On the Resiliency of Protected Masked S-Boxes Against Template Attack in the Presence of Temperature and Aging Misalignments

Md Toufiq Hasan Anik, *Student Member, IEEE*, Jean-Luc Danger, *Member, IEEE*, Sylvain Guilley, *Senior Member, IEEE*, and Naghmeh Karimi, *Senior Member, IEEE*

***Abstract*— Profiling side-channel analysis (SCA) attacks have received a lot of attention in the recent years. To perpetrate these attacks, the adversary creates a profile of a sensitive device at her disposal, and uses it to model a target device with a similar implementation to extract its key. Template attacks are recognized to be the most powerful profiling attacks when the measurement noise is Gaussian. To tackle SCA attacks, different countermeasures have been proposed in the literature, among which masking schemes have received the utmost attention. By adding randomness to the circuit, masking schemes prevent the adversary from relating the power consumption to the evaluated data, thus making the attack more difficult. In this article, we study the protection provided by several masking schemes against template attacks. More precisely, we investigate how the success of the template attack is changed when there is a misalignment between the target and profiling devices in terms of temperature and process variations. As another innovative analysis angle, we extensively study the impact of device aging on the template attack and demonstrate quantitatively how aging misalignments in side-channel traces, between the profiling and the target devices, do hinder the attack. The main objective of this study is to get accurate and numerous results allowing the designer to compare different implementations of masking and accordingly choose one which corresponds to the best compromise among complexity, security, and sensitivity to temperature and aging. We target the S-Box module of the unprotected PRESENT cipher along with its five masking variants including global lookup table (GLUT), rotating S-Box masking (referred to as RSM-LOG hereafter), RSM with read-only memory (RSM-ROM), Ishai–Sahai–Wagner masking (ISW), and threshold implementation (TI). The unprotected circuit gets impacted by such aging misalignments with ≈12.5% increase in the number of traces needed to reach 80% success rate (SR) in the course of 20 weeks of aging at 105 °C. Such increase is 23.3%, 37.19%, and 38.24% for ISW, GLUT, and RSM-LOG, respectively. For RSM-ROM such increase is 193.37% for ten weeks of aging. Interestingly, TI is not much affected by aging in this regard.**

Manuscript received 7 December 2023; revised 26 January 2024; accepted 19 February 2024. Date of publication 18 March 2024; date of current version 26 April 2024. This work was supported in part by the NSF CAREER Award under Grant NSF CNS-1943224 and in part by the Bilateral French-German "APRIORI" project (MESRI-BMBF call) under Grant ANR-20-CYAL-0007. *(Corresponding author: Md Toufiq Hasan Anik.)*

Md Toufiq Hasan Anik and Naghmeh Karimi are with the Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, MD 21250 USA (e-mail: toufiqhanik@umbc.edu; nkarimi@umbc.edu).

Jean-Luc Danger is with the Institut Polytechnique de Paris/Télécom Paris, 91120 Paris, France (e-mail: jean-luc.danger@telecom-paris.fr).

Sylvain Guilley is with Think Ahead Business Line of Secure-IC S.A.S., 35510 Paris, France, and also with the Institut Polytechnique de Paris/Télécom Paris, 91120 Paris, France (e-mail: sylvain.guilley@secure-ic.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TVLSI.2024.3374257.

Digital Object Identifier 10.1109/TVLSI.2024.3374257

***Index Terms*— Common criteria, impact of aging, innate protection against aging, masking at hardware-level, maximum likelihood distinguisher, PRESENT, resistance to side-channel analysis (SCA), S-Box, template attack.**

I. Introduction

SYMMETRIC cryptographic algorithms make use of a secret key to conceal information. This key shall not be disclosed, as it is the cornerstone of the cryptosystem security. Side-channel attacks have been known for a long time to be a realistic threat for the secret key. In particular, supervised attacks (typically template attacks [1]) are the most powerful ones.

It, therefore, becomes paramount to resort to a sound protection against such assaults. Masking schemes have a fair momentum in the industry because they can be formalized [2]. Incidentally, the American NIST has recently launched a consultation about the requirements in terms of masking countermeasures as a mitigation for side-channel attacks [3]. The main focus is placed on the choice for the masking order, that is, on the amount of entropy involved in masking schemes. However, there is also a different consideration: how to get the most of a masking scheme in the context of temperature and aging at any order? and specifically, in this article, we evaluate this question on first-order security masked designs.

The difficulty of an attack depends on several factors. The number of side-channel traces to collect and analyze in order to recover the key is one of such factors. Clearly, at one point, any attacker will decide whether or not an attack is feasible within its budget. The common criteria (CC—ISO/IEC 15408:2022) have formalized such way of measuring the security level. It is based on a *quotation* system, which takes into account multiple considerations such as the *elapsed time*, the *expertise*, the *knowledge of the target*, the *window of opportunity*, and the *equipment*. All those factors impact the success probability of attacks. This is the reason why, in this article, we have undertaken to quantify by which amount, attacks success rates (SRs) are influenced by the environmental conditions. As we shall show, different protections against side-channel attacks have different cost, which increases as the security they bring increases. Thus device cost and security are both strategic, albeit antinomic, variables in a successful product launch on the market. In order to choose the correct protection matching the budget allocated to a given part (the silicon area translated to device cost, proportionally), it is important to know and predict the impact of environment and aging on the most-known implementation of masking schemes.

Most of the scholar papers study in-depth one given countermeasure. However, there are several such countermeasures. Considering that industry is very much cost-driven, we present a horizontal study across several hardware-level masking schemes. Here the novelty is to benchmark six representative countermeasures in terms of cost versus security trade-off.

A. Contributions

This is why we conduct in this article a comparative study of several first-order masking schemes. This leads to clearly point out the characteristics in terms of security, complexity, aging, and temperature impact of every masking method. We place ourselves in the case of low-cost devices, where the signal-to-noise ratio (SNR) is large. As well, we study aging on devices implementing first-order masking.

In this article, the main contribution is to *leverage the comparison of different implementations based on the SR of template attacks deployed on them.* Such a comparison is fair as template attacks are one of the most powerful profiling attacks from the information theory perspective. Indeed, template attacks implement the *maximum likelihood rule* to distinguish among key candidates [4]. This method is applied to several implementations of the same (combinational) design, namely a substitution box. In practice, we compare the security of these masked architectures against template attacks using both HSpice simulations and FPGA emulations. Our results are precious inputs for a design when facing the decision to opt for a given masking implementation style.

The second contribution is to quantitatively investigate the resiliency of the considered masking schemes in the presence of aging misalignments, that is, when the template and target devices have been aged differently. As aging affects the device power consumption, taking aging into account when analyzing the security of the device against template attack is highly crucial. To the best of the authors' knowledge, this is the first time that these mentioned masking schemes are compared in terms of their resiliency against template attack in the presence of environmental and aging misalignments. Such information has value for designers, so that they do not overestimate the security of the protections they choose to implement.

B. Threat Model

In this article, the target of attacks is cryptographic algorithms protected by masking. The masks themselves are generated by a true- or a pseud-orandom number generator (abridged: TRNG or PRNG). We do not consider the mask generation module as part of the leakage because the TRNG or PRNG is working standalone. As the attacker does not control it, he/she cannot properly synchronize with it. On the contrary, we consider that the attacker (attempts to) correlates to the leakage of the algorithm because its plaintext/ciphertext is, depending on the context, either known or even chosen.

We aim at getting rid of practical imperfections, which would hinder the template attacks. Thus we consider the most powerful attacker, namely the one who has access to noiseless traces. Those are obtained by SPICE simulation of the targeted implementations' instantaneous power consumption.

Regarding template attacks, we assume that attackers can train their model using ideal traces. Indeed, power as a side-channel is relevant in this respect, as it is possible to capture identical power traces in profiling (i.e., training) and matching (i.e., inference) stages. However, this assumption would not hold for electromagnetic (EM) traces, whose waveforms heavily depend upon the probe's position in space.

The discrepancies between the profiled and matching traces are modeled by either adding (independent) noise to the traces, introducing aging, considering temperature misalignments between profiling and target devices, or process mismatches between these devices (all five) attested in simulation and the noise and process variation impacts are also attested in real FPGA. Indeed in most cryptographic circuits, the key is not chosen by the user. It is either produced by a physical unclonable function (PUF), or arises from a key management system. Thus it takes time for the adversary to work around those difficulties and collect "training" traces to launch machine-learning-based attacks, for example, template attacks. Such effort can result in a discrepancy between the aging of the target device and the one deployed for modeling. Accordingly, we focus on the aging effects in this article.

C. Comparison With the State-of-the-Art

Bhasin et al. [5] also looked into template attack, however from the attack portability aspect. They launched machine-learning-based template attacks and argued that considering similar devices for profiling and target is not realistic and thus the profiling circuit should be different from the target one. Then to increase their attack success in real silicon, they proposed to build the template based on multiple devices (not just one). Note that our attack is not machine-learning based; rather we benefit from template matching in our attack (discussed in Section II-C of this article). Moreover, the success of our attacks (in both simulation and real silicon, i.e., FPGA here) shows that template attacks remain powerful even in the presence of process variations.

Breuer and Levi [6] demonstrated in another research that the point of interest and leakage model in template attacks differ among various process corners. Also, maximum likelihood varies when there is a mismatch in process corner as well as environmental conditions between profiling and targeted devices for profiling attacks that utilize maximum likelihood distinguishers. Furthermore, they pointed out that statistical information changes across process corners. For the attacks conducted using Jensen–Shannon divergence as a distinguishing factor, statistical distance value rises when there is a mismatch in process corner between the profiling and targeted devices. Comparing [6] with this article, the former conducted the whole analysis on unprotected devices, whereas our analyses are done on devices protected by various masking schemes. Additionally, we thoroughly investigated the impact of environmental variations on the success of template attacks, and we even moved one step further and considered aging dimension variant. Finally, validation of our findings in actual hardware distinguishes our work from [6].

Moradi [7] reports that temperature variations might cause changes in leakage current, hence he advises keeping the temperature constant while measuring static power. Employing a climate chamber is suggested to address these temperature-dependent variations in static power measurement setup [7], [8]. De Cnudde et al. [9] focus on the impact of temperature on the first-order leakage from masking schemes; yet does not consider aging impacts. The effect of temperature on static power leakage is also investigated in [10] and [11]. Moreover, Hwang et al. [12] discussed how static power leakage of protected hiding scheme such as wave dynamic differential logic (WDDL) [13] can be increased through raising the temperature. Similarly, it is critical to investigate how different forms of countermeasures, such as masking techniques, are

affected by temperature variations not just on static but also on dynamic power leakage. Here the goal of this study is to determine how the SR differs for template attacks conducted against protected masking implementations, as well as if attackers benefit from altering the temperature while executing such attacks. Furthermore, the influence of aging on variable temperature is investigated in order to get a definitive conclusion on how protection varies for masking countermeasures when other factors such as temperature, aging, and process variation are varied.

D. Methodology and Outline

This article focuses on security evaluation when facing the most powerful attacker, namely capable of realizing profiled attacks, which are sketched in Section II.

Recall that we aim to quantify the resistance of masking countermeasures (described in Section III) when the attacker faces operational difficulties. Accordingly, after discussing our simulation setup (see Section IV), we first analyze template attacks in ideal conditions, that is, when experimental conditions and aging exactly match. Such baseline security evaluation, accounted for in Section V, is carried out leveraging SPICE simulations. Then from this ideal case, we analyze in Section V how the attacks decrease in effectiveness when either temperature or aging differ between profiled and attack devices. We also aim at understanding whether SPICE simulations are realistic, hence we extend this study and investigate the effectiveness of countermeasures when the implementation departs from the simulation model and realized in real silicon, that is, FPGA in this article. This study, carried out in Section VI, reveals that the targeted masking countermeasures behave almost similarly in real silicon except in one case [i.e., global lookup table (GLUT)] that we will discuss the reason in Section VI. This confirms the validity of the simulation results. Eventually, summary of findings is presented in Section VII and conclusions in Section VIII.

II. Preliminaries

A. Background on Physical Attack Countermeasures

In cryptographic devices, power, heat, time, and EM signals can enable the adversary to perform physical attacks through statistical analysis [14]. Physical attacks can be passive, targeting key disclosure without altering device properties, or active, involving manipulation of device properties (such as operating voltage, temperature, timing, etc.) for key disclosure. Passive attacks, such as power analysis (PA) attacks, involve monitoring power consumption to extract information, while active attacks, such as fault injection attacks (FIAs), introduce faults in devices during operation to induce malfunctions and reveal secret information.

Countermeasures against PA attacks involve hiding [14, Sec. 7] strategies, which reduce the SNR to conceal information mostly by power balancing (e.g., sense amplifier based logic (SABL) [15] and WDDL [13]) and masking [3] techniques (e.g., threshold implementation (TI)) that utilize secret sharing and multiparty computing to generate random intermediate values. In contrast, to detect FIAs, one may use concurrent error-detection (CED) schemes. CEDs include hardware-redundancy-based schemes (e.g., dual modular redundancy (DMR) and triple modular redundancy (TMR) [16], [17]), time-redundancy schemes (e.g., [18]), information redundancy-based schemes (e.g., [19]), and several other schemes that detect the faults during the normal operation of the circuit. A structure-independent parity-based fault detection scheme for the Advanced Encryption Standard (AES) SubBytes transformation has been presented in [20]. A modified S-Box and inverse S-Box structures tailored for parity-based fault detection schemes are introduced in [21]. An additional lightweight parity-based concurrent FIA detection scheme that uses composite field and normal basis is presented in [22]. FIAs injected via altering operating conditions (such as voltage, clock frequency, and temperature) can be detected using digital on-chip sensors [23]. Das et al. [25] introduced Razor II as a solution to identify and rectify variation-induced delay errors. This method detects clock glitching but cannot detect laser-based FIAs. To solve that, Ebrahimabadi et al. [26] proposed DELFINES to detect FIAs by sensing laser-induced IR-drops.

The above countermeasures either protect the device against active or passive attacks. However in certain scenarios, combined passive and active attacks may be feasible. By introducing a computational disturbance via fault injection, a passive power attack can be executed during the perturbed execution [27]. Kulikowski et al. [28] introduced a combined attack, which includes injecting faults to disturb the power consumption balance of logically balanced gates. Several countermeasures have been proposed to address combined PA and FIA attacks. However, a considerable portion of these schemes is either designed for software implementations or provides limited security while introducing significant overheads as a trade-off. However, these countermeasures tend to be application-specific, lacking a universal solution that comprehensively addresses both PA attack and FIAs.

B. Background of Device Aging

Aging mechanisms result in performance deterioration and eventual failure of digital circuits over time. Among those, the effect of negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), and hot-carrier injection (HCI) are more prominent in CMOS technologies, resulting in increased switching and path delays [31].

1) BTI Aging: NBTI occurs in a pMOS device when a negative voltage is supplied to its gate. Indeed a pMOS faces two phases of NBTI depending on its operational conditions. The Stress phase occurs when the transistor is "ON." Here, positive interface traps are formed at the Si-SiO_2 interface, resulting in a rise in the transistor's threshold voltage (V_{th}). The second phase (Recovery) occurs when the transistor is "OFF." Here, the V_{th} drift occurred during the stress phase partially recovers. The physical characteristics of the transistor, supply voltage, temperature, and stress time all affect the BTI rate [32]. PBTI affects nMOS transistors in a similar way that NBTI affects pMOS transistors.

2) HCI Aging: HCI arises in nMOS devices when hot carriers are injected into the gate dielectric during transistor switching and remain there. HCI degrades the circuit by shifting V_{th} and drain current of stressed transistors. The HCI-induced V_{th} drift depends on the activity factor of the transistor under stress, the temperature, clock frequency, and usage duration [33].

C. Background on Template Attack

Profiling side-channel analysis (SCA) attacks [34], [35] in which the adversary has access to the secret key of a cryptographic device and uses such information along with

the chips' side channel data (power consumption, runtime, EM emanations) to deduce the key of another device with the same implementation have been shown to be highly promising. Profiling attacks, and in particular template attacks pursue two phases. In the training phase, the attacker builds the model of the profiling device by recording a significant number of its traces corresponding to a varying set of input (plaintexts and keys) data. Then, in the attack phase, the traces are categorized based on the key values, and template matching is used to determine the target device's key [1].

If traces are represented as a matrix X of $D \times Q$ real numbers (Q traces of $D = 300$ samples as an example), and the learned model Y_k is also a $D \times Q$ matrix, then the attacker guesses the key as in [36, Th. 1]

$$\hat{k} = \underset{k \in \{0,1\}^4}{\operatorname{argmin}} \operatorname{tr}\left((X - Y_k)^{\mathsf{T}} \Sigma^{-1} (X - Y_k)\right) \tag{1}$$

where Σ is the $D \times D$ noise covariance matrix, tr.is the trace operator, and argmin operator selects the value of k (4 bit) that results in the minimum value of its following function (= argument). In order to avoid statistical biases, we select uniformly distributed plaintexts, hence Q is always shown as a multiple of 16. This template attack works on masked devices, where the attacker aims at exploiting first-order leakage. The more similar the profiling and target devices in terms of operating conditions, the more successful the attack. Accordingly, this article quantitatively investigates the success of the template attack on unprotected and masked-protected PRESENT cipher's S-Box in the presence of temperature, process variations, and aging mismatches between the profiling and target devices. Such an approach allows to standardize an attack-based methodology to compare different implementations on a fair basis.

The impact of aging on the success of template attacks launched on the unprotected circuits has been studied before in [37], which is discussed again in this article as a baseline.

D. Background on PRESENT Cipher

PRESENT is a lightweight block cipher with 64-bit blocks and a standard bit oriented permutation layer [38]. It has 31 rounds and can work with two key lengths of 80 and 128 bits. A bitwise XOR operation, a non-linear substitution layer, and a linear permutation layer are included in every encryption cycle. In this study, we focus on the S-Box; due to its contrasted confusion coefficient [39], it is a highly appealing target for attackers. All the PRESENT S-Box implementations we targeted in this article can be performed in one clock cycle even after masking protection. In particular, the TI implementation is not pipelined.

E. Background on Other Cryptos

Standard cryptographic algorithms, such as AES and secure hashing algorithm (SHA), are widely utilized in computer systems for general purpose computing. Nevertheless, these conventional algorithms prove unsuitable for deployment in small IoT/embedded devices due to their resource constraints. Hence, researchers are actively engaged in the development of lightweight cryptography (LWC) algorithms, specifically designed for implementation in low-area environments with minimal power consumption. Several LWC algorithms have been introduced by researchers among which PRESENT stands out due to its minimal area overhead; allowing implementation with fewer than 2000 gates in ASIC implementations [38]. Other widely recognized LWC algorithms comprise PRINCE [40], GIFT [41], ASCON [42], Midori [43], and more. Hence, additional research on physical attacks and corresponding countermeasures is imperative for enhancing the security of LWC. Therefore, our study, focusing on the innovative integration of temperature and aging misalignment in PA template attacks, also sheds more light on the secure development of LWCs.

Current cryptographic schemes may be compromised by quantum computing, thanks to its capability to efficiently solve mathematical problems. Hence, algorithms in the field of post-quantum cryptography (PQC) are formulated to ensure security throughout the quantum computing era. PQC algorithms are constructed upon mathematical problems that prove challenging for quantum computers to solve. Various types of PQC, such as lattice-based, hash-based, multivariate polynomial, and code-based schemes, are being developed as alternatives to traditional cryptographic methods. Numerous PQC schemes have been introduced in academic literature, including NTRUEncrypt, NTRUSign, Crystals Kyber, BIKE, SPHINCS+, among which lattice-based PQC schemes, such as Kyber, are gaining increased attention within the research community. Number theoretic transform (NTT) is essential in lattice-based cryptographic systems within PQC, where it optimizes polynomial operations, enhancing both computational efficiency and the security of the algorithms. Nonetheless, PQCs and NTT are susceptible to FIA and PA attack [46], [48]. Researchers are employing error-detection mechanisms [50] and investigating error-resistant NTT architectures [51] to safeguard PQC schemes from FIAs. Additionally, they are exploring masking schemes [52] as a countermeasure against power attacks. Our research on factoring in the impact of aging misalignments in the success of template attacks launched on masking-protected circuits is expected to have similar applicability in such PQC masking countermeasures. As a continuation of this study, we will focus on the effects of aging on PQCs.

III. PRESENT S-BOX IMPLEMENTATIONS

The S-Box (referred to as S hereafter) is a non-linear module included in PRESENT and several other block ciphers. Thus, it grabs the attacker's attention to leak the sensitive key by compromising it. One possible protection could be *dual-rail balancing logic*, whereby each functional bit-level activity is compensated by a dummy one [53]. However, the security of such protection cannot be ensured since imperfections in the activity occultation mechanism can lead to exploitable leakage arising from any gate of the netlist [54]. Moreover, balancing in dual rails can be impacted by aging leading to more leakage by increasing attack SR [15], [55]. This is the reason why masking logic is preferred. In this article, we target S-Box of an unprotected PRESENT along with five masking-protected counterparts whose details are discussed below.

A. Description of S-Box Implementations

1) LUT: LUT is a simple data-flow description of the unprotected S-Box, serving as a baseline for our comparisons.

2) GLUT: The GLUT described in [56, Sec. 3.2, p. 234] is a masking implementation realizing a function $\mathbb{F}_2^4 \times \mathbb{F}_2^4 \times \mathbb{F}_2^4 = \mathbb{F}_2^{12} \rightarrow \mathbb{F}_2^4$ that satisfies

$$Y = \text{GLUT}(A, \text{MI}, \text{MO})$$

such that

$$Y \oplus \mathrm{MO} = S(A \oplus \mathrm{MI})$$

where A and Y are the masked input and output, respectively, and MI and MO refer to the input and output masks. GLUT is the most straightforward masking scheme, as both inputs and outputs are masked; its internals are not optimized though.

A noticeable example of *optimized* GLUT on AES has been that of Canright [57]. This work heavily exploits the structure of the AES S-Box (which has a very simple algebraic expression [58, Sec. 4.3.2]) to make the GLUT implementation compact. Nevertheless, this optimization unfortunately opened the door to some exploitable glitches [59]. In this article, as the PRESENT S-Box is 4×4 and not 8×8 as AES and is essentially random (it has been designed without method), we do not seek (in the first place) an optimization and leave it to the compiler to find a compact implementation. As a byproduct, sensitive glitches are not expected to show up in this case either.

3) RSM: Rotating S-Box masking [60] (referred to as RSM-LOG hereafter where LOG refers to logic) aims at saving area and randomness compared to GLUT in two different ways. To achieve so, the following features should be met.

1) The masks set is a subset of the full mask set (the masking scheme is referred to as "low-entropy" [61]),
2) The masks used at the output of the S-Box are deduced deterministically from those at the input, namely by using the next one (in a circular manner) within the mask set.

In our RSM implementations, we avoid employing a strict subset of the mask (as there are only 16 of them in PRESENT 4-bit S-Box). However, we construct the output mask depending on the input mask [60] in the following way:

$$\mathrm{MO} = (\mathrm{MI} + 1) \bmod 16$$

where $0 \leq \mathrm{MI}, \mathrm{MO} \leq 15$ are integers computed via $M \in \mathbb{F}_2^4 \mapsto \sum_{i=0}^{3} M_i \cdot 2^i \in \{0, 1, \ldots, 15\}$. Indeed RSM is represented as a function $\mathbb{F}_2^4 \times \mathbb{F}_2^4 \to \mathbb{F}_2^4$ such that

$$\mathrm{RSM}(A, \mathrm{MI}) = \mathrm{GLUT}(A, \mathrm{MI}, (\mathrm{MI} + 1) \bmod 16). \quad (2)$$

As a consequence, in practice, RSM masking style is expected to be more compact than GLUT.

The RSM countermeasure can be implemented by a direct RTL synthesis of (2). However, notice that this is not faithful to the original paper [60], as there is no "ROM" block, which would prevent spurious glitches from showing up. Therefore, we consider such implementation as lazy engineering and call it "Logic RSM" or RSM-LOG in short. A correct implementation is referred to as RSM with read-only memory (RSM-ROM), discussed next.

4) RSM-ROM: ROM-based RSM-LOG scheme tries to realize the RSM-LOG goals of being secure against first-order (and even second and third order attacks [62]) via implementation in ROM. ASIC design kits and FPGA circuits propose ROM primitives, which are implemented in regular matrices that are optimal in complexity. In this architecture, we consider the logic designs realized by the instantiation of gates from a Boolean library, similar to the implementation in [63]. Initially the datapath is synchronized for any input configuration, which makes input-related deviations of leakage small. Then, the structure is designed with a one-hot strategy such that only the required logic is activated, which further contributes to reduce the side-channel leakage. The mapping into gates shall be faithful, even though electronic design automation (EDA) tools restructure the logic.

5) ISW: In order to resist compromise of masking rationale by EDA tools, Ishai et al. [64] and Covic et al. [65] introduce a bottom-up approach. When dealing with the non-linear gates, they propose to start from an optimized netlist in terms of AND/OR usage [66, Sec. 3], and gradually replace the gates by their gadgets. The gadget for the AND gate requires 1 bit of uniform randomness, that is, R. Namely, given a random sharing (A_0, A_1) of bit A (where $A = A_0 \oplus A_1$), and a similar sharing for bit B, one computes Y, the AND of A and B in a masked way using the following formula:

$$\begin{cases} Y_0 = ((A_1 \wedge B_1) \oplus R) \oplus (A_0 \wedge B_0) \\ Y_1 = ((A_0 \wedge B_1) \oplus R) \oplus (A_1 \wedge B_0). \end{cases}$$

In those equations, the implementation must (at least statically) respect the specified order in the netlist gates instantiation devised by the parenthesis in the above formula. In our implementation, we implemented OR via benefiting from De Morgan's law $\mathrm{OR}(a, b) = \neg\mathrm{AND}(\neg a, \neg b)$. In principle, Ishai–Sahai–Wagner masking (ISW)'s security proof is valid, but when considering the real features of combinational gates, it encounters several difficulties. Gates can evaluate in a non-natural order as a result of races, resulting in first-order leakage [67].

6) TI: Owing to the shortcomings of ISW, TI has been proposed as a more stringent leakage previous technique. Namely, TI is an algorithmic countermeasure against power SCA, and benefits from multiparty computation and secret sharing [68]. A TI implementation holds three properties: non-completeness, correctness, and uniformity. In TI, similar to ISW, each input bit is distributed into $n + 1$ shares. Moreover, on top of ISW, TI has the following characteristics.

1) TI does not require gate ordering, thus allowing for aggressive logic optimization by EDA tools.
2) TI mandates that each output share be based on only n shares of each input to ensure that no race/glitch may unintentionally reveal (by design) an unmasked value (i.e., incompleteness property of TI).

TI protects efficiently against leakage through glitches. However, on the downside, it does not work out-of-the-box (manual writing of netlists is necessary) and it is also noticeably more expensive than former masking schemes (namely: GLUT, RSM-LOG, RSM-ROM, ISW). Notice that TI imposes strong requirements on the shares distributions, which are relaxed for instance in "domain-oriented masking" (DOM [69]). The strategy of DOM is to insert pipeline stages, which stop glitches propagation and allow to restart afresh. The composition of DOM gadgets is error-prone, hence an important state-of-the-art refinement is "hardware private circuit" (HPC [70]), which is provable even under composition. However, such gate-level solutions (TI, DOM, and HPC) are optimal in terms of security, but are costing more than the masking styles we consider. Indeed, in this research, our target is to study the security/cost trade-off for lighter masking schemes. Table I compares the investigated S-Box implementations regarding number of gates (with 2–4 inputs), equivalent gates (#gates normalized by the number of equivalent two-input NAND gates), random bits, and max propagation delay (ps).

B. Leakage Sources Exploited in This Study

Two kinds of first-order leakage can be exploited: glitches carrying information of unmasked variables (all implementa-

TABLE I
GATE-LEVEL SPEC. OF THE TARGETED S-BOX IMPLEMENTATIONS

	LUT	GLUT	RSM-LOG	RSM-ROM	ISW	TI
# AND	2	580	134	0	16	737
# OR	2	180	74	0	0	71
# XOR	9	0	0	0	34	449
# INV	1	12	20	510	7	67
# BUF	0	0	0	0	0	1
# NAND	0	0	0	16	0	0
# NOR	0	0	0	716	0	0
# XNOR	0	0	0	0	0	2
Total Gates	14	772	228	1242	57	1325
Total Equ. Gates	29	1183	373.5	987	112.5	2369
Max. Delay (ps)	280	420	350	1700	470	480
# Random Bits	0	8	4	4	4	12

tions but TI), and Hamming weight/distance parity: indeed, in any sharing of a sensitive variable $A \in \mathbb{F}_2$ into $A_0, \ldots, A_n$, one has that $\mathrm{LSB}(w_H(A_0, \ldots, A_n)) = \mathrm{LSB}(\sum_{i=0}^{n} A_i) = \bigoplus_{i=0}^{n} A_i = A$ (unmasked A), where LSB is the least significant bit (or parity) of $w_H(A_0, \ldots, A_n) \subseteq \{0, \ldots, n\} \in \mathbb{N}$. Such first-order leakage is often overlooked because it is "one-bit" hence is perceived as "buried in noise," but is not in the case of little noised traces.

C. Evaluation Roadmap

As shown in Table I, different countermeasures have different costs. The questions we address next are the following.

1) Does indeed more cost bring more security (in terms of resistance to template attack)?
2) How is this security level affected by temperature, aging, and real capture of traces (using actual FPGA) discrepancy between profiling and attacking?

IV. EXPERIMENTAL SETUP

We implemented the add-round-key and S-Box operations in the first round of PRESENT cipher with 80-bit keys in the transistor level for the unprotected and five masking implementations of PRESENT S-Box, mainly ISW, GLUT, RSM-LOG, RSM-ROM, and TI protections, using a 45-nm technology extracted from the open-source NANGATE library.

We used Synopsys HSpice for the transistor-level simulations and the HSpice built-in MOSRA Level 3 model to assess the effect of BTI and HCI aging. For all six implementations, power traces were extracted for new and aged devices. The effect of aging was evaluated for 20 weeks of device operation in steps of 1 week. The experiments were conducted for $\{-40, -20, 0, 25, 45, 65, 85, 105\}$ °C (that is until grade 2 of AEC Q-100 [71, Sec. 1.3.3]). The supply voltage (V_{dd}) was 1.2 V.

For the HSpice simulations, the targeted masking implementations were fed with the initial value of all high (for both masks and S-Box inputs), while all possible combinations of final values for the masks and S-Box inputs were simulated. Thereby, we considered 16 input traces for the unprotected circuit (as the PRESENT S-Box is 4 bits), 256 combinations for RSM-LOG and RSM-ROM, 4096 combinations of inputs for GLUT and ISW, and finally 65 536 traces for TI (four shares each 4 bits). We categorized the power traces in all implementations (unprotected and masked) based on the value of the unmasked inputs (e.g., $A \oplus \mathrm{MI}$ in GLUT, recall Section III). Then we used the mean trace of each category to build the power templates. The key is fixed in all simulations. We used the same fixed key and the same set of inputs for both fresh (no aging) and aging simulations.

As mentioned, the simulated traces contain two parts: the results of key addition and S-Box outputs for each initial (fix in our case) as well as its following n-bit value (including masks and S-Box inputs). For the attack, we considered only the second clock cycle, when the transition takes place from *initial* to *final* value for the cryptographic circuit. To emulate the real-silicon measurement, we added Gaussian noise to the power traces extracted from simulations. In order to obtain a robust value for the attack's SR even in the presence of noise, we attacked each circuit 1000 times using such noisy environment (randomly selected traces each time and added Gaussian noise to the selected traces) and reported the average SR. To validate the simulation results, we implemented the targeted masked circuits along with the unprotected counterpart in FPGA. We used two SPARTAN6 XC6SLX75 FPGAs soldered on a SAKURA-G board [72], with Xilinx ISE 14.7 software. One of the FPGAs was utilized to apply input values, and the other was used to collect the power traces required to launch the template attack.

V. EXPERIMENTAL RESULTS

A. Power Templates for Each S-Box Implementation

The first set of results deals with the template/profile generation for the attacks on the S-Boxes. To generate the power templates, we performed extensive HSpice transistor-level simulation on both protected and unprotected S-Boxes.

As discussed earlier, for each implementation, we classified the power traces based on the value of the unmasked input (16 classes totally) and built the template based on the mean trace of each class. Note that traces are assumed to start from a given fixed value such as "`11,...,1`," which models a preset (i.e., the initial values are considered as "1"). Fig. 1 shows the 16 mean traces of each implementation. In the depicted figures, each color represents one class of power traces. Fig. 1(a) depicts the power template used in the unprotected circuit (referring to as LUT). As expected, in this case the traces are very different from each other, resulting in more leakage (as will be discussed later). In contrast, for the protected implementations, the traces are less differentiated from each other, making the attack more difficult.

B. Signal-to-Noise Ratio Calculation of Power Traces From the Targeted S-Box Implementations

The power traces from each S-Box implementations were collected from transistors-level simulations conducted by HSpice without noise. In reality, electronic devices suffer from noise in the system. Thus, to perform the analysis, it is important to consider noise in the collected power traces. Indeed, artificial Gaussian noises need to be added to the collected power traces. However, the noise level should be similar to the noise level from a real chip. According to the ISO standard for IT security techniques [73], the SNR level should be in the range of 0.001–0.010 for first-order attack.

Note that HSpice simulations are not noisy while, in practice, there is some non-algorithmic noise, which is modeled as $\mathcal{N}(0, \sigma^2)$ and is independent of the algorithmic noise whose variance is denoted as Var(Power_Templates). Here σ^2 is the non-algorithmic noise variance. In this context, SNR is defined as below

$$\mathrm{SNR} = \frac{\mathrm{Var(Power\_Templates)}}{\mathrm{Var(Power\_Templates)} + \sigma^2}. \tag{3}$$

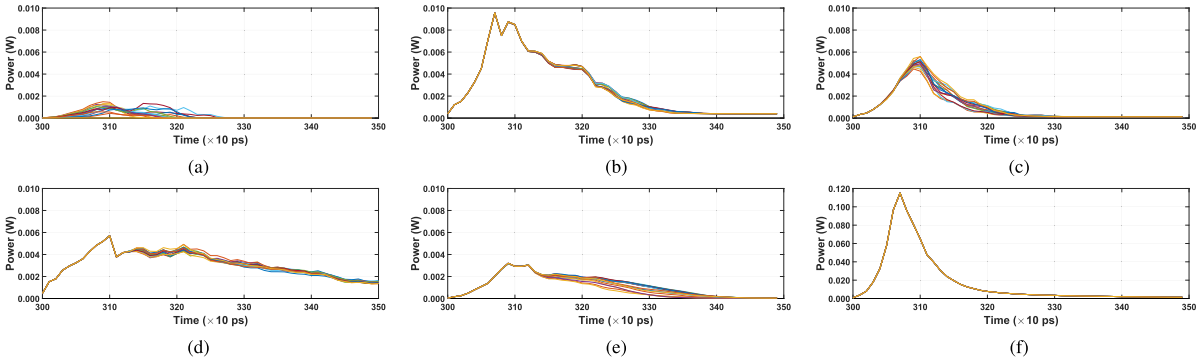

Fig. 1. Power templates in different S-Box implementations at 105 °C. Devices are fresh (not aged). Each color represents one class of power traces, where class i denotes the mean of the power traces extracted when the unmasked input is i. (a) LUT. (b) GLUT. (c) RSM-LOG. (d) RSM-ROM. (e) ISW. (f) TI.

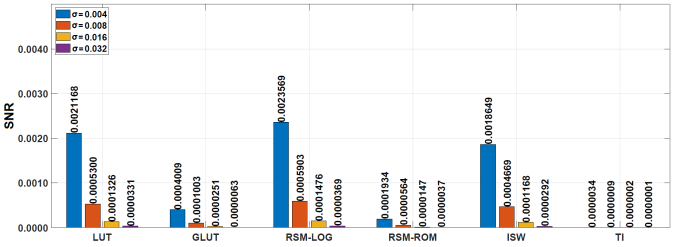

Fig. 2. SNR of S-Box implementations with mask effect. *LUT does not include any mask and just included in this figure as a baseline for SNR comparison.

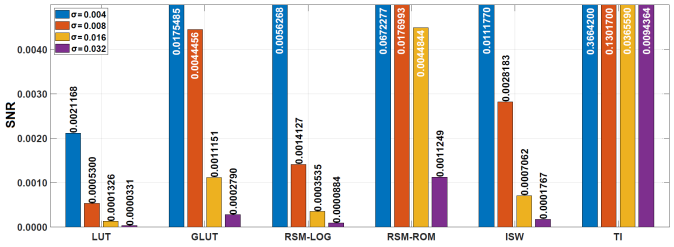

Fig. 3. SNR of S-Box implementations without mask effect.

Each Power_Template (say PTi) denotes the mean of the power traces related to unmasked input i.

Before adding the noise, we need to calculate the SNR for different σ and find the suitable one that follows the standard noise level. Here, we performed two different levels of SNR calculation for sigma (σ) values {0.004, 0.008, 0.016, 0.032}; one while masking ON and another one while masking OFF.

Fig. 2 represents the SNR values of the targeted implementations [using (3)], while the mask is ON. The idea is to find a suitable σ value that follows the ISO standard discussed earlier. In these implementations, LUT is our baseline. As shown, for the sigma value 0.004, the SNR for LUT is ≈0.0021, which is in the ISO standard range of [0.001, 0.010]. For all other sigma values, the SNR is very low. Thereby, we do not continue with sigma values other than 0.004. However, we performed some analysis for the effect of noise discussed in Section V-G. Note that we have used (3) to reflect SNR to take into account not only the measurement noise impact but also the impact from the random masks (which are not known to the attacker). Notice that this equation follows the same concept as the one presented in [14, Sec. 4.3.2, p. 73].

As Fig. 2 shows, with the increase of sigma value, the SNR of all the S-Boxes is decreasing, that is, the higher the sigma value the lower the effect of signal. As expected, in most cases, protected implementations have lower SNR compared to the unprotected LUT. Among them, TI has the lowest SNR for all sigma values. Although a system with a lower SNR is considered as more secure against attacks, this assumption might not be always true as in some cases a lower SNR can be due to design complexity or the type of the launched attack. As can be observed, RSM-LOG has a higher SNR than LUT. However, RSM needs to be implemented in memory-based implementation. By comparing RSM-LOG with RSM-ROM, we want to demonstrate how a poor implementation might affect security.

Fig. 3 investigates the resiliency of the targeted S-Box implementations through the SNR value when the effect of the mask is removed by a powerful attacker. As shown, for $\sigma = 0.004$, the removing mask increases the SNR around 2.4 and 6 times for RSM-LOG and ISW, respectively. These rates are comparatively lower than other structures as the SNR in GLUT and RSM-ROM increased 43.8 and 347.6 times comparing to the case of mask ON. These results are alarming for a secure implementation. The most interesting circuitry is the TI, which is considered the most secure one; here SNR increased 107 770 times compared to the mask ON case. These results have an interesting takeaway: there is also a "big reward" to disable the masking. Indeed, the leakage is thus exacerbated, with respect to "no countermeasure" case. Of course, the fact that the masks are randomly distributed must be a prerequisite. It is noteworthy to mention that the effect of disabling the mask is usually "overlooked" while designing the masked implementations. Please note that without mask the leakage also depends on the circuit complexity. Thereby, TI is the most leaking circuit when the masks are OFF. This means that attackers facing a TI as a protection are especially inclined to alter the TRNG generating the masks as a preliminary step in their attack path. The increase of SNR while mask is OFF relates to the presence of first-order leakage.

C. Attack Success Rate

The goal of this experiment is to perform a quantitative security analysis for different masking schemes and find out the suitable design for the user by considering security versus area trade-offs. It shall be noted that side-channel attacks leverage a "divide-and-conquer" approach, whereby keys are extracted by chunks (e.g., nibble by nibble). Therefore, the successful reconstruction of the complete key implies that each chunk be extracted with high confidence (see for instance [74]). Thus even a small change on the SR at chunk level can have a large impact on the global key recovery success. Formally, assuming that there are $N = 16$ chunks of 4 bits, as in PRESENT with 128-bit key, the global success in recovering the full key $k^\star = (k_1^\star, \ldots, k_N^\star)$ is $\Pr(k = k^\star) = \prod_{i=1}^{N} \Pr(k_i = k_i^\star)$. Indeed, each of the N key chunks is

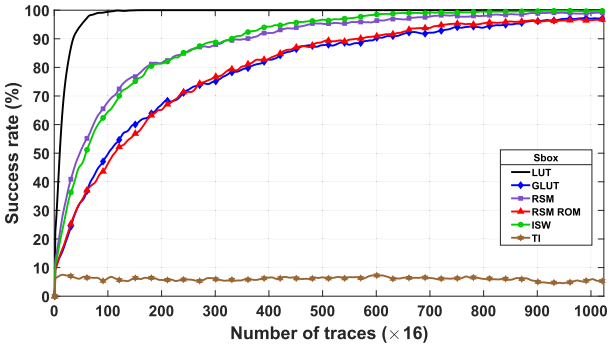

Fig. 4. SR after 1000 attacks on different S-Box implementations. Profiling and target devices are new; both operating at 105 °C, and $\sigma = 0.004$.

independent of each other. If we assume that all probabilities of success at chunk level are equal (i.e., $\Pr(k_i = k_i^\star)$ are the same for all $i \leq i \leq N$), then $\Pr(k = k^\star) = \Pr(k_1 = k_1^\star)^N$. Hence even a small change (e.g., owing to environmental change or aging) of the chunk-level SR translates *polynomially* on the global SR. One way to compare the security with a "scalar metric" is to consider the number of traces required for an attack SR to be greater than 80%. The larger this metric, the more secure the implementation.

This set of results compares the security of the investigated implementations regarding their resiliency against the template attack. In this experiment, both profiling and target devices are new and operating at 105 °C. To emulate real-silicon measurements, we added a Gaussian noise with $\sigma = 0.004$ (based on the above discussion) to the traces extracted from the target device. We consider first no process mismatch, which will be studied in Section V-F. Fig. 4 shows the SR of attacks launch on each circuity. Note that we considered a large amount of noise compared to the signals, resulting in SNR of ≈0.002 (in the unprotected circuitry), which is in the typical range [73] according to the literature.

As shown, among all implementations, TI presents higher security while LUT is the least secure against template attack. We consider the unprotected LUT-based implementation as the baseline. As depicted, the unprotected circuit is more vulnerable than all other implementations. It needs only 24(×16) traces to reach 80% SR. While for GLUT, the adversary needs 363(×16) traces to attain the SR of 80%. RSM-LOG, RSM-ROM, and ISW need 170(×16), 347(×16), and 176(×16) traces, respectively, for an attack with SR = 80%. As shown, TI is not attackable at all. This strong resiliency can be explained by its complex mask sharing with highly balanced power templates as shown in Fig. 1(f), where inter class variance is hardly observed from the extracted power traces. Moreover, SNR from Fig. 2 shows that masking effect makes the SNR lowest among all the masking schemes presented.

The takeaway point from these observations is that although masking schemes have been proposed to ensure security by randomizing the power traces, an adversary may mitigate the randomization impact by extracting many traces and averaging them while he/she still benefits from inclusion of a large number of gates prone to leakages enumerated in Section III-B. In order to illustrate the "security-cost" trade-off, we present in Fig. 5 the security *versus* the area, expressed in gate equivalent. The security is extracted from Fig. 4, and the area from Table I. It can be seen that each countermeasure is relevant, in that more area strictly entails indeed more security. Therefore, in the rest of the article, we continue our investigation with the six countermeasures.

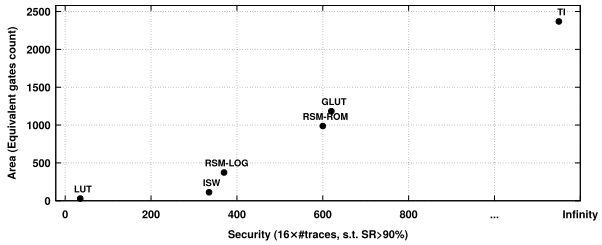

Fig. 5. Security versus area quantitative trade-off illustration.

D. Impact of Aging on the Attack Success Rate

This set of results quantifies the effect of aging on the considered circuits when launching template attacks. The aim of this experiment is to analyze how the security level of masked devices is impacted by device aging when they are exposed to template attack. To have a clear picture of the aging impact over time, first in Fig. 6, we show the difference of the power traces extracted from 0- and 20-week old circuits. Indeed, such mismatch as shown in Fig. 7 hinders the attack in both circuits. Recall that aging is not part of the attack procedure, but is rather an encumbrance met by the attacker.

The results depicted in Fig. 7 have been extracted when the profiling and attack temperatures are similar; yet there is an aging misalignment between the devices used for profiling and the target one. Here, the profiling traces were gathered from a fresh device (shown in Fig. 1) while the target device is between 0- and 20-week old (0w–20w in Fig. 7). For the sake of space, we only illustrate the zoomed version of the SR plots around SR = 80%. As shown, in both protected and unprotected circuits, the SR decays with aging. Indeed as the attacked circuits become older and older, the SR drops. Fig. 7 illustrates that the SR in both circuits falls quickly after a week. After that the SR continues to decline, but at a moderate rate.

As shown in Fig. 7 at 105 °C, the number of traces used to reach SR = 80% for LUT increases to 27(×16) for a 20-week old device comparing to 24(×16) traces for a new one (12.5% increase). Note that this graph shows the results after smoothing. The effect of aging is very similar for ISW, where the number of traces increases around 23.3% over the course of 20 weeks. RSM-LOG and GLUT required 38.24%, and 37.19% more traces, respectively. For the sake of space, for these cases, gradient decay is not shown as the concept is similar. However, effect of aging on number of traces for reaching SR = 80% is shown on Fig. 8, described later in this section. For the case of TI, as the attack is not successful even without aging misalignments (recall Fig. 4), we do not show the aging impacts as aging misalignments even hinder the template attack. Accordingly, from now on, we skip showing the TI results for the sake of space.

To show how aging impact changes in different temperatures, Fig. 8 compares the number of traces required to attain SR = 80% in the presence of aging misalignment between 0 and 20 weeks for unprotected and protected circuits when the profiling and attack devices were measured at 105 °C with the case of 65 °C. As expected, the aging impact is more prominent in higher temperatures. For example, as mentioned earlier, in ISW, we observed 23.3% increase in number of traces at 105 °C (to attain SR = 80%) in a course of 20 weeks while such increase is around 7% at 65 °C. RSM-LOG and GLUT showed 22.4% and 15.11% increase at 65 °C, respectively, for 20 weeks of aging compared to 38.24%, and 37.19% more traces at 105 °C for the same

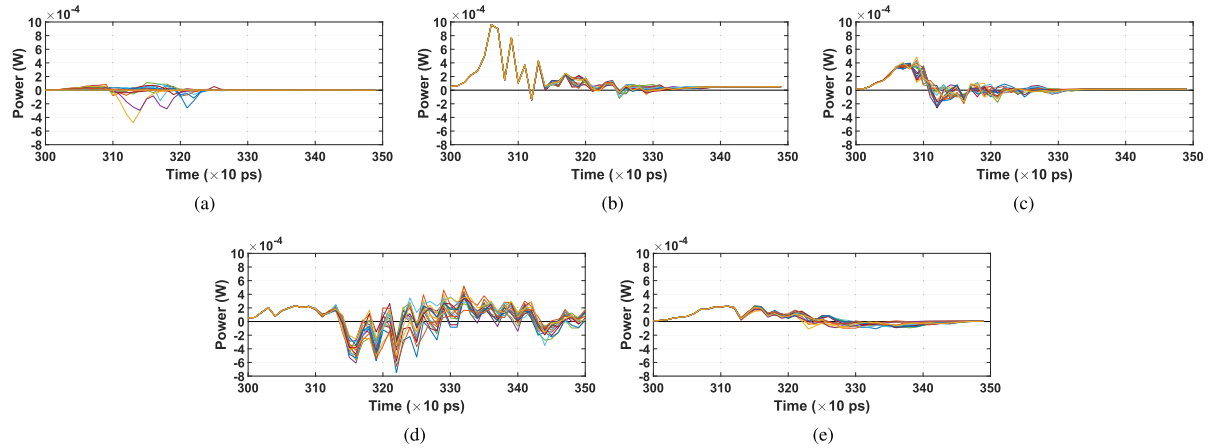

Fig. 6. Difference of power templates between week 0 and 20 of usage at 105 °C (TI is not included as the template attack was not successful on it). (a) LUT diff P(0W)–P(20W). (b) GLUT diff P(0W)–P(20W). (c) RSM-LOG diff P(0W)–P(20W). (d) RSM-ROM diff P(0W)–P(20W). (e) ISW diff P(0W)–P(20W).

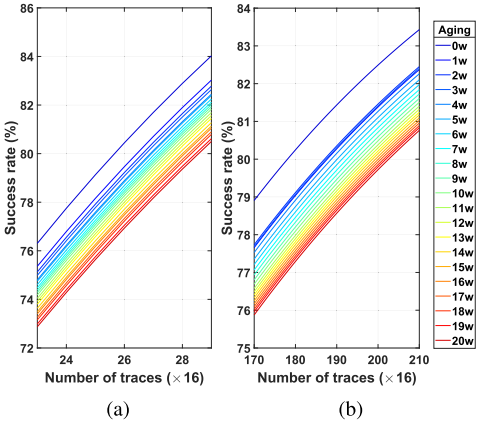

Fig. 7. SR of S-Boxes with aging. (a) LUT. (b) ISW.

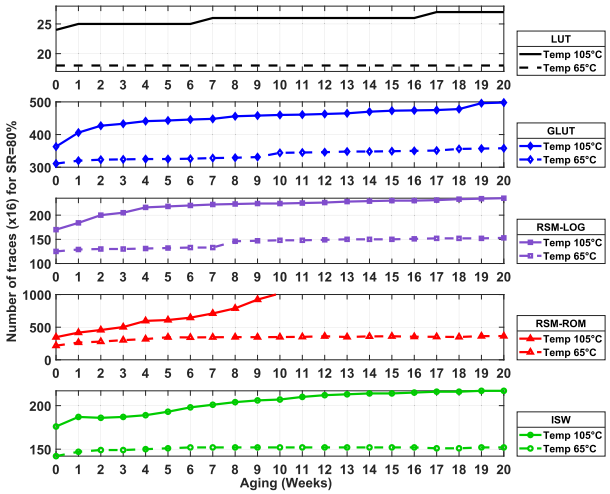

Fig. 8. Number of traces required to attain SR = 80% ($\sigma = 0.004$) when the target device has different ages. In each case, profiling and attack temperatures are equal. The profiling device is new (age = 0). The x-axis shows the age of the target device.

aging duration. For RSM-ROM such increase is 48.32% at 65 °C for 20 weeks of aging, however for 105 °C it increased 193% for ten weeks of aging but the attack failed to reach 80% SR after ten weeks of aging for maximum 1024(×16) traces. Indeed the practicality level of template attacks depends on the masking scheme. In practice, netlists with a long propagation time (e.g., RSM-ROM) have a different leakage profile between training and attacking phase, which degrades

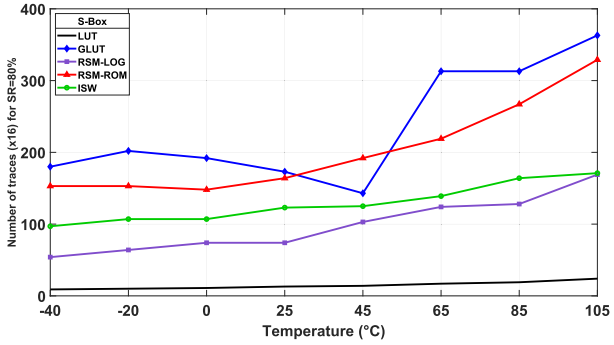

Fig. 9. Mean number of traces to reach SR = 80% when template and attack temperatures are the same. Both template and target devices are new.

the attack efficiency. On the other hand, the incompleteness property of TI makes it secure against template attacks. For the LUT circuit, the increase of traces is much less in lower temperatures, for example, ≈ 0% increase with 20 weeks of aging at 65 °C when $\sigma = 0.004$. The takeaway point from these observations is that the more the aging misalignments, the more difficult the template attack. Moreover, such effect is more prominent in higher temperatures.

Note that hiding countermeasures, for example, WDDL, are also affected by aging. These protected circuits will also be more resilient against template attacks when there is an aging misalignment between the profiling and target devices [75].

E. Effect of Temperature Mismatch

The influence of temperature on template attacks is demonstrated in this set of results. The goal of this experiment is to analyze the effect of temperature on the success of the template attacks launched on masked devices. We want to find out if the adversary can benefit from changing the temperature when attacking the device or not.

We begin by demonstrating the number of traces necessary to achieve SR = 80% when the template and target devices are both performing at the same temperature. Fig. 9 shows the findings, where the mean number of traces for each unprotected LUT circuit changes between 9(×16) and 24(×16) when the temperature changes between −40 °C and 105 °C. This range is 180(×16) to 363(×16) for the GLUT masking. In sum, as shown, GLUT affected the most and LUT affects the least by temperature change. However in all cases, the higher the temperature, the more difficult the attack. A minor variation in the findings is due to the unpredictability of the

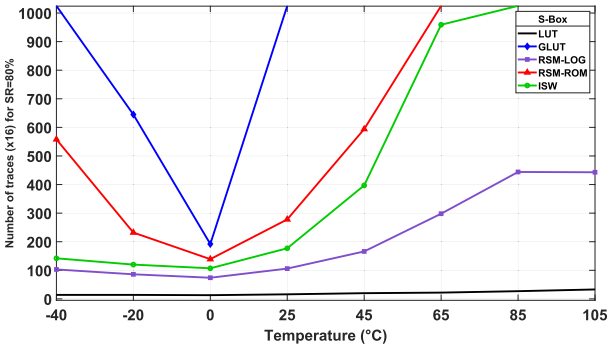

Fig. 10. Mean number of traces to reach SR = 80% when template and attack temperatures are different. Profiling temp. = 0 °C. Attack temp. varies between −40 °C and 105 °C. Template and target devices are new ($\sigma = 0.004$).

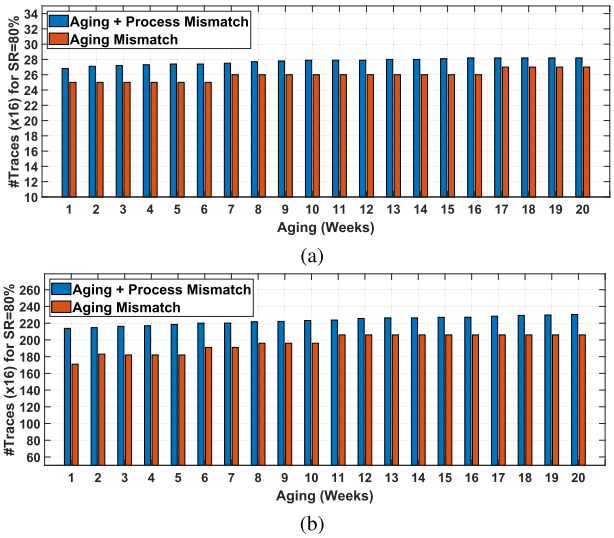

Fig. 11. Mean number of traces for SR = 80% with/without process variations. Profiling temp. = 105 °C, attack temp. = 105 °C, and $\sigma = 0.004$. (a) LUT. (b) ISW.

measurement noise that was intentionally introduced to the retrieved simulated data.

The next set of results deals with the impact of temperature mismatches between profiling and target devices where the model is built at 0 °C while the attack is performed in different temperatures. A new device was used for profiling and attack. The results depicted in Fig. 10 show that the impact of temperature mismatch is more prominent in GLUT. In all the circuits, the attack is hindered if there is such a temperature mismatch. For example, targeting ISW, if the attack is performed at 65 °C it needs around nine times more traces compared to the case when launched at 0 °C if in both cases the model has been built at 0 °C. Such rate is 4× for RSM-LOG and 1.7× for LUT.

The takeaway from these observations is that to decrease the number of traces required for the template attack, first, the training and matching devices must be at the same temperature. Then, ideally, aging should be balanced. The latter is, however, not always achievable, since trainee and attacker activities are not identical.

F. Effect of Process Mismatch

The process variations that occur during the fabrication should be taken into account while assessing the attack success. Indeed, the results reported so far were not including the process mismatch as in this article we are to show how each of operational, environmental, and process imperfections contribute to SR of the template attack. Fig. 11 depicts how much process mismatch between the profiling and target devices contributes to the number of traces needed for SR = 80% when there is already an aging mismatch between those devices. Here, we conducted Monte Carlo simulations for 11 chips using a Gaussian distribution: transistor gate length L: $3\sigma = 10\%$; threshold voltage V_{th}: $3\sigma = 30\%$, and gate-oxide thickness t_{OX}: $3\sigma = 3\%$.

The template was built based on the power traces extracted from chip 2 at 105 °C when it was new (i.e., age = 0). Then the number of traces for the SR of 80% for attacking each of the other ten chips at different ages was computed. Fig. 11 shows the average of the number of the traces required to reach SR = 80% for LUT and ISW when there are both process and aging mismatches. The takeaway point is that process mismatch inhibits the template attack as expected. Let us illustrate this conclusion quantitatively. On average, for LUT, the process plus aging mismatch results in 4.8% increase of the number of traces over 20 weeks of usage on top of the aging mismatch increase of 8% to attain 80% success. In contrast, ISW needs 11.75% more traces when both aging and process mismatch are taken into account on top of the 20.47% increase of traces due to aging only.

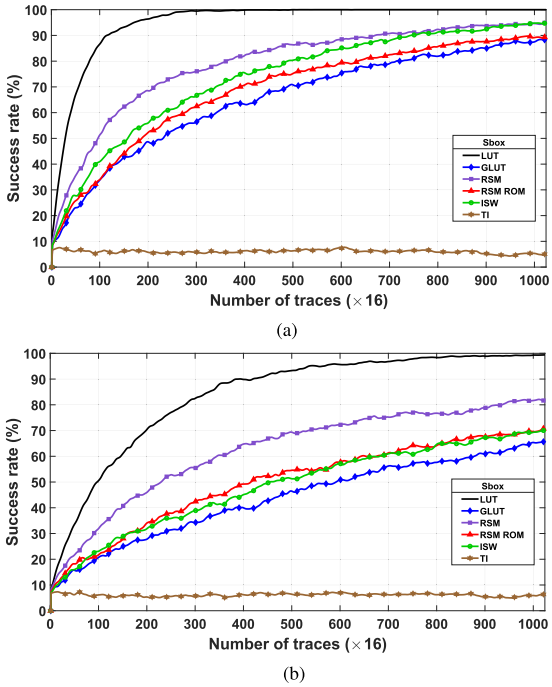

Fig. 12. SR with different noise level. Profiling temp. = 105 °C, attack temp. = 105 °C. Both devices are new (no aging). (a) $\sigma = 0.008$. (b) $\sigma = 0.016$.

G. Noise Level Impact

The noise level is affected by the number of the components running in parallel as well as the complexity of the device. In reality, the noise level can be higher for the cryptographic module running under a complex system. Generally, attack complexity changes with noise. The aim of this experiment is to find out how the relative security changes in different noise levels. To show the impact of noise, we repeated the experiments with the Gaussian noises with $\sigma \in \{0.008, 0.016\}$. Fig. 12 shows the outcome. As expected, with the increase of noise, the attack becomes more difficult. However, the relative

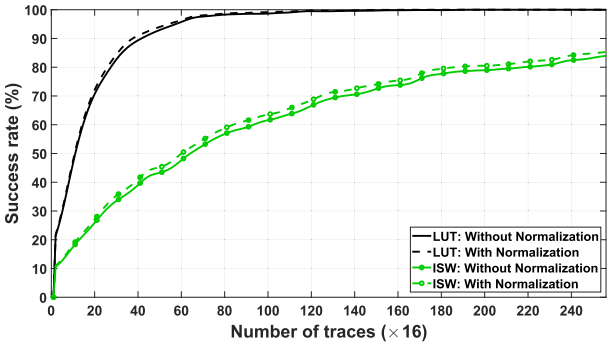

Fig. 13. SR after 1000 attacks on 20-week old circuits with and without applying normalization. Profiling and attack temp. are 105 °C, and $\sigma = 0.004$. Profiling device is new (i.e., not aged).

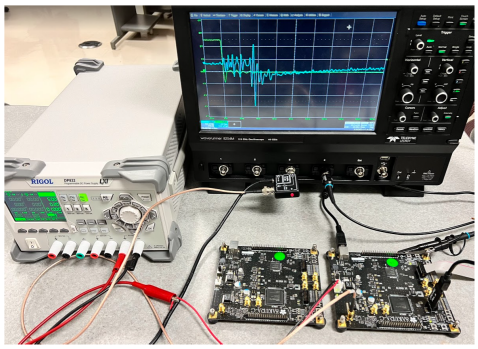

Fig. 14. Hardware setup for collecting power traces to launch template attack.

security of the considered implementations remains similar in all three noise levels, that is, TI remains as the most secure one, and then the GLUT.

H. Improving Attack Success Rate With Normalization

Template normalization [76] can be taken into account to reduce discrepancies between profiling and matching phases. Here both 2^n templates $\hat{p}$ (for all the values of the key) and 2^n online distributions $\tilde{p}$ are transformed by homothety such that they have the same *zero mean* and *unit variance*. We refer to Fig. 13 to depict how our results are affected by such normalization when the template device is new while the target device is 20 weeks old. As depicted, the normalization does not affect the number of traces required to break the key significantly. Here we showed the results for ISW and LUT. The other targeted circuits followed the very same trend.

VI. Template Attacks on Real Silicon

This section validates our findings in the real silicon, more specifically in FPGAs.

A. Experimental Setup for the Attacks on Real Silicon

Fig. 14 shows an overview of our FPGA setup. The boards are powered by an external power source. The power traces are collected from the dedicated power measuring pin in the SAKURA-G board while running the operations. A Tektronix WaveRunner 825AM oscilloscope has been utilized to measure the power traces. The plaintexts are sent and ciphertexts are collected by a separate FPGA working as the controller. We implemented each of the protected and unprotected circuits (one at a time) in two FPGA boards (FPGA_a and FPGA_b) and used the power traces from FPGA_a to build the template and used such template to attack the target circuit implemented on FPGA_b. Note that these experiments confirm our simulation results of template attacks in the considered protected and unprotected circuits rather than showing the aging impact.

B. Power Templates Collected From FPGA

Fig. 15 shows the 16 classes of power templates collected from FPGA_a for each targeted implementation. Each template is used to attack the same circuit implemented on FPGA_b. Comparing the power traces from the ASIC implementation (here HSpice) shown in Fig. 1 with the power traces extracted from FPGA confirms a similar trend in both implementations. In both cases, the unprotected circuit (i.e., LUT) exhibits the highest inter-class differences, making this circuit most vulnerable to the template attack compared to its masked protected counterparts. On the other hand, for TI the power classes are not differentiable from each other. Other implementations also follow a similar trend between ASIC simulation and FPGA implementations. This observation confirms the relevance to ASIC of our HSpice simulations. Note that due to the technology differences as well as the LUT-based structure of circuit implementations in FPGA, each implementation shows a different power consumption in FPGA versus HSpice; thus non-identical power traces. However, the inter-class variance of power traces follows a very same trend.

To depict the effect of process variations, we show the inter-class variance of power traces for all unprotected and masked-protected circuits implemented in both FPGAs in Fig. 16. As discussed earlier, the process variation hinders the attacks. As depicted in Fig. 16, the variance of power traces in these two FPGAs follows a very similar trend. This confirms the insignificant impact of process variations in our attack. Also this similarity depicts that the measurement noise impact is similar for both devices. As shown in Fig. 16, the power traces in RSM-ROM have less inter-class variance than all other implementations except the TI. Thereby, attacking RSM-ROM is more difficult than other S-Boxes (other than TI).

C. Attack Success Rate on FPGA

In this experiment, a template attack is launched on FPGA_b using the power model built based on power traces extracted from FPGA_a. Attacks are performed by increasing the number of traces by 16 each time; increasing from $1(\times 16)$ to $200(\times 16)$ traces. To have a fair comparison with the attacks launched on HSpice traces, each attack is repeated 1000 times using randomly collected traces from FPGA_b in each experiment. This removes the bias in attack success as not all traces result in a rewarding scenario for the adversary.

Fig. 17 shows the SR of the launched attacks on unprotected and protected circuits. The trend shown here is very similar to the trend observed in Fig. 4. As expected, in both cases, among all targeted circuits, the unprotected LUT is the most insecure circuit against the template attack while TI is the most secure one. The only difference between these two experiments is GLUT. This can be explained by the high number of fan-out branches in GLUT compared to other S-Box implementations targeted in this study; the more the fan-outs the more the load per gate, and thus the more the leakage and the easier the attack. Referring to Fig. 18, we observe that RSM-LOG has the highest average fan-out nets among masked protected designs, followed by GLUT. These three protected designs precisely reflect the SR trends seen in Fig. 17 comparing their average fan-out nets. RSM-ROM and TI, on the other hand, are hard to attack due to the complexity of their masked designs. Note

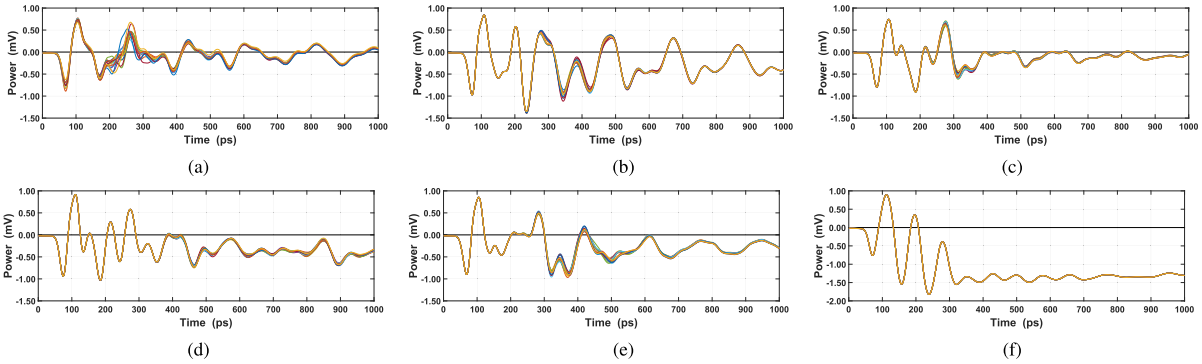

Fig. 15. Power templates of different S-Box implementations from FPGA$_a$. (a) LUT. (b) GLUT. (c) RSM-LOG. (d) RSM-ROM. (e) ISW. (f) TI.

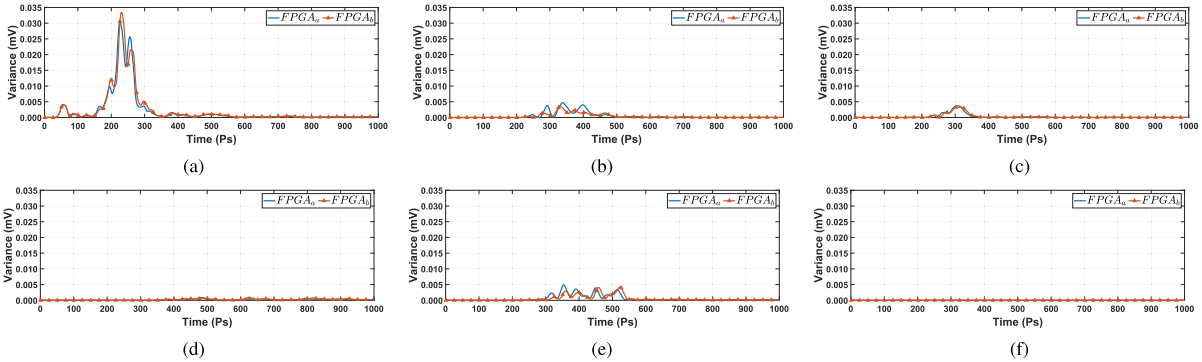

Fig. 16. Inter-class variance of power templates from S-Box operations collected from FPGA$_a$ and FPGA$_b$. (a) LUT. (b) GLUT. (c) RSM-LOG. (d) RSM-ROM. (e) ISW. (f) TI.

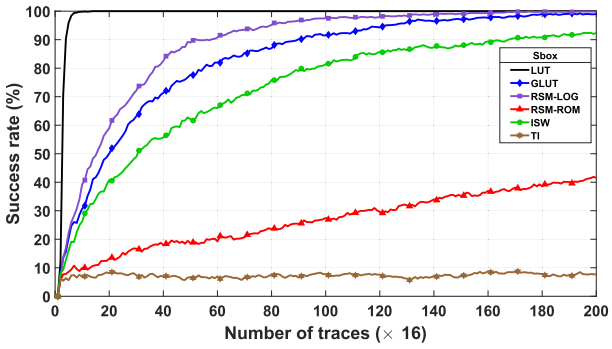

Fig. 17. SR after 1000 attacks on different S-Box implementations while profiling was performed based on the FPGA$_a$ power traces and the attack was launched on FPGA$_b$.

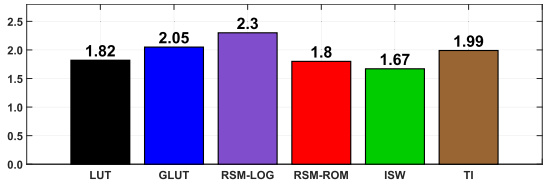

Fig. 18. Average fan-out nets of the S-Boxes in FPGAs.

that, in HSpice, the wires were ideal (no *RC* model for wires); thus, in HSpice, the impact of such high fan-outs in GLUT is not seen and GLUT was shown to be more secure than other circuits (except TI) in HSpice. This finding highlights that the nature of the logic style is the primary factor for selecting a logic style, but that the implementation details (such as the "fan-out") can, as a secondary factor, affect the ordering between logic styles. The takeaway points from this set of results is that attacks on both simulation and real-silicon traces show a similar trend, thus confirming the validity of the HSpice simulations.

VII. Summary of Findings

This section summarizes the findings throughout this study. We found that as expected masked S-Boxes have lower SNR compared to the base LUT S-Box in most cases. However, the result shows that there is a "big reward" disabling the masks for protected masked S-Boxes. Indeed, the leakage is exacerbated by increasing the SNR, with respect to "no countermeasure" case (LUT). This issue is more alarming for TI. We found that considering the attack SR at 80% (HSpice simulation), security level provided by masked S-Boxes quantize as TI, RSM-ROM, ISW, GLUT, and RSM-LOG, respectively, from high to low.

Our experimental results show that the number of traces required to attain SR equal to 80% for profiling attacks (template attack here) increases with aging. Such effect is more prominent in higher temperatures. Indeed, we observed that template attacks profiled and launched on higher temperatures require higher traces for a successful attack. Moreover, we found that template attacks profiled and launched on similar temperature require less traces (thus a more successful attack) compared to the case where there is temperature misalignment between the profiling and target devices.

Regarding process variation mismatch, as shown, it inhibits the template attack. Aging mismatch on top of the process mismatch even makes template attacks harder. The results show that template attack becomes more difficult with the increase of noise. However, the relative security of the considered masked implementations remains similar even when the noise level changed (tested on $\sigma \in \{0.004, 0.008, 0.016\}$).

Note that the overhead in an attack, in terms of number of traces to extract the key, is directly related to the overhead in terms of attack cost. In practice, one shall bear in mind that attackers behave rationally: they refrain to execute an attack if its gain is not positive. Unlike computational power

TABLE II
RATING FOR ELAPSED TIME IDENTIFICATION EXPLOITATION

Time \ Attack phase	Identification	Exploitation
< one hour	0	0
< one day	1	3
< one week	2	4
< one month	3	6
> one month	5	8
Not practical	*	*

growth in cryptanalyses, more time to perform an attack yields a linear manpower increase. For instance, a "+20%" in terms of attack time directly turns into "+20%" more budget to realize the attack. The growth of the attack time and budget can be either a deterrent or a weapon. In order to provide concrete usage of quantitative security estimation, a summary of the security rating under common criteria is given. The elapsed time devoted by the evaluator is tackled in [77, Annex 7, Sec. 4.2. p. 190] for smartcards and similar devices. It is recalled in Table II, considering "identification" and "exploitation" phases. The former phase constitutes the time required to collect traces to build the templates (under known keys), whereas the latter constitutes the time required to perform the online attack (key extraction). The scores given in the two columns quantify the difficulty of the attack—the large the score the more difficult the attack. The exact values attributed to the scores result from a consensus established by the common criteria community (refer to [77]).

To comply to a given common criteria quotation, it could thus be relevant to use a better countermeasure if the quotation is just below one of thresholds of Table II. For instance, it is possible to choose the most adequate (i.e., less expensive) countermeasure to reach the "not practical" rating. Alternatively, if the security level (across all environmental and aging conditions) falls well between two thresholds, then it could be opportunistic to resort to a less costly countermeasure, without demoting the security level according to common criteria scale.

Moreover, attacking the same masked architectures in FPGA showed that the power traces collected from the real hardware (FPGA) follow the very same trend as the ones extracted from HSpice regarding the order of their power's variance. Finally, attacks launched on both simulation and real-silicon traces show a similar trend (in terms of security of the considered masked architectures against template attack), confirming the validity of the HSpice simulations.

VIII. CONCLUSION

The state-of-the-art masking schemes comprise multiple proposals of implementation. In this article, we compared the resiliency of six representative schemes against template attacks. They feature different costs, in inverse proportion of their innate security (except for GLUT where its high rate of fan-outs per gate results in more leakage). We further quantitatively investigated how the success of a template attack is affected when there is a misalignment between the target and profiling devices in terms of temperature and aging. The experimental results confirm that such misalignment hinders the attack in both unprotected and masked circuits. Among the investigated masked circuits, RSM-ROM is affected the most by aging while TI experiences less aging-induced impacts. We also showed (based on FPGA experiments) that the practicality of template attacks depends on the masking scheme. In practice, netlists with a long propagation time (e.g., RSM-ROM) have typically (in practice) a different leakage profile between training and attacking phase, which degrades the attack efficiency. On the other hand, the incompleteness property of TI makes it secure against template attacks.

REFERENCES

[1] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Proc. Cryptograph. Hardw. Embedded Syst. (CHES)*, 2002, pp. 13–28.

[2] J. Blömer, J. Guajardo, and V. Krummel, "Provably secure masking of AES," in *Proc. Sel. Areas Cryptography*, 2004, pp. 69–83.

[3] NIST. (Sep. 2021). *Masked Circuits for Block-Ciphers*. [Online]. Available: https://csrc.nist.gov/Projects/masked-circuits

[4] A. Heuser et al., "Good is not good enough-deriving optimal distinguishers from communication theory," in *Proc. CHES*, 2014, pp. 55–74.

[5] S. Bhasin et al., "Mind the portability: A warriors guide through realistic profiled side-channel analysis," in *Proc. NDSS*, 2020, pp. 1–14.

[6] R. Breuer and I. Levi, "How bad are bad templates? Optimistic design-stage side-channel security evaluation and its cost," *Cryptography*, vol. 4, no. 4, p. 36, Dec. 2020.

[7] A. Moradi, "Side-channel leakage through static power: Should we care about in practice?" in *Proc. CHES*, 2014, pp. 562–579.

[8] T. Moos et al., "Static power side-channel analysis of a threshold implementation prototype chip," in *Proc. DATE*, 2017, pp. 1324–1329.

[9] T. De Cnudde, M. Ender, and A. Moradi, "Hardware masking, revisited," in *Proc. CHES*, 2018, pp. 123–148.

[10] T. Moos, "Static power SCA of sub-100 nm CMOS ASICs and the insecurity of masking schemes in low-noise environments," in *Proc. CHES*, 2019, pp. 202–232.

[11] I. Levi et al., "Ask less, get more: Side-channel signal hiding, revisited," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 4904–4917, 2020.

[12] D. D. Hwang et al., "AES-based security coprocessor IC in 0.18-μm CMOS with resistance to differential power analysis side-channel attacks," *IEEE J. Solid-State Circuits*, vol. 41, no. 4, pp. 781–792, 2006.

[13] K. Tiri et al., "Prototype IC with WDDL and differential routing—DPA resistance assessment," in *Proc. CHES*, vol. 3659, 2005, pp. 354–365.

[14] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing The Secrets of Smart Cards*. Cham, Switzerland: Springer, 2006.

[15] B. Fadaeinia, M. T. Hasan Anik, N. Karimi, and A. Moradi, "Masked SABL: A long lasting side-channel protection design methodology," *IEEE Access*, vol. 9, pp. 90455–90464, 2021.

[16] R. Vadlamani et al., "Multicore soft error rate stabilization using adaptive dual modular redundancy," in *Proc. DATE*, 2010, pp. 27–32.

[17] C. Carmichael. (Jul. 12, 2016). *Triple Module Redundancy Design Techniques for Virtex FPGAs*. [Online]. Available: https://www.xilinx.com/support/documentation/application_notes/xapp197.pdf

[18] M. Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," in *Proc. VTS*, 1999, pp. 86–94.

[19] S. Bhunia and M. Tehranipoor, *Hardware Security: A Hands-on Learning Approach*. Amsterdam, The Netherlands: Elsevier Science, 2018.

[20] M. M. Kermani and A. R. Masoleh, "A structure-independent approach for fault detection hardware implementations of the advanced encryption standard," in *Proc. FDTC*, 2007, pp. 47–53.

[21] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Fault detection structures of the S-boxes and the inverse S-boxes for the advanced encryption standard," *J. Electron. Test.*, vol. 25, nos. 4–5, pp. 225–245, Aug. 2009.

[22] M. M. Kermani et al., "A lightweight concurrent fault detection scheme for the aes S-boxes using normal basis," in *Proc. CHES*, 2008, pp. 113–129.

[23] M. T. H. Anik et al., "Detecting failures and attacks via digital sensors," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 7, pp. 1315–1326, 2020.

[24] M. R. Muttaki et al., "FTC: A universal sensor for fault injection attack detection," in *Proc. HOST*, 2022, pp. 117–120.

[25] S. Das et al., "RazorII: In situ error detection and correction for PVT and SER tolerance," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, 2008.

[26] M. Ebrahimabadi et al., "DELFINES: Detecting laser fault injection attacks via digital sensors," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 43, no. 3, pp. 774–787, 2023.

[27] F. Amiel et al., "Passive and active combined attacks: Combining fault attacks and side channel analysis," in *Proc. FDTC*. IEEE, 2007, pp. 92–102.

[28] K. J. Kulikowski et al., "DPA on faulty cryptographic hardware and countermeasures," in *Proc. FDTC*. Springer, 2006, pp. 211–222.

[29] Á. Kiss et al., "Algorithmic countermeasures against fault attacks and power analysis for RSA-CRT," in *Proc. COSADE*. Springer, 2016, pp. 111–129.

[30] T. Schneider, A. Moradi, and T. Güneysu, "Parti–towards combined hardware countermeasures against side-channel and fault-injection attacks," in *Proc. Adv. Cryptology–CRYPTO*, 2016, pp. 302–332.

[31] M. T. H. Anik, S. Guilley, J.-L. Danger, and N. Karimi, "On the effect of aging on digital sensors," in *Proc. VLSID*, 2020, pp. 189–194.

[32] S. Khan et al., "NBTI monitoring and design for reliability in nanoscale circuits," in *Proc. DFTS*, 2011, pp. 68–76.

[33] F. Oboril et al., "Extratime: Modeling and analysis of wearout due to transistor aging at microarchitecture-level," in *Proc. DSN*, 2012, pp. 1–12.

[34] S. Chowdhury et al., "Physical security in the post-quantum era: A survey on side-channel analysis, random number generators, and physically unclonable functions," *J. Cryptograph. Eng.*, Feb. 2021.

[35] L. Batina et al., "In hardware we trust: Gains and pains of hardware-assisted security," in *Proc. DAC*, 2019, pp. 1–4.

[36] N. Bruneau, S. Guilley, A. Heuser, D. Marion, and O. Rioul, "Optimal side-channel attacks for multivariate leakages and multiple models," *J. Cryptograph. Eng.*, vol. 7, no. 4, pp. 331–341, Nov. 2017.

[37] N. Karimi, S. Guilley, and J.-L. Danger, "Impact of aging on template attacks," in *Proc. GLSVLSI*, 2018, pp. 455–458.

[38] A. Heuser et al., "Side-channel analysis of lightweight ciphers: Does lightweight equal easy?" in *Proc. RFIDSec*. Springer, 2016, pp. 91–104.

[39] Y. Feiet et al., "A statistics-based success rate model for DPA and CPA," *J. Cryptograph. Eng.*, vol. 5, no. 4, pp. 227–243, 2015.

[40] J. Borghoff et al., "Prince—A low-latency block cipher for pervasive computing applications," in *Proc. ASIACRYPT*, 2012, pp. 208–225.

[41] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "GIFT: A small present," in *Proc. CHES*, 2017, pp. 321–345.

[42] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, "Ascon v1. 2," *CAESAR Competition*, vol. 5, no. 6, p. 7, 2016.

[43] S. Banik et al., "Midori: A block cipher for low energy," in *Proc. ASIACRYPT*, 2015, pp. 411–436.

[44] K. Ramezanpour et al., "Active and passive side-channel key recovery attacks on Ascon," in *Proc. NIST LWC Workshop*, 2020, pp. 1–27.

[45] A. Jana, "Differential fault attack on Ascon cipher," Cryptol. ePrint Arch., Tech. Paper 2023/1923, 2023. [Online]. Available: https://eprint.iacr.org/2023/1923

[46] A. C. Canto, J. Kaur, M. M. Kermani, and R. Azarderakhsh, "Algorithmic security is insufficient: A comprehensive survey on implementation attacks haunting post-quantum security," 2023, *arXiv:2305.13544*.

[47] Y.-T. Kuo et al., "A lattice attack on crystals-Kyber with correlation power analysis," in *Proc. ICISC*. Springer, 2023, pp. 202–220.

[48] P. Ravi et al., "Fiddling the twiddle constants-fault injection analysis of the number theoretic transform," in *Proc. CHES*, 2023, pp. 447–481.

[49] A. Sarker et al., "Hardware constructions for error detection of number-theoretic transform utilized in secure cryptographic architectures," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 3, pp. 738–741, 2018.

[50] A. Sarker et al., "Error detection architectures for hardware/software co-design approaches of number-theoretic transform," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 7, pp. 2418–2422, Nov. 2022.

[51] S. Khan et al., "Efficient, error-resistant NTT architectures for CRYSTALS-kyber FPGA accelerators," in *Proc. VLSI-SoC*. IEEE, 2023, pp. 1–6.

[52] S. Özeren and O. Yayla, "Methods for masking crystals-kyber against side-channel attacks," in *Proc. ISCTürkiye)*, 2023, pp. 1–6.

[53] J.-L. Danger et al., "Overview of dual rail with precharge logic styles to thwart implementation-level attacks on hardware cryptoprocessors," in *Proc. SCS*, 2009, pp. 1–8.

[54] S. Guilley et al., "Evaluation of power constant dual-rail logics countermeasures against DPA with design time security metrics," *IEEE Trans. Comput.*, vol. 59, no. 9, pp. 1250–1263, Sep. 2010.

[55] M. T. H. Anik et al., "On the impact of aging on power analysis attacks targeting power-equalized cryptographic circuits," in *Proc. ASP-DAC*, 2021, pp. 414–420.

[56] E. Prouff and M. Rivain, "A generic method for secure SBox implementation," in *Proc. WISA*, 2007, pp. 227–244.

[57] D. Canright and L. Batina, "A very compact 'perfectly masked' S-box for AES," in *Proc. ACNS*, 2008, pp. 446–459.

[58] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*. Cham, Switzerland: Springer, 2002.

[59] S. Takarabt et al., "Formal evaluation and construction of glitch-resistant masked functions," in *Proc. HOST*, 2021, pp. 304–313.

[60] M. Nassar et al., "RSM: A small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs," in *Proc. DATE*, 2012.

[61] M. Nassar, S. Guilley, and J. Danger, "Formal analysis of the entropy/security trade-off in first-order masking countermeasures against side-channel attacks," in *Proc. INDOCRYPT*, 2011, pp. 22–39.

[62] C. Claude et al., "Side-channel indistinguishability," in *Proc. HASP*, 2023, pp. 1–8.

[63] M. Giaconia et al., "Area and power efficient synthesis of DPA-resistant cryptographic S-Boxes," in *Proc. VLSI Design*, 2007, pp. 731–737.

[64] Y. Ishai, A. Sahai, and D. Wagner, "Private circuits: Securing hardware against probing attacks," in *Proc. CRYPTO*, 2003, pp. 463–481.

[65] A. Covic, F. Ganji, and D. Forte, "Circuit masking schemes: New hope for backside probing countermeasures?" *SRC TECHCON*, Sep. 2020.

[66] N. Courtois, D. Hulme, and T. Mourouzis, "Solving circuit optimisation problems in cryptography and cryptanalysis," *IACR Cryptol. ePrint Arch.*, vol. 2011, p. 475, Jan. 2011.

[67] D. B. Roy et al., "CC meets FIPS: A hybrid test methodology for first order side channel analysis," *IEEE Trans. Comput.*, vol. 68, pp. 347–361, 2019.

[68] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold implementations against side-channel attacks and glitches," in *Proc. 8th Int. Conf. Inf. Commun. Secur. (ICICS)*, vol. 4307, 2006, pp. 529–545.

[69] H. Gross et al., "Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order," in *Proc. TIS*, 2016, p. 3.

[70] G. Cassiers et al., "Hardware private circuits: From trivial composition to full verification," *IEEE Trans. Comput.*, vol. 70, no. 10, pp. 1677–1690, 2021.

[71] Automotive Electronics Council. (2007). *AEC-Q100, Rev-G: Failure Mechanism Based Stress Test Qualification For Integrated Circuits*. [Online]. Available: http://www.aecouncil.com/Documents/AEC_Q100_Rev_G_Base_Document.pdf

[72] H. Guntur et al., "Side-channel attack user reference architecture board SAKURA-G," in *Proc. GCCE*, 2014, pp. 271–274.

[73] *Information Technology—Security Techniques—Test Tool Requirements and Test Tool Calibration Methods for Use in Testing Non-invasive Attack Mitigation Techniques in Cryptographic Modules—Part 1: Calibration Method and Apparatus*, Standard ISO/IEC 20085-1:2020, ISO/IEC JTC 1/SC 27/WG 3, 2020, p. 17. [Online]. Available: https://www.iso.org/standard/70082.html

[74] N. Veyrat-Charvillon et al., "An optimal key enumeration algorithm and its application to side-channel attacks," in *Proc. Sel. Areas Cryptogr.*, 2012, pp. 390–406.

[75] F. Niknia et al., "Aging effects on template attacks launched on dual-rail protected chips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 41, no. 5, pp. 1276–1289, 2022.

[76] M. A. Elaabid and S. Guilley, "Portability of templates," *J. Cryptograph. Eng.*, vol. 2, no. 1, pp. 63–74, May 2012.

[77] (May 2021). *Cybersecurity Certification. EUCC, a Candidate Cybersecurity Certification Scheme To Serve As a Successor To the Existing SOG-IS*. [Online]. Available: https://www.enisa.europa.eu/publications/cybersecurity-certification-eucc-candidate-scheme-v1-1.1/at_download/fullReport