aware Integrated Buck Voltage Regulator [51] [52]	• TVLA • CPA	• SCA by remote power measurement
Fully Integrated Inductive Voltage Regulator [53]	• TVLA • CPA	• SCA by remote power measurement
Noise Injection [71]	• DPA • CPA	• SCA by remote power measurement • CPA with increased number of traces
Clock noise and voltage noise combination [72]	• Protection against SCA when noise is injected, and Clock Randomizer (CR) is utilized. • Noise injection alone or CR alone does efficiency against CPA	• Vulnerable to CPA/DPA SCA with remote measurement if the trojan logic is adequately located

7.1 Threat models and vulnerability hypothesis

The models of the threats analyzed in this section assume the insertion of a voltage measuring logic circuit as a trojan RTL inside an FPGA or the insertion of a voltage monitoring circuit inside an ASIC by a contracting third-party house, unbeknown to the design owner. With the trojan RTL measuring the local IC PDN grid voltage, will the PDN-based countermeasures be effective in preventing remote SCA? Or will they only increase the number of plaintexts or ciphertexts necessary to successfully attack the system?

7.2 *On-chip VR as a Countermeasure*

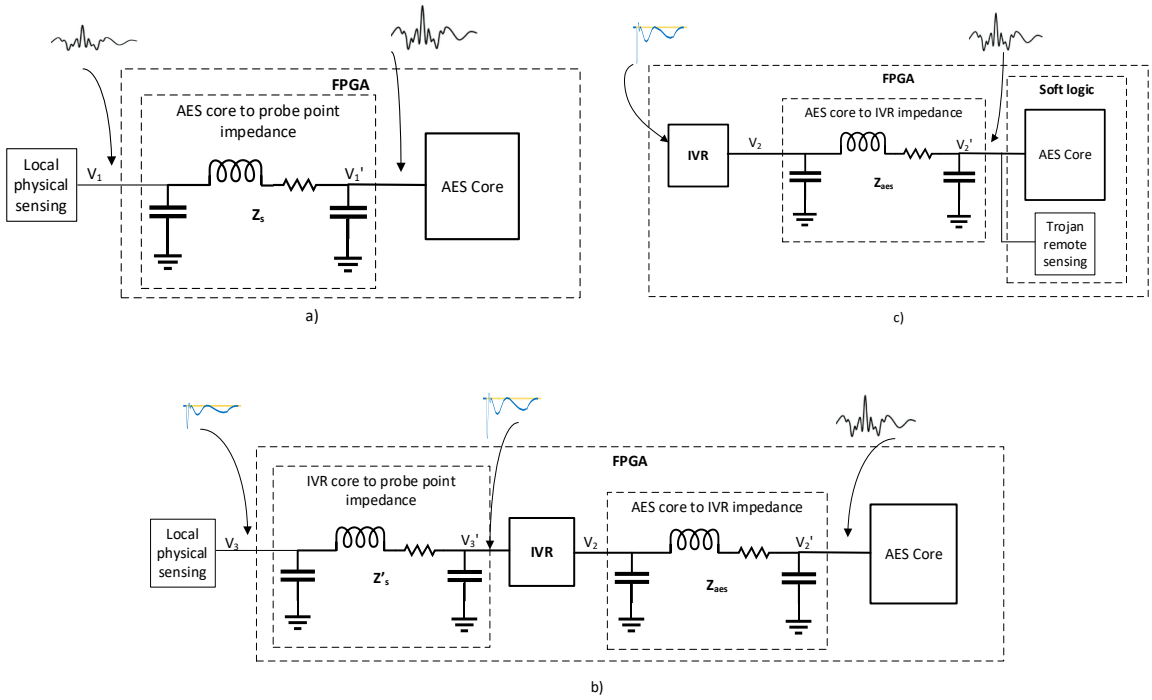


Fig. 26. Modeling of: a) - local attack without IVR; b) - local attack with IVR; c) - remote attack with a trojan IP, in the presence of IVR

The concept of using an IVR as a countermeasure works on the premise that the attacker measures the leaked power information locally by sensing the device power pins or somewhere between the power pins and the external voltage regulator. The IVR aims to scramble the device's input current to reduce or remove the correlation with the internal operation. In Fig. 26a, the voltage or current measured at the local sensing point (which is representative of the IC power consumption) is a linear transformation of the current at the AES engine and thus will show a good correlation to this internal current of the AES engine. However, with the IVR integration, Fig. 26b, the transformation is active nonlinear and, therefore, there will be a poor correlation between the internal AES engine current and the leaked current measured by the attacker [53]. However, for cases where

the encryption is implemented in FPGA softcore logic, the attacker can implement a trojan logic that measures the voltage locally at the power grid and send it to an offline processing center to run DPA or CPA [10][33]. In Fig. 26c, the voltage measured by the malicious agent remotely is tightly coupled to the cryptographic engine current and thus will exhibit a good correlation to this current.

Let us quantify the correlation impact for the various scenarios outlined in Fig. 26. Let us use the voltage at the node as a representative of the current through the node. This is a valid assumption because there is a linear relationship between both quantities, and hence, a strong correlation exists between the two.

For a local attack scenario, the leaked measured voltage and the voltage at the engine are linked as follows:

$$\begin{matrix} V_1 = G_s V_1' \\ \sigma_{V_1} = G_s \sigma_{V_1'} \end{matrix} \quad \begin{matrix} 24 \\ 26 \end{matrix}$$

$$cov(V_1, H) = E[V_1 H] - E[V_1]E[H] = E[G_s V_1' H] - E[G_s V_1']E[H] = G_s cov(V_1', H) \quad 25$$

where

G_s is the power delivery network impedance gain from the encryption engine to the local onboard measurement point,

H is the power estimation made by the attacked for CPA,

σ_{V_1} and $\sigma_{V_1'}$ are the standard deviations of the random quantities V_1 and V_1', respectively.

Similarly, the correlation coefficients are linked by the following relationships:

$$\rho_{V_1H} = \frac{cov(V_1,H)}{\sigma_{V_1}\sigma_H} = \frac{cov(V_1',H)}{\sigma_{V_1'}\sigma_H} = \rho_{V_1'H} \quad 27$$

$$V_2' = V_1' + G_{aes}V_2 \quad 28$$

$$cov(V_2',H) = cov(V_1' + G_{aes}V_2, H) = E[(V_1' + G_{aes}V_2)H] - E[V_1' + G_{aes}V_2]E[H] = E[V_1'H] - E[V_1']E[H] + G_{aes}(E[V_2H] - E[V_2]E[H]) = cov(V_1',H) + G_{aes}cov(V_2,H) \quad 29$$

$$\sigma_{V_2'}^2 = \sigma_{V_1'}^2 + |G_{aes}|^2\sigma_{V_2}^2 \quad 30$$

$$\rho_{V_2'H} = \frac{cov(V_2',H)}{\sigma_{V_2'}\sigma_H} = \frac{cov(V_1',H)+G_{aes}cov(V_2,H)}{\sigma_H\sqrt{\sigma_{V_1'}^2+|G_{aes}|^2\sigma_{V_2}^2}} \quad 31$$

ρ_{V_1H} and $\rho_{V_2'H}$ are the correlation coefficients between the locally onboard measured voltage and the estimated device power and between the remotely measured voltage and the estimated device power, respectively.

G_{aes} is the gain of the power delivery network impedance from the encryption engine to the IVR output.

σ_{V_2}, $\sigma_{V_2'}$, and σ_H are the standard deviations of the random quantities V_2, V_2' and H, respectively.

The IVR is designed to generate a voltage containing signature patterns that can scramble the encryption engine signature to protect against power SCA. It is an independent random variable and uncorrelated to the device's estimated power consumption. Hence, the correlation factor between the crypto device voltage and the estimated power consumption is written as:

$$\rho_{V_2'H} = \frac{cov(V_2',H)}{\sigma_{V_2'}\sigma_H} = \frac{cov(V_1',H)}{\sigma_H\sqrt{\sigma_{V_1'}^2+|G_{aes}|^2\sigma_{V_2}^2}} = \frac{\rho_{V_1H}}{\sqrt{1+|G_{aes}G_s|^2\left(\frac{\sigma_{V_2}}{\sigma_{V_1}}\right)^2}} \quad 32$$

7.2.1 Impact of Integrated VR as countermeasure on FPGA remote attacks

The relationship between the correlation coefficients in the remote measurement scheme in the presence of the IVR and the local measurement derived in Equation (9) above yields the following conclusions:

- Introducing the IVR reduces the magnitude of the correlation coefficients in the remote attack scenario and thus reduces the probability of uncovering the secret key.
- It shows how the new correlation factor can reject even the right key candidate. For example, if the IVR noise level is 1000x higher than the noise level at the local measurement point, the correlation factors are 100 to 1,000 times smaller. Hence, the number of plaintexts/ciphers required to successfully attack the implementation is significantly increased.
- The impedance of the power delivery network between the IVR and the remote sense location (i.e., the physical location of the encryption device) impacts the correlation coefficients and the probability of uncovering the secret key. As the gain G_{aes} approaches 1, i.e., the impedance Z_{aes} approaches zero, and the correlation coefficients increase, signaling the increased effectiveness of the IVR in scrambling the voltage at the output of the crypto engine and thus reducing the probability of recovering the secret key.

Let us assume the following notations:

- The IVR random voltage source standard deviation normalized to the local sense voltage standard deviation, $\sigma = \frac{\sigma_{V_2}}{\sigma_{V_1}}$,
- The product of the gain of the impedance networks is called:

$$G = G_{aes}G_s \qquad 33$$

- The ratio of the correlation coefficients is denoted$\rho = \frac{\rho_{V_2'H}}{\rho_{V_1H}}$

Equation (38) above can be written as:

$$\rho = \frac{1}{\sqrt{1+|G|^2\sigma^2}} \qquad 34$$

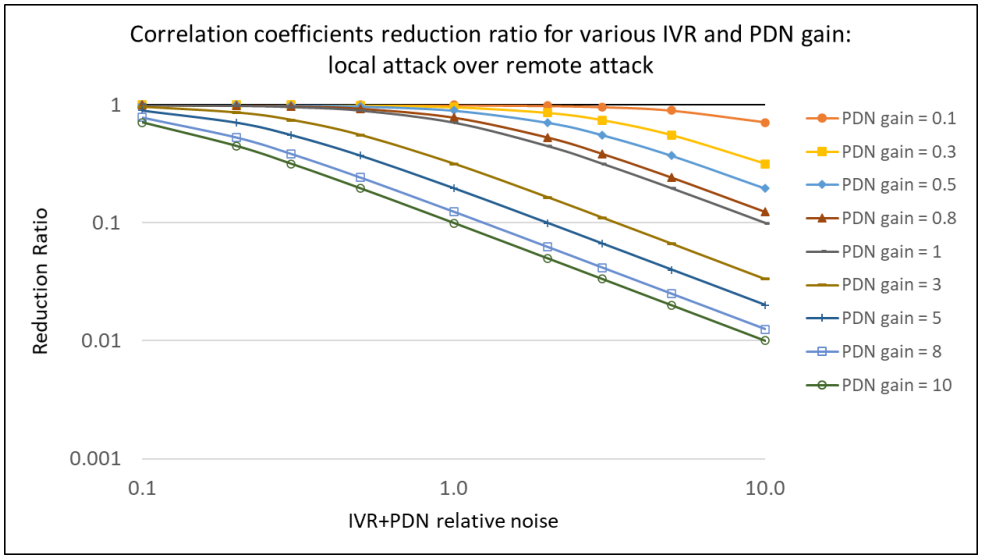


Fig. 27. Correlation factors reduction ratio as a function of the IVR (or other noise sources) relative noise, for various PDN impedance attenuations.

Let us illustrate the impact of the IVR as a countermeasure with Fig. 27. The figure shows the plot of the IVR normalized correlation factor ρ vs. the IVR voltage standard deviation, normalized to that of the voltage noise of a local attack, parametrized by the

product of network impedance attenuations. This shows that the IVR can reject the correlation factors used by the attacker in a remote scheme by attenuating them. The higher the IVR voltage spread (standard deviation) is, the higher the rejection. However, lower PDN attenuation renders IVR integration ineffective. As seen on the chart, the higher the gain G is, the lower the correlation coefficients. As an example, for an IVR relative noise level of 10 (10x higher than the voltage at the sense point of a local attack scheme), a reduction in impedance network attenuation from 0.9 to 0.1 (9x) results in an increase in the correlation coefficients from 0.11 to 0.7 (6.4x). Therefore, the use of IVR as a countermeasure is not an effective method, as the device PDN network may still allow the correlation coefficients to yield the secret key in a cryptanalysis case.

However, even in the presence of high attenuation, the IVR correlation coefficients, although reduced, may only increase the number of plaintexts or ciphertexts necessary to attack the system. Therefore, the effectiveness of the countermeasure hinges on the ability of the attacker to successfully mount an attack with an increased quantity of captured data.

7.2.2 Comparison of correlation power attack with prior art

[53]has demonstrated that with the integration of an IVR as a countermeasure, the reduction in the correlation factors is between 5x and 30x, resulting in increases in the minimum traces to discover (MTD) from ~5,000 to more than 500,000. Let us reiterate that the attack in their analysis is a local attack, i.e., the traces are now measured at the input of the IVR after it has scrambled the AES block signature.

Table 7 - Comparison of correlation factor reduction between an IVR implementation on a local attack, to potential IVR implementation with remote attack

	IVR and local attack [53]	**IVR with Loop Randomizer and local attack [53]**	**IVR with remote attack** ***IVR rel. noise: 10 PDN att: 0.1***	**IVR with remote attack** ***IVR rel. noise: 10 PDN att.: 0.9***	**IVR with remote attack** ***IVR rel. noise: 100 PDN att.: 0.1***	**IVR with remote attack** ***IVR rel. noise: 100 PDN att.: 0.9***
Correlation coefficients reduction ratio	1/5	1/30	1/1.4	1/9.1	1/10.1	1/90.1

Table 7 summarizes the correlation factor reduction obtained in the prior art by [53] with IVR used as a countermeasure against cryptanalysis of the implementation of the AES128 algorithm. In addition to a standard IVR, they also introduce the concept of a loop randomizer to randomize all transformations through the IVR. Thus, the IVR input current signature has an increased noise level, as seen by the local attacker, because there is no constant relationship between the captured measurements. The standard IVR produces a correlation factor reduction of 5x, whereas the introduction of a loop randomizer improves the reduction to 30x. However, as in our analysis, a remote attack on an IVR implementation has a reduction of ~1.4x - 9.1x when the relative IVR noise is 10 and 10.1x – 90.1x when the relative IVR noise is 100. A relative IVR noise of 100 amounts to an IVR feature with an efficiency higher than the loop randomizer. To our knowledge, such a feature does not yet exist in the current literature.

In conclusion, an IVR with a remote attack only results in increasing the minimum trace to detection but still leaves it vulnerable to power side-channel attacks with correlation power analysis (CPA). The use of an IVR as a countermeasure is effective for a local attack, as shown in [53], but remote attacks can still be very effective and hence break the IVR countermeasure.

7.3 *Voltage noise and clock noise generation*

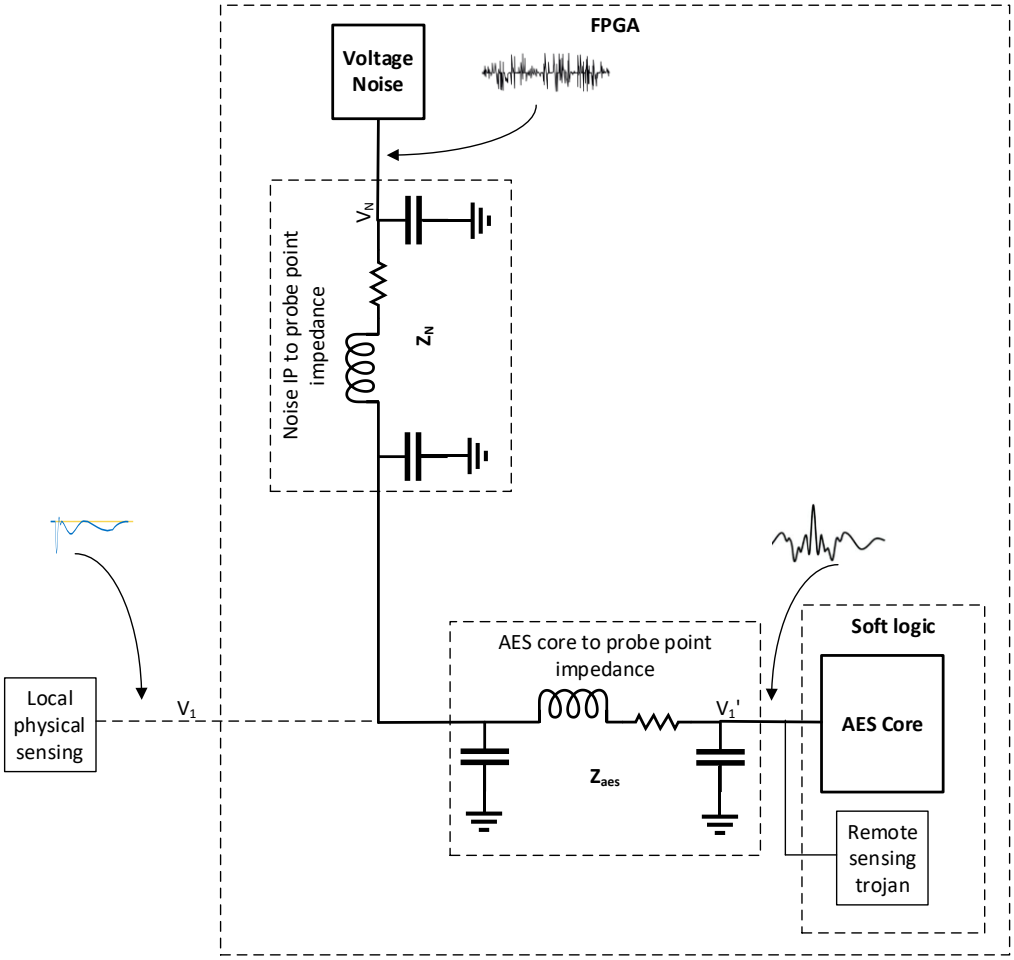


Fig. 28. Model of a noise injection as a power SCA countermeasure

Adding noise to a system to counter power side-channel attacks can be modeled as shown in Fig. 28. If v_n denotes the noise injected, Z_n denotes the impedance of the subcircuit from the noise injection point to the local measurement point, and Z_{aes} denotes the impedance of the subcircuit from the AES core location to the local measurement point, then the voltages are related by the equation below:

$$V_1' = \frac{Z_{aes}+Z_n}{Z_n} V_1 - \frac{Z_{aes}}{Z_n} v_n \qquad 35$$

Assuming that the estimated power and the injected noise are uncorrelated, the correlation coefficients of the remotely measured voltage and the estimated power are:

$$\rho_{V_1'H} = \frac{cov(V_1',H)}{\sigma_{V_1'}\sigma_H} = \frac{cov\left(\frac{Z_{aes}+Z_n}{Z_n}V_1,H\right)-cov\left(\frac{Z_{aes}}{Z_n}v_n,H\right)}{\sigma_H\sqrt{\sigma_{V_1}^2+\sigma_{v_n}^2}} = \frac{\rho_{V_1H}}{\sqrt{1+G'^2\left(\frac{\sigma_{v_n}}{\sigma_{V_1}}\right)^2}} \quad 36$$

Denoting ρ as the correlation coefficient reduction ratio between the local board-level measurement and the remote silicon-level measurement:

$$\rho = \frac{\rho_{V_1'H}}{\rho_{V_1H}} = \frac{1}{\sqrt{1+G'^2\sigma^2}} \quad 37$$

where $\sigma = \frac{\sigma_{v_n}}{\sigma_{V_1}}$ is the relative noise standard deviation, i.e., normalized to the locally measured voltage standard deviation,

$$\text{and } G' = \left|\frac{Z_{aes}}{Z_{aes}+Z_n}\right| \quad 38$$

This result is similar to that of the IVR integration presented above when the gain of the impedance is equal to G, (Fig. 27) and *G=G'*.

Table 8 - Comparison of correlation factor reduction between noise injection and local attack against noise injection with remote attack

	Noise Addition Only, local attack [72]			**Noise Addition w/ Attenuated Signature, local attack [72]**			**Noise addition and remote attack, G'=0.5**				**Noise addition and remote attack, G'=1**			
Relative noise power	0.317	2.38	3.7	0.006	0.013	0.053	0.1	1	2.38	3.7	0.1	1	2.38	3.7
Reduction ratio	1/4.5	1/22.5	<1/36	1/7.5	1/12	<1/45	1	1/1.1	1/1.6	1/2.1	1/1	1/1.4	1/2.6	1/3.8

An observation can be made by analyzing the comparative data in Table 8: for a sample noise level of 2.38, a local attack scenario achieves a reduction ratio of 1/22.5 with noise injection only [72], and a remote attack scenario, assuming G'= 0.5 (equal impedance between the AES path and the noise injection path), achieves a correlation coefficient

reduction of 1/1.6. Stretching the impedance gain ratio to 1 improves this reduction to 1/2.6. In the next chapter, the practical experiments of this research focus on analyzing the impact of such a reduction on the minimum number of traces required to discover the secret key in a remote attack scenario.

7.4 Effect of on-package decoupling capacitors as side channel attack resistance

This section studies the impact that the integration of on-package decoupling capacitors (OPDs) has on the success of power SCAs for local attacks (onboard trace capture) and remote attacks (on-die trace capture). OPDs are incorporated into system designs to reduce the voltage droop from high-frequency switching activities.

Fig. 29 illustrates OPDs on a package soldered on a motherboard with an onboard voltage regulator. The transient response to a step load is illustrated for measurements made at the power grid and the board voltage regulator decoupling capacitors. The transient response of the PDN is divided into four parts. The 1st droop is the initial response to the current step provided by the on-die decoupling capacitors and sometimes the OPDs because of their low impedance path to the die. The frequency range is typically in the 10 s or 100 s of MHz. The 2nd droop is the response provided by the OPDs after the charge from the on-die caps has been depleted. The 2nd droop frequency is in the single digit MHz range. Similarly, once the OPD charge is depleted, the onboard regulator capacitors kick in, which creates the 3rd droop [142]. Here, the frequency ranges from KHz to single digit MHz. The high-frequency noise riding on the average waveform is from the logic switching activities. Its frequency ranges from 100 s MHz to multiple GHz.

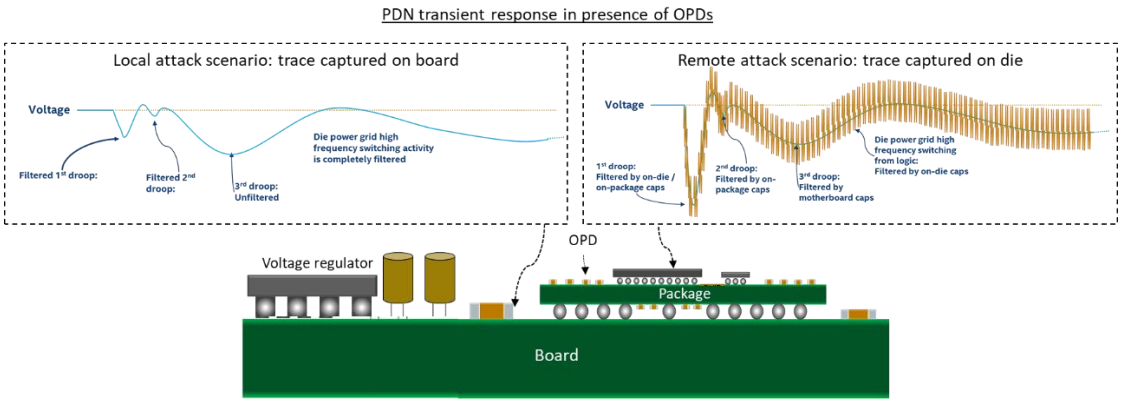


Fig. 29. On-board and on-die traces of system PDN transient response.

The system model of on-package decoupling capacitors integration is similar to that of Fig. 28, with the noise source removed and the impedance Z_{aes} replaced with the capacitor equivalent circuit of Fig. 30. The correlation coefficients in a remote attack scenario are defined by:

$$\rho_{V_1 H} = \frac{cov(V_1,H)}{\sigma_{V_1}\sigma_H} = \frac{cov\left(\frac{V_1'}{G_{OPD}},H\right)}{\frac{\sigma_{V_1'}}{G_{OPD}}\sigma_H} = \rho_{V_1' H} \tag{39}$$

With:

$$V_1' = G_{OPD} V_1$$

and G_{OPD} is the gain of the OPD circuit in Fig. 30.

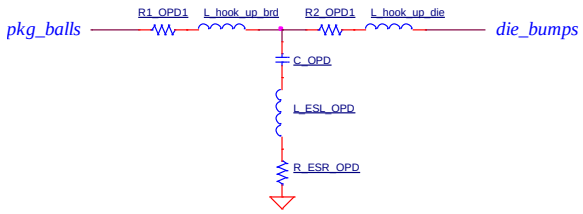


Fig. 30. Equivalent circuit modeling of the OPD with hook up impedances.

It is derived from the analysis that the linear effect of decoupling filtering has no impact on the correlation coefficients and thus the power side-channel resistance. This is because filtered versions of the 1st and 2nd droop are propagated to the board level and are thus captured by the malicious agent in a local attack scenario. Furthermore, the 3rd droop signal that is seen at the board level by the malicious agent has a magnitude independent of the OPD scheme. Thus, the 3rd droop magnitude is the main carrier of sensitive distinguishing information in a local attack scenario.

7.5 Experiments

The experiments to ascertain the impact of a remote attack on the correlation coefficients are carried out in three steps: (i) generating the current profile of an AES algorithm; this is performed by measuring the current of an Artix 7 FPGA while running an AES256 algorithm implemented with the ChipWhisperer side channel attack environment; (ii) applying the current profile to a generic FPGA platform SPICE model, then running simulations with target victim and malicious agent models attached to the FPGA die; (iii) CPA run, then computing the correlation coefficients for local attacks and remote attack scenarios.

7.5.1 FPGA remote attack modeling framework

In a remote attack scenario, a malicious agent remotely uploads its trojan program into the FPGA to attempt to monitor the IC power grid voltage. The ability of the on-die silicon power grid to act as a filter for a high-frequency signal crossing over from the victim's location to the attacker’s trojan logic location determines how successful the malicious agent will be in guessing the victim's secret information.

Fig. 31 highlights the silicon layout of an FPGA used in datacenter cloud applications, modeled based on the FPGA architecture shared in [141]. The FPGA is divided into its core, a 2x5 sector array, two transceivers, and two IO and embedded external memory interfaces (EMIFs). The malicious agent logic and the victim logic are physically placed as far away as possible around sectors 10 and 1, respectively.

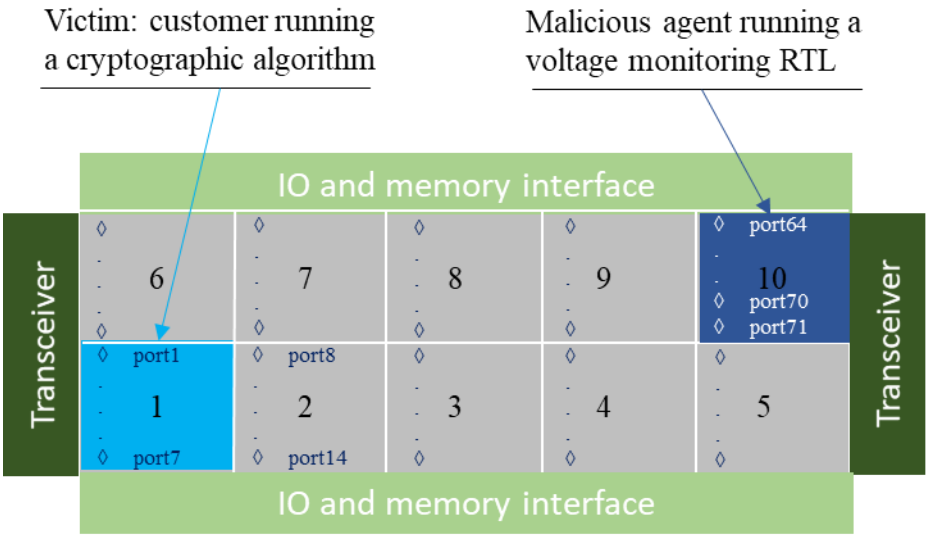


Fig. 31. Representative floorplan of the FPGA partition with two independent applications

7.5.2 Power delivery network modeling

The device PDN modeling consists of three parts: the die+metal-insulator-metal (MiM) capacitors, the package substrate, and the voltage regulator. The die, on-chip MiM, and package substrate are extracted as a distributed model with 71 ports each. The equivalent circuit model of the FPGA on-chip MiM is represented by the simplified RC model, derived from the equivalent model of [140] but with the parasitic elements (the series inductance L_s and the oxide capacitance C_{ox}) neglected. For each distributed port x (x=1,2,…,71), the MiM capacitor is thus represented with $R_{mimx}/C_{mimx,}$ as shown in Fig.

32. The vertical contact to other layers is represented by the resistance R_{vert}. Similar to the MiM, the die is extracted as an RC model, as shown in the figure. Table 9 summarizes the values of the components, which are also included in the SPICE models.

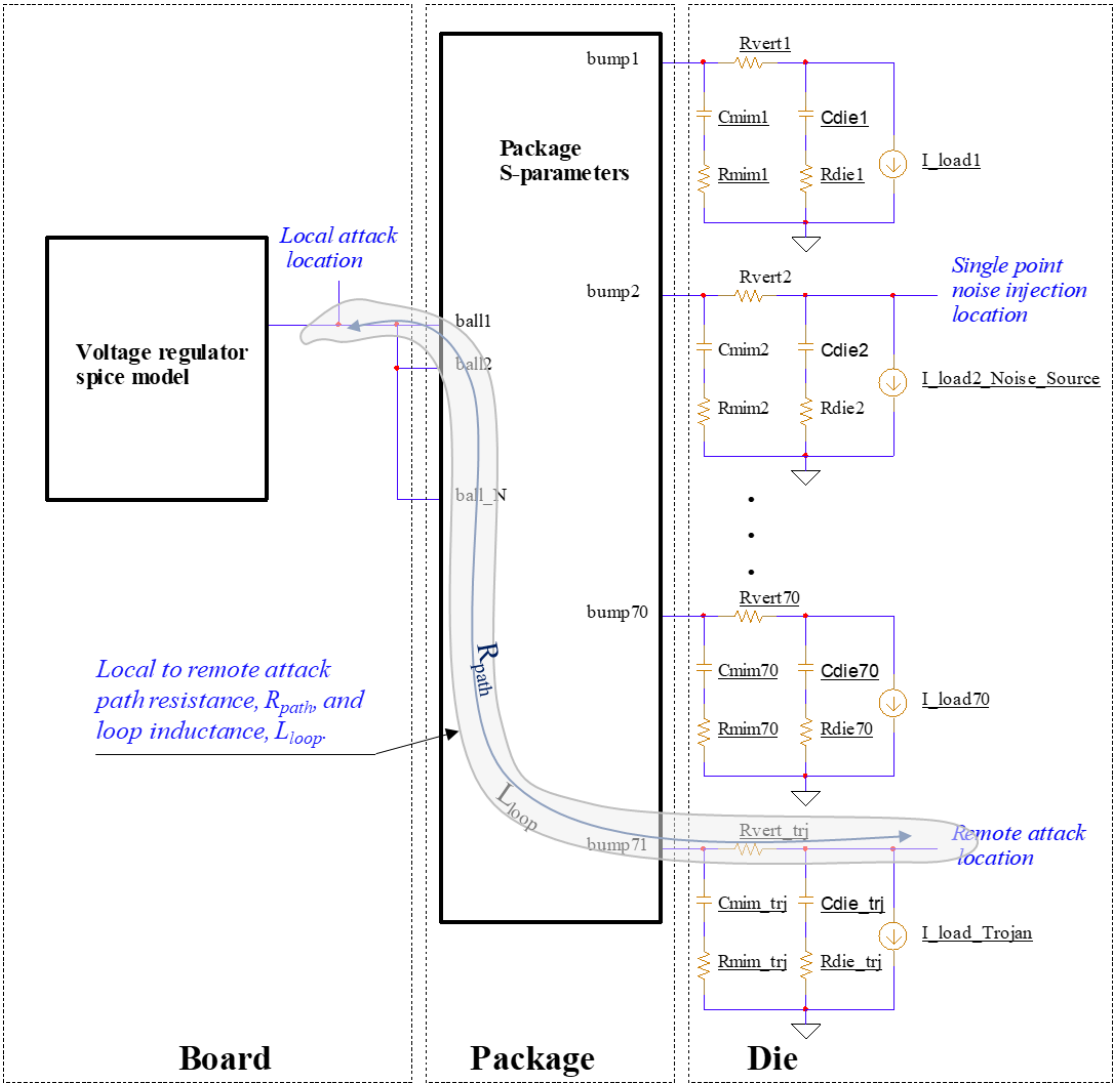


Fig. 32. Distributed PDN modeling of local and remote side channel attacks

Table 9 - Die and MiM caps SPICE model parameters

	C_{min}	R_{min}	C_{die}	R_{die}	R_{vert}

Value	5.2nF	142mΩ	35.3nF	3.6mΩ	0.003mΩ

Each sector of the FPGA and the corresponding power grid is distributed into seven ports. The traces are probed in the SPICE model at the load locations to illustrate the attacker's remote sensing of the victim's actual voltage. For the local attack scenario, the voltage is measured on the PCB, which corresponds to where the adversary measures the voltage when they have physical access to the device.

7.5.3 Simulation setup

In practical applications, the malicious agent implements trojan logic such as a ring oscillator or a time-to-digital converter in the vicinity of the victim's logic to measure the voltage that serves as a trace for the differential power analysis. However, the effectiveness of such a circuit depends on the algorithm topology and the accuracy of the instrumentation portion of the circuit. We removed this complexity from the scope of this research and instead measured the voltages directly at the FPGA power grid and package balls in the SPICE simulations.

As shown in Fig. 33, the side channel attack (AES current capture on Artix 7 and correlation coefficient computations) on the FPGA is carried out in the ChipWhisperer environment [21]. The environment provides certain APIs for random plaintexts and random key generation. The traces captured during the AES256 core encryption are passed to the Hspice simulator via text files. The Hspice simulator is embedded within the time domain traces capture subblock and invoked within the Python notebook framework. The simulator is invoked in a loop for each trace captured. The outputs of the

simulator are the traces captured at various locations: at the die bumps closest to the malicious agent trojan logic (remote attack scenario) and at the board level (local attack scenario). The correlation power analysis (CPA) and the computation of the correlation coefficients are carried out according to methods and principles developed in [59][102]. The attack on the AES256 algorithm is performed in the last round using the side channel attack leak functions and the corresponding Hamming distance shared in [102].

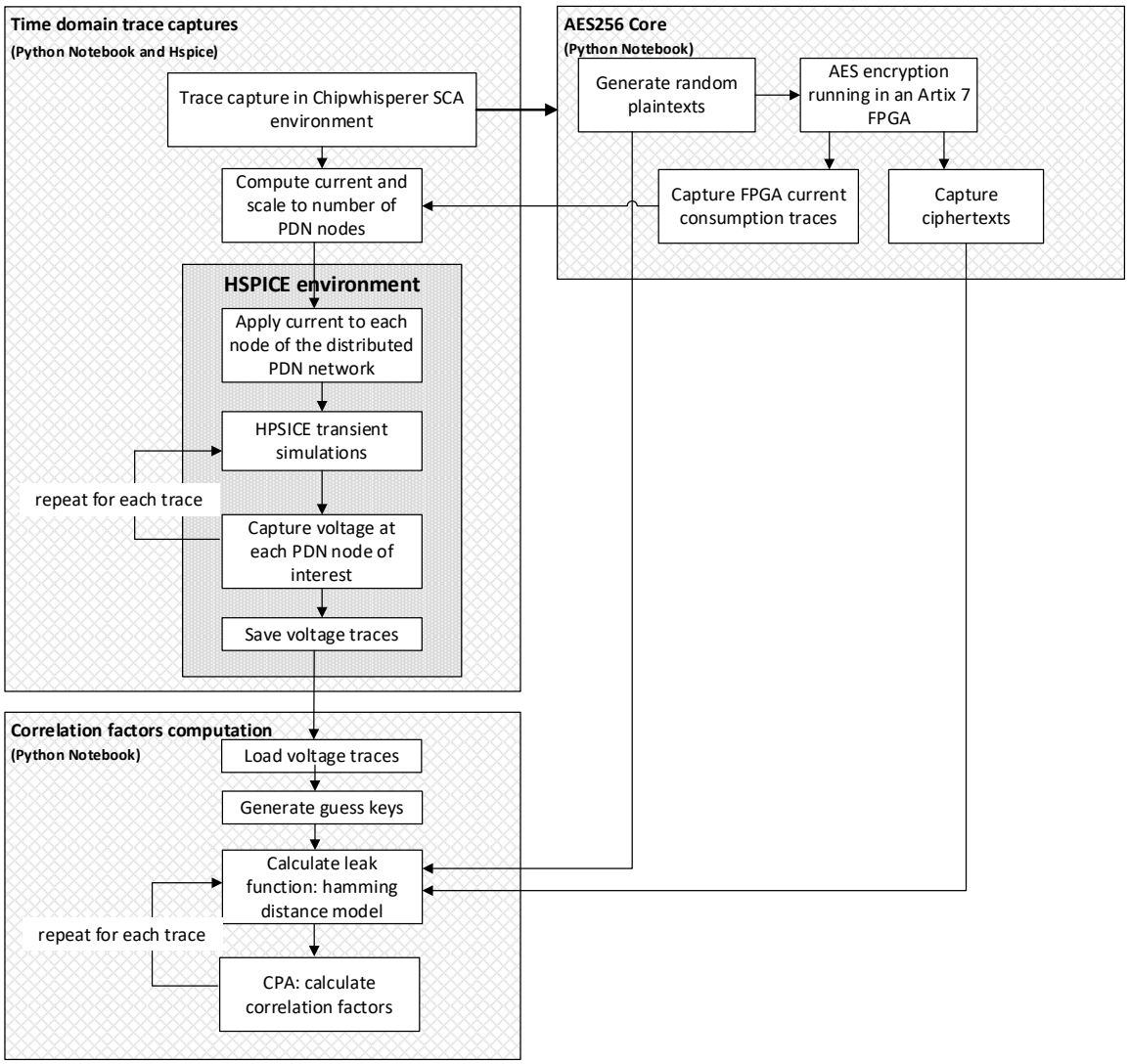
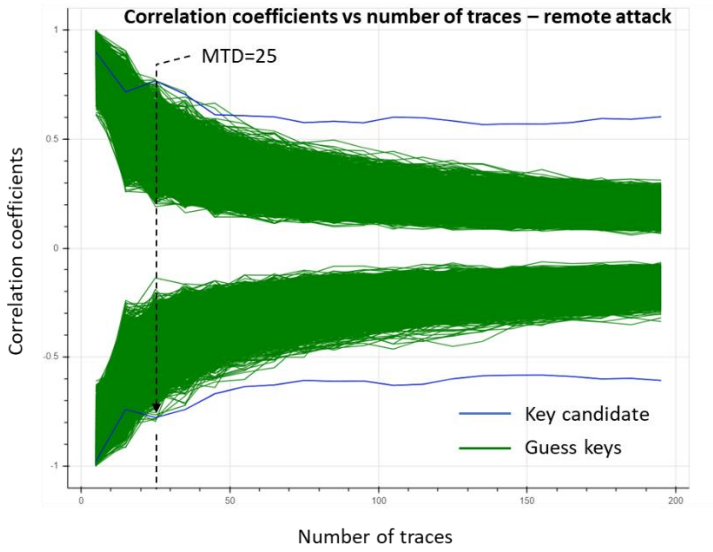


Fig. 33. Remote power side channel attack experiment framework

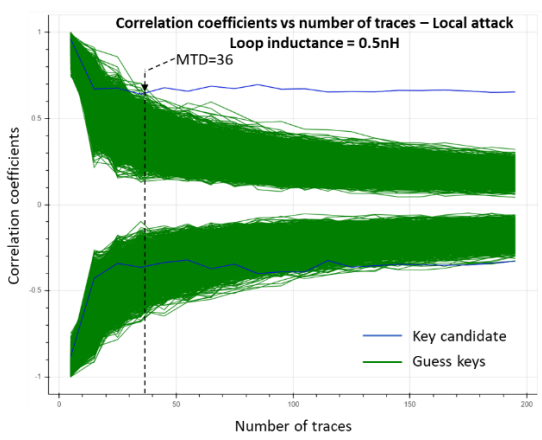
7.5.4 Local vs remote attack results

The nature of FPGAs provides malicious agents opportunities to remotely configure or reconfigure a portion of the fabric with a trojan IP that serves as a telemetry agent, monitoring the IC power grid voltage fluctuations in its vicinity.

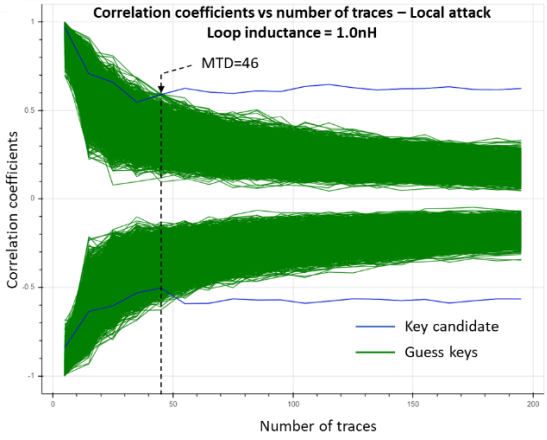
The path between the power grid and the physical onboard attack point is characterized by the package's physical dimensions and the substrate stack-up. These physical characteristics present a loop inductance between the Si power grid and the onboard measurement location. In addition, the package substrate stack-up composition, such as the number of CU layers and the CU layer thicknesses, defines the path resistance. The impedance parameters of various package sizes and stack-up compositions were extracted, and each of them was characterized by loop inductance and path resistance. For the same resistance packages (R_{path} = 0.5 mΩ), a remotely carried attack requires only 25 traces to discover the secret encryption key, whereas 36, 46, 38, 45, and 82 traces are required for loop inductances L_{loop} of 0.5 nH, 1.0 nH, 1.5 nH, 2.0 nH, and 2.5 nH, respectively. However, the package resistance has little effect on the MTD, as shown in Fig. 34.



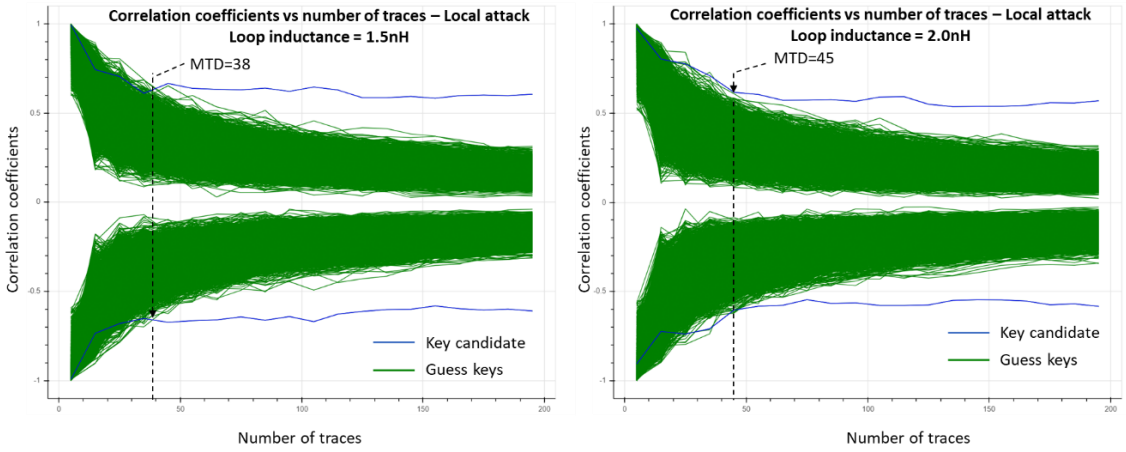
a) Remote attack



b)

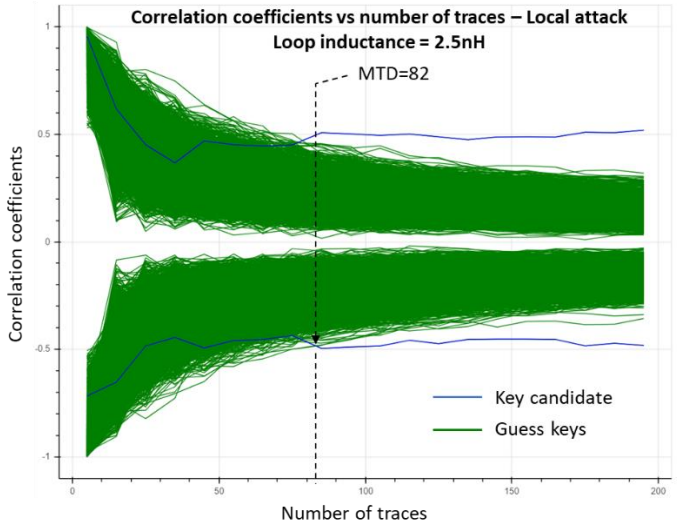


c)



d)

e)



f)

Fig. 34. Impact of package inductance on local attack success. Constant resistance R_{path} = 0.5m : a) – Remote attack; Local attack with various package loop inductances: b) – L_{loop} = 0.5n; c) – L_{loop} = 1.0n; d) – L_{loop} = 1.5n; e) – L_{loop} = 2.0n; f) – L_{loop} = 2.5n.

The experiment carried out reveals that at constant loop inductance, package resistance does not impact the MTD, but the MTD increases with the inductance (irrespective of resistance), as shown in Fig. 35. Hence, with larger packages (higher loop inductance), it takes more captures to uncover the secret key, as evidenced by the surface tilted upward

on the inductance axis. In summary, the extra PDN impedance between the IC power grid and the external local attack measurement point acts as a countermeasure against local power SCA. Thus, remote attacks are more effective than local attacks, assuming that the attacker can maximize the trojan IP telemetry accuracy.

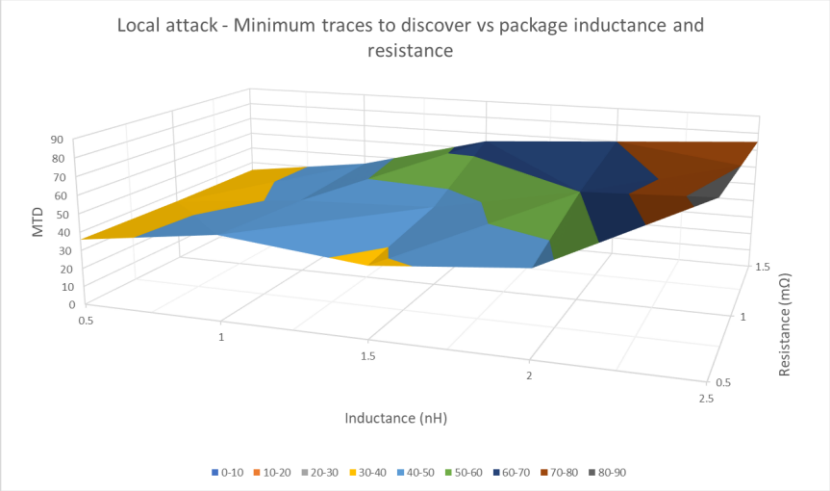


Fig. 35. MTD vs package impedance in a local attack.

7.5.5 *Impact of PDN noise injection on power side channel attack success, with remote attacks*

For power SCA experiments, the signal-to-noise ratio (SNR) is introduced as [32][108]:

$$SNR = \frac{\sigma_{trace}}{\sigma_{noise}} \quad (40)$$

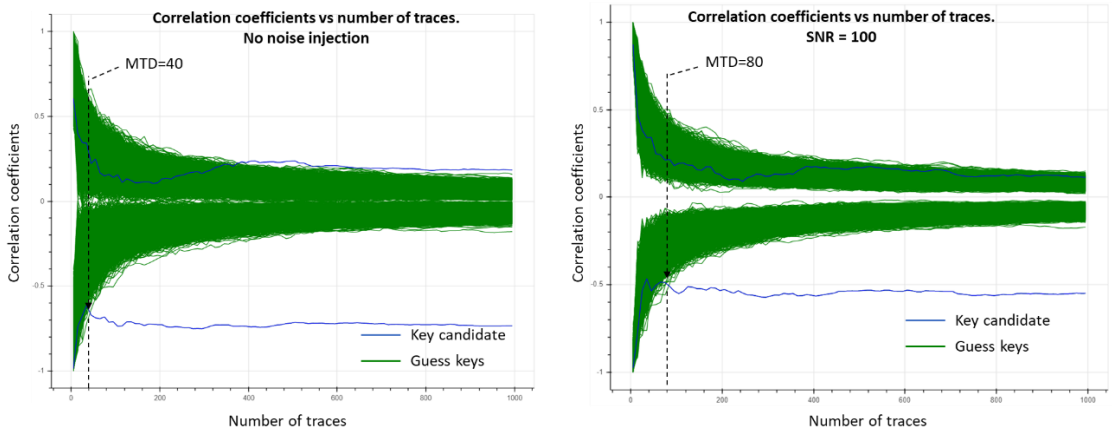
where σ_{trace} and σ_{noise} represent the standard deviation of the IC power consumption and the injected noise, respectively.

Noise is injected into the system at the injection point shown in Fig. 32. In practical applications, voltage traces are a collection of signals from various IPs running

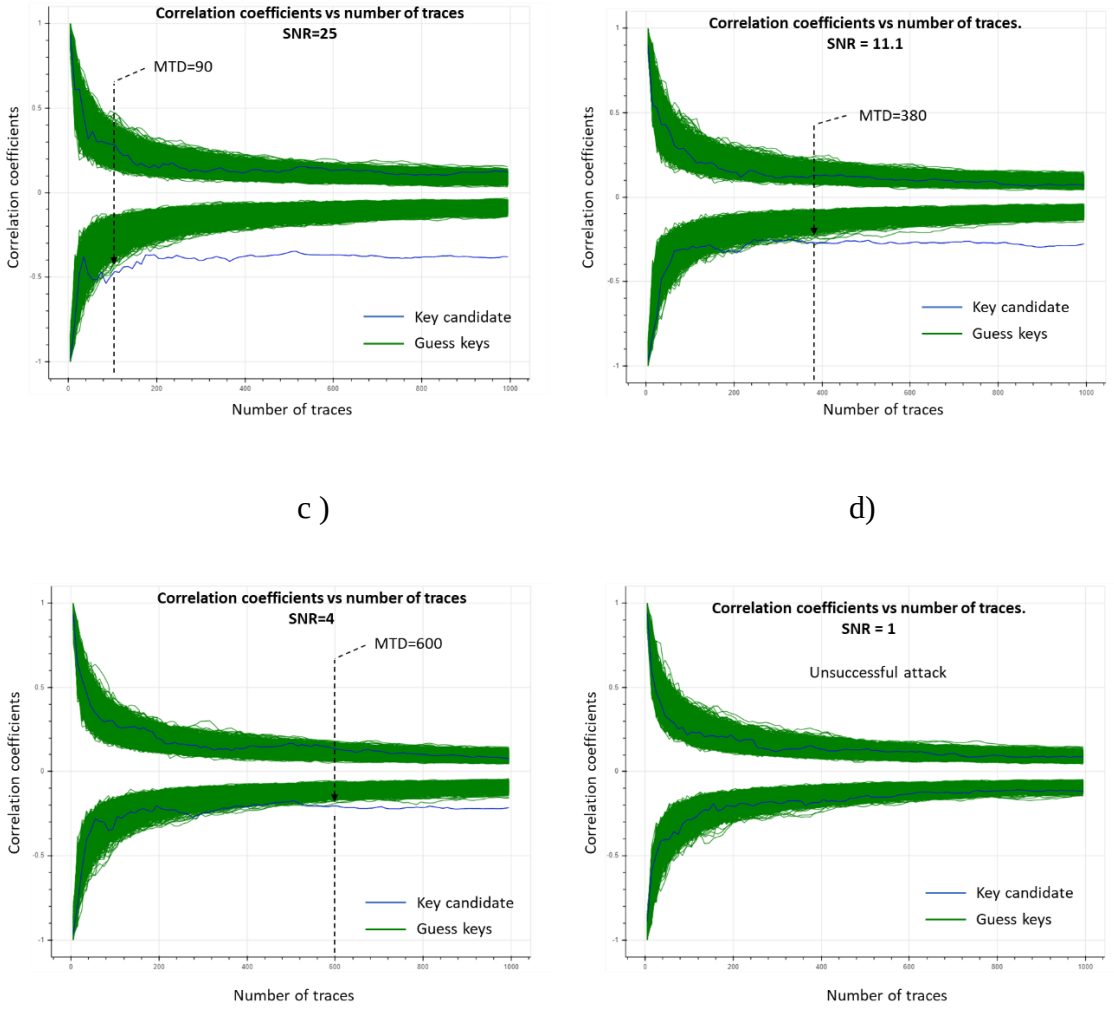
concurrently with the victim IP. Hence, for real-life applications with multiple IPs, the traces from other IPs constitute the noise that provides SCA countermeasures.

To quantify the impact of a noise source on SCA success, a Gaussian noise source is injected into the extracted model. Then, the simulation is run, and the measurements taken at the C4 bumps closest to the attacker trojan IP, to emulate a remote attack. A statistical analysis (CPA) is then performed to compute the correlation coefficients and the MTD for various SNR levels. Based on the results of the previous section, attacks carried out remotely are far more effective than local attacks; thus, it is predicted that with noise injected into the PDN network, a local attack will still require more traces to uncover the secret key.

The MTD for the baseline without noise injection is computed and plotted for five SNR levels: 10, 5, 3, 2, and 1 (Fig. 36). Note from the figure that the MTD increases gradually as we go from no noise to noise injection of SNR = 10 and 5. Then, there is an exponential increase as the SNR decreases from 5 to 1. We could not mount a successful attack with 1,000 traces when the SNR is equal to 1.



a) b)



c) d)

e) f)

Fig. 36. MTD with and without PDN noise injection: a) – Baseline, no noise injection; b) – Noise injection: SNR=100; c) – Noise injection: SNR=25; d) – Noise injection: SNR=11.1; e) – Noise injection: SNR=4; f) – Noise injection: SNR=1.

Fig. 37 summarizes the impact of the noise injection by plotting the maximum correlation coefficients, the experimental and theoretical correlation coefficient reduction ratio, and the MTD versus the noise relative magnitude (which is the inverse of the SNR). The plot also shows a linear interpolation of the MTD. The maximum correlation is attained for

each trace where the estimated leak function correlates with the measurements, which is during the last round of encryption. The correlation coefficient reduction represents the ratio of the max correlation coefficients for the noise level over the baseline max coefficient without noise injection. As expected, the max correlation coefficient decreases as the noise magnitude increases, as does the reduction ratio. A comparison between the theoretical reduction ratio for a system impedance with gain G=8, as defined in equation (10), shows a close match with the experimental results. Therefore, an empirical deduction is made that our system PDN network has a gain of G=8. Likewise, the MTD shows a similar trend, with a marked exponential increase above a relative noise magnitude of 0.3. In summary, the experimental results show that the presence of noise in the PDN is an effective countermeasure to power SCA.

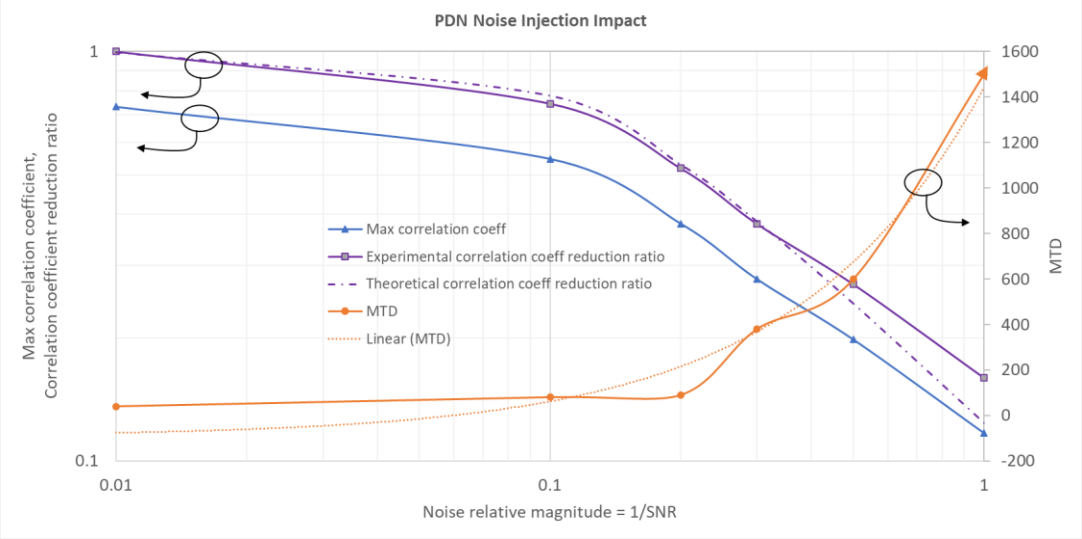


Fig. 37. Impact of noise injection on power side channel attack success.

7.5.6 Impact of on-package decoupling capacitors on side channel attack success

The OPD filters the 1st and 2nd droop signals seen at the die level. With OPDs modeled as shown in Fig. 38, simulations of the AES256 cryptosystem are run, and on-die and onboard waveforms are captured with OPD scenarios. Comparing the voltages with no OPD and with 20 OPDs, in Fig. 39, it is apparent that the OPDs significantly reduced the magnitude of the voltage measurement at the board level, from 14 mVpp to 1.6 mVpp (8.75x). They have also impacted the magnitude of the on-die voltage, albeit with a lower ratio, reducing it from 2 mVpp to 0.6 mVpp (3.33x).

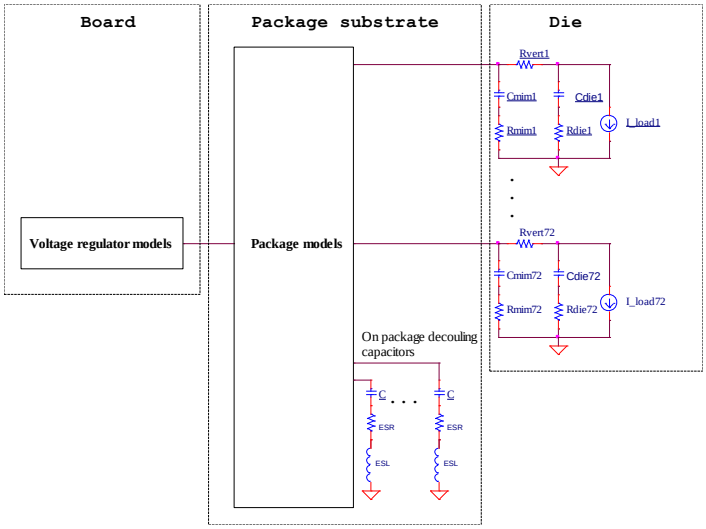


Fig. 38. System modeling with on-package decoupling capacitors

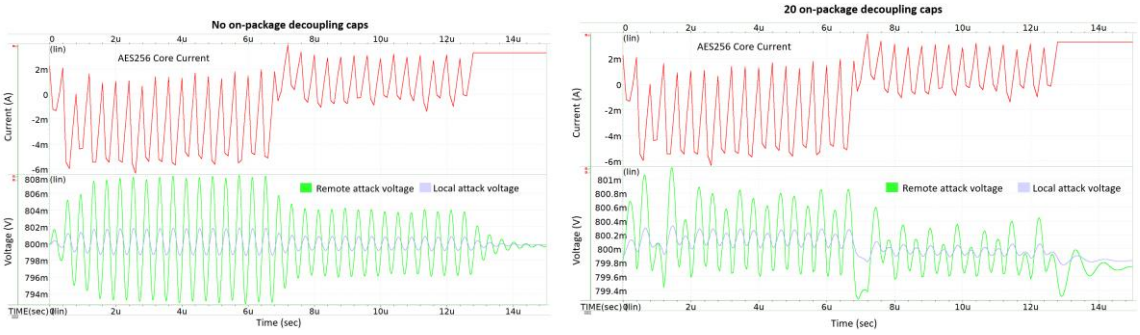
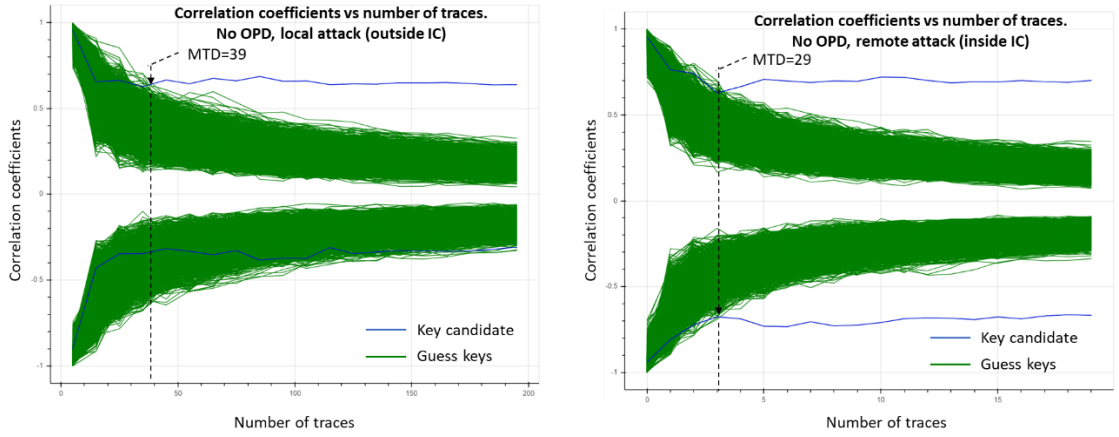


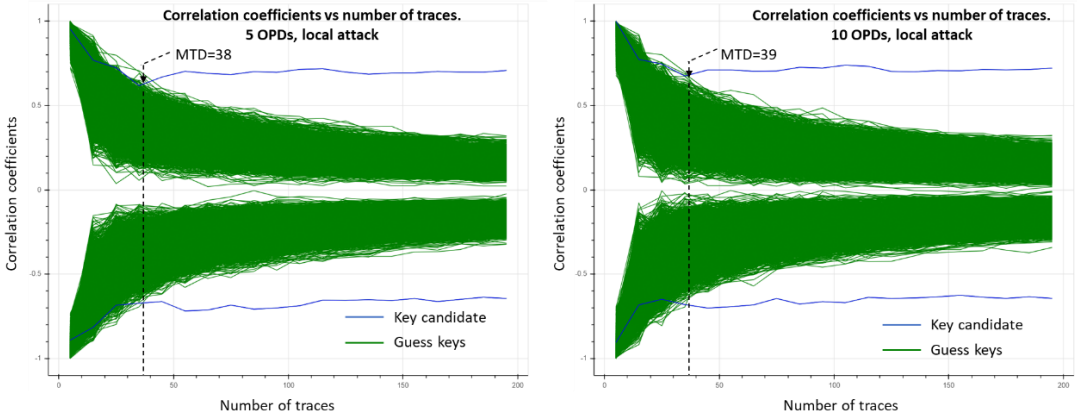
Fig. 39. Simulated waveforms of AES256 engine. Left: No OPD; trace measured at the die (green) and at the board (blue), vs AES256 current (red). Right: 20 OPDs, trace measured at the die (green) and at the board (blue), vs AES256 current (red).

Simulations were run with multiple settings of OPDs to gauge their impact on the success rate or the probability of an attacker uncovering the secret key while mounting either a local attack (capturing traces onboard with physical presence at the scene) or a remote attack (capturing the voltage at the die level with a trojan IP). With no OPDs, 39 and 29 traces are required to mount a successful local and remote attack, respectively. This is a reduction of 25.6% from local to remote attack scenarios (Fig. 40a, and Fig. 40b). With 5, 10, 15, 20, 30, and 40 OPDs, the MTDs for a local attack are 38, 39, 42, 48, 49, and 49, respectively, as shown in Fig. 41. Although the locally measured waveforms in the presence of OPDs show a gain attenuation, it should be observed that there is little distortion present on those waveforms compared to the waveforms without OPDs. This explains the CPA results that show only a small increase in the MTD: 39 to 48. The gain attenuation is a linear transformation that has no impact on the correlation coefficients. This is rooted in the principle of Pearson correlations, which constitute the basis for CPA [2][59]. When the estimated power accurately models the measured power, a deviation in the magnitude of the measured power, in the same direction as the no-OPD scenario, will lead to similar correlation coefficients as the no-OPD case. However, for remote attacks, the MTD remains constant at 29-30, regardless of the number of OPDs.

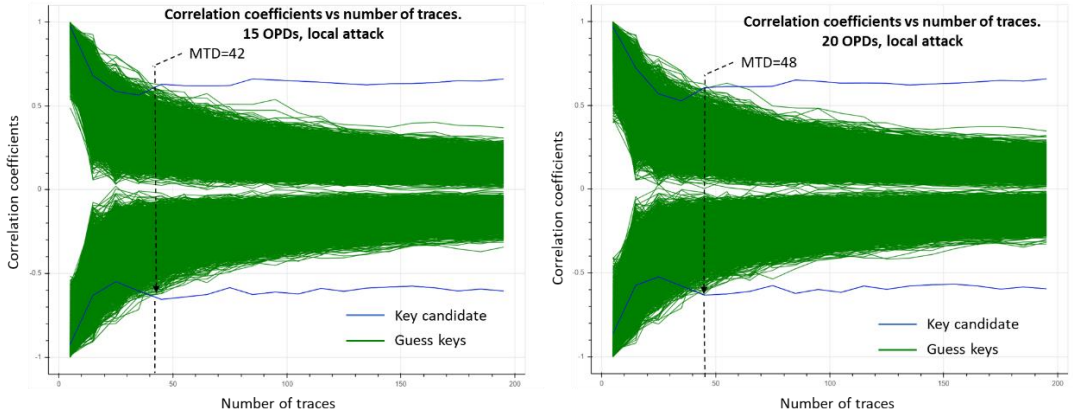


a) b)

Fig. 40. Correlation coefficients vs number of traces: MTD with no OPD for local attack (a) and remote attack (b).

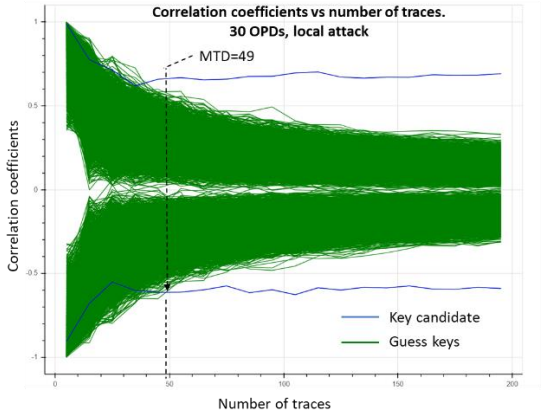


a) b)

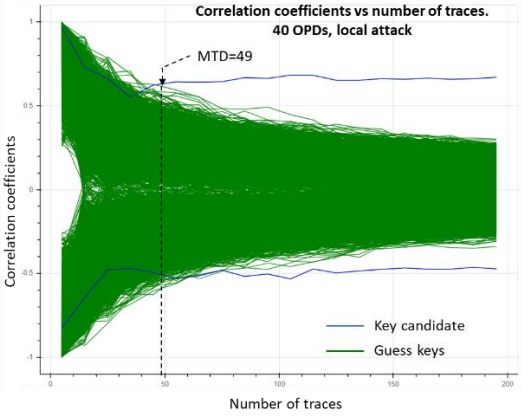


c)

d)



e)



f)

Fig. 41. Correlation vs number traces for various OPD settings: a) – 5 OPD; b) – 10 OPDs = ; c) 15 OPDs; d) 20 OPDs; e) - 30 OPDs ; f) - 40 OPDs.

A summary of the impact of the OPDs on the MTD is presented in Fig. 42. The local attack MTD increases from 39 to 49 (or ~25%) from no OPD to 40 OPDs but remains constant for remote attacks.

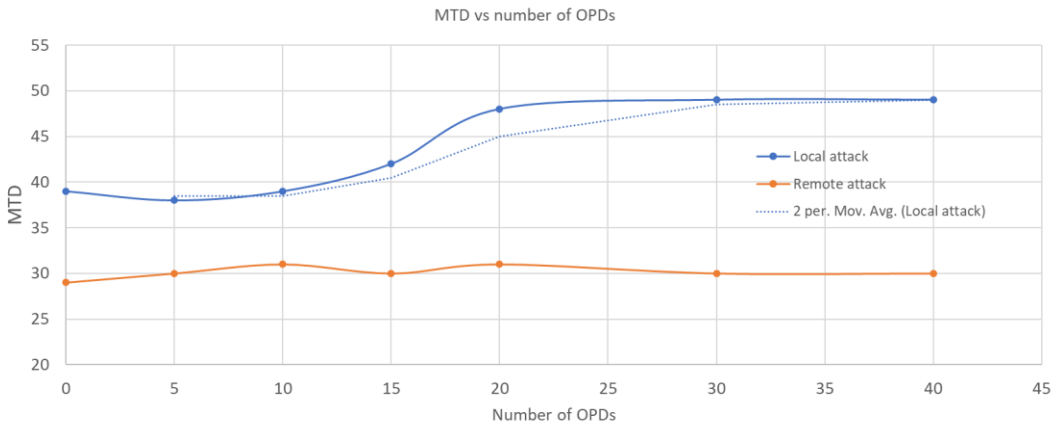


Fig. 42. MTD vs number of OPDs.

7.5.7 Summary of PDN countermeasures experimental findings

For the practical system considered herein, the design space is defined by the acceptable values of the design parameters that can be practically implemented to keep the product viable and realistic. The range of realistic values for the number of OPDs is 0 to 40, and the max implementable package size yielded a loop inductance of 2.5 nH and path resistance of 1.5 mΩ after extraction with broadband spice. Additionally, the maximum magnitude of the noise that can be injected into the design is set to be equal to the signal magnitude, hence a relative noise magnitude of 1. Therefore, the design space is defined as the trivariate (relative noise magnitude, number of OPDs, package impedance):

Design space

- Relative noise magnitude: [0,1]
- Number of OPDs: [0,40]
- Package impedance:
  - Loop inductance: [0.5nH, 2.5nH]
  - Path resistance: [0.5mΩ, 1.5mΩ]

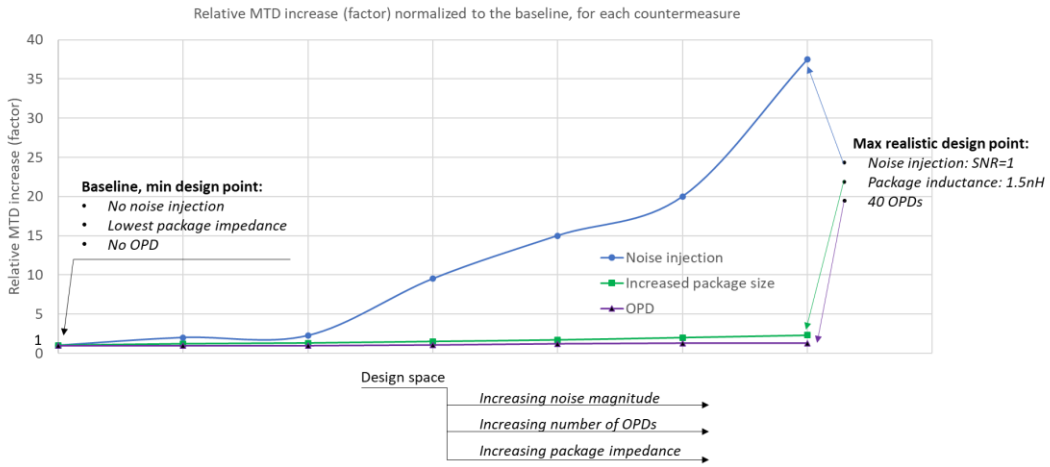


Fig. 43. Relative MTD increase compared to the baseline, for each PDN-based countermeasure.

The normalized MTD versus each of the design space variables is plotted in Fig. 43. The MTD is normalized to the following minimum value for each parameter: no noise injection (noise magnitude), no OPD (number of OPDs), and 0.5 nH, 0.5 mΩ (impedance). The Y-axis shows the relative increase in MTDs, and the X-axis shows increasing design parameter values. It can be observed that in the range of practical values, OPDs and larger packages provide only 1.3x and 2.3x increases in the MTD. However, the noise injection in the PDN yields a 37x increase in the MTD. In summary, one should not rely on increasing the number of OPDs or the distance between locally measured power and die location afforded by a larger package size as efficient power SCA countermeasures. Noise injection is by far the best countermeasure mechanism.

Chapter 8

Lightweight Ciphers in IoT Applications

Major sections of this chapter are part of a journal article accepted for publication in *IET Computers and Digital Techniques.*

Mozipo, A.T., and Acken, J.M.: Residual vulnerabilities to power side channel Attacks of lightweight ciphers cryptography competition finalists. *IET Comput. Digit. Tech.* 1-14 (2023). https://doi.org/10.1049/cdt2.12057

Authors: Aurelien. T. Mozipo, John. M. Acken

CRediT Taxonomy:

Aurelien T. Mozipo:

Conceptualization, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing

John M. Acken:

Conceptualization, Methodology, Supervision, Visualization, Writing – review & editing

DOI: 10.1049/cdt2.12057

Due to the exponential rise of communication networks implemented on small internet of things (IoT) devices, there has been an urgent need to secure these networks to protect both consumers' and cloud service providers' private information. With the implementation of cryptographic algorithms, the need arises to protect them against malicious attacks. Power side channel attacks (SCAs) are of great concern on IoT devices. This is stemming from the fact that malicious agents can implement power measurements and run cryptanalysis algorithms such as differential power analysis (DPA) to extract secret information from the device. Although power SCAs have been extensively studied, they have been applied mainly to the advanced encryption standard (AES) for regular full power applications. The AES is not suited for IoT devices because of its complexity and power dissipation. Multiple lightweight, low-power, compact cipher algorithms have been proposed for such devices. Likewise, traditional countermeasures against a power SCA proposed for AES implementations yield relatively significant area, performance, and power overheads when implemented on lightweight ciphers such as SIMON, PRINCE, and PRESENT. But there are optimal countermeasures or modes of operation targeted for lightweight ciphers that lead to acceptable results. Particularly, SIMON with a round unrolled datapath architecture that enhances vulnerability against a power SCA and yet increases throughput and reduces energy per encryption (pJ/encryption), has been presented in [32] and [110]. Likewise, PRINCE with unrolled architecture implementation with countermeasures against power SCA has also been proposed [37].

Although these lightweight ciphers represent a viable and safe alternative to the power-hungry AES, their proliferation and the indecision in the industry around the choice of a common encryption technique and mode of operation have prompted the US National Institute of Standards and Technology (NIST) to undertake the creation of standard, resilient lightweight ciphers. They should encompass confidentiality, security, and authentication. They must either have built-in countermeasures to side channel attacks or show a strong resistance against power SCAs through the algorithm constructs.

8.1 Current lightweight ciphers

Because IoT devices are typically low-power, low-resource devices, there's the need to use lightweight cryptography algorithms to encrypt data before transmission to the cloud. A sample set of countermeasures implementation of power SCA on three lightweight ciphers, PRESENT, SIMON, and PRINCE is shown in Table 10. Masking and hiding techniques which are effective countermeasures techniques for regular cryptographic algorithms (AES/DES) are not practical for lightweight ciphers because they require significant resources. Algorithmic implementation of a parallel data path in lightweight ciphers is one of the countermeasures against SCA proposed here. Serial implementations tend to have high leakage. But [39] shows that SIMON 64/96 can be made resilient to CPA at the cost of 66.6% cost to area and 13.4% performance penalties. For SIMON128, the MTD (Minimum Traces to Detection) is 1300 for bit-serial implementation and 20000 for 64-bit parallel implementation. However, a residual vulnerability of this SIMON 64/96 implementation is a collision attack. Collision-correlation attacks are still possible though difficult.

Round unrolling which, has been previously proposed for DES, is another countermeasure proposed for lightweight ciphers. SIMON128 implemented with a 6-round unrolled data path had MTD 384x the baseline bit-serial design. Also, with 500 000 traces, there's no successful CPA attack [32]. Even though the MTD is higher for SIMON128-bit parallel and very high (>500K) for 6-round unrolled datapath, these algorithms could still be vulnerable to CPA if the number of traces is high enough. However, a high degree of round unrolling (i.e 6) makes CPA infeasible. Thus, a residual vulnerability of SIMON128 implementation with 6-round unrolled is hard to implement with power SCA given that the MTD is a lot more than 500 000. However, given the advancement in computer performances, it's within the realm of possibilities that in the next one or two generations of high power processors, computers might be able to process millions of traces within a time acceptable to successfully perform an attack.

Table 10 - Residual and added vulnerabilities of some Lightweight Ciphers

	Countermeasure, SCA protected Against	**Residual vulnerabilities**	**Added vulnerabilities**
Energy Efficient and Side-Channel Secure Cryptographic Hardware for IoT-Edge Nodes [32]	• SIMON protected against CPA • Round unrolled SIMON 64b datapath (2r, 3r, 4r, 6r unrolled)	• CPA with a higher number of traces >500k • HO-DPA • DPA in presence of glitches	None
Extracting Side-Channel Leakage from Round Unrolled Implementations of Lightweight Ciphers [40]	Unrolled datapaths of lightweight cryptographic algorithms are resistant to SCA	Unrolled architecture in PRINCE, SIMON, and other lightweight ciphers remain vulnerable to CFA (Correlation Frequency Analysis).	PRINCE with glitch canceler is more vulnerable to SCA
A look into SIMON from a side-channel perspective [39]	CPA (SIMON 64/96) on masked implementations of SIMON	Collision-correlation attacks are difficult but still possible	
Statistical Power Analysis for IoT Device	Study tamper resistance of glitch	None observed	PRINCE in IoT, with glitch canceler, has

Oriented Encryption with Glitch Canceller [37]	canceler used for power reduction		increased vulnerability to SCA
Lightweight ciphers with serial architectures: CLEFIA [103], PRESENT [104], PRINCE [105]	CPA	• CPA because of high SNR: no algorithm noise stemming from parallel implementations, thus increased side channel power leakage	None

8.2 A First look at residual vulnerabilities to power side channel attacks of lightweight cryptography competition finalists

This section proposes a comprehensive evaluation of the ten algorithm finalists of the National Institute of Standards and Technology (NIST) IoT lightweight cipher competition.

8.2.1 Introduction

8.2.1.1 Relevant studies on the security of IoT communications

Much of this research is focused on studying the residual vulnerabilities to power SCA of the NIST lightweight ciphers cryptography competition (LWC) finalists. First, we are dedicating this section to introducing similar relevant prior art as well as the necessary background knowledge helpful to readers in understanding the concepts at hand. Many researchers have published studies to address the security challenges of lightweight cryptographic protocols. The authors of [146] published a comparative survey of lightweight cryptographic algorithms, their strengths, weaknesses, and general security requirements, such as integrity, confidentiality, and authentication. [147] compared the 32 LWC second-round candidates for features such as performance and power. [148]

focuses on surveying certain lightweight block ciphers that can easily be implemented in resource challenge devices; such ciphers include PRESENT, SIMON, and GRAIN. The authors of [149] propose a SCA categorization system, particularly for enabling analysis of SCA on mobile devices. The study goal is also the facilitation of the development of new countermeasures.

Protecting the integrity of the communications between IoT devices goes beyond the protection of the device themselves. Malicious agents have also intercepted the communications and tried to exploit the weaknesses in the protocol. [150] proposes a survey of existing protocols and analyzes methods to establish secure communications between IoT devices. Direct attacks on lightweight cipher implementations are also a threat to IoT devices' data. [151] proposes a differential attack on the family of lightweight block ciphers SKINNY. [152] has demonstrated a successful collision fault attack on GIFT with only 64 faulty ciphertexts.

To help understand the theory and algorithms behind power side channel cryptanalysis, the review in [153] and study in [154] provide foundations that summarize the concepts of power analysis distinguishers. They focus on distinguishers used in non-template attacks, including correlation power analysis (CPA), which is one of the most efficient distinguishers. They also introduce the notion of test vector leakage assessment (TVLA). TVLA, based on Welch’s T-test, uncovers leakage of information without mounting an attack. Other distinguishers summarized in the paper are simple power analysis (SPA), differential power analysis (DPA), and mutual information analysis (MIA).

However, none of these prior studies address the issue of resistance to power side channel attacks of the LWC finalists, thus our analysis is the first with such a goal.

8.2.1.2 Related surveys and work on side channel attacks of lightweight ciphers

In this section, we discuss surveys of IoT and mobile devices, as well as surveys on SCA distinguishers, applied to lightweight ciphers. We also present studies dealing with multiple cryptanalysis aspects of a single lightweight cipher. Multiple prior arts have also performed comparative studies of SCA on multiple lightweight ciphers, which we are also summarizing in this section. Table 11 summarizes the prior art covering surveys and studies on lightweight ciphers' vulnerabilities to power SCA, with references for readers.

Table 11 - Previous Surveys/work on Side Channel Attacks on Symmetric Ciphers

Year	Article	Main topic covered
2021	Khan M N [146]	Lightweight cryptographic protocols, focusing on IoT devices
2018	Spreitzer R [149]	Classification of side channel attacks, focusing on mobile devices
2020	Randolph M [153]	Exploration of the foundation of power SCA distinguishers.
2020	Fei Y [155]	Evaluation of WAGE vulnerability to CPA and comparison with LWC competition 2nd round candidates.
2022	Liu Z [156]	Root cause of power leakage, compared to AES, in three candidates of LWC competition.
2022	Abdulgadir A [157]	Study the impact on cost and performance, of applying Domain-Oriented Masking on three LWC competition finalists.
2022	Babinkostova L [158]	Study of side channel leakage of GIFT-COFB by applying CPA with the Hamming distance model.
2016	Nalla Anandakumar, N [159]	Study SCA resistance of FPGA implementations of MAC-PHOTON.
2016	Biryukov A [160]	Analysis of the efficiency of common leak functions used in CPA to attack AES and seven lightweight ciphers.
2021	Zhang J [161]	Power attack method against the diffusion layer of GIFT implemented in an FPGA.
2017	Samwel N [162]	Presents first DPA attack on Keyak S-box and first CPA attack on Ascon S-box.
2022	Windarta S [163]	Analysis of cryptographic areas and cryptanalysis attacks of various hash functions suitable for lightweight ciphers.
2022	Batina L [164]	Side channel attack evaluation of software implementations of ASCON, Xoodyak and ISAP

2021	Miteloudi K [165]	First application of ROCKY as a countermeasure against SCA.
2018	Diehl W [166]	Study of protection against DPA of a few authenticated ciphers.
2017	Heuser A [167]	Study of side channel analysis metrics used to determine resistance to SCA.

8.2.1.2.1 Surveys on IoT and mobile devices

[146] surveyed lightweight cryptographic protocols focusing on IoT devices. But, SCA on these protocols is not a focus of the study. However, [149] presents a classification of side channel attacks focusing on mobile devices. They allow and facilitate the development of new countermeasures. But this paper fails to address most lightweight ciphers and certainly not the NIST LWC candidates, which is the focus of our study.

8.2.1.2.2 Surveys and studies on SCA distinguishers applied to lightweight ciphers

The study of [155] evaluates and analyses authenticated lightweight cipher WAGE vulnerability to CPA and compares against LWC 2nd round candidates. [158] focuses on the study of side channel leakage of GIFT-COFB by applying CPA with the Hamming distance model. Then, they use the attack results to rate the reliability of several side-channel leakage assessment metrics: transparency order, revisited transparency order, and signal-to-noise ratio, amongst others. [159] studies SCA resistance of FPGA implementations of MAC-PHOTON. They implement three concept architectures (iterative, folding, and unrolling), then analyze their security against SCA. They also elaborate on MAC-PHOTON Threshold Implementation (TI) resistance against first-order power analysis. [161] covers power attack methods against the diffusion layer of GIFT implemented in an FPGA. [162] presents the first DPA attack on Keyak S-box and the first CPA attack on Ascon S-box. The difference with our work is that we propose a

method to attack the 320-bit state of ascon-128. In [165], they show the first application of ROCKY as a countermeasure against SCA, on four architectures of Xoodoo implemented in an FPGA.

Contrary to the above-mentioned studies and surveys that focus only on a single cipher, our study focuses on exposing residual vulnerabilities on multiple LWC finalists, namely all seven ciphers that do not have built-in SCA countermeasures.

8.2.1.2.3 Surveys of the comparative studies of SCA on multiple lightweight ciphers.
GIFT-COFB, Xoodyak, and Grain-128, three finalists of the LWC are covered in [156]. This research studies the root cause of power leakage in those ciphers and compares it to AES. [157] studies the impact on cost and performance, of applying Domain-Oriented Masking on three LWC competition finalists: Elephant, TinyJambu, and Xoodyak. In [160], the authors analyze the efficiency of common leak functions used in CPA to attack symmetric ciphers. The study case is the implementation of AES and seven lightweight ciphers (Fantomas, LBlock, Piccolo, PRINCE, RC5, SIMON, and SPECK) in an 8-bit processor. None of these is amongst the finalists of the NIST LWC, which is the focus of our study. [163] focuses on the analysis of cryptographic areas and cryptanalysis attacks of various hash functions suitable for lightweight ciphers. They have also conducted a comparative study and presented research challenges on hardware and software implementations of those lightweight cryptography hash functions. However, this work does not focus on power SCA. [164] proposes side channel attack evaluation of software implementations of ASCON, Xoodyak, and ISAP. [166] is a study of protections against DPA of a few authenticated ciphers (ACORN, ASCON, CLOC, SILC, JAMBU, AES-

GCM). In that paper, the authors use TVLA to demonstrate vulnerability to 1st-order DPA and to demonstrate improved resistance of the protected versions. Then, they compare the cost of implementing countermeasures on those ciphers. [167] is a study of side channel analysis metrics used to determine resistance to SCA. Particularly, they attack the first, last, and both rounds of several 4-bit S-boxes ciphers (KLEIN, Midori, Mysterion, LED, Piccolo, PRESENT, PRIDE, PRINCE, RECTANGLE, SKINNY) and 8-bit S-boxes ciphers (AES, Zorro, Robin).

Amongst the above-mentioned studies that deal with the same ciphers of interest as us, the NIST LWC finalists, a maximum of three ciphers is analyzed in any one paper. Therefore, none comprehensively covers the SCA vulnerabilities of all of them; which is what we address in this paper.

8.2.1.3 Organization of this section on LWC finalists

This paper is organized as follows: section 2 presents a detailed account of our contribution to knowledge while section 3 is the background summary which gives the theoretical knowledge necessary to understand the analysis throughout this section. In section 4, we discuss our evaluation of the residual vulnerabilities against power SCA of the seven LWC finalists that do not integrate a built-in countermeasure against side channel attacks. We conclude our analysis in section 5.

8.2.2 Our Contribution

The novelty of this research resides in the fact that we identify vulnerabilities to power SCA in seven (out of ten) LWC finalists and propose methodologies for attacking five of

them. We also propose the leakage functions needed to perform CPA on those lightweight ciphers.

This study defines a method for attacking Ascon by reducing the key search space to a practically implementable size. We also propose a leakage function used in a CPA to attempt to uncover the state. Leveraging a methodology shared in [102], we introduce two hamming distance-based leakage functions for attacking the first and last rounds of GIFT_COFB. We highlight the Hamming distance-based leakage model for attacking GRAIN-128-AEADv2. The study also proposes methodologies for launching power SCA on PHOTON-Beetle, Romulus, and Schwaemm.

The study begins with a comprehensive comparative study and evaluation of the 10 LWC finalists to evaluate their hardware implementations' residual vulnerability against a power SCA. To our knowledge, a study of this kind has never been performed on these ciphers, so this will be the first proposal. Many generalized analyses of lightweight ciphers have been proposed. Some general studies focus on security aspects, performance, power consumption [147], area, and validations of advertised features of confidentiality, authentication, and integrity [146]. Unlike [147], which proposed a general, broad survey targeting the 32 second-round candidates of the LWC competition, our research goes in-depth into the level of resistance to power side channel attacks, targeting the 10 candidates of the final round. We aim to provide the evaluators of these algorithms, the NIST community, and IoT device designers with the tools that will help educate and inform on the weaknesses of those algorithms. The authors of [112] have launched a call to side channel security labs to propose an evaluation against side channel

attacks of the 10 finalists. Hence, our comprehensive vulnerability evaluation is intended to serve as a lantern to those who aim to develop power side channel attack proposals against the 10 finalists to evaluate their robustness before the final selection by NIST. Some of the residual vulnerabilities uncovered are based on demonstrated, previously published literature. Others are based on our initial theoretical assessment.

8.2.3 Background

To address the critical issue of standardization of lightweight ciphers, NIST has initiated a competition to solicit lightweight ciphers suitable for low-power, compact, or otherwise highly constrained devices. After two preliminary selection rounds, NIST reduced the initial 57 submissions to a final round of 10 candidates.

Table 12 - Security characteristics of the 10 finalists of the lightweight cipher cryptography competition

	MAC, Hash functions or primitives	**Vulnerabilities and cryptanalysis features reinforcing security**
Ascon [121] **Type: Block cipher;** **Key size: 128**	Ascon-Hash, Ascon-HashA	• No countermeasure is implicitly implemented. However, the algorithm architecture offers protection against repeated nonces. • Ascon round function is amenable to the application of masking.
Elephant [137] **Type: Tweakable block cipher (Elephant);** **Key size: 128**	A variant of the protected counter sum MAC function	Masked using LSFR
GIFT-COFB [138] **Type: Block cipher (GIFT-128)** **Key size: 128**	No integrated hash functionality. If needed, the authors propose a 256-bit hash function from another research.	• Masked using LSFR • GIFT Ascon round function is amenable to the application of masking.
Grain-128AEADv2 [120] **AEAD stream cipher;** **Key size:128**	Based on non linear feedback shift registers (NLFSR) and LSFR pre-output generator	None

	MAC, Hash functions or primitives	Vulnerabilities and cryptanalysis features reinforcing security
ISAP: [129] Isap-A-128a, and Isap-A-128 Isap-K-128a, and Isap-K-128 Key Size:128	320-bit Ascon-p permutation 400-bit Keccak-p[400] permutation	Sponge based rekeying
PHOTON-Beetle Authenticated Encryption and Hash Family [126] Key size: 128	P_{256} ($PHOTON_{256}$ Hash)	None
Romulus [139] Type: Tweakable Block Cipher (SKINNY); Key size: 128	Romulus-H	SKINNY round function is amenable to the application of masking.
SPARKLE (SCHWAEMM and ESCH [132] Type: block cipher; Key size: 128	Esch	• No countermeasure • Collision resistant • Long Trail Strategy (LTS) provides security against differential and linear cryptanalysis
TinyJambu [134] Type: block cipher Key sizes: 128, 192, 256	Keyed permutation P_n N rounds of state update, based on nonlinear feedback shift register	TinyJambu round function is amenable to the application of masking.
Xoodyak [136] Type: Stream cipher; Key size: >= 128	Xoodoo permutations	Built-in countermeasures: • Cyclist: DPA countermeasure that absorbs the session counter used for a nonce. It limits the number of selection functions an attacker can use. • A key replacement scheme. Instead of a counter, a new key is generated and saved for the next instantiation of Xoodyak • A method similar to “Forget”, which is a ratchet mechanism offered by Cyclist: prevents the recovery of the secret key before the use of the ratchet. • Xoodoo round function is amenable to the application of masking. However, masking is implemented in the LWC proposal.

8.2.3.1 Lightweight cipher competition finalists

Table 12 summarizes the main characteristics of the 10 proposals selected by NIST for the final round of evaluations. They are based on authenticated encryption with associated data (AEAD), which are symmetric encryption algorithms that provide both confidentiality and authentication. The following three LWC competition finalist algorithms have integrated countermeasures against side channel attacks: ISAP, Elephant,

and Xoodyak. Elephant implements masking using linear-feedback shift registers (LSFRs) [137]. ISAP features sponge-based rekeying [129]. Xoodyak's built-in countermeasure, called Cyclist, implements a DPA countermeasure by absorbing the session counter that is used for a nonce. It limits the number of selection functions an attacker can use [136]. However, the other ciphers, Ascon, GIFT-COFB, Grain, PHOTON-Beetle, Romulus, SPARKLE, and TinyJambu, do not feature such built-in side channel protections and will constitute the focus of this work.

8.2.3.2 Security Metrics for Sample Classes of Attacks

Before diving into the cipher analysis, let us state some security metrics which are classes of attacks that constitute the basics of some vulnerabilities exposed in a few ciphers [125].

8.2.3.2.1 Chosen ciphertext attack and leakage in encryption only, with nonce-respecting adversary (CCAL1):

The malicious agent performs several encryption/decryption operations that leak the algorithmic implementation of the authenticated encryption scheme. Then, s/he chooses two new messages and receives the corresponding ciphertexts while measuring the leaked information. The system is considered insecure when the agent can match the ciphertext to the plaintext with a reasonable advantage. The CCAL1 security variant is when the chosen ciphertext has nonce-respecting and leakage is measured during encryption operations only.

8.2.3.2.2 Chosen ciphertext attack and leakage in encryption only, with nonce-misuse resilience adversary (CCAmL1):

Same as CCAL1 but with a fresh challenge nonce.

8.2.3.2.3 Chosen ciphertext attack and leakage in encryption only, with nonce-misuse resilience (CCAmL2):

Same as CCAL1 but with a fresh challenge nonce and leakage during both encryption and decryption.

8.2.3.2.4 Ciphertext integrity with leakage during encryption only (CIL1), with nonce-respecting adversary:

For this security metric, the malicious agent also performs encryptions/decryptions while capturing the leaked information. The implementation is considered secure if the malicious agent cannot guess a valid plaintext with good probability. The CIL1 security variant is non-respecting and leaks only during encryption.

8.2.3.2.5 Ciphertext integrity with leakage during encryption and decryption (CIML2), with nonce misuse resistance:

Similar to CIL1, except there is no constraint on nonces and leakage during both encryption and decryption.

8.2.4 Evaluation of residual vulnerabilities

Several studies have demonstrated the effectiveness of authenticated encryption with advanced data (AEAD), in protecting communications with IoT devices [168][169][170]. They provide security, authentication, and confidentiality, all in one algorithm implementation. However, the proposed LWC algorithms still displayed residual vulnerabilities against power SCA, which we expose in the next few sections.

8.2.4.1 Ascon-128/Ascon-128a

Ascon-128 and Ascon-128a are suites of lightweight ciphers that provide AEAD, in addition to hash functions Ascon-Hash and Ascon-Hasha and extendable output functions Ascon-Xof and Ascon-Xofa. The primary recommendation for the NIST competition is

the suite set Ascon-128/Hash-128/Hash-Xof. The parameters for this authenticated encryption scheme include a key size and permutation length of 128 bits and 320 bits, respectively. The algorithm also features two permutations p^a and p^b used in the AEAD and the hash functions of lengths 12 and 6 in the AEAD, and lengths of 12 each in the hashing algorithm [121].

The construction of Ascon has an initialization stage that generates the state by manipulating the encryption key (*K*), the initialization vector (*IV*), the nonce (*N*), and the permutation p^a as follows:

$$S \leftarrow IV \parallel K \parallel N \tag{1a}$$

$$S \leftarrow p^a(S) \oplus (0^{320-k} \parallel K) \tag{41b}$$

$$p^a = p_C \circ p_S \circ p_L \tag{1c}$$

where *S* is the 320-bit state, K is the *k*-bit key, *k*=128, p^a is a permutation with *a* rounds (*a*=12), p_C is the constant addition layer, p_S is the substitution layer, p_L is the linear diffusion layer and ‖ represents the concatenation operation.

8.2.4.1.1 Proposed scheme for attacking Ascon-128/Ascon-128a

A power SCA works on the premise of developing a predictable relationship between the algorithm's internal operations, the encryption key, and other input/output data. Thus, based on the initialization stage in the equations above and the Hamming distance model developed in [102], we propose the following leakage function for an attack on Ascon-128 using the correlation power analysis (CPA) distinguisher :

$$Leak_{ascon-128} = HD(p'^a(S) \oplus (0^{320-k} \parallel K), S) \tag{42}$$

where *HD(x,y)* represents the Hamming distance between *x* and *y*.

The success of this leakage function in recovering the state largely depends on the signal-to-noise ratio (SNR) of the measurements, which in turn depends on the algorithm implementation. The authors of Ascon have stated that recovering the state during data processing may not directly lead to recovery of the secret key, and recovery of the state during the initialization stage will lead to recovery of the secret key.

To reduce the complexity of guessing the 320-bit state *S*, we decompose the guessing phase into 64-bit substates to align with the structure of the 64-bit register words (x_0, x_1, x_2, x3, x_4):

$$S = x_0 \parallel x_1 \parallel x_2 \parallel x_3 \parallel x_4 \quad (3)$$

We can divide Ascon state *S* into 5 64-bit words and guess each word by replacing the permutation p^a with the permutation p'^a defined as follows:

$$p'^a = p_C \circ p_L \quad (4)$$

By removing the substitution layer p_S from the permutation p^a, we are ensuring that the result of each substitution p'^a on x_i does not depend on the remaining 4 words. Given that the substitution layer p_S acts on a 5-bit column word across all 5 words x_i, it mixes the 5 64-bit words, and thus its output is no longer solely dependent on the 5-bit words x_i. Thus, the attack on the state is reduced into 64-bit operations, therefore reducing the search from 2^{320} to $5x2^{64}$.

8.2.4.1.2 On the confidentiality and integrity of Ascon under the security game "Chosen ciphertext attack and leakage in encryption only, with nonce-misuse resilience adversary (CCAmL1)"

The message processing part of Ascon is simple power analysis (SPA) secure under CCAmL1 assumptions. However, without DPA protected implementation of the

verification phase [125], it is possible to successfully attack secure bootloading applications [127] by estimating valid messages without knowledge of the encryption key [128].

8.2.4.2 GIFT-COFB

GIFT-128, which is the block cipher used in GIFT-COFB LWC, is a larger version of PRESENT [113]. Thus, weaknesses of PRESENT against power SCA are also vulnerabilities of GIFT-COFB against power SCA. PRESENT implements a bit-oriented permutation layer and has a 64-bit block size. Each encryption/decryption round consists of layers AddRoundKey, sBoxLayer (substitution layer), pLayer (permutation layer). One more key addition is performed after the encryption rounds. Similarly, each encryption round of GIFT-128 (and GIFT-COFB) consists of 3 three layers: SubCells (32-bit state cell substitution), PermBits (bitwise permutations, different for each 32-bit state cell), and AddRoundKey (round key addition to the state).

Proposed leakage function for attacking GIFT-COFB

The round constructions are also similar to AES rounds, especially the last round which does not feature a MixColunm layer. Thus, we are proposing that GIFT-COFB can be attacked with a CPA targeting the first and/or last round with leak functions defined in the equations below:

$$Leak_{first\_round} = HD\big(PermBits(SubCells(P)) \oplus K_{r\_fisrst}, P\big) \quad (5)$$

$$Leak_{last\_round} = HD\big(inv\_SubCells(inv\_PermBits\big(C \oplus K_{r\_last}\big), C\big) \quad (6)$$

where *HD* represents the Hamming distance, K_r is the round key, *P* (plaintext) is the input to the first round, and *C* (ciphertext) is the output of the last round. Successful

uncovering of the encryption key in an AES implementation has been demonstrated with practical experiments, with similar leakage functions [102].

Additionally, an analysis performed on 4000 traces of PRESENT in an ASIC without any countermeasure yielded a test vector leakage assessment (TLVA) of 12.28 [114], which is higher than the threshold of 4.5 required by NIST to be accepted for secure cryptographic implementations. This means that the measured traces of GIFT-COFB implementations will be said to carry sensitive distinguishing information that could be exploited by a malicious agent to uncover the secret key.

8.2.4.3 GRAIN-128-AEADv2

The authors of this algorithm proposal have argued that Grain-128a (the raw encryption algorithm of GRAIN-128a-AEADv2) is resistant to a fast correlation attack, the classical method that was designed to exploit the state of the LFSR inside the algorithm [120]. Although [118] have demonstrated successful attacks on smaller grain-like stream ciphers, those attacks do not apply to Grain-128a. Furthermore, a revised fast correlation attack from the same authors revealed that the Grain-128a state can be recovered with data and time complexity of 2^{114} [119]. However, this revised fast correlation attack does not apply to Grain-128a in authentication mode because only every other keystream bit can be recovered by the malicious agent [120].

But, GRAIN-128-AEADv2 is an AEAD stream cipher that derives from GRAIN-128-AEAD, which is a common stream cipher previously studied in the literature, from an SCA perspective [117]. For any successful SCA, the malicious agent needs to have a deterministic relationship between the input data and the encryption key. As

demonstrated in [102][117], the Hamming distance model is a very reliable method to estimate the power dissipation of the system for CPA. It is possible to define a leakage model based on the Hamming distance of the state.

Also, it has been demonstrated that one can construct a fast and automated process through Z3, a publicly available satisfiability modulo theory (SMT) solver, with the leakage model and the publicly available keystream, which leads to key recovery in a few seconds [117].

8.2.4.4 PHOTON-Beetle

PHOTON-Beetle authenticated encryption and hash are made of the sponged-based mode Beetle and the PHOTON256 permutation [126]. Although its mode is designed to be side channel resistant, PHOTON-Beetle is only strongly protected against SPA without averaging [125]. Given that the nonce repetition is prevented under the CCAL1 and CIL1 security hypothesis [125], the resistance to SPA is thus at its possible maximum. Thus, PHOTON-Beetle can be implemented in the flat, leveled architecture shown in Fig. 44.

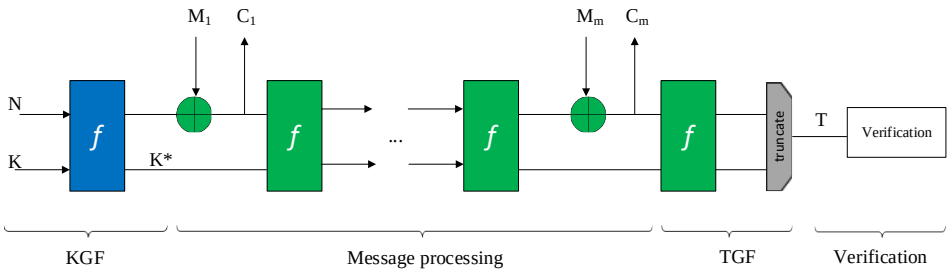


Fig. 44. PHOTON-Beetle leveled implementation for an *m*-block message, with CCAL1 and CIL1 security targets

The following defines the variables used in the above Fig. 44. *N*: nonce, *K*: master key, *M*: plaintext divided into *m* blocks of *r* bits each, with the last padded with 0's if it is

smaller than *r*, *r*=128 is the rate of the message absorption, *T*: tag, ***f:*** PHOTON256 permutation function [126], *KGF*: key generation function, and *TGF*: tag generation function.

However, the PHOTON-Beetle message processing section shows residual vulnerability against DPA with the following scenarios: define a fix nonce and ephemeral key K*, generate multiple plaintexts blocks M_1 or multiple ciphertexts C_1, uncover the capacity section along with the plaintext M_1 or ciphertext C_1, and then perform the inverse permutation to uncover the key *K*. Thus, more uniform protection is needed to obtain a security level stronger than CCAL1 and CIL1 [125].

Comparing the vulnerability to a power SCA of PHOTON-Beetle S-Box vs. Elephant S-Box and GIFT S-Box:

S-Box operations in symmetric cryptographic algorithms are frequently the main target of malicious agents who wish to extract information about the secret key. The authors of [131] have developed theoretical metrics to evaluate the vulnerability against a power SCA of 4x4 S-Box operations in PHOTON-Beetle and several other lightweight ciphers: revisited transparency order (VTO), confusion coefficient variance (CCV), and minimum confusion coefficient (MCC). Based on theoretical analysis, PHOTON-Beetle is the least vulnerable to a power SCA when evaluated with VTO and CCV. Even with the MCC metric, PHOTON still shows the 2nd highest resistance to a non-profiled power SCA among the nine ciphers studied (including NIST LWC finalists GIFT and Elephant). Additionally, practical experiments evaluating the minimum number of traces to achieve 90% confidence of attack showed that PHOTON requires ~800 traces (for a non-profiled attack and noise level of $\log_2(\sigma^2) = 5$) and 300 (for a profiled attack and noise level $\sigma=2$).

For a non-profiled attack, PHOTON ranks second least vulnerable after the Elephant cipher. But for the profiled attack, the position compared to Elephant and GIFT is inconclusive, as it varies depending on the trace noise level [131].

However, although these results might indicate a low level of vulnerability for PHOTON, it is worth pointing out that the number of traces required to reach a high confidence level is nonetheless very low compared to what state-of-the-art attack scenarios are capable of today [102][127].

In summary: barring the realistic aspect of implementing a practical attack targeting solely PHOTON-Beele S-box operations, a malicious agent will merely need to increase the number of traces to successfully uncover the encryption key.

8.2.4.5 Romulus

Romulus is based on a tweakable block cipher modeled over the SKINNY family of ciphers. Precisely, the version proposed in the LWC competition, Romulus-N, implements a change in the number of rounds compared to SKINNY-128-384. Romulus-N adopts 40 rounds of encryption, which is the same SKINNY-128-384+ [121]. Similar to GIFT-COFB, Romulus will be vulnerable to the same power SCA methodologies that have been demonstrated on its parent algorithm. Specifically, a power SCA run with a CPA distinguisher and the Hamming distance leakage function, on an unprotected SW implementation of SKINNY-128, has shown that the minimum traces to discover (MTD) is only 80 traces. This means that only 80 traces are required to attack an unmasked SKINNY-128, although a masked version could not be successfully attacked with 1000 traces [115]. However, Romulus' proposal does not integrate masking to protect against

an SCA. Furthermore, a power SCA mounted on an HW implementation of SKINNY with a Hamming distance model showed a success rate of close to 100% with only 60 traces [167]. However, masking scheme implementation on SKINNY has shown an increase in the MTD to more than 1000 traces [115]. But, 1000 traces is not much of a deterrent with today's state-of-the-art computers and capture equipment because we have shown capabilities to mount SCAs with over 100 000 traces [102]. Thus, it still goes to show that Romulus implementations will need to be coupled with a countermeasure to be resistant to a power SCA.

8.2.4.6 Sparkle (Esc/Schwaemm)

The Sparkle proposal to the NIST LWC is a family of permutations closely related to the block cipher SPARX but with a fixed key and wider block size. The submission comprises the hash functions Esch256 and Esch384, based on the permutation family SPARKLE384 and SPARKLE512, respectively, which produce digests of 256 bits and 384 bits, respectively, and yield security levels of 128 bits and 192 bits respectively. The AEAD cipher family proposed is Schwaemm. The main implementation within the family is Schwaemm256-128, which accepts a key of length 128 bits, a nonce of length 256 bits, and produces a tag of length 128 bits. The encryption construction accepts the plaintext and outputs the ciphertext. Three other variants with different key, nonce, and tag lengths are proposed: 128-128, 192-192, and 256-256 [132].

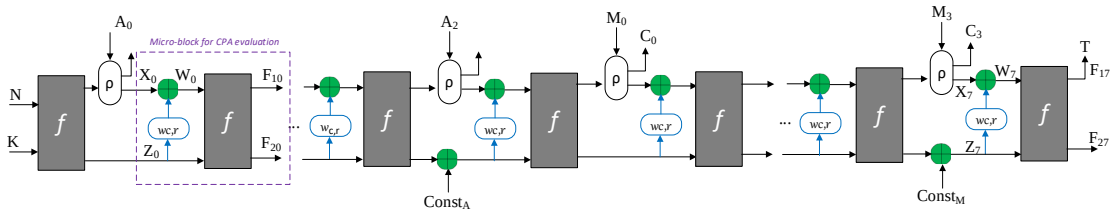


Fig. 45. Schwaemm AEAD construction with 3 associated data blocks and 4 message blocks, showing the addition of the whitening block vs. Beetle [132]

Fig. 45 represents Schwaemm authenticated encryption construction with 3 associated data blocks and 4 message blocks, showing the addition of the whitening block vs. Beetle. The function ***f*** represents one of the permutations Sparkle256$_s$, Sparkle384$_s$, or Sparkle512$_s$; s represents the number of steps in the permutation, ρ is the combined feedback function and $w_{c,r}$ is the whitening function as defined in [132].

The Schwaemm AEAD algorithm is based on a modified version of the Beetle mode for authenticated encryptions. Beetle is based on a duplexed sponge that provides additional security by using combined feedback to create a difference between the ciphertext output and the input of the permutation calls [133]. One of the main differences between Beetle and Schwaemm is that Schwaemm makes use of rate whitening, which consists of XORing the capacity to the rate before the permutation starts, as shown in Fig. 45. However, half of the branches in the state are not modified. Another deviation from Beetle is making the Schwaemm key length the same as the capacity, which alters how the tag is handled.

Despite the differences between Beetle and Schwaemm and given that half of the branches in the state are identical, the security of the Schwaemm algorithm follows the security of the underlying cryptographic algorithms from which it is derived.

Specifically, the security of Sparkle is based on the security of sponge-based hashing and the Beetle mode. The differences mentioned above have no impact on the potential relation between the leaked trace and the encryption key. Thus, most vulnerabilities observed on Beetle still apply to Schwaemm [133].

Other vulnerabilities of Schwaemm to power SCA

The addition of the whitening function to Schwaemm does not change leakage prevention under the CCAL1 and CIL1 security hypothesis. Thus, the resistance to SPA is maximum as with Beetle. However, Beetle is shown to be vulnerable to DPA under the scenario defined in section 8.2.4.4. Therefore, the Schwaemm algorithm will also be vulnerable to DPA when the capacity recovery step is changed to accommodate the inclusion of the combined feedback function ρ. Thus, instead of recovering the capacity straight up, we will need to perform the inverse combined feedback function to recover the capacity and then perform the inverse permutation to uncover the key *K*. Table 13 summarizes the difference between Beetle and Schwaemm DPA vulnerability under the CCAL1 and CIL1 security games.

Table 13 - Difference between Beetle and Schwaemm scenarios to uncover DPA vulnerability

Attack steps	**Beetle [125]**	**Schwaemm**
Step 1	Define a fix nonce and ephemeral key K*	Same as Beetle
Step 2	Generate multiple plaintexts blocks M_1 or multiple ciphertexts C_1	Same as Beetle

Step 3	Uncover the capacity section along with the plaintext M_1 or ciphertext C_1,	Perform inverse feedback function, then uncover the capacity section along with the plaintext M_1 or ciphertext C_1,
Step 4	Then perform the inverse permutation to uncover the key K	Same as Beetle

8.2.4.7 TinyJambu

TinyJambu is a family of AEAD ciphers derived from Jambu that comprises three key size options: 256 bits, 192 bits, and 128 bits. They all feature a 128-bit keyed permutation, a message block size of 32 bits, and a state size of 128 bits, as shown in Fig. 46 [134].

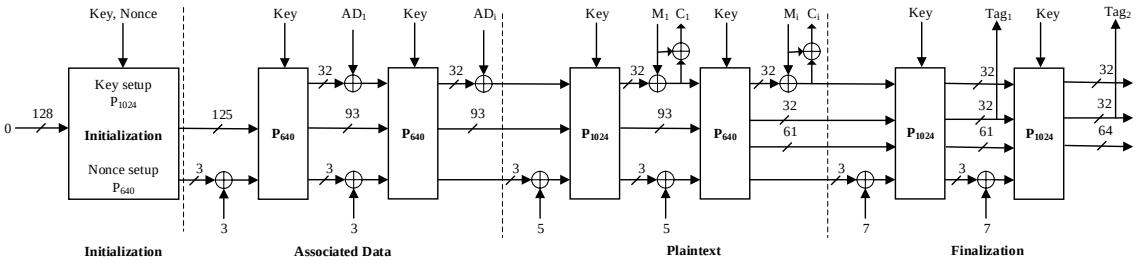


Fig. 46. TinyJambu AEAD cipher, indicating the number of rounds of each permutation [134]

StateUpdate(S, K, i):

$$feedback = s_0 \oplus s_{47} \oplus \left(\sim(s_{70} \& s_{85})\right) \oplus s_{91} \oplus k_{i \bmod klen}$$

For j from 0 to 126: $s_j = s_{j+1}$

$$s_{127} = feedback$$

end

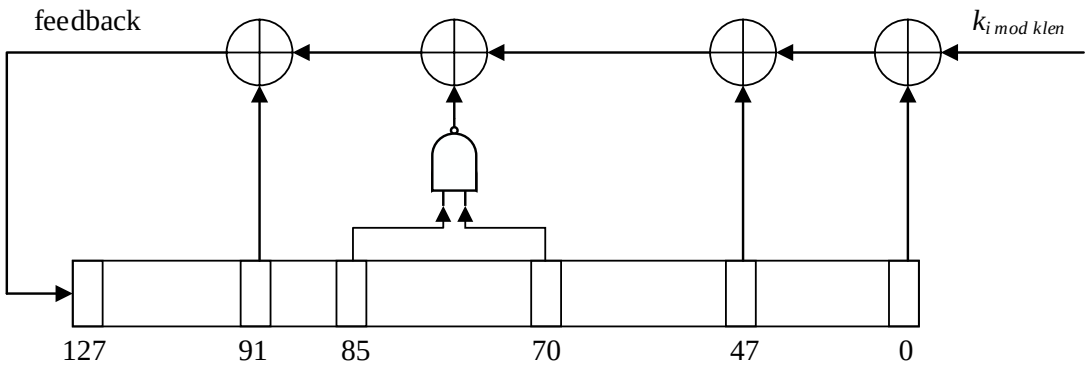


Fig. 47. TinyJambu keyed permutation algorithm (top), and graphical feedback implementation, with the nonlinear feedback shift register (bottom) [134]

TinyJambu constructs features of the 128-bit keyed permutation $\mathbf{P_n}$ at every step of its operation: initialization, associated data processing, plaintext processing, and tag generation steps. However, the number of permutation rounds, **n**, varies for each step. The nonlinear feedback shift register (NLFSR) and the elementary state update function (Fig. 47), are executed n times for a permutation $\mathbf{P_n}$. In 32-bit processors commonly used in IoT devices, 32 rounds of permutations can be implemented in parallel. Additionally, in a typical HW implementation, the key, nonce, and associated data are input on a 32-bit bus width to match the algorithm block size.

Scheme for attacking TinyJambu

Thus, implementations of unprotected TinyJambu with a block size of 32 bits [135] require the key to be accepted during the initialization phase in at least 4 words of 32 bits each maximum. The process of accepting and storing the 32-bit data and then implementing parallel computations of 32 feedback bits with the NLFSR will generate power consumption that a malicious agent can exploit to run CPA. In the worst case, the

key search space is reduced to a complexity of $4x2^{32}$, meaning 17 billion key guesses are needed to fully uncover all 128 bits of the encryption key. With such a reduction, a traditional CPA can be carried out (with modern computers), analogous to the key search space reduction from 2^{128} to $16x2^{8}$ of attacks on AES128 implementations [21]. Whether the computational complexity of such computations will result in a timely uncovering of the secret key is left to the next steps of this work. For a case of high-frequency implementation where the NLFSR computes 1 feedback bit per clock cycle, the algorithm implementation will additionally be vulnerable to SPA. If only one feedback bit is computed by the NLSFR in each clock cycle, the power consumption of the device will be different whether the feedback bit computed in Fig. 47 results in a 1 or a 0. The computation result of the most significant bit (MSB) will then create a discernable power consumption difference that can be visually analyzed by the malicious agent. Such SPA weakness, which borrows similarities to the conditional jump weakness in a data encryption standard (DES) algorithm and demonstrated in [9], allows the malicious agent to uncover the full state one bit at a time. Then, the full key can be deducted with the reverse computation of the initialization steps.

In a nutshell, TinyJambu implementations, and particularly its initialization phase, are vulnerable to CPA with key search space reduced from 2^{128} to $4x2^{32}$ and/or bit-by-bit simple power analysis attacks on its state when the feedback is computed one bit at a time. The above vulnerabilities are ubiquitous because the algorithm construct does not integrate any SCA countermeasure, such as masking or hiding. This thus makes it susceptible to leaking information that can be easily analyzed with first-order DPA to

uncover secret information. In fact, [135] shows an unprotected implementation of TinyJambu, on which an experiment with 10,000 traces yielded a test vector leakage assessment (TVLA) higher than 5. This is above the threshold of 4.5 widely accepted as the limit to which an implementation said is considered secure. This indicates that the implementation of an insecure TinyJambu leaks identifiable information with a probability greater than 99.999%.

8.2.5 Summary of Power Side Channel Attacks Vulnerabilities

The practical assessment of power SCA vulnerabilities that must be considered in evaluating the security of the seven lightweight ciphers is summarized in Table 14. For each cipher, we have shown the proposed integrated SCA countermeasure and our assessment of the residual vulnerabilities a malicious agent could exploit to extract secret information from the device. Most information is supported by prior art demonstrated with proven practical experiments, while others are novel concepts developed and demonstrated theoretically based on well-known general art concepts on power side channel cryptanalysis.

Table 14 - Residual vulnerability assessment of LWC finalist candidates

Ciphers	Residual vulnerabilities
Ascon **Type: Block cipher** **Key size: 128**	Round reduced (7 out of 12) implementations are vulnerable to attacks [121]. Ascon is not considered secured under the CCAmL1 security game. Without DPA-protected implementation of the verification phase [125], it is possible to successfully attack secure bootloading applications [127] by estimating valid messages without knowledge of the encryption key [128]. The attack on the state can be reduced to 64-bit operations, therefore reducing the search from 2^{320} to $5x2^{64}$.
GIFT-COFB **Type: Block cipher (GIFT-128)** **Key size: 128**	GIFT-COFB is vulnerable to CPA on a reduced number of rounds (11 vs. 40). However, the authors claim that 40 round implementation is resistant to DPA [113]. GIFT-128 looks like a larger version of PRESENT, thus vulnerabilities of PRESENT can be present here as well. GIFT S-box is susceptible to CPA when assessed with the transparency order (TO) metric[167].
Grain-128AEADv2	Grain-128AEADv2 is vulnerable to a power SCA with the Hamming distance model [117]. It has been demonstrated that one can construct a fast and automated process through Z3, a

Ciphers	Residual vulnerabilities
AEAD stream cipher Key size:128	publicly available satisfiability modulo theory (SMT) solver, with the leakage model and the publicly available keystream, that lead to key recovery in a few seconds [117]. Grain128AEADv2 is not resistant to fault attacks. The authors expect the users to implement protection mechanisms.
PHOTON-Beetle Authenticated Encryption and Hash Family Key size: 128	The message processing section shows residual vulnerability against DPA under CCAL1 and CIL1 [125]. Targeting S-box with an increased number of traces may lead to the recovery of secret information.
Romulus Type: Tweakable Block Ciper (SKINNY) Key size: 128	Romulus is vulnerable to the same CPA as SKINNY, with a leak function defined as the Hamming distance of the input/output of the target round. A power SCA mounted on an HW implementation of SKINNY with a Hamming distance model showed a success rate of close to 100% with approximately 60 traces only [167].
SPARKLE (SCH WAEMM and ESCH Type: block cipher Key size: 128	Sparkle and Beetle share similar residual vulnerabilities to a power SCA. Sparkle is vulnerable to a power SCA under the security game CCAL1 and CIL1. The main difference lies in the fact that instead of recovering the capacity straight up as with Beetle [125], we need to perform the inverse combined feedback function to recover the capacity, then perform the inverse permutation to uncover the key K, with Schwaemm.
TinyJambu Type: block cipher Key sizes: 128, 192, 256	An unprotected implementation of TinyJambu yielded a TVLA higher than 5, which is above the threshold of 4.5 [135]. TinyJambu implementations, and particularly its initialization phase, are vulnerable to CPA with key search space reduced from 2^{128} to $4x2^{32}$, and/or bit by bit simple power analysis attacks on its state when the feedback is computed one bit at a time.

Ascon, Sparkle, and PHOTON-Beetle security vulnerability can generally be assessed with the security assumptions CCAmL1 and CCAL1/CIL1, respectively. However, the security vulnerability of GIFT-COFB, Grain, Romulus, and TinyJambu can be evaluated more straightforwardly with proposed leakage functions or publicly available leakage models (Hamming distance model). These latter four ciphers' security vulnerability can also be evaluated with a solver (satisfiability modulo theory) or with a more computer-intensive approach that consists of significantly increasing the number of traces collected to launch the attack.

We can further note that the ISAP, Elephant, and Xoodyak modes of operation provide built-in approaches to preventing side channel attacks against algorithm implementations. For instance, one of the most powerful tools used in power SCA, DPA, operates by accumulating information on the secret key by measuring the power consumption of the

device during multiple encryption operations on different data. To counter this, ISAP has integrated a sponge-based rekeying in the encryption and MAC parts, which generates a fresh key for each new input. Doing so significantly decreases the vulnerability of ISAP implementations against a power SCA 167. This is demonstrated in 167, where it is shown that this out-of-the-box security meets the highest security level defined by the authors, which is CCAmL2.

9 SUMMARY AND CONCLUSIONS

9.1 Conclusion

This research characterized the implementations of cryptographic algorithms and uncovered residual vulnerabilities to power SCA (unintentionally not addressed by the algorithm implementation) and induced vulnerabilities (unintentionally created by the algorithm implementation). Previous research only broadly addressed the classification of general SCA with a case study applied to mobile devices. The work provided an in-depth analysis of PDN related countermeasures against power SCA. It benefits cryptographic IC designers while being a reference to easily identify vulnerabilities of cryptographic algorithm implementations. It also proposed a comprehensive evaluation of the residual vulnerabilities of the algorithm finalists of the NIST IoT lightweight cipher competition and proposed frameworks to launch power SCA against some of them.

The study on PDN related countermeasures has analyzed the impact of IVR, noise injection, OPDs, and circuit impedance on the ability of a cryptographic system PDN to reduce the amount of leaked identifiable information in a remote side channel attack scheme. The prior art narrowly focused on IVR and noise injection countermeasures against local attacks, with a physical presence. However, this study showed that remote attacks with traces captured at the IC power grid are significantly less impacted by IVR and OPD. The proximity and low impedance of the remote trojan IP to the victim are great security vulnerabilities, as it is shown that it requires fewer traces to uncover the secret key than a locally carried attack that captures the traces farther away on the system board.

However, it was demonstrated that noise injection constitutes an effective countermeasure to remote SCA as it increases the MTD by 37x, compared to 1.3x for OPDs increase. Additionally, a local attack requires 2.3x fewer traces to discover the secret key than a remote attack. Considering circuit loop impedance as a factor for remote vs. local attack analysis, which is a novel art, circuit impedance alterations, including IVR, are not effective at reducing the correlation between the measured traces and the encryption key. This PDN based countermeasure cryptanalysis was performed on the full-size cipher AES, but we have also characterized lightweight ciphers for power SCA residual vulnerabilities.

Power side channel attacks are of great concern on IoT devices because malicious agents have physical access to the device and thus can run cryptanalysis algorithms after the products are deployed. The finalists selected by NIST at the LWC competition each have their residual vulnerabilities, of which we brought to light a few relevant ones. The expectation is that this comprehensive study will be useful to SCA vulnerability testers/analyzers. Furthermore, these finalist ciphers or related variants have been previously been proposed and used in applications. Therefore, future IoT IC designers can leverage this work to evaluate the resilience of their products during the design phase.

9.2 Research contributions and publications

The contribution to research is classified into five areas as shown in Table 15.

Table 15: Contributions to research

Area	Contributions	Existing literature	Status
Contribution to knowledge: SCA on AES implementations	Attack on AES first round with experiments. Definition of the power estimation.	Attacks on AES last round	Complete
Improvement of prior concepts: Correlation Power Analysis	Reformulation for non-exhaustive key searches: convergence of correlation coefficients	Formulated only for exhaustive key search: max correlations coefficients correspond to the candidate key	Complete
Contribution to knowledge: Kullback-Leibler Rank	Introduction of Kullback-Leibler Rank distinguisher	Kullback-Leibler divergence as a distinguisher	Complete
Contribution to knowledge: Characterization of algorithms implementations:	Uncover residual vulnerabilities Uncover added vulnerabilities	Classification of general SCA: a case study of mobile devices. [1] Study of popular SCA and latest countermeasures. [84]	Complete
Contribution to knowledge: Analysis of the impact of PDN on side channel leakage	Demonstration of the effectiveness of remote attacks on PDN based countermeasures designed for local attacks	None is known to our knowledge	Complete
Contribution to knowledge: Proposing means to identify vulnerabilities to power SCA in NIST LWC finalists	Identify vulnerabilities and propose leakage functions needed to perform CPA on those lightweight ciphers	Surveys of the comparative studies of SCA on multiple lightweight ciphers [156][157][163][166][167]	Complete

On the publications front, the articles shown in Table 16 were published or submitted at the respective outlets.

Table 16: Publications

Publication Title/Proposal	Type	Conference/Journal	Timeline
Residual Vulnerabilities to Power Side Channel Attacks of Lightweight Ciphers Cryptography Competition Finalists	Journal	IET Computers and Digital Techniques Wiley 2023	Accepted for publication, April 2023
Power Side Channel Attacks of AES FPGA Implementation with Experimental Results using Full Keys	Conference	IEEE Design and Test of Integrated Micro and Nano Systems 2021	Published - Spring 2021

Remote vs. Local Power Side-Channel Attack Against On-Package Decoupling Capacitors, Noise Injection, or Power Delivery Network-Based Countermeasures	Journal	IEEE Transactions on Dependable and Secure Computing	Original submission: Sept 2022. Revised submission: May 2023

9.3 Future Work

Follow up work to this research could be applying a combination of the findings of the last two chapters of this thesis. One could determine the vulnerability to power SCA of implementations of NIST LWC finalists to remote attacks.

And for practical experiments, one could implement the winner of the NIST LWC competition, Ascon, in an FPGA and determine the vulnerability of the PDN to remote power SCA.

References

[1] Spreitzer, Raphael, Veelasha Moonsamy, Thomas Korak, and Stefan Mangard. "Systematic classification of side-channel attacks: A case study for mobile devices." *IEEE Communications Surveys & Tutorials* 20, no. 1 (2017): 465-488.

[2] Brier, Eric, Christophe Clavier, and Francis Olivier. "Optimal Statistical Power Analysis." IACR Cryptology ePrint Archive 2003 (2003): 152.

[3] De Mulder, Elke, Pieter Buysschaert, S. B. Ors, Peter Delmotte, Bart Preneel, Guy Vandenbosch, and Ingrid Verbauwhede. "Electromagnetic analysis attack on an FPGA implementation of an elliptic curve cryptosystem." In *EUROCON 2005-The International Conference on" Computer as a Tool"*, vol. 2, pp. 1879-1882. IEEE, 2005.

[4] Carlier, Vincent, Hervé Chabanne, Emmanuelle Dottax, and Hervé Pelletier. "Electromagnetic side channels of an FPGA implementation of AES." In *CRYPTOLOGY EPRINT ARCHIVE, REPORT 2004/145*. 2004.

[5] Messerges, Thomas S., Ezzat A. Dabbish, and Robert H. Sloan. "Examining smart-card security under the threat of power analysis attacks." IEEE Transactions on Computers 51.5 (2002): 541-552.

[6] Messerges, Thomas S., Ezzy A. Dabbish, and Robert H. Sloan. "Investigations of Power Analysis Attacks on Smartcards." Smartcard 99 (1999): 151-161.

[7] Socha, Petr, Jan Brejník, and Matěj Bartik. "Attacking AES implementations using correlation power analysis on ZYBO Zynq-7000 SoC board." 2018 7th Mediterranean Conference on Embedded Computing (MECO). IEEE, 2018.

[8] Spreitzer, Raphael, et al. "Systematic classification of side-channel attacks: a case study for mobile devices." IEEE Communications Surveys & Tutorials 20.1 (2017): 465-488.

[9] Kocher, Paul, Joshua Jaffe, and Benjamin Jun. "Differential power analysis." Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1999.

[10] Zhao, Mark, and G. Edward Suh. "FPGA-based remote power side-channel attacks." 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018.

[11] R. Lumbiarres-López, M. López-García and E. Cantó-Navarro, "Hardware Architecture Implemented on FPGA for Protecting Cryptographic Keys against Side-Channel Attacks," in IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 5, pp. 898-905, 1 Sept.-Oct. 2018, doi: 10.1109/TDSC.2016.2610966.

[12] Zode, Pravin, and Raghavendra Deshmukh. "Side channel attack resistant architecture for elliptic curve cryptosystem." Cyber-Physical Systems 4.4 (2018): 205-215.

[13] Macé, François, François-Xavier Standaert, and Jean-Jacques Quisquater. "Information theoretic evaluation of side-channel resistant logic styles." International Workshop on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg, 2007.

[14] Yang, Shengqi, et al. "Power attack resistant cryptosystem design: A dynamic voltage and frequency switching approach." Design, Automation and Test in Europe. IEEE, 2005.

[15] Sengupta, Abhrajit, et al. "Logic Locking with Provable Security Against Power Analysis Attacks." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2019).

[16] Pramstaller, Norbert, Elisabeth Oswald, Stefan Mangard, Frank K. Gürkaynak, and Simon Häne. A Masked AES ASIC Implementation. na, 2004.

[17] Lumbiarres-Lopez, Ruben, Mariano López-García, and Enrique Canto-Navarro. "Hardware architecture implemented on FPGA for protecting cryptographic keys against side-channel attacks." IEEE Transactions on Dependable and Secure Computing 15, no. 5 (2016): 898-905.

[18] Niu, Yongchuan, Jiawei Zhang, An Wang, and Caisen Chen. "An efficient collision power attack on AES encryption in edge computing." IEEE Access 7 (2019): 18734-18748.

[19] Alasad, Qutaiba, Jiann-Shuin Yuan, and Yu Bi. "Logic locking using hybrid CMOS and emerging SiNW FETs." Electronics 6, no. 3 (2017): 69.

[20] Alasad, Qutaiba, and Jiann Yuan. "Logic obfuscation against ic reverse engineering attacks using plgs." In 2017 IEEE International Conference on Computer Design (ICCD), pp. 341-344. IEEE, 2017.

[21] NewAE Technology Inc, ChipWhisperer® by, https://wiki.newae.com/Main_Page

[22] S. D. Putra, A. D. W. Sumari, I. Asrowardi, E. Subyantoro and L. M. Zagi, "First-Round and Last-Round Power Analysis Attack Against AES Devices," 2020 International Conference on Information Technology Systems and Innovation (ICITSI), Bandung - Padang, Indonesia, 2020, pp. 410-415, doi: 10.1109/ICITSI50517.2020.9264976.

[23] Longo, Jake, Daniel P. Martin, Luke Mather, Elisabeth Oswald, Benjamin Sach, and Martijn Stam. "How low can you go? Using side-channel data to enhance brute-force key recovery." IACR Cryptol. ePrint Arch. 2016 (2016): 609.

[24] Veyrat-Charvillon, Nicolas, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert. "An optimal key enumeration algorithm and its application to side-channel attacks." In International Conference on Selected Areas in Cryptography, pp. 390-406. Springer, Berlin, Heidelberg, 2012.

[25] Cai, Xiaomin, Renfa Li, Shijie Kuang, and Jinhui Tan. "An Energy Trace Compression Method for Differential Power Analysis Attack." IEEE Access 8 (2020): 89084-89092.

[26] Mather, Luke, Elisabeth Oswald, and Carolyn Whitnall. "Multi-target DPA attacks: Pushing DPA beyond the limits of a desktop computer." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2014.

[27] NIST, “Lightweight Cryptography, round 2 Candidates.” https://csrc.nist.gov/projects/lightweight-cryptography/round-2-candidates

[28] CSCR, NIST, “Applications and standardization of Light Cryptography”. https://csrc.nist.gov/CSRC/media/Presentations/Applications-and-Standardization-of-Lightweight-Cr/images-media/Talk-SAC-SummerSchool-meltem-Aug2018.pdf

[29] Patranabis, Sikhar, Jakub Breier, Debdeep Mukhopadhyay, and Shivam 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), pp. 25-32. IEEE, 2017.

[30] Moradi, Amir, David Oswald, Christof Paar, and Pawel Swierczynski. "Side-channel attacks on the bitstream encryption mechanism of Altera Stratix II: facilitating black-box analysis using software reverse-engineering." In *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, pp. 91-100. 2013.

[31] Mahmoud, Dina, and Mirjana Stojilović. "Timing violation induced faults in multi-tenant FPGAs." In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1745-1750. IEEE, 2019.

[32] Singh, Arvind, Nikhil Chawla, Jong Hwan Ko, Monodeep Kar, and Saibal Mukhopadhyay. "Energy efficient and side-channel secure cryptographic hardware for IoT-edge nodes." *IEEE Internet of Things Journal* 6, no. 1 (2018): 421-434.

[33] Zick, Kenneth M., Meeta Srivastav, Wei Zhang, and Matthew French. "Sensing nanosecond-scale voltage attacks and natural transients in FPGAs." In *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, pp. 101-104. 2013.

[34] Federal Information Processing Stands Publication 197: Advanced Encryption Standard (AES), November 26, 2001

[35] Tim Good, Mohammed Benaissa, “AES on FPGA from the fastest to the smallest.” CHES ’05: Proceedings of the 7th International Workshop on Cryptographic Hardware and Embedded systems. Pages 427-440, 2005

[36] https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm

[37] Takemoto, Shu, Yusuke Nozaki, and Masaya Yoshikawa. "Statistical Power Analysis for IoT Device Oriented Encryption with Glitch Canceller." 2019 IEEE 11th International Workshop on Computational Intelligence and Applications (IWCIA). IEEE, 2019.

[38] Moradi, Amir, and Tobias Schneider. "Side-channel analysis protection and low-latency in action." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2016.

[39] Bhasin, Shivam, Tarik Graba, Jean-Luc Danger, and Zakaria Najm. "A look into SIMON from a side-channel perspective." In 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 56-59. IEEE, 2014.

[40] Chawla, Nikhil, Arvind Singh, Nael Mizanur Rahman, Monodeep Kar, and Saibal Mukhopadhyay. "Extracting side-channel leakage from round unrolled implementations of lightweight ciphers." In 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 31-40. IEEE, 2019.

[41] Beaulieu, Ray, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. "The SIMON and SPECK lightweight block ciphers." In Proceedings of the 52nd Annual Design Automation Conference, pp. 1-6. 2015.

[42] Borghoff, Julia, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander et al. "PRINCE–a low-latency block cipher for pervasive computing applications." In International conference on the theory and application of cryptology and information security, pp. 208-225. Springer, Berlin, Heidelberg, 2012.

[43] Paar, C. and Pelzl, J. “Understanding Cryptography”, 2nd corrected printing, Springer, 2010

[44] Federal Information Processing Stands Publication 202: SHA-3 Standard: permutation-based Hash and extendable output functions, August 2015.

[45] Iokibe, Kengo, Tetsuo Amano, Kaoru Okamoto, and Yoshitaka Toyota. "Equivalent circuit modeling of cryptographic integrated circuit for information security design." IEEE transactions on electromagnetic compatibility 55, no. 3 (2013): 581-588.

[46] S. D. Dhia, M. Ramdani, and E. Sicard, Electromagnetic Compatibility of Integrated Circuits. New York, NY, USA: Springer-Verlag, 2006.

[47] Tsukioka, Akihiro, Karthik Srinivasan, Shan Wan, Lang Lin, Ying-Shiun Li, Norman Chang, and Makoto Nagata. "A Fast Side-Channel Leakage Simulation Technique Based on IC Chip Power Modeling." IEEE Letters on Electromagnetic Compatibility Practice and Applications 1, no. 4 (2019): 83-87.

[48] Yu, Weize, Orhun Aras Uzun, and Selçuk Köse. "Leveraging on-chip voltage regulators as a countermeasure against side-channel attacks." In Proceedings of the 52nd Annual Design Automation Conference, pp. 1-6. 2015.

[49] Yu, Weize, and Selçuk Köse. "Exploiting voltage regulators to enhance various power attack countermeasures." IEEE Transactions on emerging topics in Computing 6, no. 2 (2016): 244-257.

[50] Yang, Shengqi, Wayne Wolf, Narayanan Vijaykrishnan, Dimitrios N. Serpanos, and Yuan Xie. "Power attack resistant cryptosystem design: A dynamic voltage and frequency switching approach." In Design, Automation and Test in Europe, pp. 64-69. IEEE, 2005.

[51] Kar, Monodeep, Arvind Singh, Sanu Mathew, Anand Rajan, Vivek De, and Saibal Mukhopadhyay. "8.1 Improved power-side-channel-attack resistance of an AES-128 core via a security-aware integrated buck voltage regulator." In 2017 IEEE International Solid-State Circuits Conference (ISSCC), pp. 142-143. IEEE, 2017.

[52] Singh, Arvind, Monodeep Kar, Sanu K. Mathew, Anand Rajan, Vivek De, and Saibal Mukhopadhyay. "Improved power/EM side-channel attack resistance of 128-bit aes engines with random fast voltage dithering." IEEE Journal of Solid-State Circuits 54, no. 2 (2018): 569-583.

[53] Kar, Monodeep, Arvind Singh, Sanu K. Mathew, Anand Rajan, Vivek De, and Saibal Mukhopadhyay. "Reducing power side-channel information leakage of AES engines using fully integrated inductive voltage regulator." IEEE Journal of Solid-State Circuits 53, no. 8 (2018): 2399-2414.

[54] Zick, Kenneth M., Meeta Srivastav, Wei Zhang, and Matthew French. "Sensing nanosecond-scale voltage attacks and natural transients in FPGAs." In Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays, pp. 101-104. 2013

[55] Gnad, Dennis RE, Fabian Oboril, Saman Kiamehr, and Mehdi B. Tahoori. "An experimental evaluation and analysis of transient voltage fluctuations in FPGAs." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 26, no. 10 (2018): 1817-1830.

[56] Gravellier, Joseph, Jean-Max Dutertre, Yannick Teglia, Philippe Loubet Moundi, and Francis Olivier. "Remote side-channel attacks on heterogeneous soc." In International Conference on Smart Card Research and Advanced Applications, pp. 109-125. Springer, Cham, 2019.

[57] Örs, Sıddıka Berna, Elisabeth Oswald, and Bart Preneel. "Power-analysis attacks on an FPGA–first experimental results." International Workshop on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg, 2003.

[58] Standaert, O-X., Eric Peeters, Gaël Rouvroy, and J-J. Quisquater. "An overview of power analysis attacks against field programmable gate arrays." Proceedings of the IEEE 94, no. 2 (2006): 383-394.

[59] Brier, Eric, Christophe Clavier, and Francis Olivier. "Correlation power analysis with a leakage model." *International workshop on cryptographic hardware and embedded systems*. Springer, Berlin, Heidelberg, 2004.

[60] Veyrat-Charvillon, Nicolas, and François-Xavier Standaert. "Mutual information analysis: how, when and why?." International Workshop on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg, 2009.

[61] Kullback, Solomon, and Richard A. Leibler. "On information and sufficiency." The annals of mathematical statistics 22.1 (1951): 79-86.

[62] Fisher, R. (1925). Theory of Statistical Estimation. Mathematical Proceedings of the Cambridge Philosophical Society, 22(5), 700-725. doi:10.1017/S0305004100009580

[63] Chong, Kwen-Siong, K. Z. L. Ne, Weng-Geng Ho, Nan Liu, A. H. Akbar, Bah-Hwee Gwee, and Joseph S. Chang. "Counteracting differential power analysis: Hiding encrypted data from circuit cells." In 2015 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC), pp. 297-300. IEEE, 2015.

[64] Diehl, William, Abubakr Abdulgadir, Jens-Peter Kaps, and Kris Gaj. "Comparing the cost of protecting selected lightweight block ciphers against differential power analysis in low-cost FPGAs." Computers 7, no. 2 (2018): 28.

[65] Sengupta, Abhrajit, Bodhisatwa Mazumdar, Muhammad Yasin, and Ozgur Sinanoglu. "Logic locking with provable security against power analysis attacks." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 39, no. 4 (2019): 766-778.

[66] https://www.hpcwire.com/2019/08/19/ayar-labs-to-demo-photonics-chiplet-in-fpga-package-at-hot-chips/

[67] https://blogs.intel.com/psg/intel-releases-royalty-free-high-performance-aib-interconnect-standard-to-spur-industrys-chiplet-adoption-and-grow-the-ecosystem/

[68] Kison, Christian, Jürgen Frinken, and Christof Paar. "Finding the AES bits in the haystack: Reverse engineering and sca using voltage contrast." In International Workshop on Cryptographic Hardware and Embedded Systems, pp. 641-660. Springer, Berlin, Heidelberg, 2015.

[69] Mangard, Stefan, Norbert Pramstaller, and Elisabeth Oswald. "Successfully attacking masked AES hardware implementations." In International workshop on cryptographic hardware and embedded systems, pp. 157-171. Springer, Berlin, Heidelberg, 2005.

[70] 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," in IEEE Journal of Solid-State Circuits, vol. 53, no. 8, pp. 2399-2414, Aug. 2018, doi: 10.1109/JSSC.2018.2822691.

[71] D. Das, S. Maity, S. B. Nasir, S. Ghosh, A. Raychowdhury and S. Sen, "High efficiency power side-channel attack immunity using noise injection in attenuated signature domain," 2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2017, pp. 62-67, doi: 10.1109/HST.2017.7951799.

[72] J. Lagasse, C. Bartoli and W. Burleson, "Combining Clock and Voltage Noise Countermeasures Against Power Side-Channel Analysis," 2019 IEEE 30th International Conference on Application-specific Systems, Architectures and Processors (ASAP), 2019, pp. 214-217, doi: 10.1109/ASAP.2019.00009.

[73] McEvoy, Robert, Michael Tunstall, Colin C. Murphy, and William P. Marnane. "Differential power analysis of HMAC based on SHA-2, and countermeasures." In International Workshop on Information Security Applications, pp. 317-332. Springer, Berlin, Heidelberg, 2007.

[74] De Meyer, Lauren, Oscar Reparaz, and Begül Bilgin. "Multiplicative masking for AES in hardware." IACR Transactions on Cryptographic Hardware and Embedded Systems (2018): 431-468.

[75] Sasdrich, Pascal, Begül Bilgin, Michael Hutter, and Mark E. Marson. "Low-latency hardware masking with application to aes." IACR Transactions on Cryptographic Hardware and Embedded Systems (2020): 300-326.

[76] Moos, Thorben. "Static power SCA of sub-100 nm CMOS asics and the insecurity of masking schemes in low-noise environments." IACR Transactions on Cryptographic Hardware and Embedded Systems (2019): 202-232.

[77] Jaffe, Josh. "A first-order DPA attack against AES in counter mode with unknown initial counter." In International Workshop on Cryptographic Hardware and Embedded Systems, pp. 1-13. Springer, Berlin, Heidelberg, 2007.

[78] Kocher, Paul, Joshua Jaffe, and Benjamin Jun. "Introduction to differential power analysis and related attacks." (1998): 3.

[79] Chong, Kwen-Siong, Aparna Shreedhar, Ne Kyaw Zwa Lwin, Nay Aung Kyaw, Weng-Geng Ho, Chao Wang, Jun Zhou, Bah-Hwee Gwee, and Joseph S. Chang. "Side-channel-attack resistant dual-rail asynchronous-logic AES accelerator based on standard library cells." In 2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), pp. 1-7. IEEE, 2019.

[80] Nabeel, Mohammed, Mohammed Ashraf, Satwik Patnaik, Vassos Soteriou, Ozgur Sinanoglu, and Johann Knechtel. "2.5 D root of trust: Secure system-level integration of untrusted chiplets." IEEE Transactions on Computers 69, no. 11 (2020): 1611-1625.

[81] Nabeel, Mohammed, Mohammed Ashraf, Satwik Patnaik, Vassos Soteriou, Ozgur Sinanoglu, and Johann Knechtel. "An interposer-based root of trust: Seize the opportunity for secure system-level integration of untrusted chiplets." arXiv preprint arXiv:1906.02044 (2019).

[82] Awad, Amro, and Rujia Wang. "Guest Editors' Introduction to the Special Issue on Hardware Security." IEEE Transactions on Computers 69, no. 11 (2020): 1556-1557.

[83] Y. -S. Won, D. -G. Han, D. Jap, S. Bhasin and J. -Y. Park, "Non-Profiled Side-Channel Attack Based on Deep Learning Using Picture Trace," in IEEE Access, vol. 9, pp. 22480-22492, 2021, doi: 10.1109/ACCESS.2021.3055833.

[84] M. Taouil, A. Aljuffri and S. Hamdioui, "Power Side Channel Attacks: Where Are We Standing?," 2021 16th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS), 2021, pp. 1-6, doi: 10.1109/DTIS53253.2021.9505075.

[85] Prest T., Goudarzi D., Martinelli A., Passelègue A. (2019) Unifying Leakage Models on a Rényi Day. In: Boldyreva A., Micciancio D. (eds) Advances in Cryptology – CRYPTO 2019. CRYPTO 2019. Lecture Notes in Computer Science, vol 11692. Springer, Cham. https://doi.org/10.1007/978-3-030-26948-7_24

[86] Standaert FX., Malkin T.G., Yung M. (2009) A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In: Joux A. (eds) Advances in Cryptology - EUROCRYPT 2009. EUROCRYPT 2009. Lecture Notes in Computer Science, vol 5479. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-01001-9_26

[87] Batina, Lejla, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. "Mutual information analysis: a comprehensive study." Journal of Cryptology 24, no. 2 (2011): 269-291.

[88] Martin, Daniel P., et al. "Counting keys in parallel after a side channel attack." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2015.

[89] David, Liron, and Avishai Wool. "A bounded-space near-optimal key enumeration algorithm for multi-subkey side-channel attacks." Cryptographers' Track at the RSA Conference. Springer, Cham, 2017.

[90] Poussier, Romain, François-Xavier Standaert, and Vincent Grosso. "Simple key enumeration (and rank estimation) using histograms: an integrated approach." International Conference on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg, 2016.

[91] Mangard, Stefan, Thomas Popp, and Berndt M. Gammel. "Side-channel leakage of masked CMOS gates." In Cryptographers' Track at the RSA Conference, pp. 351-365. Springer, Berlin, Heidelberg, 2005.

[92] Suzuki, Daisuke, Minoru Saeki, and Tetsuya Ichikawa. "Random Switching Logic: A Countermeasure against DPA based on Transition Probability." IACR Cryptol. ePrint Arch. 2004 (2004): 346.

[93] Lerman, Liran, Gianluca Bontempi, and Olivier Markowitch. "Power analysis attack: an approach based on machine learning." International Journal of Applied Cryptography 3, no. 2 (2014): 97-115.

[94] Hospodar, Gabriel, Benedikt Gierlichs, Elke De Mulder, Ingrid Verbauwhede, and Joos Vandewalle. "Machine learning in side-channel analysis: a first study." Journal of Cryptographic Engineering 1, no. 4 (2011): 293.

[95] Chari, Suresh, Josyula R. Rao, and Pankaj Rohatgi. "Template attacks." In International Workshop on Cryptographic Hardware and Embedded Systems, pp. 13-28. Springer, Berlin, Heidelberg, 2002.

[96] Maghrebi, Houssem. "Deep Learning based Side Channel Attacks in Practice." IACR Cryptol. ePrint Arch. 2019 (2019): 578.

[97] Roy, Debapriya Basu, Shivam Bhasin, Sylvain Guilley, Annelie Heuser, Sikhar Patranabis, and Debdeep Mukhopadhyay. "Leak me if you can: Does tvla reveal success rate." Cryptology ePrint Archive, Report 2016/1152 (2016).

[98] Chen, Zhimin, and Yujie Zhou. "Dual-rail random switching logic: a countermeasure to reduce side channel leakage." In International Workshop on Cryptographic Hardware and Embedded Systems, pp. 242-254. Springer, Berlin, Heidelberg, 2006.

[99] Popp, Thomas, and Stefan Mangard. "Masked dual-rail pre-charge logic: DPA-resistance without routing constraints." In International Workshop on Cryptographic Hardware and Embedded Systems, pp. 172-186. Springer, Berlin, Heidelberg, 2005.

[100] Tiri, Kris, and Ingrid Verbauwhede. "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation." In Proceedings Design, Automation and Test in Europe Conference and Exhibition, vol. 1, pp. 246-251. IEEE, 2004.

[101] Kotipalli, Siva, Yong-Bin Kim, and Minsu Choi. "Asynchronous advanced encryption standard hardware with random noise injection for improved side-channel attack resistance." Journal of Electrical and Computer Engineering 2014 (2014).

[102] A. T. Mozipo and J. M. Acken, "Power Side Channel Attack of AES FPGA Implementation with Experimental Results using Full Keys," 2021 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), 2021, pp. 1-6, doi: 10.1109/DTS52014.2021.9497976.

[103]Proença, Paulo, and Ricardo Chaves. "Compact CLEFIA implementation on FPGAs." In 2011 21st International Conference on Field Programmable Logic and Applications, pp. 512-517. IEEE, 2011.

[104] Bogdanov, Andrey, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. "PRESENT: An ultra-lightweight block cipher." In International workshop on cryptographic hardware and embedded systems, pp. 450-466. Springer, Berlin, Heidelberg, 2007.

[105]Borghoff, Julia, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander et al. "PRINCE–a low-latency block cipher for pervasive computing applications." In International Conference on the Theory and Application of Cryptology and Information Security, pp. 208-225. Springer, Berlin, Heidelberg, 2012.

[106]Beaulieu, Ray, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. "The SIMON and SPECK lightweight block ciphers." In Proceedings of the 52nd Annual Design Automation Conference, pp. 1-6. 2015.

[107]So, Jaewoo. "Deep learning-based cryptanalysis of lightweight block ciphers." Security and Communication Networks 2020 (2020).

[108]Guilley, Sylvain, Houssem Maghrebi, Youssef Souissi, Laurent Sauvage, and Jean-Luc Danger. "Quantifying the quality of side channel acquisitions." COSADE, February (2011).

[109] Mangard, Stefan. "Hardware countermeasures against DPA–a statistical analysis of their effectiveness." In Cryptographers' Track at the RSA Conference, pp. 222-235. Springer, Berlin, Heidelberg, 2004.

[110]Singh, Arvind, Nikhil Chawla, Monodeep Kar, and Saibal Mukhopadhyay. "Energy efficient and side-channel secure hardware architecture for lightweight cipher SIMON." In 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 159-162. IEEE, 2018.

[111] Lightweight Cryptography, https://csrc.nist.gov/Projects/lightweight-cryptography/finalists

[112]https://cryptography.gmu.edu/athena/index.php?id=LWC

[113]Banik, Subhadeep, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. "GIFT: a small present." In International Conference on cryptographic hardware and embedded systems, pp. 321-345. Springer, Cham, 2017.

[114]Slpsk, Patanjali, Prasanna Karthik Vairam, Chester Rebeiro, and V. Kamakoti. "Karna: A gate-sizing based security aware EDA flow for improved power side-channel attack protection." In 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 1-8. IEEE, 2019.

[115]Ge, Jing, Yifan Xu, Ruiqian Liu, Enze Si, Ning Shang, and An Wang. "Power attack and protected implementation on lightweight block cipher SKINNY." In 2018 13th Asia Joint Conference on Information Security (AsiaJCIS), pp. 69-74. IEEE, 2018.

[116]Heuser, Annelie, Stjepan Picek, Sylvain Guilley, and Nele Mentens. "Lightweight ciphers and their side-channel resilience." IEEE Transactions on Computers 69, no. 10 (2017): 1434-1448.

[117]Baksi, Anubhab, Satyam Kumar, and Santanu Sarkar. "A New Approach For Side Channel Analysis On Stream Ciphers And Related Constructions." IEEE Transactions on Computers (2021).

[118]Zhang, Bin, Xinxin Gong, and Willi Meier. "Fast correlation attacks on Grain-like small state stream ciphers." IACR Transactions on Symmetric Cryptology (2017): 58-81.

[119]Todo, Yosuke, Takanori Isobe, Willi Meier, Kazumaro Aoki, and Bin Zhang. "Fast correlation attack revisited." In Annual International Cryptology Conference, pp. 129-159. Springer, Cham, 2018.

[120] M. Hell, et al, ”Grain-128AEADv2 – A lightweight AEAD stream cipher”, https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/grain-128aead-spec-final.pdf

[121]C. Dobraunig, M Eichlseder, F. Mendel, M. Schlaffer: “Ascon”, https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/ascon-spec-final.pdf

[122] Levi, Itamar, Davide Bellizia, David Bol, and François-Xavier Standaert. "Ask less, get more: Side-channel signal hiding, revisited." IEEE Transactions on Circuits and Systems I: Regular Papers 67, no. 12 (2020): 4904-4917.

[123] Yu, Weize, and Selçuk Köse. "A lightweight masked AES implementation for securing IoT against CPA attacks." IEEE Transactions on Circuits and Systems I: Regular Papers 64, no. 11 (2017): 2934-2944.

[124]Guo, Chun, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. "Towards low-energy leakage-resistant authenticated encryption from the duplex sponge construction." Cryptology ePrint Archive (2019).

[125]Bellizia, Davide, Olivier Bronchain, Gaëtan Cassiers, Vincent Grosso, Chun Guo, Charles Momin, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. "Mode-level vs. implementation-level physical security in symmetric cryptography." In Annual International Cryptology Conference, pp. 369-400. Springer, Cham, 2020.

[126]Z. Bao, et al. : ” PHOTON-Beetle Authenticated Encryption and Hash Family”, https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/photon-beetle-spec-final.pdf

[127]O’Flynn, C., Chen, Z.D.: Side channel power analysis of an AES-256 bootloader. In: CCECE, pp. 750–755. IEEE (2015)

[128]Berti, F., Pereira, O., Peters, T., Standaert, F.: On leakage-resilient authenticated encryption with decryption leakages. IACR Trans. Symmetric Cryptol. 2017(3), 271–293 (2017)

[129]C. Dobraunig, et al. : ”ISAP v2.0”, https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/isap-spec-final.pdf

[130]W. Unger, L. Babinkostova, M. Borowczak and R. Erbes, "Side-channel Leakage Assessment Metrics: A Case Study of GIFT Block Ciphers," 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2021, pp. 236-241, doi: 10.1109/ISVLSI51109.2021.00051.

[131]Li, H., Yang, G., Ming, J. et al. Transparency order versus confusion coefficient: a case study of NIST lightweight cryptography S-Boxes. Cybersecur 4, 35 (2021). https://doi.org/10.1186/s42400-021-00099-1

[132]C. Beierle, et al., ”Schwaemm and Esch: Lightweight Authenticated Encryption and Hashing using the Sparkle Permutation Family”, https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/sparkle-spec-final.pdf

[133] Chakraborti, Avik, Nilanjan Datta, Mridul Nandi, and Kan Yasuda. "Beetle family of lightweight and secure authenticated encryption ciphers." Cryptology ePrint Archive (2018).

[134] H. Wu, T. Huang, “TinyJambu: A Family of Lightweight Authenticated Encryption Algorithms (Version 2)”, https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/tinyjambu-spec-final.pdf

[135]Abdulgadir, Abubakr, Sammy Lin, Farnoud Farahmand, Jens-Peter Kaps, and Kris Gaj. "Side-channel Resistant Implementations of a Novel Lightweight Authenticated Cipher with Application to Hardware Security." In Proceedings of the 2021 on Great Lakes Symposium on VLSI, pp. 229-234. 2021.

[136] Joan Daemen, Seth Hoffert, Silva Lella, Michael Peeters, Gilles Van Assche and Ronny Van Keer: “Xoodyak, a lightweight cryptographic scheme.”

https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/xoodyak-spec-final.pdf

[137] T. Beyne, Y.L Chen, C. Dobraunig, B. Mennink: "Elephant v2", https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/elephant-spec-final.pdf

[138] S. Subhadeep, et al: "GIFT-COFB", https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/gift-cofb-spec-final.pdf

[139] C. Guo, T. Iwata, M. Khairallah, K. Minematsu, T. Peytin: "Romulus, v1.3", https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/romulus-spec-final.pdf

[140] Song, Seong-Sik, Seung-Wook Lee, Joonho Gil, and Hyungcheol Shin. "Simple wide-band metal-insulator-metal (MIM) capacitor model for RF applications and effect of substrate grounded shields." Japanese journal of applied physics 43, no. 4S (2004): 1746.

[141] Chromczak, Jeffrey, Mark Wheeler, Charles Chiasson, Dana How, Martin Langhammer, Tim Vanderhoek, Grace Zgheib, and Ilya Ganusov. "Architectural enhancements in intel® agilex™ fpgas." In Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp. 140-149. 2020.

[142] P. Pant and J. Zelman, "Understanding Power Supply Droop during At-Speed Scan Testing," 2009 27th IEEE VLSI Test Symposium, 2009, pp. 227-232, doi: 10.1109/VTS.2009.46.

[143] V. Van der Veen et al., 'Drammer deterministic rowhammer attacks on mobile platforms', *in proc. Conf. Comput. Commun. Security (CCS)*, Vienna, Austria, 2016 pp. 1657-1689

[144] Yaron Y, Falkner K, "FLUSH-RELOAD: A high resolution, low noise, L3 cache side channel attack." *In proc: USENIX Security Symp.* San Diego, CA, USA, 2014. Pp. 719-732

[145] O'Flynn C, "Fault injection using crowbars on embeded systems", *IACR Cryptology ePrint archive. Report 2016/810,* 2016. [Online]. Available: https://eprint.iacr.org/2016/810

[146] Khan M N, Rao A and Camtepe S, "Lightweight Cryptographic Protocols for IoT-Constrained Devices: A Survey," in IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4132-4156, 15 March15, 2021, doi: 10.1109/JIOT.2020.3026493.

[147] Fotovvat A, Rahman G M E, Vedaei S S and Wahid K A, "Comparative Performance Analysis of Lightweight Cryptography Algorithms for IoT Sensor Nodes," in IEEE Internet of Things Journal, vol. 8, no. 10, pp. 8279-8290, 15 May15, 2021, doi: 10.1109/JIOT.2020.3044526.

[148]Philip M A and Vaithiyanathan, "A survey on lightweight ciphers for IoT devices," 2017 International Conference on Technological Advancements in Power and Energy (TAP Energy), 2017, pp. 1-4, doi: 10.1109/TAPENERGY.2017.8397271.

[149]Spreitzer R, Moonsamy V, Korak T and Mangard S, "Systematic Classification of Side-Channel Attacks: A Case Study for Mobile Devices," *in IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 465-488, Firstquarter 2018, doi: 10.1109/COMST.2017.2779824.

[150]Granjal J, Monteiro E and Silva J Sá, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," in IEEE Communications Surveys & Tutorials, vol. 17, no. 3, pp. 1294-1312, third quarter 2015, doi: 10.1109/COMST.2015.2388550.

[151]Yang D, Qi WF, Chen HJ. Impossible differential attacks on the SKINNY family of block ciphers. IET Information Security. 2017 Nov 2;11(6):377-85.

[152]Liu S, Guan J, Hu B. Fault attacks on authenticated encryption modes for GIFT. IET Information Security. 2022 Jan;16(1):51-63.

[153]Randolph M, and Diehl W, "Power side-channel attack analysis: A review of 20 years of study for the layman." Cryptography 4, no. 2 (2020): 15.

[154]Mangard S, Oswald E, Standaert FX. One for all–all for one: unifying standard differential power analysis attacks. IET Information Security. 2011 Jun 1;5(2):100-10.

[155]Fei Y, Gong G, Gongye C, Mandal K, Rohit R, Xu T, Yi Y, Zidaric N. Correlation power analysis and higher-order masking implementation of WAGE. InSelected Areas in Cryptography: 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers 27 2021 (pp. 593-614). Springer International Publishing.

[156]Liu Z, Schaumont P, "Root-Cause Analysis of Power-Based Side-Channel Leakage in Lightweight Cryptography Candidates": https://csrc.nist.rip/presentations/2022/root-cause-analysis-of-power-based-side-channel-le; NIST 5th Lightweight Cryptography Workshop, (2022)

[157]Abduladir A, Haeussler R, Lin S, Kaps J P, Gaj K, "Side-Channel Resistant Implementation of Three Finalists of the NIST Lightweight Cryptography Standardization Process: Elephant, TinyJambu, and Xoodyak"; https://csrc.nist.rip/presentations/2022/side-channel-resistant-implementations-of-three-lw; NIST 5th Lightweight Cryptography Workshop, (2022)

[158]Unger W, Babinkostova L, Borowczak M, Erbes R, Srinath A, "TVLA, Correlation Power Analysis and Side-Channel Leakage Assessment Metrics"; https://csrc.nist.rip/presentations/2022/tvla-correlation-power-analysis-side-channel-leaka; NIST 5th Lightweight Cryptography Workshop, (2022)

[159]Nalla Anandakumar N. (2015). SCA Resistance Analysis on FPGA Implementations of Sponge Based MAC−PHOTON . In: Bica, I., Naccache, D., Simion, E. (eds) Innovative Security Solutions for Information Technology and Communications.

SECITC 2015. Lecture Notes in Computer Science(), vol 9522. Springer, Cham. https://doi.org/10.1007/978-3-319-27179-8_6

[160]Biryukov A., Dinu D., Großschädl J. (2016). Correlation Power Analysis of Lightweight Block Ciphers: From Theory to Practice. In: Manulis, M., Sadeghi, AR., Schneider, S. (eds) Applied Cryptography and Network Security. ACNS 2016. Lecture Notes in Computer Science(), vol 9696. Springer, Cham. https://doi.org/10.1007/978-3-319-39555-5_29

[161]Zhang J., Li L., Li Q., Zhao J., Liang X. (2021). Power Analysis Attack on a Lightweight Block Cipher GIFT. In: Liu, Q., Liu, X., Li, L., Zhou, H., Zhao, HH. (eds) Proceedings of the 9th International Conference on Computer Engineering and Networks . Advances in Intelligent Systems and Computing, vol 1143. Springer, Singapore. https://doi.org/10.1007/978-981-15-3753-0_55

[162]Samwel N, Daemen J, "DPA on hardware implementations of Ascon and Keyak", Proceedings of the Computing Frontiers Conference, May 2017, Pages 415–424 https://doi.org/10.1145/3075564.3079067

[163]Windarta S, Suryadi S, Ramli K, Pranggono B and Gunawan T S, "Lightweight Cryptographic Hash Functions: Design Trends, Comparative Study, and Future Directions," in IEEE Access, vol. 10, pp. 82272-82294, 2022, doi: 10.1109/ACCESS.2022.3195572.

[164]Batina L, Buhan I.R., Chmielewski L.M., Gunnarsdottir E., Jahandideh V., Stock T., Weissbart L.J.A., "Side-Channel Evaluation Report on Implementations of Several NIST LWC Finalists", https://repository.ubn.ru.nl/handle/2066/253567

[165]Miteloudi K, Chmielewski Ł, Batina L and Mentens N, "Evaluating the ROCKY Countermeasure for Side-Channel Leakage," 2021 IFIP/IEEE 29th International Conference on Very Large Scale Integration (VLSI-SoC), Singapore, Singapore, 2021, pp. 1-6, doi: 10.1109/VLSI-SoC53125.2021.9606973.

[166]W. Diehl, A. Abdulgadir, F. Farahmand, J. -P. Kaps and K. Gaj, "Comparison of cost of protection against differential power analysis of selected authenticated ciphers," 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), Washington, DC, USA, 2018, pp. 147-152, doi: 10.1109/HST.2018.8383904.

[167]Heuser A, Picek S, Guilley S, and Mentens N, "Lightweight ciphers and their side-channel resilience." IEEE Transactions on Computers 69, no. 10 (2017): 1434-1448.

[168]Nguyen ND, Bui DH, Tran XT. A Lightweight AEAD encryption core to secure IoT applications. In2020 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS) 2020 Dec 8 (pp. 35-38). IEEE.

[169]Rostampour S, Bagheri N, Bendavid Y, Safkhani M, Kumari S, Rodrigues JJ. An authentication protocol for next generation of constrained Iot systems. IEEE Internet of Things Journal. 2022 Jun 20;9(21):21493-504.

[170]De Santis F, Schauer A, Sigl G. ChaCha20-Poly1305 authenticated encryption for high-speed embedded IoT applications. InDesign, Automation & Test in Europe Conference & Exhibition (DATE), 2017 2017 Mar 27 (pp. 692-697). IEEE.

[171]J. S. Ng, J. Chen, N. Aung Kyaw, N. K. Zwa Lwin, W. G. Ho, K. S. Chong, and B. H. Gwee. "A highly efficient power model for correlation power analysis (cpa) of pipelined advanced encryption standard (aes)." In 2020 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-5. IEEE, 2020.

[172]N. Chawla, A. Singh, N. Mizanur Rahman, M. Kar, and S. Mukhopadhyay. "Extracting side-channel leakage from round unrolled implementations of lightweight ciphers." In 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 31-40. IEEE, 2019.

[173]S. Bhasin, T. Graba, J. -L. Danger and Z. Najm, "A look into SIMON from a side-channel perspective," 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2014, pp. 56-59, doi: 10.1109/HST.2014.6855568.

[174]K. Shahbazi, and S. B. Ko. "High throughput and area-efficient FPGA implementation of AES for high-traffic applications." IET Computers & Digital Techniques 14, no. 6 (2020): 344-352