Beware Your Standard Cells! On Their Role in Static Power Side-Channel Attacks

Jitendra Bhandari, Likhitha Mankali, Mohammed Nabeel, Ozgur Sinanoglu, *Senior Member, IEEE*,
Ramesh Karri, *Fellow, IEEE*, and Johann Knechtel, *Member, IEEE*

***Abstract*—The increase in leakage power from advanced tech nodes elevates the risk of static power side-channel (S-PSC) attacks. While protective measures exist, they involve a security-cost trade-off. Hardware Trojans, particularly PSC-based ones, represent another significant threat. Despite acknowledging the link between static power leakage, advanced tech nodes, and vulnerability to S-PSC attacks, the role of the components at the heart of this sensitive interplay – the standard cells – has not been extensively studied in commercial-grade IC design. We analyze this relationship for commercial 28nm and 65nm nodes using a regular AES design. Our CAD framework permits design optimization while assessing S-PSC vulnerability. Contrary to the belief that high-performance designs are more vulnerable, we find timing constraints and threshold-voltage cell ratios are pivotal factors. Also, we discover that an attacker can deploy highly effective, stealthy PSC-based Trojans without any gate overheads or compromising timing paths.**

***Index Terms*—Hardware Security, Power Side-Channel, CAD**

I. Introduction

As technology in general advances, there is significant innovation in microelectronics to meet arising demands. Modern technologies for integrated circuits (ICs) manufacturing, also referred to as technology nodes, offer a large range of so-called standard cells which are providing Boolean algebra as well as memories. Importantly, there are multiple versions of the same standard cells but with different profiles for power consumption, performance, and area (PPA). PPA requirements vary across different IC applications, hence such different profiles are are needed. High-end applications like GPUs prioritize performance and favor fast cells for timing closure, despite power overheads, due to their design complexity and long exploration runtimes. On the other hand, embedded/edge devices with power constraints prefer performance trade-offs to save power, supported by lower design complexity that eliminates the need for fast cells as a timing fix. Different PPA profiles are enabled by transistor-level tuning, including but not limited to tuning of the threshold voltage (VT).

Side-channel attacks represent a significant threat for the security of ICs, even if their underlying logic is cryptographically robust. Prior research has demonstrated that hardware can leak information through various side-channels, such as electromagnetic, timing, power, and others [1]–[3].

Power side-channel attacks in particular have been extensively studied by the community [4]. Such attacks can be conducted in two different ways, namely by either focusing on dynamic power consumption while the IC is running or by focusing on static power consumption while the IC is halted. The latter, static power side-channel (S-PSC) attacks, are especially concerning for modern technology nodes [5].

Hardware Trojans represent a substantial security risk. Trojans are malicious alterations to ICs and have two parts: a trigger and a payload. Under rare and carefully crafted conditions, the trigger activates the payload, which then executes malicious actions on parts of the IC. Such actions can cause system failures or leak sensitive information.

Hardware Trojans have attracted attention throughout decades [6]; this has only become more pronounced due to the outsourced supply chain of ICs. Since Trojan modifications can occur at any early point in the supply chain (i.e., design and manufacturing), detecting hardware Trojans before deployment in the field is a challenge for secure and trustworthy ICs. This holds true despite that outsourced assembly and test facilities (OSATs) verify the proper IC functionality, as the trigger condition is unknown and often based on specific, rare conditions, which are unlikely covered during regular testing.

Scope and Contributions: Here, we study the role of standard cells, in particular their VT level and timing constraints, on S-PSC attacks from both attack and defense perspectives.

We propose a simple, side-channel-based Trojan that makes it easy for attackers to extract the secret key of an AES circuit using the S-PSC attack. The Trojan is created by replacing certain register cells with functionally equivalent cells that have a low/ultra-low threshold-voltage profile (LVT/ULVT). While these cells switch faster, they leak higher currents, making them vulnerable to S-PSC attack. [1] This Trojan is practical and stealthy, requiring no design changes.

The proposed Trojan highlights a challenge in protecting ICs from side-channel attacks: balancing performance and S-PSC vulnerability when using ULVT/LVT and RVT/HVT cells, a trade-off previously overlooked. We develop a security-focused design-space exploration framework using commercial CAD tools, for providing design guidelines and implementing the Trojan. This framework is essential for design-space exploration from both offense and defense perspectives.

This work was supported in part by the NYU Center for Cybersecurity (CCS) and the NYUAD CCS.

J. Bhandari, L. Mankali, and R. Karri are with New York University, New York City, NY, 11201 USA. E-mail: {jb7410, lm4344, rkarri}@nyu.edu

J. Knechtel, M. Nabeel, and O. Sinanoglu are with New York University Abu Dhabi, UAE. E-mail: {johann, mtn2, ozgursin}@nyu.edu

[1] PSC attacks do not benefit per se from higher power consumption. As we show in this study, resilience (or lack thereof) against S-PSC attacks is based on an IC's ratio for utilizing different VT cells, the resulting interspersion of power profiles, and the timing constraints which dictate these ratios.

Our contributions are threefold:

1) We propose a security-focused design-space exploration framework that utilizes commercial CAD tools. (Our intention is to make our scripts publicly available after omitting technology-specific configurations, as the details of these libraries are confidential.)
2) We put forward a straightforward, highly effective concept for zero-gate Trojans. This has practical applications in facilitating S-PSC attacks.
3) We analyze the security-versus-PPA design-space across multiple technology nodes with all their VT cell options.

Section II provides an overview of fundamental concepts and inspirational aspects for this study, whereas Section III offers a review of related works. Our methodology is described in detail in Section IV, and Section V presents our experimental studies. Section VI offers our conclusions and perspectives.

II. Background

A. Power Side-Channel Attacks

Power side-channel attacks exploit variations in a device's power consumption to extract sensitive data like cryptographic keys. By analyzing this consumption and understanding the device's operations, attackers can infer the internal workings and associate actual power use with possible secret data profiles. This method is well-researched in cryptographic hardware implementation due to its potential security vulnerabilities [4].

There are various power analysis attacks, such as simple power analysis (SPA), differential power analysis (DPA), and correlation power analysis (CPA) [7]. SPA profiles power consumption during crypto operations, while DPA compares power consumption between similar operations to derive a secret key. CPA uses Pearson correlation coefficient to relate predicted and actual power profiles, inferring data and keys.

Dynamic power side-channel attacks (D-PSCA) extract sensitive information by monitoring and analyzing power consumption during a device's operation. Static power side-channel attacks (S-PSCA) analyze power usage when a device is at a halt or "idle" state, leveraging data-dependent power profiles even without data processing. D-PSCA demands precise timing, whereas S-PSCA needs sophisticated equipment but no timing synchronization, just a halted clock. As static/leakage power becomes more significant with advanced technology nodes, the importance of S-PSCA grows [8], [9].

Designers counteract attacks by integrating masking [10], shuffling, and balancing [11], [12] into IC design. These measures complicate extraction of secret keys and reduce power consumption variations during cryptographic operations, making it harder to distinguish power traces. While studies attempt to minimize the leakage of information through the power side-channel, doing so incurs overhead. However, this may not always be practical, particularly when the cost of silicon per mm^2 is high. It is necessary to consider the trade-offs between mitigation and PPA overhead during design time.

B. Hardware Trojans

Hardware Trojans have been studied extensively in the research community to be aware of any unexpected behavior that can occur due to intentional, malicious modifications in the hardware. Such modifications can lead to various sorts of threats like denial of service (DoS), system failure, unwanted privileged access, et cetera [6], [13]. This sort of malicious modification can be realized by some untrusted entity at any stage of the design and manufacturing cycle, which makes it difficult to avoid with the current distributed ICs supply chain, where IPs from the design house travel all the way offshore for fabrication. Naturally, such threats can lead to some serious incidents in critical missions where the reliability of the underlying IC is of utmost importance.

TABLE I

Variation of static power of a DFF in two different commercial nodes, reported in nW. The power depends on both the VT cells as well as input and output data.

CLK	D	Q	28nm Node			65nm Node		
			LVT	RVT	HVT	LVT	RVT	HVT
0	0	0	101.6	8.1	0.9	78.1	23.0	16.3
0	0	1	171.3	12.8	1.2	118.4	26.1	15.5
0	1	0	161.6	11.9	1.1	106.9	29.5	20.6
0	1	1	135.0	9.9	1.1	113.8	28.3	18.6
1	0	0	104.8	8.2	0.9	86.6	24.1	16.7
1	0	1	117.1	9.1	0.9	136.2	29.2	17.5
1	1	0	142.0	10.5	1.0	88.4	25.9	18.1
1	1	1	109.4	8.3	0.9	103.8	26.6	17.8

Hardware Trojans can be difficult to detect and remove because they can be designed to activate only under specific conditions, such as a particular input signal or after a certain period of time. It has been shown that, to make some Trojans avoid conventional testing, it requires some rare trigger conditions to get activated [14]. Since such malicious modifications will be done on top of the baseline design, there are limitations for placement and routing; Trojans should require as few gates as possible and occupy as small an area as possible.

C. Leakage Power for Modern Technologies

Commercial technology nodes offer diverse standard cell versions for different PPA demands. These cells have unique physical properties, like area, power consumption, and propagation time, facilitating optimization for design constraints. The inclusion of low and ultra-low VT cells enhances performance, especially for time-constrained paths.

Although faster, these LTV/ULVT cells leak more static power than standard cells (Table I). For the 28nm node, the increase in leakage power when going from HVT to RVT as well as when going from RVT to LVT cells for a D-flip-flop (DFF) is a factor of around 10x, and in total (i.e., when going from HVT to LVT) it is a factor of around 120x. For the 65nm node, the increase is much less pronounced, with a factor of around 1.5x when going from HVT to RVT, but with a factor of still around 4x when going from RVT to the fastest LVT cell, and a corresponding factor of around 6x in total.

D. Design Automation and Security as New Objective

CAD tools have evolved to be more complex and adaptable, and are guided by designers to optimize based on past designs. These tools use heuristics and user constraints to generate an optimized netlist meeting the user's specifications. As these constraints traditionally only target PPA goals, and there are no clear standards for formulating other types of constraints, security concerns are not considered at this stage.

Accordingly, there are few if any commercial settings where one considers security first hand – this renders IC layouts susceptible to various threats that can be exploited in the field, like PSC attacks. Prior research work advocated considering security objectives during the design phase, e.g., Recent to protect against Trojans and other threats [15]–[17]. However, as indicated, the important role of standard cells has not been studied thoroughly yet in context of PSC attacks.

III. Related Works

[18] showed, for the first time, the potential of the S-PSC as a security threat. [9] have conducted one of the first practical experiments for S-PSC attacks using FPGAs. [8] highlighted the importance of leakage power and its effect on the PSC especially for more advanced nodes. [19] experimentally studied the role of various measurement factors on the success of S-PSC attacks. [20] have shown the important effect of aging on smaller technology nodes, further compromising the security of modern devices under S-PSC attacks. [21] conducted a multivariate analysis on the S-PSC. [5] studied the static and dynamic PSC for the 65nm node, where they have shown that S-PSC can undermine protection efforts even for this old node.

[12] studied various countermeasures against S-PSC attacks, e.g., balancing logic, which can come at considerable overheads. Their study is based on an 28nm IC. Furthermore, they have indicated on the important role of different VT cells. [22] proposed standard cell delay-based dual-rail precharge logic (SC-DDPL) as specific countermeasures against S-PSC, where NAND gates are used to implement every other logic stage, thus increasing the symmetry in the design and the resulting power profiles. This approach has the shortcoming/limitation of being not compatible with commercial CAD tool optimization flows. [23] propose a CAD framework aiming to reduce PSC vulnerability in a design by assigning vulnerability scores to different parts and iteratively optimize the design. A limitation of that work is that it assumes timing slacks are available for any security-centric optimization – this is not realistic in real applications where designs are pushed to limits to meet performance requirements. Furthermore, they do not conduct actual S-PSC evaluation.

[24] studied Trojan-based PSC attacks and countermeasures. [25], [26] show the feasibility of Trojan-based PSC attacks on FPGAs. [25] proposed a masking scheme as protection which incurs overheads. [27] proposed a CAD framework for insertion of PSC Trojans in the late design stages. [28] have proposed design procedures to reduce overhead and increase stealth of Trojans against PSC-based detection.

IV. Methodology

Our methodology aims to systematically study the role of VT cells in S-PSCA. While the approach applies to any IC design, we focus our case studies on 128-bit AES cipher, in a vanilla hardware implementation without protections in place. In addition, we adopted the concept of side-channel-based Trojans as a threat model, to emphasize the importance of our work. Such a Trojan does not introduce malicious activities, but rather enables an attack in a fielded system.

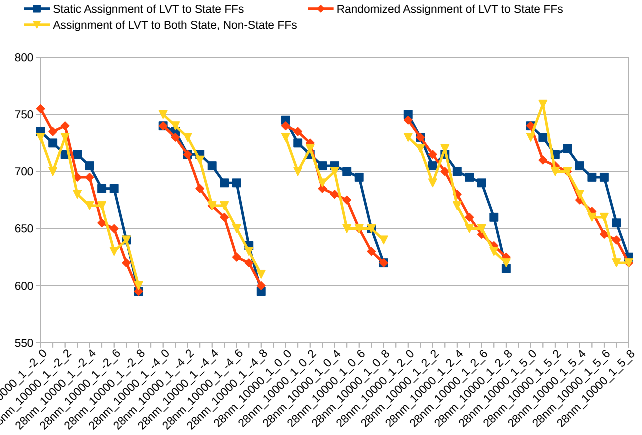

Fig. 1. Number of traces required until disclosure for different VT settings, for a commercial 28nm node, with the baseline timing constraint set to 1ns.

A. Exploratory Study

To commence, we conducted an empirical investigation that helped us determine the general role of various VT cells for S-PSCA. This study is based on a commercial 28nm technology node (the same outlined in Table I), and it follows our security-aware design-space exploration framework. (An overview as well as more details for the framework are provided in Sec. IV-C and IV-E, respectively.)

We started with a baseline implementation of the AES netlist, with timing constraints set to 1ns, which would fulfill the most basic performance requirements while using only HVT cells. Accordingly, the design got optimized for area and power but not for performance; indeed, an inspection of the netlist confirms that no LVT cells are instantiated.

Next, we revise the netlist as follows, which we also refer to as scenario *Randomized Assignment of LVT to State FFs*. For the *state_reg* registers – the registers holding the state/intermediate texts that a classical PSCA is aiming at – we stepwise replace more and more HVT with LVT cells. Since registers are grouped by bytes in the hardware implementation (as well as for the modeling part in the S-PSCA), we replace HVT FFs with LVT counterparts in the range of 0–8 registers per byte. For any given number of registers to replace, for each byte, we randomly select the actual FFs within each byte, i.e., we select here in no particular order of bits.

As expected (from Table I), such substitutions impact the static power profile of the design. This directly impacts the prospects for S-PSCA, as demonstrated in Fig. 1: the number of traces required until disclosure of the secret key (y-axis) decreases consistently with an increase of LVT registers employed in the registers (x-axis; usage of LVT registers increases, within each group of curves, from left to right).

The different cases listed along the x-axis of Fig. 1, are to be read as follows. First, there is the technology node (*28nm*), followed by the number of total traces available to the S-PSCA (*10000*), followed by the identifier of the secret key (*1*), followed by a variation of the baseline timing constraint in percentage (e.g., *-2*), and finally followed by the number of LVT registers employed per byte (e.g., *2*). Note that, for better readability, we have grouped all cases according to the varying timing constraints (ranging from -2% to +5%). Also note that, for a fair comparison, we employ the same random but fixed key across all cases, as well as the same random but fixed set of ciphertexts. Furthermore, we conduct a thorough sampling of the number of traces required until disclosure, which prevents us from reporting findings based on some "lucky guess" for some set of "relatively weak" ciphertexts. (More details for the S-PSCA experimentation are given in Sec. IV-E and V-A.)

As mentioned, for the baseline implementation where HVT registers are stepwise and randomly replaced with LVT counterparts – represented by the red curves covering the scenario 1) *Randomized Assignment of LVT to State FFs* in Fig. 1 – we find a consistent trend as in more LVT registers are employed, fewer traces are required until disclosure. Next, we sought to cross-check this observation through other scenarios (Fig. 1):

2) *Static Assignment of LVT to State FFs:* Here we replace FFs in a systematic way, namely in order of bits. For 3 HVT registers to be replaced by LVT counterparts, we replace bits 0, 1, and 2 for all the 16 state bytes. This scenario is important to understand the impact of bit-level position of LVT cells in the state bytes.
3) *Assignment of LVT to Both State, Non-State FFs:* Here we are modifying state registers as well as other, non-state registers. We replace the same number of cells for both state and non-state registers, and we follow the above strategies of static assignment for state registers versus randomized assignment across all other registers. This scenario is important to understand the role of LVT cells in state versus non-state registers, if any.

Across all three scenarios, we observe that neither the order of HVT versus LVT cells in state registers nor HVT versus LVT cells for state registers versus other/non-state registers impacts the trend. From the one corner case of 0 LVT cells per byte to the other, 8 LVT cells per byte, the reduction of traces required until disclosure is around 20%. Timing impacts the trend, as the more relaxed timing constraints are, the slower the reduction in resilience becomes.[2] Overall, replacing HVT registers with LVT registers within state bytes undermines the IC's resilience against S-PSCA.

B. Scope and Threat Model

Different VT cells can significantly impact the S-PSC. Notably, the replacement/modification of HVT cells with LVT cells for a few selected gates, namely the AES state registers, induces a significant reduction in resilience. Naturally, this issue can go two ways: while a cautious designer could use it to evaluate and enhance the resilience of IC designs in a security-aware framework, a malicious entity in the design/manufacturing process could exploit it to create a PSC Trojan, given they have similar tools.

We assume classical threat models for PSCA and hardware Trojans. That is, for PSCA we assume that adversaries can obtain power measurements through physical access to the IC. We further assume that attackers can observe – but not control – the externally accessible data – but not any internal data. Specifically, they can only observe the AES outputs, i.e., cipher-texts. For Trojans, the design can be compromised at any stage in the design and manufacturing cycle of ICs. Considering an untrusted foundry, e.g., adversaries can replace VT cells as outlined. This attack vector is not destructive, as no logic is removed or altered in any way, rendering this a stealthy attack, especially since there is zero impact on the area, no negative impact on timing, and only marginal impact on static power (as only a few selected gates are replaced).

Furthermore, we assume a threat model of collusion: this type of PSC-based Trojan assumes an attacker in the field, who is significantly supported by the reduced number of traces required until disclosure. There are many cases where an attacker would benefit from such inherently more leaky power side-channel.[3] From a defense viewpoint, the challenge lies in avoiding this increased leakage while adhering to strict PPA budgets. A straightforward avoidance of LVT cells is impractical for high-performance ICs with tight timing constraints. Thus, designers must carefully evaluate the susceptibility of the LVT cell usage scenario to PSC attacks.

C. Security-Aware Design-Space Exploration

1) Overview: Our security-aware design-space exploration framework Fig. 2 inputs the register-transfer level (RTL) of the design and different VT standard-cell libraries. It outputs the number of power traces until the secret key is disclosed. The more traces, the less susceptible the IC is to PSC attacks.

2) Research Questions: Our framework helps a designer check for possible vulnerabilities in their design against PSC attacks early on so that they can make necessary judgments to mitigate the vulnerability. However, doing so is more difficult than the attacker model of simply swapping selected cells to leak more information (Sec. IV-D). Because, as a defender, we have to look at various parts of the design and search for a relatively good solution with little impact on PPA. This challenge raises some questions in the mind of the designer:

1) How to make the design more resilient to power side-channel attacks without undermining PPA optimization?
2) Which part of the design plays a more dominant role when trying to reduce the leaking information through the power side-channel?
3) What if my security-aware design modifications lead to some violations of the previous optimization efforts performed by the tools?

[2] The role of timing will be more pronounced/diverse and more dominant for other, practical, and more challenging design scenarios; we cover this in Sec. V in more detail.

[3] First, attackers may have only limited access or could measure only over a limited number of traces before risking detection. Second, for the same number of traces collected, attackers would obtain a dataset with an inherently better signal-to-noise ratio which could help them, e.g., with a higher-order correlation test. Third, security protocols implemented at the system level, e.g., key update/replacement after so and so many AES rounds, would be much easier to bypass for such an attacker.

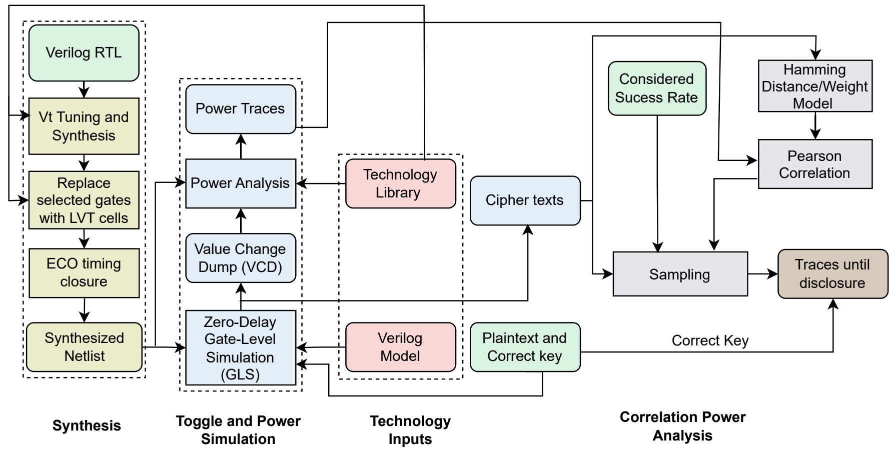

Fig. 2. Overview of the proposed security-aware design-space exploration framework.

These questions are all valid when it comes to security along with conventional PPA targets. This is not as "easy" to achieve as utilizing well-established heuristics like we do to meet our PPA targets – it rather requires a thorough understanding of different corners in the design space on both PPA as well as security, e.g., different usage profiles of VT cells on PPA as well as PSC information leakage.

To tackle these questions and resulting new challenges, we argue the following. Initially, the simplest solution to Questions 1 and 2 is to limit or disable the uses of LVT cells for state registers – the key finding from the exploratory study in Sec. IV-A would suggest this. However, doing so is not viable under aggressive timing constraints. Consequently, more sophisticated approaches are required, such as 1) optimizing timing closure in other design parts, 2) re-evaluating the role of LVT cells in state versus non-state registers, and 3) establishing design guidelines for optimal PPA and security trade-offs. These challenges necessitate comprehensive empirical studies using the proposed framework, which we present in Sec. V.

Related to the points just raised above, as well as to answer Question 3, we need to understand that any post-synthesis modification of VT cells can result in violations. More specifically, swapping faster (but more leaky) LVT cells with slower (but less leaky) RVT or HVT cells may well result in timing issues for the affected paths. Such timing violations would adversely affect the functionality of the design and cause erroneous behavior – they must be addressed also in a security-aware design-space exploration campaign, and we provide more details in Sec. IV-E.

D. Trojan Design

As already indicated, the proposed Trojan is a simple, yet highly effective, type of Trojan that aids PSC attacks in the field, by increasing the related information leakage. The Trojan is easy to realize on top of any design under attack, simply by replacing some instances of standard cells with other cells of the very same functionality but with low or (if available) even ultra-low VT profile (LVT/ULVT). This type of Trojan is stealthy and practical for the following reasons.

1) These cells are faster than regular cells, implying that:
   a) An attacker can implement such a Trojan (i.e., replace some RVT or HVT cells of interest with LVT/ULVT cells) right away, without any other changes needed in the design, and without risking any timing violations.
   b) Detecting the Trojan is impossible during functional testing.
   c) Detecting such a Trojan is difficult even for parametric, timing-based testing (which is costly and applied only selectively anyway); improved timing in these select locations can be masked by cells that intersect along common timing paths[4].
2) LVT/ULVT cells have the very same footprint as other cells, implying that:
   a) Trojan is obscure for regular optical inspection;
   b) They are easy to integrate by any adversary within design and manufacturing entities, even just at the end like through mask configuration in the foundry.

Based on our findings, the attacker would want to specifically replace the state register cells with such LVT/ULVT cells.

For the actual Trojan implementation, i.e., the number and location of the state register that an attacker would want to replace, this depends on the overall design characteristics, especially on the timing paths. Our proposed framework for security-aware design-space exploration would help the attacker tackle with these very questions.

E. Implementation Details

Recall the overview of our methodology as outlined in Fig. 2. Next, we provide some more implementation details, followed by the actual setup for experiments in Sec. V-A.

[4] An empirical study on this claim is left for future work. Even for parametric testing that focuses on static power, it remains to be seen whether the observed profile can be identified as malicious. For one, assuming a few more LVT/ULVT cells are introduced among millions or even billions of gates in modern SoCs, their contribution to the overall static power profile can be small. For another, the use of LVT/ULVT cells cannot be considered as malicious per se, implying that significant variations in power profiles are expected even for benign ICs, e.g., where designers optimized VT cell usage for different operation corners.

1) Simulation-Based Power Analysis: First, the cipher RTL is synthesized using the technology libraries of choice, considering all VT cell options. Then, using a testbench, the functionality of the gate-level, post-synthesis netlist is verified. After the user specifies a set of plain-texts and a key (or set of keys), the testbench generates the corresponding set(s) of cipher-texts required for verification. During these gate-level simulations, a Value Change Dump (VCD) file is generated; it captures the switching activity of every gate/node within the netlist in a user-defined time resolution (e.g., 1 ps). Next, the VCD file is used for power simulation, along with the post-synthesis netlist and libraries of choice.

The power simulation tool calculates the power consumption of each cell by adding 1) static/leakage power, 2) internal power (from input-pin switching), and 3) switching power (from output-pin switching). The library power characterization stores data for leakage and internal power of the cells, and the tool uses the VCD's input/output state information to calculate the total leakage power of the design. Zero-delay simulations are used instead of full-timing simulations as our interest lies in capturing static power in a specific clock cycle, not average leakage power. Such static power value will be the same regardless if it is a full-timing simulation using a standard delay format (SDF) or a zero-delay simulation. We then sequentially obtain power traces while processing AES texts for the given secret key(s), focusing on the last-round operations [7]. This procedure is repeated separately for each scenario and technology setup considered in this work.

2) Correlation Power Analysis Attack, CPA: This attack uses the Pearson correlation coefficient (PCC) to measure the relationship between observed power consumption during cryptographic tasks and the secret data [7]. This involves comparing real power consumption and predicted power profiles for different key values across a range of observations. Then, the key is byte-wise inferred from the most promising candidates, i.e., those with the highest PCC values.

More specifically, predicted power profiles are derived from a power model of choice [7]. For S-PSCA, typically either the Hamming weight (HW) for the cipher-texts or the Hamming distance (HD) for the cipher-texts to the prior last-round operation are chosen, depending on the power profiles of the underlying technology nodes. For the nodes considered in our study, we observe a considerable dependency of the FFs static power on both the input and output data (Table I), thus the HD model seems more promising. Indeed, while we did run all experiments for both HW and HD models, better results are achieved using the HD model for all scenarios and all nodes.

In our detailed study on the role of VT cells, we conduct sampling campaigns for each scenario separately. We provide increasing numbers of random traces to the CPA and calculate the success rate over several CPA trials. We stop when we achieve a desired confidence level, such as a 90% success rate, and report the corresponding number of traces. To improve computational efficiency, we first conduct coarse sampling with fewer trials and a larger step size, which provides a reasonable starting point. We then conduct thorough sampling with more trials and smaller step-size, starting from the point identified in the coarse sampling.

Algorithm 1: ECO VT replacement, timing closure

```
Input: List of failing paths $FT$, VT constraints (number of LVT/RVT/HVT
       cells allowed)
Output: Report $file$
Function CorrectTiming($path$):
    $startPoint \leftarrow path.startpoint()$; // Start point of path
    $endPoint \leftarrow path.endpoint()$; // End point of the path
    $gates \leftarrow path.gates()$; // List of gates in the path
    $gates \leftarrow$ SORT($gates$); // Sort the gates in decreasing
      order of delay
    foreach $i \in gates$ do
        $x \leftarrow$ GETALTERNATIVECELLS($i$); // List of
          alternative cells available, considering
          driver strength and VT cells
        foreach $j \in x$ do
            if $j.delay() < i.delay()$ and VT constraints still met then
                REPLACE $i$ by $j$ ; // Replace old one with
                  new cell
            else
                KEEP $i$ ;
    $STATUS \leftarrow$ REPORTTIMING($startPoint, endPoint$) ;
    return $STATUS$; // Return the status
$TEMP \leftarrow$ COPY($FT$);
$UnresolvedPath \leftarrow \emptyset$;
foreach $path \in TEMP$ do
    $STATUS \leftarrow CorrectTiming(path)$;
    if $STATUS == PASS$ then
        COMMIT; // Commit to the changes
    else
        $UnresolvedPath.append(path)$
$file \leftarrow \{UnresolvedPath\}$; // List of paths for manual
  analysis
return $file$
```

3) Engineering Change Order (ECO) Modifications, Timing Closure: We use commercial synthesis tools to create baseline netlists for various scenarios. Then, through Tcl scripts, we make further ECO modifications on these netlists, specifically replacing gates of interest – the registers holding the AES intermediate texts – with different VT cells as outlined in Sec. IV-A and Sec. V. These ECO modifications may lead to violations, particularly timing violations. To handle this, we implement the following procedures (Algorithm 1).

Initially, we identify all failing paths and sort the gates on these paths in decreasing order of delay. Next, we explore gates with 1) greater drive strengths and/or 2) different VT cells to replace them. While Option 2) is based on the VT scenario under investigation and its constraints for the use of different VT cells, Option 1) is applied as needed. Next we re-evaluate the failing paths. If there are paths that fail to meet the timing, a manual analysis is necessary. Since many timing paths intersect, in some cases, revising other paths can help fix those violating paths. If failing paths remain, related timing constraints are infeasible, and setup for scenario is revised.

V. Experimental Investigation

A. Setup

All experimentation (including the exploratory study in Sec. IV-A) are conducted on an AMD EPYC 7542 server with 128 CPUS, 512KB caches, and 1TB RAM, operating with Red Hat Enterprise Linux Server Release 7.9 (Maipo). In our implementation of the CAD flow, we employ commercial tools as follows. We use Synopsys VCS M-2017.03-SP1 for functional simulations at RTL and gate level, Synopsys DC M-2016.12-SP2 for logic synthesis, and Synopsys PrimeTime PX

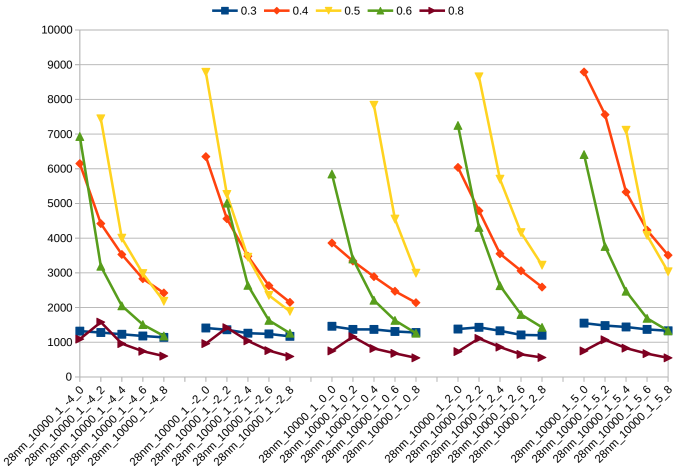

Fig. 3. Number of traces required until disclosure for a commercial 28nm node using LVT and HVT cells. The legend refers to reference timing constraints in ns. For reading the naming scheme of different cases listed on the x-axis, please refer back to Sec. IV-A. Any missing data point indicates that CPA could not succeed with 10,000 traces.

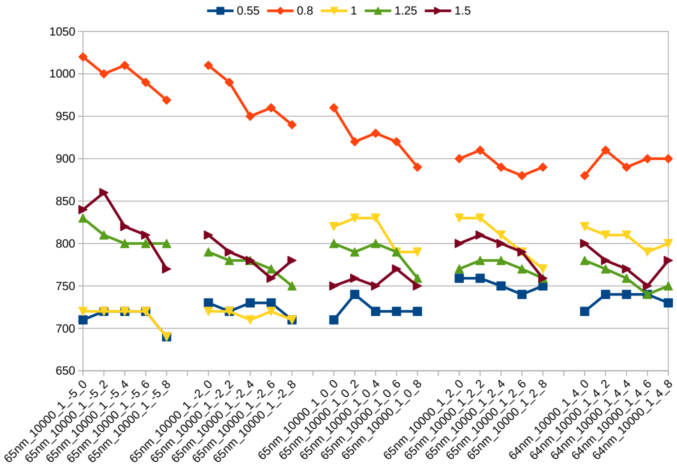

Fig. 4. Number of traces required until disclosure for a commercial 65nm node using LVT and HVT cells. See also Fig. 3's caption.

TABLE II

PPA RESULTS FOR 28NM NODE, TIMING CONSTRAINT OF 0.3NS, WITH H(VT) AND L(VT) CELLS USED. 'TIME' REFERS TO % VARIATION OF THE TIMING CONSTRAINT. LEFT 'LVT' COLUMN REFERS TO NUMBER OF LVT CELLS IN EACH BYTE OF THE REGISTERS HOLDING AES STATE. 'LVT' AND 'HVT' REFER TO RATIO OF ALL GATES IN THE DESIGN IMPLEMENTED BY THESE VT CELLS. 'STATE FFS' AND 'NON-STATE FFS' (NS FFS) REFER TO THE COUNTS OF VT CELLS IN THESE REGISTERS.

Time	L	Area	WNS	Power	L	H	State FFs		NS FFs	
		(μm^2)	(ns)	(mW)	(%)	(%)	L	H	L	H
-4	0	15534	-0.05	23.89	50	50	0	128	318	206
	2	15534	-0.04	23.91	51	49	32	96	318	206
	4	15534	-0.04	23.94	51	49	64	64	318	206
	6	15534	-0.03	23.96	52	48	96	32	318	206
	8	15534	-0.01	23.98	52	48	128	0	318	206
-2	0	15599	-0.05	22.25	43	57	0	128	277	247
	2	15599	-0.03	22.27	44	56	32	96	277	247
	4	15599	-0.03	22.29	44	56	64	64	277	247
	6	15599	-0.01	22.32	45	55	96	32	277	247
	8	15599	-0.01	22.34	45	54	128	0	277	247
0	0	14867	-0.01	21.52	43	57	0	128	273	251
	2	14867	-0.01	21.54	44	56	32	96	273	251
	4	14867	-0.01	21.57	44	56	64	64	273	251
	6	14867	0	21.59	45	55	96	32	273	251
	8	14867	0	21.61	45	55	128	0	273	251
2	0	15038	0	20.84	41	59	0	128	261	263
	2	15038	0	20.86	41	59	32	96	261	263
	4	15038	0	20.89	42	58	64	64	261	263
	6	15038	0	20.91	42	58	96	32	261	263
	8	15038	0	20.93	43	57	128	0	261	263
5	0	14097	0	18.73	39	61	0	128	299	225
	2	14097	0	18.75	40	60	32	96	299	225
	4	14097	0	18.77	40	60	64	64	299	225
	6	14097	0	18.80	41	59	96	32	299	225
	8	14097	0	18.82	41	59	128	0	299	225

M-2017.06 for power simulations. For synthesis, we employ regular optimization techniques. The design-space exploration framework is implemented in Tcl scripts. The CPA code is based on the open-source release of [29].

For the AES design, we leverage a regular RTL, working on 128-bit keys and 128-bit texts, using look-up tables for the AES substitution box, and without any PSC countermeasures. We employ commercial libraries for a 65nm and a 28nm technology node. For both nodes, we consider their respective TT corners which are characterized for 25 degree Celsius and for 0.9V/1.0V for the 28nm/65nm node. We utilize VT cells as desired/appropriate for the different scenarios under study. For all CPA runs, we consider 10,000 traces in total. We increase the number of traces made available to the CPA stepwise by 10. We employ coarse versus thorough sampling considering 64 versus 640 trials, respectively, and we report final results of t traces until disclosure for a 90% success rate for thorough sampling, i.e., CPA must succeed to infer all key bytes correctly for at least 576 out of 640 randomly selected subsets of t traces. We consider HW and HD models separately, and found that HD model has better results in all scenarios; thus final results are reported for the HD model.

B. *Results I: HVT versus LVT*

In the first set of experiments, we study the impact of VT cells. We focus on extreme cases: HVT vs LVT. Like the preliminary study, we adjust timing constraint between -4% and +5%, while replacing varying numbers of HVT cells with LVT in AES state registers. We consider different timing scales, such as (0.3, 0.4, 0.5, 0.6, 0.8)ns for 28nm vs. (0.55, 0.8, 1.0, 1.25, 1.5)ns for 65nm node. These values push limits, with faster constraints failing even with majority of LVT use.

Security Analysis: The CPA results are provided in Figures 3 and 4, respectively, for the 28nm and the 65nm node.

The observed trend is consistent here as well. The number of traces reduces with an increase in the number of LVT cells in the state registers. However, for the 28nm node, the slope of the curves describing this correlation varies significantly, given the varied scales of timing constraints. This finding is interesting, as prior works anticipated that high-performance designs will be leakier, in terms of static power (which is correct) and in terms of information leakage (which is wrong).

Furthermore, the number of traces required for breaking the two corner cases in the 28nm node (i.e., 0.3ns versus 0.8ns) are both in lower resilience ranges, whereas the middle ranges for timing (i.e., 0.4–0.6ns) are much more varied and running in much higher resilience ranges. For the 65nm node, while the different timing constraints generally induce less strong variations, it still holds true that some "middle range"

TABLE III
PPA RESULTS FOR 28NM NODE, TIMING CONSTRAINT OF 0.5NS. SEE ALSO TABLE II'S CAPTION.

Time	L	Area	WNS	Power	L	H	State FFs		NS FFs	
		(μm^2)	(ns)	(mW)	(%)	(%)	L	H	L	H
-4	0	13448	0	11.08	9	91	0	128	25	499
	2	13435	0	11.08	9	91	32	96	25	499
	4	13424	0	11.07	10	90	64	64	25	499
	6	13408	0	11.06	10	90	96	32	25	499
	8	13401	0	11.06	11	89	128	0	25	499
-2	0	13386	0	10.61	8	92	0	128	12	512
	2	13375	0	10.61	8	92	32	96	12	512
	4	13362	0	10.60	9	91	64	64	12	512
	6	13352	0	10.60	9	91	96	32	12	512
	8	13339	0	10.59	10	90	128	0	12	512
0	0	13230	0	10.42	8	92	0	128	10	514
	2	13221	0	10.42	8	92	32	96	10	514
	4	13214	0	10.43	9	91	64	64	10	514
	6	13206	0	10.43	9	91	96	32	10	514
	8	13199	0	10.44	10	90	128	0	10	514
2	0	13178	0	10.16	8	92	0	128	25	499
	2	13170	0	10.17	8	92	32	96	25	499
	4	13162	0	10.17	9	91	64	64	25	499
	6	13157	0	10.18	9	91	96	32	25	499
	8	13153	0	10.19	10	90	128	0	25	499
5	0	13129	0	9.85	6	94	0	128	8	516
	2	13120	0	9.84	6	94	32	96	8	516
	4	13116	0	9.85	7	93	64	64	8	516
	6	13109	0	9.86	7	93	96	32	8	516
	8	13101	0	9.86	8	92	128	0	8	516

TABLE IV
PPA RESULTS FOR 65NM NODE, TIMING CONSTRAINT OF 0.55NS. SEE ALSO TABLE II'S CAPTION.

Time	L	Area	WNS	Power	L	H	State FFs		NS FFs	
		(μm^2)	(ns)	(mW)	(%)	(%)	L	H	L	H
-5	0	67678	-0.03	38.17	86	14	0	128	389	135
	2	67678	-0.03	38.17	87	13	32	96	389	135
	4	67678	-0.01	38.17	87	13	64	64	389	135
	6	67678	-0.01	38.17	87	13	96	32	389	135
	8	67678	-0.01	38.18	88	12	128	0	389	135
-2	0	64324	0	34.77	84	16	0	128	381	143
	2	64324	0	34.77	84	16	32	96	381	143
	4	64324	0	34.77	85	15	64	64	381	143
	6	64324	0	34.77	85	15	96	32	381	143
	8	64324	0	34.78	86	14	128	0	381	143
0	0	66453	0	35.35	86	14	0	128	367	157
	2	66453	0	35.35	86	14	32	96	367	157
	4	66453	0	35.36	86	14	64	64	367	157
	6	66453	0	35.36	87	13	96	32	367	157
	8	66453	0	35.36	87	13	128	0	367	157
2	0	66464	0	34.50	82	18	0	128	367	157
	2	66464	0	34.50	83	17	32	96	367	157
	4	66464	0	34.50	83	17	64	64	367	157
	6	66464	0	34.50	84	16	96	32	367	157
	8	66464	0	34.51	84	16	128	0	367	157
4	0	65420	0	33.07	84	16	0	128	368	156
	2	65420	0	33.07	84	16	32	96	368	156
	4	65420	0	33.07	84	16	64	64	368	156
	6	65420	0	33.08	85	15	96	32	368	156
	8	65420	0	33.08	85	15	128	0	368	156

TABLE V
PPA RESULTS FOR 65NM NODE, TIMING CONSTRAINT OF 0.8NS. SEE ALSO TABLE II'S CAPTION.

Time	L	Area	WNS	Power	L	H	State FFs		NS FFs	
		(μm^2)	(ns)	(mW)	(%)	(%)	L	H	L	H
-5	0	56163	0	19.77	27	73	0	128	263	261
	2	56163	0	19.77	28	72	32	96	263	261
	4	56163	0	19.78	28	72	64	64	263	261
	6	56163	0	19.78	29	71	96	32	263	261
	8	56163	0	19.78	29	71	128	0	263	261
-2	0	55415	0	19.10	25	75	0	128	232	292
	2	55415	0	19.11	25	75	32	96	232	292
	4	55415	0	19.11	26	74	64	64	232	292
	6	55415	0	19.11	26	74	96	32	232	292
	8	55415	0	19.11	27	73	128	0	232	292
0	0	54241	0	18.29	23	77	0	128	224	300
	2	54241	0	18.29	24	76	32	96	224	300
	4	54241	0	18.30	24	76	64	64	224	300
	6	54241	0	18.30	25	75	96	32	224	300
	8	54241	0	18.30	25	75	128	0	224	300
2	0	54228	0	17.59	22	78	0	128	221	303
	2	54228	0	17.59	23	77	32	96	221	303
	4	54228	0	17.60	23	77	64	64	221	303
	6	54228	0	17.60	24	76	96	32	221	303
	8	54228	0	17.60	24	76	128	0	221	303
4	0	54064	0	17.45	21	79	0	128	232	292
	2	54064	0	17.45	22	78	32	96	232	292
	4	54064	0	17.45	22	78	64	64	232	292
	6	54064	0	17.45	23	77	96	32	232	292
	8	54064	0	17.46	23	77	128	0	232	292

for timing (i.e., 0.8ns) induces the largest resilience, whereas faster and slower constraints induce lower resilience ranges.

Therefore, we can conclude that the S-PSCA information leakage is not solely determined by the number of LVT cells, but rather by both the number of LVT cells as well as the timing constraint. In fact, the latter will dictate the distribution or ratio of LVT to HVT cells, which is an important factor of resilience. After reviewing the CPA results along with the design properties of all these different scenarios, we argue the following for an explanation of this observation. First, for CPA in general, a set of power traces that exhibits some consistent patterns for power consumption is easier to correlate to the correct key. Second, such consistent power patterns are more likely for designs with greater homogeneity, that is when the usage/ratio of different VT cells is either quite balanced or almost exclusively one-sided/singular. For example for the 28nm node, while we observe a balance in LVT and HVT cells for the overall design with the 0.3ns constraint, for 0.8ns there is a clear dominance of HVT cells over LVT cells. In contrast, for constraints of 0.4–0.6ns, LVT cells are utilized to a somewhat larger degree, yet far from balanced with HVT cells. Third, these trends are less pronounced for the 65nm node, which is expected from the less varied static power profiles (Table I). In short, a small number of LVT cells results in diverse power profiles, which challenges the CPA.

Attacker's Perspective for Trojan Design: It holds true here as well that more LVT cells are preferable. While resilience values are converging for a maximum number of LVT cells across all timing constraints, some differences do remain here. These differences arise from the imbalance in LVT/HVT cells usage for the whole design; thus, an attacker might further reduce this by assigning more and more LVT cells to other gates as well, but larger-scale modifications might also be easier detected later on in the field. These differences across timing constraints are smaller than the reduction in resilience achievable by using the maximum of LVT cells. In short, the threat is practical across wide timing ranges and is powerful when using LVT cells for all 8 bits in state-register bytes.

PPA Analysis: For the 28nm node, results are given in Tables II–III and for 65nm node in Tables IV–V.

Despite our efforts (Algorithm 1), we encountered cases where timing was not met, which challenges the assumption in prior art [23] that sufficient slack is available for any VT optimization. For high-performance design settings (0.3ns for 28nm, Table II; 0.55ns for 65nm, Table IV), timing closure was

TABLE VI
PPA RESULTS FOR 28NM NODE, WITH 3 LVT CELLS USED IN EACH BYTE OF THE STATE. 'TIME' REFERS TO THE TIMING CONSTRAINT. 'NON-STATE FF SETTING' (NS FFS SETTING) REFERS TO THE TOTAL NUMBER OF LVT CELLS USED FOR NON-STATE FFS. OTHER COLUMNS ARE THE SAME AS BEFORE.

Time	NS FFs	Area	WNS	Power	L	H	State FFs		NS FFs	
	Setting	(μm^2)	(ns)	(mW)	(%)	(%)	L	H	L	H
0.3	0	14867	-0.05	21.55	44	56	48	80	273	251
	60	14867	-0.01	21.60	45	55	48	80	333	191
	120	14867	-0.01	21.63	46	54	48	80	393	131
0.5	0	13211	0	10.41	8	92	48	80	10	514
	60	13211	0	10.44	9	91	48	80	70	454
	120	13211	0	10.47	10	90	48	80	130	394
0.75	0	12336	0	6.09	1	99	48	80	0	524
	60	12336	0	6.10	2	98	48	80	60	464
	120	12336	0	6.12	3	97	48	80	120	404
1	0	12085	0	4.29	1	99	48	80	0	524
	60	12085	0	4.32	2	98	48	80	60	464
	120	12085	0	4.33	3	97	48	80	120	404

difficult for negative ranges for timing variation, although we were able to get close, with -0.01–0.05 WNS remaining. This indicates that timing constraints are pushing the respective technology and synthesis limits. There is an increase in total power for these constraints (Tables III and V, respectively); these constraints may be considered too power-hungry by designers. Since the constraints yield lower resilience, this indicates that this part of the design space is unfavorable for PPA-and-security co-optimization.

C. Results II: State versus Non-State FFs

To examine the role of non-state registers in more detail, here we sweep the overall number of LVT instances for those registers, for different sets of LVT cells for state registers, and for varying timing constraints.

TABLE VII
PPA RESULTS FOR 28NM NODE, WITH 8 LVT CELLS USED IN EACH BYTE OF THE STATE. SEE ALSO TABLE VI'S CAPTION.

Time	NS FFs	Area	WNS	Power	L	H	State FFs		NS FFs	
	Setting	(μm^2)	(ns)	(mW)	(%)	(%)	L	H	L	H
0.3	0	14867	-0.01	21.61	45	55	128	0	273	251
	60	14867	0	21.65	46	54	128	0	333	191
	120	14867	0	21.69	47	53	128	0	393	131
0.5	0	13199	0	10.44	10	90	128	0	10	514
	60	13199	0	10.47	11	89	128	0	70	454
	120	13199	0	10.49	12	88	128	0	130	394
0.75	0	12336	0	6.12	2	98	128	0	0	524
	60	12336	0	6.13	4	96	128	0	60	464
	120	12336	0	6.16	5	95	128	0	120	404
1	0	12085	0	4.32	2	98	128	0	0	524
	60	12085	0	4.34	4	96	128	0	60	464
	120	12085	0	4.36	5	95	128	0	120	404

Security Analysis: The CPA results for thorough sampling are provided in Figure 5 for the 28nm node. CPA results for the 65nm node follow similar trends, albeit less pronounced; these are not reported separately here. The general trend is valid. That is, both the number of LVT cells in state registers and the timing constraints dominate the resilience. This is true again for the "middle range" timing constraints. The number of LVT cells in non-state registers has negligible impact.[5]

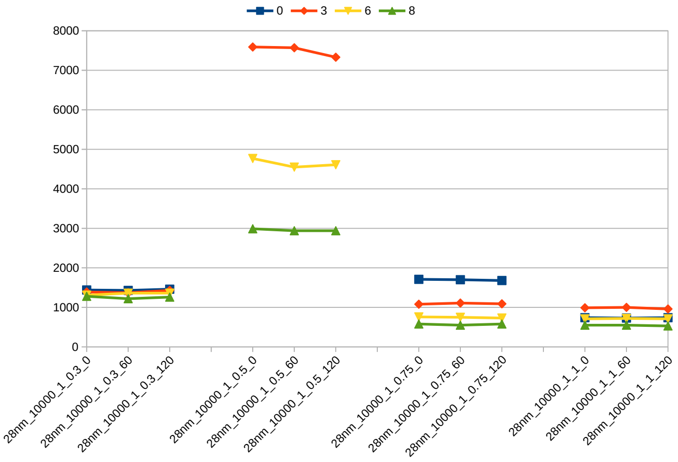

Fig. 5. Number of traces required until disclosure for a commercial 28nm node using LVT and HVT cells. The legend refers to the number of LVT cells in state-register bytes. Cases listed on the x-axis differ here as follows in naming: the second-to-last element is timing constraints and the last element is the number of LVT cells in non-state registers. Missing data point indicates CPA fails after 10,000 traces.

Attacker's Perspective for Trojan Design: It holds true again that more LVT cells are preferable, but, as indicated, differences across timing constraints are more pronounced now for the attacker's best-case of all state FFs using LVT cells.

PPA Analysis: For the 28nm node, PPA results are provided in Tables VI–VII. Findings here are similar to those for Results I, e.g., timing closure becomes only somewhat more challenging for the most aggressive setting, and power cost is considerable for those aggressive constraints. PPA results for the 65nm node are similar here as well to those for Results I and are, thus, not reported separately.

D. Results III: LVT, RVT, and HVT

We study a practical scenario where all types of VT cells are used.[6] Otherwise, setting is similar to the experiments above.

Security Analysis: CPA results are provided in Figures 6 and 7 for 28nm and the 65nm nodes. The general trend remains valid here as well. The variation in resilience becomes more pronounced for different timing constraints. For the 28nm node, the trends are more pronounced when locally sweeping/revising the aggressive 0.35ns constraint and the medium 0.75ns constraint. For the 65nm node, note the lower resilience for negative timing variations for lower and upper ranges of baseline timing constraints.

Such trends help designers. For example for the 28nm node, securing high-performance design options is promising compared to using LVT, HVT cells. The improvement in resilience over the corresponding cases in Results I is around 3.5x (for 0.35ns timing constraint). For the 65nm node, for the most promising "mid range" of 0.8ns, the improvement over corresponding cases in Results I is around 2.8x. Timing constraints should be explored for resilience.

[5] this does not necessarily hold true for all other, combinational gates. This will be studied in more detail in the next set of experiments.

[6] However, we use RVT cells only for all other parts of the design. For state registers, we employ the same procedure of sweeping the number of HVT cells to be replaced by LVT cells. This is important to better understand the role of other parts of the design, which will be impacted to a larger degree by the more varied VT cell options than only the state registers.

TABLE VIII
PPA RESULTS FOR 28NM NODE, TIMING CONSTRAINT OF 0.35NS, WITH HVT ('H'), RVT ('R'), AND LVT ('L') CELLS USED. SEE ALSO TABLE II'S CAPTION.

Time	L	Area	WNS	Power	L	R	H	State FFs			NS FFs		
		(μm^2)	(ns)	(mW)	(%)	(%)	(%)	L	R	H	L	R	H
-4	0	14529	-0.03	17.99	19	24	57	0	0	128	219	17	288
	2	14529	-0.03	18.01	20	24	57	32	0	96	219	17	288
	4	14529	-0.03	18.04	20	24	56	64	0	64	219	17	288
	6	14529	-0.03	18.06	21	24	56	96	0	32	219	17	288
	8	14529	-0.03	18.08	21	24	55	128	0	0	219	17	288
-2	0	13890	-0.02	17.03	18	26	55	0	0	128	209	26	289
	2	13890	-0.02	17.05	19	26	55	32	0	96	209	26	289
	4	13890	-0.02	17.07	19	26	54	64	0	64	209	26	289
	6	13890	-0.02	17.09	20	26	54	96	0	32	209	26	289
	8	13890	-0.02	17.11	20	26	53	128	0	0	209	26	289
0	0	14636	-0.01	16.79	13	24	63	0	0	128	179	48	297
	2	14636	-0.01	16.81	13	24	63	32	0	96	179	48	297
	4	14636	-0.01	16.83	14	24	62	64	0	64	179	48	297
	6	14636	-0.01	16.85	14	24	62	96	0	32	179	48	297
	8	14636	-0.01	16.87	15	24	61	128	0	0	179	48	297
2	0	14330	0	15.97	13	24	63	0	0	128	219	37	268
	2	14330	0	15.99	14	24	63	32	0	96	219	37	268
	4	14330	0	16.01	14	24	62	64	0	64	219	37	268
	6	14330	0	16.03	15	24	62	96	0	32	219	37	268
	8	14330	0	16.06	15	24	61	128	0	0	219	37	268
5	0	14560	0	15.65	10	24	65	0	0	128	111	75	338
	2	14558	0	15.67	11	24	65	32	0	96	111	75	338
	4	14558	0	15.69	11	24	64	64	0	64	111	75	338
	6	14558	0	15.71	12	24	64	96	0	32	111	75	338
	8	14558	0	15.73	12	24	63	128	0	0	111	75	338

TABLE IX
PPA RESULTS FOR 28NM NODE, TIMING CONSTRAINT OF 0.5NS, WITH HVT ('H'), RVT ('R'), AND LVT ('L') CELLS USED. SEE ALSO TABLE II'S CAPTION.

Time	L	Area	WNS	Power	L	R	H	State FFs			NS FFs		
		(μm^2)	(ns)	(mW)	(%)	(%)	(%)	L	R	H	L	R	H
-4	0	13291	0	11.21	3	19	79	0	0	128	3	62	459
	2	13283	0	11.21	3	19	78	32	0	96	3	62	459
	4	13276	0	11.21	4	19	78	64	0	64	3	62	459
	6	13268	0	11.21	4	19	77	96	0	32	3	62	459
	8	13260	0	11.21	5	19	77	128	0	0	3	62	459
-2	0	13229	0	10.48	3	16	82	0	0	128	5	73	446
	2	13222	0	10.49	3	16	81	32	0	96	5	73	446
	4	13213	0	10.48	4	16	80	64	0	64	5	73	446
	6	13203	0	10.48	4	16	80	96	0	32	5	73	446
	8	13197	0	10.49	5	16	79	128	0	0	5	73	446
0	0	13155	0	9.99	2	14	84	0	0	128	3	68	453
	2	13155	0	10.01	3	14	83	32	0	96	3	68	453
	4	13155	0	10.03	3	14	83	64	0	64	3	68	453
	6	13155	0	10.04	4	14	82	96	0	32	3	68	453
	8	13155	0	10.06	4	14	82	128	0	0	3	68	453
2	0	13089	0	9.82	2	14	84	0	0	128	4	83	437
	2	13089	0	9.83	2	14	84	32	0	96	4	83	437
	4	13089	0	9.84	3	14	83	64	0	64	4	83	437
	6	13089	0	9.86	4	14	83	96	0	32	4	83	437
	8	13089	0	9.87	4	14	82	128	0	0	4	83	437
5	0	13082	0	9.60	1	12	87	0	0	128	0	32	492
	2	13082	0	9.62	2	12	86	32	0	96	0	32	492
	4	13082	0	9.62	3	12	86	64	0	64	0	32	492
	6	13082	0	9.63	3	12	85	96	0	32	0	32	492
	8	13082	0	9.64	4	12	85	128	0	0	0	32	492

Since we did not, on purpose, use RVT cells for state FFs, these observations indicate that diverse options for all other gates play an important role. For example, for the 65nm node, there is a larger benefit now for positive timing variations. This is due to the flexibility of using RVT and HVT for relaxed timing. These observations do not contradict Results II, but extend the related perspective: VT options for other gates can matter, but require more VT options.

TABLE X
PPA RESULTS FOR 65NM NODE, TIMING CONSTRAINT OF 0.8NS, WITH HVT ('H'), RVT ('R'), AND LVT ('L') CELLS USED. SEE ALSO TABLE II'S CAPTION.

Time	L	Area	WNS	Power	L	R	H	State FFs			NS FFs		
		(μm^2)	(ns)	(mW)	(%)	(%)	(%)	L	R	H	L	R	H
-5	0	55109	0	20.05	21	30	50	0	0	128	257	24	243
	2	55109	0	20.05	21	30	49	32	0	96	257	24	243
	4	55109	0	20.05	22	30	49	64	0	64	257	24	243
	6	55109	0	20.05	22	30	48	96	0	32	257	24	243
	8	55109	0	20.06	23	30	48	128	0	0	257	24	243
-2	0	56525	0	19.96	14	29	57	0	0	128	229	9	286
	2	56525	0	19.96	15	29	56	32	0	96	229	9	286
	4	56525	0	19.96	15	29	56	64	0	64	229	9	286
	6	56525	0	19.97	16	29	55	96	0	32	229	9	286
	8	56525	0	19.97	16	29	55	128	0	0	229	9	286
0	0	56003	0	19.41	13	30	57	0	0	128	256	10	258
	2	56003	0	19.41	14	30	56	32	0	96	256	10	258
	4	56003	0	19.42	14	30	56	64	0	64	256	10	258
	6	56003	0	19.42	15	30	55	96	0	32	256	10	258
	8	56003	0	19.42	15	30	55	128	0	0	256	10	258
2	0	55514	0	18.39	11	30	58	0	0	128	221	3	300
	2	55514	0	18.39	12	30	58	32	0	96	221	3	300
	4	55514	0	18.39	12	30	57	64	0	64	221	3	300
	6	55514	0	18.39	13	30	57	96	0	32	221	3	300
	8	55514	0	18.40	13	30	56	128	0	0	221	3	300
4	0	54888	0	18.10	11	28	61	0	0	128	215	7	302
	2	54888	0	18.10	11	28	60	32	0	96	215	7	302
	4	54888	0	18.10	12	28	60	64	0	64	215	7	302
	6	54888	0	18.11	12	28	59	96	0	32	215	7	302
	8	54888	0	18.11	13	28	59	128	0	0	215	7	302

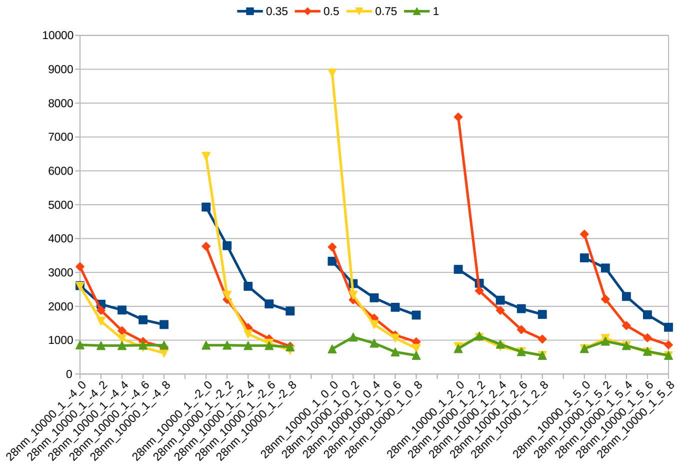

Fig. 6. Number of traces required until disclosure for a commercial 28nm node using LVT, RVT, and HVT cells. The legend refers to reference timing constraints in ns. For reading the naming scheme of different cases listed on the x-axis, please refer back to Sec. IV-A.

Attacker's Perspective for Trojan Design: It holds true again that more LVT cells in state registers are preferable, but the local variation of timing constraints also plays an important role now. This can also be explained by the more varied options for VT cells, especially with a more varied use of RVT and HVT cells for more relaxed constraints (Table X).

Conversely, this also means that an attacker may benefit as well from employing more LVT cells in other parts of the design. We deduct this from the fact that, for negative timing variations, we had to employ more LVT cells in other parts as well (Table X) in order to meet timing (Algorithm 1) – this had a considerable detrimental effect on resilience, and an attacker could achieve a similar outcome by adding more LVT cells in other design parts. This can be interesting when trying to avoid

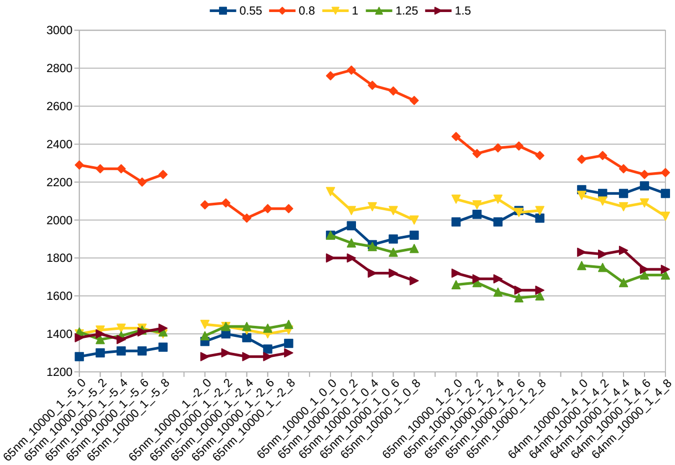

Fig. 7. Number of traces required for PSC for a commercial 65nm node using LVT, RVT, HVT cells. See also Fig. 6's caption.

malicious modifications directly of the sensitive parts of a design such as the AES state registers, as these registers might be particularly vetted by runtime detection against Trojans.

PPA Analysis: For the 28nm node, PPA results for time constraints of 0.35ns and 0.5ns are reported in Table VIII and Table IX, respectively. Note that the constraints considered vary somewhat at both lower and upper ends from those considered in Result I; this is on purpose to allow for better utilization of RVT cells as appropriate. For the 65nm node, PPA results are provided in Table X. Overall, the findings are similar to those in Results I. In short, most aggressive timing constraints do not offer promising trade-offs for PPA cost (especially not for power) versus security, whereas more relaxed constraints are promising.

VI. Conclusions

We studied the role of VT cells on the resilience of an AES design (implemented for 28nm and 65nm nodes) against static power side-channel attacks. First, we developed a security-aware design-space exploration framework using commercial CAD tools and an open-source CPA tool. This framework can be used by designers and security experts to trade-off between design cost and security and importantly for security tuning before tape-outs. Our experiments offer insights into S-PSC attacks. It was assumed that high-performance designs are vulnerable to S-PSC. But, timing constraints and distribution and ratio of threshold-voltage cells play a dominant role. We postulate guidelines for security-aware design:

1) Limit LVT cells in critical gates like AES state registers.
2) Use LVT cells in other parts where necessary, such as for managing timing of the design. This won't compromise resilience, although it depends on the technology node.
3) Use all types of VT cells. This promotes a diverse application of VT cells by CAD tool, leading to a range of power profiles that are difficult to compromise.
4) Aim for moderate ranges and/or slightly aggressive timing constraints. Avoid being too lax. This aligns well with goals for PPA optimization.
5) Given the variability in the design space, which is influenced by timing constraints and VT cell use in the design – particularly for the 28nm node – it is advised to explore the PPA-security trade-offs. Such efforts are supported by our framework.

Concerning the offense perspective, an attacker can implement highly effective PSC-based Trojan by revising only few gates in a design to use LVT cells. Such Trojans are stealthy as they are using zero additional gates and do not undermine timing paths if designed properly. From an attacker's perspective, we postulate that one should use as many LVT cells in the sensitive gates as possible; the more LVT cells are used for sensitive gates, the lower the resilience to S-PSC attacks. While this holds true across wide ranges of timing constraints, it can be limited by varied usage of different VT cells in other parts of the design and older technology nodes.

In future work, we would seek to obtain access to commercial libraries for advanced nodes and to advanced PSC attacks, e.g., machine learning-based models. We will release our scripts later on, after modularizing procedures along with outsourcing of technology settings, as we are legally not allowed to share details for commercial libraries.

References

[1] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *International workshop on Cryptograph. Hardw. and Embedded Syst.* Springer, 2001, pp. 251–261.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Cryptology Conf.* Springer, 1999, pp. 388–397.

[3] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Annual International Cryptology Conf.* Springer, 1996, pp. 104–113.

[4] M. Randolph and W. Diehl, "Power side-channel attack analysis: A review of 20 years of study for the layman," *Cryptography*, vol. 4, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2410-387X/4/2/15

[5] S. M. Del Pozo, F.-X. Standaert, D. Kamel, and A. Moradi, "Side-channel attacks from static power: When should we care?" in *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, pp. 145–150.

[6] K. Yang, M. Hicks, Q. Dong, T. Austin, and D. Sylvester, "A2: Analog malicious hardware," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 18–37.

[7] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems - CHES 2004*, M. Joye and J.-J. Quisquater, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 16–29.

[8] M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti, "Leakage power analysis attacks: A novel class of attacks to nanometer cryptographic circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 2, pp. 355–367, 2010.

[9] A. Moradi, "Side-channel leakage through static power," in *Cryptographic Hardware and Embedded Systems – CHES 2014*, L. Batina and M. Robshaw, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 562–579.

[10] T. Popp and S. Mangard, "Masked dual-rail pre-charge logic: Dpa-resistance without routing constraints," in *Cryptographic Hardware and Embedded Systems – CHES 2005*, J. R. Rao and B. Sunar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 172–186.

[11] N. Veyrat-Charvillon, M. Medwed, S. Kerckhof, and F.-X. Standaert, "Shuffling against side-channel attacks: A comprehensive study with cautionary note," in *Advances in Cryptology – ASIACRYPT 2012*, X. Wang and K. Sako, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 740–757.

[12] T. Moos and A. Moradi, "Countermeasures against static power attacks: – comparing exhaustive logic balancing and other protection schemes in 28 nm cmos –," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2021, no. 3, p. 780–805, Jul. 2021. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/8992

[13] S. Skorobogatov and C. Woods, “Breakthrough silicon scanning discovers backdoor in military chip,” in *Cryptographic Hardware and Embedded Systems – CHES 2012*, E. Prouff and P. Schaumont, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 23–40.

[14] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, “Mero: A statistical approach for hardware trojan detection,” in *Cryptographic Hardware and Embedded Systems - CHES 2009*, C. Clavier and K. Gaj, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 396–410.

[15] J. Knechtel, J. Gopinath, M. Ashraf, J. Bhandari, O. Sinanoglu, and R. Karri, “Benchmarking security closure of physical layouts: Ispd 2022 contest,” in *Proceedings of the 2022 International Symposium on Physical Design*, ser. ISPD ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 221–228. [Online]. Available: https://doi.org/10.1145/3505170.3511046

[16] J. Bhandari, J. Gopinath, M. Ashraf, J. Knechtel, and R. Karri, “Defending integrated circuit layouts,” Cryptology ePrint Archive, Paper 2023/205, 2023, https://eprint.iacr.org/2023/205. [Online]. Available: https://eprint.iacr.org/2023/205

[17] G. Guo, H. You, Z. Tang, B. Li, C. Li, and X. Zhang, “Assurer: A ppa-friendly security closure framework for physical design,” in *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, ser. ASPDAC ’23. New York, NY, USA: Association for Computing Machinery, 2023, p. 504–509. [Online]. Available: https://doi.org/10.1145/3566097.3567923

[18] J. Giorgetti, G. Scotti, A. Simonetti, and A. Trifiletti, “Analysis of data dependence of leakage current in cmos cryptographic hardware,” in *Proceedings of the 17th ACM Great Lakes Symposium on VLSI*, ser. GLSVLSI ’07. New York, NY, USA: Association for Computing Machinery, 2007, p. 78–83. [Online]. Available: https://doi.org/10.1145/1228784.1228808

[19] T. Moos, A. Moradi, and B. Richter, “Static power side-channel analysis—an investigation of measurement factors,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 2, pp. 376–389, 2020.

[20] N. Karimi, T. Moos, and A. Moradi, “Exploring the effect of device aging on static power analysis attacks,” *TCHES*, vol. 2019, p. 233–256, May 2019. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/8295

[21] M. Djukanovic, D. Bellizia, G. Scotti, and A. Trifiletti, “Multivariate analysis exploiting static power on nanoscale cmos circuits for cryptographic applications,” in *Progress in Cryptology - AFRICACRYPT 2017*, M. Joye and A. Nitaj, Eds. Cham: Springer International Publishing, 2017, pp. 79–94.

[22] D. Bellizia, S. Bongiovanni, M. Olivieri, and G. Scotti, “Sc-ddpl: A novel standard-cell based approach for counteracting power analysis attacks in the presence of unbalanced routing,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 7, pp. 2317–2330, 2020.

[23] P. Slpsk, P. K. Vairam, C. 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)*, 2019, pp. 1–8.

[24] J. Zhang, G. Su, Y. Liu, L. Wei, F. Yuan, G. Bai, and Q. Xu, “On trojan side channel design and identification,” in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014, pp. 278–285.

[25] W. Meng, W. Zhu, C. Zhang, J. Liu, Z. Guo, D. Gu, W. Fan, H. Zhang, and Y. Yuan, “An implementation of trojan side-channel with a masking scheme,” in *2017 13th International Conference on Computational Intelligence and Security (CIS)*, 2017, pp. 566–569.

[26] L. Lin, M. Kasper, T. Güneysu, C. Paar, and W. Burleson, “Trojan side-channels: Lightweight hardware trojans through side-channel engineering,” in *Cryptographic Hardware and Embedded Systems - CHES 2009*, C. Clavier and K. Gaj, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 382–395.

[27] T. Perez, M. Imran, P. Vaz, and S. Pagliarini, “Side-channel trojan insertion - a practical foundry-side attack via eco,” in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5.

[28] X. Wang, S. Narasimhan, A. Krishna, T. Mal-Sarkar, and S. Bhunia, “Sequential hardware trojan: Side-channel aware design and placement,” in *2011 IEEE 29th International Conference on Computer Design (ICCD)*, 2011, pp. 297–300.

[29] Y. Fei *et al.* (2013) Side channel analysis library. [Online]. Available: https://tescase.coe.neu.edu/?current_page=SOURCE_CODE&software=aestool