Hardware Moving Target Defenses against Physical Attacks: Design Challenges and Opportunities

David S. Koblah
dkoblah@ufl.edu
University of Florida
Gainesville, FL, USA

Fatemeh Ganji
fganji@wpi.edu
Worcester Polytechnic Institute
Worcester, MA, USA

Domenic Forte
dforte@ece.ufl.edu
University of Florida
Gainesville, FL, USA

Shahin Tajik
stajik@wpi.edu
Worcester Polytechnic Institute
Worcester, MA, USA

ABSTRACT

The concept of moving target defense (MTD) has entrenched itself as a viable strategy to reverse the typical asymmetries in cyber warfare. MTDs are technologies that seek to make target systems dynamically change in order to limit the time and information available to complete an attack, increase the likelihood of detection, and/or deter attackers from proceeding. The benefits of MTD have been shown for network-, operating system-, and application-level security. Hardware roots-of-trust, however, are static “sitting ducks", especially against physical attacks, and can therefore benefit from the dynamics brought about by MTDs. Although many MTD concepts seem transferable to hardware applications, there has hardly been any work to establish a functioning research pipeline for countermeasures to physical attacks. The aim of this paper is to introduce viable MTD concepts, describe the issues that they can address, and chart a path towards their realization for the community.

CCS CONCEPTS

• **Security and privacy → Hardware security implementation**; **Hardware attacks and countermeasures**; **Side-channel analysis and countermeasures**.

KEYWORDS

Cyber Deception; Fault Injection; Moving Target Defense; Physical Attacks; Randomization; Side-Channel Analysis; Physically Unclonable Functions; True Random Number Generators.

ACM Reference Format:
David S. Koblah, Fatemeh Ganji, Domenic Forte, and Shahin Tajik. 2022. Hardware Moving Target Defenses against Physical Attacks: Design Challenges and Opportunities. In *Proceedings of the 9th ACM Workshop on Moving Target Defense (MTD '22), November 7, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3560828.3564010

MTD '22, November 7, 2022, Los Angeles, CA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9878-7/22/11…$15.00
https://doi.org/10.1145/3560828.3564010

1 INTRODUCTION

With the ubiquity of digital electronics in our daily lives and critical infrastructures, more assets than ever before are being stored on secure integrated circuits (ICs) as the root-of-trust (RoT). Examples of on-chip assets include secret keys, proprietary firmware and intellectual property (IP), passwords, and personal information. For cryptography, secret keys remain the single point of failure. By acquiring keys, an adversary can effectively destroy all the assurances required for various critical applications. The security of ICs can be compromised by attackers, who can gain access to these devices and mount physical attacks, such as side-channel analysis (SCA) and fault injection (FI) attacks. Similar to approaches taken in software security, current efforts for achieving secure hardware prevent attacks by mitigating vulnerabilities during design (i.e., pre-silicon phase) and/or by detecting attacks in the field (i.e., post-silicon phase).

Typically, there exist asymmetries in cyber warfare that favor attackers. For example, an attacker need only find a single vulnerability in the hardware implementation to compromise a victim while defenders must protect the entire attack surface at all times. In addition, the attacker's arsenal grows over time, while the defender is more of a static “sitting duck”. Approaches such as Cyber Deception (CD) and Moving Target Defenses (MTDs) can alleviate such issues. Systems based on rapidly changing their properties leave the attackers no time to identify their vulnerabilities and make the attack surface unpredictable. While MTD and CD have been successfully applied to software, there are several challenges and new opportunities in transferring them to hardware.

Over the last decades, several classes of countermeasures have been proposed to mitigate the vulnerabilities of hardware implementations against SCA and FI attacks. Many of these countermeasures share some features of CD and MTD, although they are not known by the same names. For instance, various algorithmic countermeasures, have been proposed which rely on the data randomization, circuit reconfigurations, or voltage/clock variations. Masking and hiding against SCA attacks and redundancy against FI attacks are examples of such CD- and MTD-like countermeasures.

Unfortunately, while MTD countermeasures are well established for achieving software and network security, they have been applied to hardware systems in an ad-hoc manner. For instance, while much attention has been paid to MTD-like countermeasures against non-invasive physical attacks (e.g., power and EM side-channel analysis), not enough attention was paid to more advanced semi- or fully-invasive attacks. Similarly, little research has been conducted on how to efficiently and adaptively combine various CD and MTD countermeasures to achieve a higher level of security for hardware against most physical attack classes. On the other hand, some assumptions (e.g., the existence of a reliable source of randomness) for software-level CD and MTD countermeasures have been made to argue about the availability of primitives supporting these countermeasures. Put differently, the existence of the required primitives is taken for granted. However, such assumptions might no longer be valid for hardware systems due to the capabilities of a physical adversary, who can influence the physical conditions of a chip and, consequently, interfere with the random number generation process.

In this paper, we review the differences in threat models of software and hardware systems. We further review various categories of physical attacks and the state-of-the-art CD- and MTD-like hardware countermeasures against physical attacks for different hardware security primitives. We discuss supporting hardware technologies for implementing these countermeasures as well as the fundamental challenges in realizing such protection schemes. Finally, we give an insight into future research directions.

2 BACKGROUND

2.1 Cyber Deception (CD)

Cyber deception (CD) techniques promote "active" defenses in order to counter or reverse asymmetries in cyber warfare. Vouk et al. [122] defines CD as planned actions meant to mislead and/or confuse attackers in order to execute decisions that aid computer security defenses. With this in mind, it's worth discussing the similarities and differences between CD and MTDs. In the literature, there is some debate over whether or not MTD fits under the CD umbrella or if it is a distinct concept that can be utilized along with CD.

Pawlick et al. [79] created a CD taxonomy of six defense detection types, which included MTD. They argued that MTD can be viewed as cryptic, intensive, and motive [79]. *Cryptic* methods prevent an attacker from being certain of a target's information by hiding its true existence, *intensive* type alters a target using its own features, and *motive* type randomizes or changes the same features over time. To this end, MTD uses randomization and reconfiguration in order to limit the attacking potential of an adversary in the time domain.

Another more comprehensive description of MTD by the NITRD program highlights its attempts to increase the required complexity and cost for a desired attack by making the system more robust and adaptable [128]. That is, the dynamics of the system change so that a vulnerability found – but not yet exploited – may not be present in the next system state. In addition, a future state might even neutralize a vulnerability's effects. Wang et al [123] distinguishes CD from MTD by arguing that MTD relies on deployment without assessment of the adversary while CD leverages active engagement, analysis, and manipulation. In this paper, we follow this definition of MTD. That is, MTD is not a CD technique in itself but can be combined with CD for improved defense.

Regardless of their classifications, one cannot deny the possibilities that either concept presents. They have already proven their usefulness to software and network systems, but our central thesis is that they can help bolster hardware security as well.

2.2 Moving Target Defense (MTD) Techniques

MTD techniques can be categorized based on their point of impact. It is important to note that although some of these categories do not apply to hardware security directly, the insights they provide are invaluable to the community. Cyber MTDs can be classified into five main categories [75, 126]:

- **Dynamic data techniques** alter the format, syntax or encoding of application data.
- **Dynamic software techniques** change application code instructions, order, grouping, or format.
- **Dynamic runtime environment techniques** change the environment that the operating system (OS) uses for an application during execution. They are divided into *address space randomization* that changes the layout of memory and *instruction set randomization* that changes interface components, such as processor and system calls, to operate I/O devices.
- **Dynamic platform techniques** modify platform properties like the OS version and CPU architecture.
- **Dynamic network techniques** affect network properties like protocols and addresses.

2.3 Cyberattack Techniques

A critical component of cybersecurity is knowledge of attack techniques. The taxonomy of attacks in the software and network domain is described in [126]. Software and network entities may present different opportunities to adversaries, but there are intersections with hardware systems that can build on existing knowledge. Below are a subset that have counterparts in the hardware security domain (see Sections 3 and 4).

- **Data Leakage Attacks** target critical information, such as cryptographic keys, by examining shared resources, e.g., Prime and Probe attacks [76].
- **Resource Attacks** exhaust or manipulate shared resources to prevent legitimate requests from being fulfilled, such as in denial-of-service (DoS).
- **Injection Attacks** force undesirable behavior at the software level. For example, code injection can be accomplished through buffer overflow while control injection chains existing code snippets to create malware.
- **Scanning Attacks** collect information from devices before launching sophisticated attacks. An example is port scanning where hackers send a message to each port. Based on the received responses, they can determine the services that are running, their associated users, which require authentication, etc.
- **Supply Chain Attacks** occur by targeting less-secure third-party software used by a system or their vendors.

Attacks without a direct hardware equivalent include Exploitation of Authentication, Exploitation of Privilege/Trust, and Spoofing.

2.4 Taxonomy of Weaknesses

MTD techniques are also susceptible to the adaptability of malicious parties. The best case scenario is an MTD type that can cancel any clever scheme an attacker develops, but no system is completely secure. Hence, it is imperative that an MTD technique's weaknesses are known to its designers and users, especially to delay successful attacks. According to [126], the taxonomy of weaknesses is:

- **Limit or Disable**: The attacker may be able to limit or completely stop the MTD technique. If a single component is the root of an MTD's functionality, the attacker's access to it presents a problem to the entire system.
- **Predict**: An MTD technique may proceed as designed, but an attacker might be able to ascertain movements. For example, is may be possible to use a machine learning model to predict its randomization mechanism.
- **Overcome**: Perhaps the worst instance for a security engineer is a working MTD technique that can be brushed aside by an attacker. The adversary may even use the MTD technique in operation to execute an attack.

3 HARDWARE THREAT MODELS

3.1 Physical Attack Threat Model

In contrast to the cyber threats in Section 2.3, where most attacks are mounted remotely on computer systems, physical attacks usually require physical access to the victim device by the adversary. In this case, the attacker cannot only intercept the observable traffic through the I/Os of the device, but she can also physically measure computation and storage based on different quantities, such as timing, power consumption, and electromagnetic (EM) radiation. On the other hand, in addition to injecting false data into the system through the I/Os, she can operate the device under non-standard physical conditions, e.g., by varying the supply voltage, temperature, clock frequency, etc. Moreover, a set of physical attacks can be launched remotely on various hardware platforms by exploiting the physical influence of adjacent IP cores on each other [4, 97]. As a solution, chip vendors provide isolation schemes that separate different IP cores on the chip by creating spatial fences to avoid any communication or fault propagation between them. However, such schemes provide logical rather than physical isolation. Although an IP has no logical access to other IP cores, it can still exploit the shared physical layer to cause damage to other IPs or to intercept information from them. Hence, the physical attacker has much more control over the device under attack compared to the traditional cyber attacker.

3.2 Hardware Supply Chain Threat Model

The hardware supply chain is susceptible to attacks similar to the software supply chain (see Section 2.3), but with additional points of attack. First, today's *IC supply chain* is global and fab-less. It typically involves multiple offshore and untrusted parties: third-party IP (3PIP) vendors, contract foundries (or fabs) and assemblies, and distributors. 3PIP vendors license pre-designed hardware IPs to design houses. The design houses integrate these IPs with their own in-house IPs. They might take care of the remaining design steps such as synthesis, verification, design-for-test, and layout or they might outsource one or more of these steps to another third party. The layout is shared with a fab to manufacture the design into chips, and those chips are packaged by the assembly. The fab and assembly also test the resulting chips. Distributors sell working chips to end users and customers. Like the software supply chain, any one of these third parties might be infiltrated by an adversary to maliciously modify or steal the chip design. In the literature, malicious changes are referred to as hardware trojans [113]. They are activated under rare, specific conditions to remain stealthy, and when triggered can degrade performance, leak information, and act as hidden kill switches. Stolen IP can be analyzed to discover weaknesses and vulnerabilities.

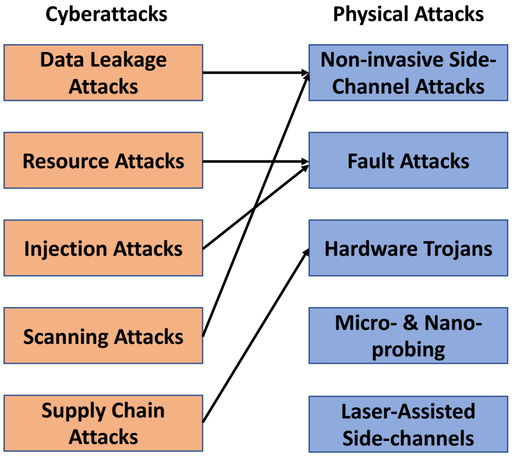


Figure 1: Attack comparisons: The left column abstracts the cyberattacks and the right column shows the closest physical attack counterparts. Note that probing and laser-assisted side-channel attacks are unique to hardware security.

Second, the *PCB supply chain* relies on untrusted third-party manufacturers, distributors, and integrators. The manufacturers fabricate the PCBs. Chips, discrete components, and sockets are purchased from distributors, and integrators solder them to the PCBs. Distributors also sell PCBs to end users and customers. Supply chain attacks might involve modifying the PCB board designs, adding or removing components and connections, or using counterfeit components [114]. For example, a 2018 Bloomberg article [92] reported that spies implanted a chip disguised as a coupler into Supermicro server motherboards that loaded malicious codes from remote attackers. Affected motherboards were allegedly found in over 30 companies and government agencies including Amazon and Apple. In another well-publicized case, Edward Snowden alleged that the National Security Agency (NSA) routinely intercepted routers and other network devices being exported to international customers and implanted backdoor surveillance tools [35].

4 PHYSICAL ATTACKS AGAINST HARDWARE

4.1 Non-invasive Attacks

The closest relatives to data leakage, injection, and scanning attacks from Section 2.3 in the hardware domain are non-invasive SCA and FI attacks, see Figure 1. Such attacks do not require package removal of the IC under attack, making them inexpensive. Power [56], EM [3], and timing [57] analysis are examples of passive SCAs where the attacker only makes observations. On the other hand, voltage glitching, clock glitching, and EM fault attacks, are examples of non-invasive, active FI attacks. These attacks usually require a few hours to succeed. The effectiveness of non-invasive SCA attacks against side-channel protected cryptographic implementations is limited due to their low resolutions and their susceptibility to noise. In other words, while a power/EM probe can capture the entire circuit's power consumption/radiation, it cannot distinguish the activity of individual transistors when the circuit is large and complex, see Figure 2. Thus, the more advanced and expensive physical attacks might be necessary.

4.2 Semi-invasive Attacks

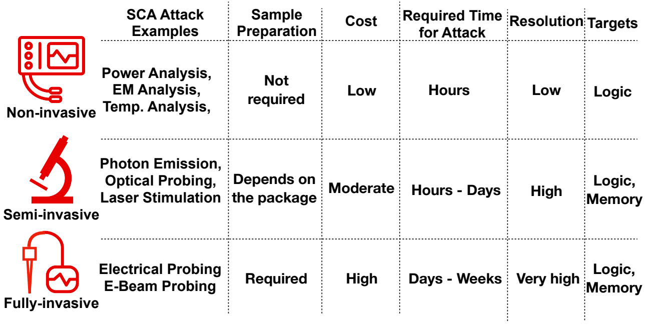
Semi-invasive SCA and FI approaches rely on known optical failure analysis (FA) techniques making them a significant departure from conventional cyber attacks. They provide much higher resolution than non-invasive SCA methods and are less costly than fully-invasive ones. Photonic emission analysis (PEM) [109, 112], laser voltage probing/imaging (LVP/I) [65, 111], laser logic state imaging (LLSI) [58, 59], and thermal laser stimulation (TLS) [59, 66] are examples of semi-invasive SCA techniques. The targets of these attacks include secret keys, PUFs' responses, and on-die transient signals. On the other hand, the most prominent semi-invasive attack is laser fault injection (LFI) [110].

To perform optical attacks, no physical contact with the transistors is necessary. Although such attacks can be carried out from both the frontside (i.e., through top-layer metals) and backside of the IC (i.e., through silicon substrate), the multiple interconnected layers on the frontside of the modern ICs obstruct the optical paths from transistors to the surface of the device. This fact makes the analysis of the target IC from its backside more attractive to the attacker. As a result, only the package's removal on the chip's backside is needed if the proper photon wavelengths are deployed. In the case of flip-chip packages, the silicon substrate on the IC backside is already exposed, and therefore, these attacks can even be mounted non-invasively [66, 86, 111]. Semi-invasive attacks can be accomplished in a matter of days (from initial analysis of an IC's activity to full data extraction), even with limited knowledge of the IC under attack.

4.3 Invasive Attacks

Invasive attacks require direct access to the internal components of an IC or PCB. They are partially or completely destructive and, thus, leave behind evidence of an attack. In the case of ICs, they begin by removing the chip package in order to expose the silicon die [117]. Then, either chemical or dry etching (e.g., focused ion beam or FIB [124]) is used to expose critical wires and/or circuits for imaging and probing. For PCBs, wires can be exposed in more inexpensive ways such as sandpaper, Dremel tools, or CNC milling machines [34].

In passive versions of invasive attacks, the hardware is not modified. Instead, data stored in read-only memory (ROM) is extracted by imaging the IC layout with a scanning electron microscope (SEM). For example, [134] presented a selective staining approach to image data from EEPROM and Flash memories with node sizes of 40nm and 250nm. Alternatively, wires can be exposed and then physically probed to steal data from a running chip or PCB [100].



	SCA Attack Examples	Sample Preparation	Cost	Required Time for Attack	Resolution	Targets
Non-invasive	Power Analysis, EM Analysis, Temp. Analysis,	Not required	Low	Hours	Low	Logic
Semi-invasive	Photon Emission, Optical Probing, Laser Stimulation	Depends on the package	Moderate	Hours - Days	High	Logic, Memory
Fully-invasive	Electrical Probing E-Beam Probing	Required	High	Days - Weeks	Very high	Logic, Memory

Figure 2: Examples of side-channel attacks, requirements, and capabilities: while non-invasive attacks are low cost and can be carried out in a short amount of time, they can be easier mitigated by combining several MTD-like countermeasures. On the other hand, while the semi- and fully-invasive attacks require more resources, they can better bypass or disable MTD-like countermeasures.

On the other hand, active versions of invasive attacks disable on-chip, security protection circuits by cutting critical, internal metal wires or destroying the entire circuit [42]. As another example, an SRAM PUF was cloned using a FIB in [41]. For PCBs, active attacks include adding wires or modchips, which have been used by end users to break DRM protections of video game consoles [102].

5 MTD-RELATED METHODS IN HARDWARE

5.1 MTD/CD as SCA Countermeasures

MTD can be thought of as an alternative defense approach, which aims to design systems with varying parameters to defeat the knowledgeable attacker. In this regard, even if the attacker could gain some information to compromise the security of the system, periodic changes made to that should prevent the attacker from extracting the secret. Although not yet fully realized, some techniques developed to impair the effectiveness of SCA can be categorized as MTD methods, see Figure 3. In this context, three main classes of such techniques are (1) hiding, (2) inducing misalignment, (3) partial reconfiguration, and (4) masking.

5.1.1 *Hiding.*

The goal of hiding countermeasures is to directly change the power characteristics in order to reduce the signal-to-noise ratio (SNR). In this class of countermeasures, reducing the SNR is achieved by either raising the noise floor, e.g., using additional noise sources or by balancing the instantaneous power consumption [61].

An example of cryptic CD: Equalizing the instantaneous power consumption. Main proposals for this have applied variants of dual-rail pre-charge logic for equalization [22]. One of the first studies devoted to this has presented dual-rail pre-charge logic styles that have been initially designed for application-specific integrated circuits (ASICs) [115]. Similarly, [19, 82, 83] have considered implementation variants relying on this concept. For instance, separated wave dynamic differential logic (SWDDL) has been used to balance the power consumption at the price of area overhead [116], although it has been experimentally shown to fail due to the glitches caused by a race between a global signal (pre-charge) and local signals (differential data pairs) cf. [37]. To deal with this, Masked Dual-Rail Pre-Charge Logic (MDPL) was proposed [83] to implement secure circuits using a standard CMOS cell library. They further have relaxed the constraints on the place-and-route since the random masking handles the difference of loading capacitance between all pairs of complementary logic gates. Nevertheless, it has been demonstrated that the leakage even occurs in the MDPL gates, similar to WDDL gates, when input signals have a difference in delay time [106]. The shortcomings identified shortly after their introduction have rendered the adoption of such methods difficult to implement on ASICs.

Unfortunately, they cannot be applied directly to field-programmable gate arrays (FPGAs) either. To propose an implementation of the balanced circuit on FPGAs, the efficacy of double WDDL (DWDDL) has been discussed in [116, 133], although these are prone to SCA due to dissimilar signal delays and wire capacities on an FPGA [127], as also studied in [107, 108]. Such problems are particularly acute when considering the implementation on FPGAs. Besides [72, 116], there are only a few proposals in this regard, which have mainly discussed specific types of FPGAs [10, 39, 40, 53, 67, 74, 96, 133]. Work presented in [10, 39, 40, 53, 67, 72, 74, 96, 133] are especially interesting since the notion of *duplication* has been put forward, where a part of a circuit is re-instantiated and placed at another location on the FPGA to act as a dual function. This can be easily supported by FPGAs containing similar blocks, where each block is formed by a couple of slices with (almost) equal inter- and intraconnections. Nevertheless, these studies have been proven flawed and susceptible to SCA cf. [127]. One of the most prominent and perhaps the most promising candidate is the GliFreD framework designed particularly for Xilinx FPGAs (a simplified sketch of its mechanism is drawn in Figure 4). Although it resolves the issues with early propagation, the glitches, and the necessity of a dual-rail routing tool, it still cannot ideally equalize the power consumption due to the process variation violating the balance between the cloned routes [73]. Despite the effort made in this line of research, it has been concluded that the SCA-resiliency of these types of hiding schemes must be boosted by combining them with other countermeasures, namely masking [63, 73].

An example of MTD: Raising the noise floor. One of the first studies that have considered generating noise to defeat SCA is [52]. In line with this work, randomly activating ring oscillators (ROs) for noise generation has been explored as an example of a generic countermeasure [64]. Recent advancements in this context include the design of an RO-based active fence between attacker and victim, when the on-chip SCA is concerned [61]. The proposed design is delicate in the sense that the fence is implemented as a row-by-row RO array, with the row activation depending on either a sensor value for power equalization or a pseudo-random number generator (PRNG) for noise increase; hence, this countermeasure can serve as an example of MTD or CD. As a continuation of that study, in [60], RO arrays are deployed around the AES modules, which are randomly activated to increase the noise level and, consequently, make the SCA more difficult. In the same vein, [130] has investigated the possibility of using programmable ROs (PROs) for injecting a random noise pattern into the design's power consumption. This study has also discussed the application of PROs in on-chip power monitoring of the fluctuations in the power distribution network (PDN). In this way, it is possible to detect anomalies, i.e., electromagnetic fault injection and power glitches, as well as hardware Trojans; however, the PROs generating the noise do not work adaptively to react to such anomalies. Both studies presented in [60, 130] have noted an important observation: placement of circuits (either malicious ones or countermeasures) on the chip, while sharing a common PDN), could have an impact on both the detection and effectiveness of the hiding-based countermeasure. Nevertheless, to the best of our knowledge, no dynamic reaction technique has been proposed to adapt the noise level in response to malicious activity (e.g., FI or SCA).

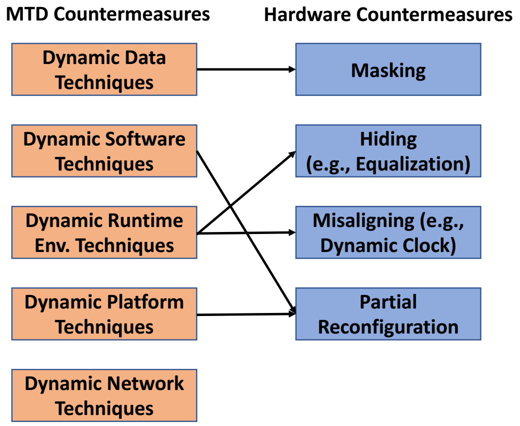

MTD Countermeasures
Hardware Countermeasures
Dynamic Data Techniques
Dynamic Software Techniques
Dynamic Runtime Env. Techniques
Dynamic Platform Techniques
Dynamic Network Techniques
Masking
Hiding (e.g., Equalization)
Misaligning (e.g., Dynamic Clock)
Partial Reconfiguration

Figure 3: MTD countermeasures comparison: The left column abstracts the MTD countermeasures, and the right column shows the hardware countermeasures counterparts.

5.1.2 Misalignment. Misalignment methods attempt to obfuscate the relation between the power consumption of the device at a certain time and the intermediate values generated or processed by the cryptographic core. When considering software implementation, insertion of random delays through dummy operations [20, 21] and shuffling [121] are among the most frequently studied countermeasures against SCA. The latter is well in accordance with the definition of dynamic software MTD techniques (see Section 2.2). Specifically, shuffling deals with the randomization of the execution order of the instructions [43, 91] and the physical resources used in the scheme, e.g., registers used to perform permutation. It has been demonstrated that without the proper randomization of hardware resources, "indirect leakages" are observable due to the different power consumption models of the hardware resources [121].

On the other hand, for hardware implementations, countermeasures have relied on the unstable clock, random hardware interruption, and clock stealing cf. [13]. Prime examples of such implementations on FPGAs are an unstable clock with randomly scattered frequencies [26, 50], phase shifts [38, 89], or execution delays [68, 71]. A more recent approach in this class has taken into account clock management and generator subsystems available in a family of FPGAs to address the resource and throughput overhead, which is the main drawback of misalignment countermeasures [45]. In spite of these efforts made to prove misalignment methods effective, numerous realignment techniques have jeopardized the security of the system depending on these countermeasures. Such techniques range from pattern matching [2], Dynamic Time Warping (DTW) [118], FFT [98] or Sliding Window (SW) integration [25] cf. [44].

5.1.3 Partial Reconfiguration. In this category, countermeasures leverage partial reconfiguration as an inherent feature of modern FPGAs to allow the circuit to be modified at certain blocks of logic during runtime without interrupting the operation of other blocks. These countermeasures are devised to stop an attacker from mounting FI and SCA attacks and most closely resemble dynamic platform MTD techniques (see Section 2.2 and Figure 5). [71] has presented one of the first approaches in this category, where temporal jitter is induced by adding or removing registers between subfunctions of an AES implementation. Furthermore, by relocating the subfunctions to four different positions on the chip, spatial jitter is created. Nevertheless, the number of possible configurations (maximum 10 reconfigurations) was not high enough to defeat adversaries. In line with this, Hettwer et al. [47] have proposed a reconfiguration-based countermeasure, where a single synthesized netlist is employed to generate different physical configurations of the Register Transfer Level (RTL) description corresponding to a cryptographic algorithm. These configurations are dynamically exchanged through partial configuration. In this respect, multiple interesting aspects have been pointed out. First, in order to get the most out of reconfiguration, it should be done within an encryption/decryption operation. The drawback of this is, however, the requirement for additional registers to store the cipher state (context storage), which must be protected by means of another countermeasure. As an alternative solution, after a number of encryption runs, the circuits can be reconfigured without modifying the RTL, although the adversary may collect several traces from the same configuration, making the attack relatively easier.

Second, the physical layout obtained for each partial bitstream should be different; on the other hand, it is recommended to keep some parts of the design (e.g., S-Boxes and corresponding registers) together to have a short routing. Additionally, the size of the reconfigurable area should be large enough to allow diverse placement and routing options, and consequently, more physically-distinct implementations. According to these observations, Hettwer et al. have switched the complete AES implementation via partial reconfiguration while only keeping the control logic static [47]. As a result, the position and wiring of all important logic elements are forced to change during each reconfiguration. Nonetheless, there are other options for reconfiguration as enumerated in [70].

For instance, [38] has introduced a countermeasure for AES, in which LUT, SRL, BRAM, and digital clock managers (DCM) are used for different purposes, namely (1) the generation of Gaussian noise by using LUTs, (2) the randomization of the clock by using DCMs, and (3) the scrambling of the S-box via BRAMs. Another proposal in [95] has introduced configurable lookup tables (CFG-LUTs) as an effective way to implement randomly reconfigurable S-boxes. The work in [94] has been devoted to countermeasures against the attacks that target the value of the intermediate signals. For this, cryptographic algorithms are divided into basic elements implemented in the BRAM blocks of an FPGA that are randomly substituted along with the random encoding of the intermediate connections cf. [70].

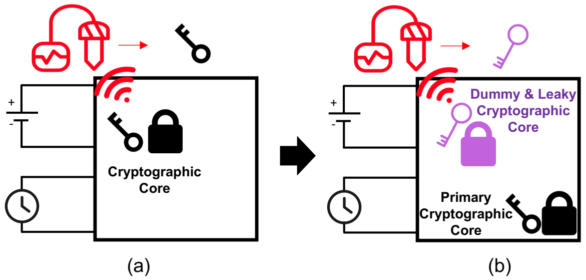


Figure 4: Hardware-based Cyber Deception: Running dummy cryptographic operations parallel to the original cryptographic core's computations to fool the adversary by leaking wrong keys.

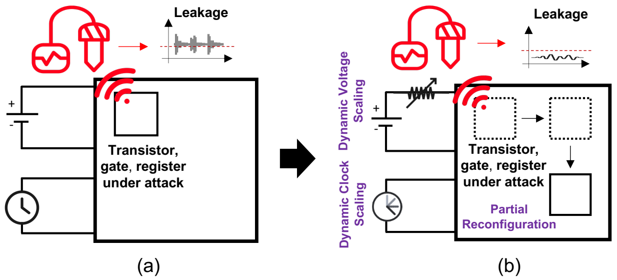


Figure 5: Hardware-based Moving Target Defense: Deploying dynamic voltage variations, clock variations, dynamic reconfigurations, etc., to add noise to the leaked information.

Finally, [54] has formulated the notion of partial reconfiguration-based countermeasures in the MTD framework. For this purpose, four realizations of the Sbox function of the AES and 64 random noise sources have been used, where each of the Sbox variants can be mapped randomly to any of those 16 partially reconfigurable regions. In line with that, [5] has suggested that modifications at the synthesis level could offer more freedom to change the structure of the circuit.

To sum up, it should be emphasized that although these techniques have been shown effective, the trade-off between the complexity of reconfiguration (including area and throughput costs) and the reduced leakage should be further studied.

5.1.4 Masking. Masking is a prime example of how data randomization, as an MTD technique, has found application in hardware security and in particular, secure execution. According to its definition, masking adds noise intentionally by randomly changing the secret [87]. Masking countermeasures have proven effective and theoretically sound, built on the pioneering work of Chari et al. [18], and Goubin et al. [33]. Their work has suggested applying the so-called XOR-secret sharing or Boolean masking countermeasure: each bit b is represented by k random bits, whose exclusive-or is equal to b. In doing so, the inputs of the circuit are *masked* with the random numbers, while the circuit's gates are replaced by clusters of gates (so-called *gadgets*), which have the same functionality as the original circuit, although being augmented by the random numbers.

The masking scheme introduced above is just one form of masking, extended to fit the purpose of different applications and provide a stronger security guarantee, e.g., security against physical attacks (e.g., glitches). To this end, polynomial masking relying on Shamir's sharing scheme and multi-party computation techniques has been introduced [14, 32, 85]. Another example of more advanced masking has been proposed in [6, 7] that offers more resilience to SCA. For this, the masked variable is represented by $2n$ shares in the form of two random vectors (L, R) of n elements each so that the sensitive variable S is equal to the inner product of L and R. Other examples include direct sum masking (DSM) [12, 17, 84] and its general instance, code-based masking [15, 16], where their elaborate algebraic structure leads to improved security properties, although at the cost of expensive computing over the encoding [36].

To conclude the discussion in this section, we stress that masking and other countermeasures devised against SCA are involved in a rich field of study, where even tools have been developed to assess the security of a design. Despite these efforts and similarities between MTD/CD techniques and side-channel countermeasures, to the best of our knowledge, cross-cutting problems across these domains have remained unanswered. For instance, it is not clear how the randomness needed in both domains should be provided. Therefore, we expect to witness a substantial increase in the number of interdisciplinary studies devoted to answering such questions.

5.2 MTD for Trojan Detection and Prevention

MTD concepts have been adapted to thwart hardware Trojan insertion, but, to our knowledge, only within stages of the FPGA flow. Zhang et al. [136] made the outputs of design mapping and place-and-route tools unpredictable. Their method uses multiple replicas of the same design, along with slice positions and submodules, to randomly configure the FPGA, thus significantly reducing the precision of a Trojan inserted via malicious design software. If a Trojan is successfully inserted, the framework composed of runtime pin grounding (RPG) and hardware moving target defense (HMTD) was proposed to detect and nullify its effect [135]. The RPG stops communication with the external environment via unused input/output pins by grounding them, while the HMTD prevents interference with FPGA replacement in legacy systems by comparing randomly-picked module-to-replace (MTR) copies and flagging inconsistent outputs as Trojans.

Undoubtedly, both frameworks rely on possessing a large pool of replica designs similar to dynamic software techniques (see Section 2.2). One issue most random-selection mechanisms face is the possibility of attacking the generator directly. The version used in [136] is a pseudo-random selector created by a user-defined arbitrary logic function. With the exponential growth of machine learning, an attacker may also model and predict the selection process (see Section 2.4).

5.3 MTD for Secure Key Generation and Storage

5.3.1 Physically Unclonable Functions (PUFs). For cryptographic protocols or primitives, it is essential to attain the objectives specified during the design process, namely secure key generation and storage. These objectives are relevant to the notion of root-of-trust introduced to deal with this by providing adequate reasoning with respect to physical security [69]. A root-of-trust is, in particular, a primitive composed of hardware and/or software to offer trusted, security-critical functions [120]. Traditionally, the root-of-trust is realized by a secret key embedded in the hardware [77], e.g., a key stored in the non-volatile memories of the IC; nevertheless, the vulnerability of such legacy key storage methods to physical attacks has been demonstrated in the literature [42, 66]. In this regard, physically unclonable functions (PUFs) have been identified as a promising solution to secure key generation and storage issues [30]. PUFs leverage the inherent characteristics of devices in terms of process variations and imperfections of metals and transistors in identical chips in order to offer a device-specific fingerprint. Mathematically formalizing this, a PUF is a mapping generating virtually unique outputs (i.e., responses) to a given set of input bits (i.e., challenges). These responses can be used either to authenticate and identify a device or to generate keys for cryptographic modules. The volatile nature of PUFs makes them more difficult to extract their secrets/responses using invasive attacks (e.g., imaging) when chips are un-powered. Various non-invasive and sem-invasive attacks, however, have proven PUFs less effective than expected [28].

An important class of such vulnerabilities include machine learning (ML) attacks, where ML algorithms are applied to determine the input/output (challenge/response) behavior of PUFs. This leads to predicting the response of the PUF to an unseen challenge and, consequently, a decrease in the entropy of the generated key or a failure in the authentication process. One of the main success factors of such attacks is the static nature of the circuitry, realizing the challenge/response behavior of PUFs. The countermeasures developed in this regard and relevant to MTD can be traced back to using the multiplexer to select PUF instances implemented on an ASIC or FPGA [93]. In another attempt, a PUF circuit is physically swapped partially or entirely through the dynamic reconfigurability feature of mainstream FPGAs [101]. This has been performed on a trial-and-error basis and in a blind fashion; however, reconfigurability has been shown to be helpful. Another issue with the proposed approach corresponds to the difficulty of resource allocation and implementation of a strong PUF that requires precise and symmetric routing constraints, not achievable by random reconfigurations. Therefore, it is preferred to partially reconfigure the PUF and swap only a few stages of it. This issue is addressed in [29], where a systematic methodology is developed to identify which PUF stages should be reconfigured. To the best of our knowledge, the proposed PUF is the first of its kind regarding not only compliance with the concept of MTD, but also a provable security guarantee.

5.3.2 True Random Number Generators (TRNGs). True random number generators have become an integral part of virtually all keyed cryptographic primitives. To generate keys, these primitives have been considered promising thanks to their specific characteristics, including unpredictability. Physical TRNGs extract randomness from physical processes, and nondeterministic processes, e.g., Johnson's noise, Zener noise, radioactive decay, photon path splitting at the two-way beam splitter, photon arrival times, etc. [103]. Along with the randomness source, an entropy harvesting mechanism should be included in the design of TRNGs, which is further equipped with a post-processing stage to provide a uniform distribution. When implementing TRNGs on FPGAs, main randomness sources include timing jitter of Ring Oscillators (ROs), Phase Locked Loops (PLLs), and metastability of logic cells cf. [119]. Implementation of TRNGs on FPGAs has been identified as challenging due to the careful placement and routing required in this case [132]. Nonetheless, such implementations are advantageous since they could rely on purely digital components; hence, the designs could be relatively simple as they leverage the computer-aided design (CAD) tools available for FPGAs [51].

Among FPGA-based TRNGs, [51] has proposed an architecture that allows on-the-fly tuning of statistical qualities of a TRNG through DPR capabilities of modern FPGAs for varying the digital clock manager (DCM) modeling parameters. The proposed tunable jitter control capability depends on dynamic partial reconfiguration (DPR) that is available on Xilinx FPGAs. This leads to the modification of the output frequency of the TRNG without reconfiguring the entire circuit. As a follow-up to this study, [27] has investigated the properties of the DCM-based TRNG to find the best possible source of randomness on the FPGA. None of these studies has mentioned the feasibility of applying their techniques to offer tolerance in the presence of environmental changes and, more importantly, tampering and attacks. This, however, seems possible and could be an interesting topic to be explored in the MTD framework.

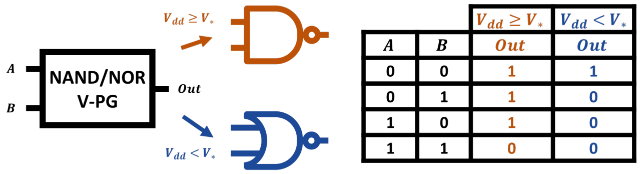
6 SUPPORTING TECHNOLOGIES FOR MTD IN HARDWARE

6.1 Random Number Generators

A source of randomness is the primary requirement of all MTD countermeasures. If the attacker can predict the next state of the circuit in an MTD countermeasure, the protection scheme becomes ineffective. Therefore, high entropy sources are needed to make unpredictable moves possible. TRNGs are the most prominent candidates introduced to exploit physical sources of noise to generate random numbers. Classical and quantum physical phenomena, e.g., thermal and shot noise, are examples of such noise sources. In addition to TRNGs, PRNGs, such as Linear Shift Feedback Registers (LFSRs), are deployed to generate pseudo-noise sequences. In contrast to TRNGs, PRNGs do not employ physical phenomena and instead rely on circuits with a large number of states, making the prediction of the following states quite impossible.

6.2 Hardware Reconfigurability

Randomization of addresses or movement of data in software is natural and can be easily achieved. While hardware MTD techniques, such as data randomization or the inclusion of jitter in hardware, are also practical, other techniques, such as physical movement of the circuit components, cannot be achieved if specific technologies are not considered. In this case, reconfigurable hardware technologies can enable MTD countermeasures. While there are different ways to obtain reconfigurable hardware, here, we mention three different candidates, which are currently used or have high potentials for future technologies.



		$V_{dd} \geq V_*$	$V_{dd} < V_*$
A	*B*	*Out*	*Out*
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0

Figure 6: Voltage-controlled polymorphic (V-PG) NAND/NOR gate and truth table. When $V_{dd} \geq V_*$, the V-PG acts as NAND. Else, it behaves as NOR.

6.2.1 *Programmable Logic.*

Reconfigurable logic devices utilize an array of identical programmable cells to allow reconfigurable implementation of logic functions in hardware. Field programmable gate arrays (FPGAs) are the prominent instances of reconfigurable hardware. Mainstream FPGAs support a feature, called dynamic or partial reconfiguration, enabling the placement and routing updates as well as replacing logic functions possible. This feature has been effectively used in hardware security to mitigate the non-invasive side-channel leakages.

Unfortunately, conventional ASICs do not benefit from the reconfigurability features. However, with the introduction of embedded FPGA (eFPGA) technologies in the last few years, dedicated FPGA fabric IPs can be included in ASICs. Naturally, such IPs can be used to enable MTD countermeasures.

6.2.2 *Multiplexers and Demultiplexers.*

MTD countermeasures can also be realized on ASICs through the realization of function or routing redundancies. In this case, the switching between various functions and signal routes can be obtained using standard multiplexers and demultiplexers. The advantage of this solution is that the MTD countermeasures can be realized using standard digital logic tools. The downside is, however, that the redundant circuits cause a large overhead. Moreover, if it becomes evident that the reconfiguration is vulnerable to a specific physical attack, the design cannot be patched to update the existing MTD scheme.

6.2.3 *Polymorphic Circuits.*

The dictionary definition of polymorphism is "the quality or state of existing in or assuming different forms" [1]. In cybersecurity, the term is now associated with a type of malware that constantly changes its features in order to evade detection as well as to describe the associated countermeasures of such malware. Based on these definitions, one can think of polymorphism in hardware in two ways: (1) *static or compile time* where no two systems are created to be exactly alike (e.g., different logic, layout, etc.). This can limit the effectiveness of the attacker's prior knowledge, especially if each design has different security weaknesses; and (2) *dynamic or runtime* where the system's behavior changes in response to runtime conditions. The former is best accomplished with programmable logic and eFPGAs (see above). While the latter can be accomplished with dynamic partial reconfiguration or multiplexers (also discussed above), polymorphic circuits are even better due to their fast response time and low overhead.

Polymorphic circuits were first proposed by Stoica et al [105]. They superimpose two or more functions into a single digital circuit such that switching between the functions is controlled by changes in the external environment (e.g., temperature, supply voltage, light, etc.), rather than digital signals (see Figure 6 for an example). In the area of hardware security, polymorphic circuits have mostly been limited to niche applications such as logic locking, camouflaging, and watermarking [78, 88, 125] to protect IP. However, Bi et al. were the first to propose them for active defenses [11]. Specifically, the polymorphic behavior of graphene-based symmetric tunneling FETs (SymFETs) was used to counter voltage fault injection (VFI) attacks. Polymorphic SymFET current and voltage protection circuits can severely limit current and voltage, respectively, *as soon as the supply voltage* leaves a range around the nominal value.

7 CHALLENGES AND FUTURE DIRECTIONS

While several MTD-like hardware countermeasures have been developed against SCA and FI attacks, there are still open challenges, which need to be addressed. Here, we elaborate on some of the challenges and research opportunities.

7.1 Randomness Source Vulnerabilities

The primary requirement for the successful realization of MTD countermeasures is the existence of randomness sources, as it makes not only the repetition and integration of the measurements infeasible but it also makes the prediction of the next states of the circuit impossible. However, depending on her capabilities, a physical adversary could deactivate the random source and neutralize the MTD countermeasures. For instance, the attacker can halt the clock of the circuit. In this case, countermeasures relying on randomness, such as masking and hiding, become ineffective, and the entire state of the circuit can be recovered using advanced static attack methods [58, 59]. Similarly, the attacker can inject faults into the TRNG of the system to either reduce its entropy by biasing or disabling it entirely [24, 131]. Therefore, new research is required to provide protection for the random source itself. One solution would be the Independence of the random source circuit from global clock signals. This might be achieved by self-timed or asynchronous circuits.

7.2 Polymorphic Circuits

Challenges and opportunities for utilizing polymorphic circuits to realize MTDs in hardware include lack of design automation, reliance on beyond-CMOS technologies, and limited demonstration of applications. Polymorphic circuits have not been widely adopted due to the challenges of designing polymorphic gates and performance-optimized circuits. Specifically, existing polymorphic gates are inefficiently created using evolutionary algorithms [104]. Further, it is extremely difficult to achieve timing closure of the circuit's multiple functions at low overhead. While polymorphic gates can be implemented in CMOS, the existing research in hardware security focuses on beyond-CMOS technologies [11, 78, 88], thereby limiting their use in current chips. Bi et al. [11] consider polymorphic circuits for fault injection, but they are likely applicable as MTDs against other physical attacks as well, which demands exploration.

7.3 Overhead and Adaptive MTD/CD

The main downside of the MTD countermeasures is their high overhead in terms of power, performance, and area. Thus, it is imperative to develop intelligent MTD countermeasures, which become active if only a threat is detected. For instance, it has been shown that cryptographic hardware might leak more information at higher temperatures or higher clock frequencies. As a result, the MTD could be activated upon the detection of such physical conditions using the on-chip sensors or polymorphism. Similarly, upon the detection of the system-level tampering using more advanced anti-tamper technologies, the MTD countermeasures could be activated to add another layer of the defense to the system. The MTD methods also could contain various levels of randomization for different situations. Incorporating more advanced decisions and policies based on game theory could also bring such MTD approaches closer to the realm of CD.

7.4 AI-assisted Attacks Against MTDs

As discussed in Section 2.4, MTD techniques must be mindful of the attacker's ability to predict behavior. Traditionally, the profile of a device has been obtained by characterizing the leakages precisely through statistical techniques, e.g., linear regression [23, 99]. Shortly after the introduction of ML to SCA [48, 49, 62], neural networks (NNs) were proposed as powerful profiling models [9, 13, 31, 46, 55, 80, 81, 90, 129]. In particular, it has been demonstrated that NNs can further defeat some countermeasures designed to protect a cryptographic implementation. Specifically, the jitter-based misalignments in the side-channel traces, i.e., creating an array of asynchronous measurements, cannot stop an attacker from launching SCA through NNs [8, 13]. Even masked implementations, with countermeasures that randomize the intermediate values, can be successfully attacked by NNs [55, 80, 129].

8 CONCLUSION

In this paper, first, we reviewed the established countermeasures developed following the concepts of CD and MTD for software and network systems to mitigate cyber-attacks. We further discussed the threat model corresponding to hardware systems and how they differ from conventional cyber threats. By reviewing various categories of physical attacks, we discussed the challenges of conventional CD- and MTD-like hardware countermeasures against such attacks. Finally, we discussed the challenges of implementing MTD countermeasures in hardware and provided some thoughts about the opportunities for future research directions.

ACKNOWLEDGMENT

This effort was sponsored in part by NSF under grant number 2117349 and in part by NSF grant number 2150123.

REFERENCES

[1] 2022. Polymorphism Definition. [Online] https://www.merriam-webster.com/dictionary/polymorphism [Accessed: Sept.9, 2022]. (2022).

[2] Karim M Abdellatif, Damien Couroussé, Olivier Potin, and Philippe Jaillon. 2017. Filtering-based CPA: a successful side-channel attack against desynchronization countermeasures. In *Proceedings of the Fourth Workshop on Cryptography and Security in Computing Systems*. 29–32.

[3] Dakshi Agrawal, Bruce Archambeault, Josyula R Rao, and Pankaj Rohatgi. 2002. The EM side—channel (s). In *International workshop on cryptographic hardware and embedded systems*. Springer, 29–45.

[4] Md Mahbub Alam, Shahin Tajik, Fatemeh Ganji, Mark Tehranipoor, and Domenic Forte. 2019. RAM-Jam: Remote temperature and voltage fault attack on FPGAs using memory collisions. In *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE, 48–55.

[5] Ali Asghar, Benjamin Hettwer, Emil Karimov, and Daniel Ziener. 2021. Increasing Side-Channel Resistance by Netlist Randomization and FPGA-Based Reconfiguration. In *International Symposium on Applied Reconfigurable Computing*. Springer, 173–187.

[6] Josep Balasch, Sebastian Faust, and Benedikt Gierlichs. 2015. Inner product masking revisited. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 486–510.

[7] Josep Balasch, Sebastian Faust, Benedikt Gierlichs, and Ingrid Verbauwhede. 2012. Theory and practice of a leakage resilient masking scheme. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 758–775.

[8] Ryad Benadjila, Emmanuel Prouff, Rémi Strullu, Eleonora Cagli, and Cécile Dumas. 2018. Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to ASCAD database. *IACR Cryptol. ePrint Arch., 2018/053* (2018).

[9] Shivam Bhasin, Anupam Chattopadhyay, Annelie Heuser, Dirmanto Jap, Stjepan Picek, and Ritu Ranjan. 2020. Mind the Portability: A Warriors Guide through Realistic Profiled Side-channel Analysis. In *NDSS 2020*.

[10] Shivam Bhasin, Sylvain Guilley, Florent Flament, Nidhal Selmane, and Jean-Luc Danger. 2010. Countering early evaluation: an approach towards robust dual-rail precharge logic. In *Proceedings of the 5th Workshop on Embedded Systems Security*. 1–8.

[11] Yu Bi, Kaveh Shamsi, Jiann-Shiun Yuan, Pierre-Emmanuel Gaillardon, Giovanni De Micheli, Xunzhao Yin, X Sharon Hu, Michael Niemier, and Yier Jin. 2016. Emerging technology-based design of primitives for hardware security. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 13, 1 (2016), 1–19.

[12] Julien Bringer, Claude Carlet, Hervé Chabanne, Sylvain Guilley, and Houssem Maghrebi. 2014. Orthogonal direct sum masking–A Smartcard Friendly Computation Paradigm in a Code, with Builtin Protection against Side-Channel and Fault Attacks. In *IFIP International Workshop on Information Security Theory and Practice*. Springer, 40–56.

[13] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. 2017. Convolutional neural networks with data augmentation against jitter-based countermeasures. In *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 45–68.

[14] Claude Carlet, Abderrahman Daif, Sylvain Guilley, and Cédric Tavernier. 2019. Polynomial direct sum masking to protect against both SCA and FIA. *J. of Cryptographic Engineering* 9, 3 (2019), 303–312.

[15] Claude Carlet, Jean-Luc Danger, Sylvain Guilley, and Houssem Maghrebi. 2012. Leakage squeezing of order two. In *International Conference on Cryptology in India*. Springer, 120–139.

[16] Claude Carlet, Jean-Luc Danger, Sylvain Guilley, Houssem Maghrebi, and Emmanuel Prouff. 2014. Achieving side-channel high-order correlation immunity with leakage squeezing. *Journal of Cryptographic Engineering* 4, 2 (2014), 107–121.

[17] Claude Carlet and Sylvain Guilley. 2015. Complementary dual codes for countermeasures to side-channel attacks. In *Coding Theory and Applications*. Springer, 97–105.

[18] Suresh Chari, Charanjit S Jutla, Josyula R Rao, and Pankaj Rohatgi. 1999. Towards sound approaches to counteract power-analysis attacks. In *Annual Intrl. Cryptology Conf.* Springer Berlin Heidelberg, Berlin, Heidelberg, 398–412.

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

[20] Jean-Sébastien Coron and Ilya Kizhvatov. 2009. An efficient method for random delay generation in embedded software. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 156–170.

[21] Jean-Sébastien Coron and Ilya Kizhvatov. 2010. Analysis and improvement of the random delay countermeasure of CHES 2009. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 95–109.

[22] Jean-Luc Danger, Sylvain Guilley, Shivam Bhasin, and Maxime Nassar. 2009. Overview of dual rail with precharge logic styles to thwart implementation-level attacks on hardware cryptoprocessors. In *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*. IEEE, 1–8.

[23] Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert. 2011. Univariate side channel attacks and leakage modeling. *Journal of Cryptographic Engineering* 1, 2 (2011), 123.

[24] Maik Ender, Samaneh Ghandali, Amir Moradi, and Christof Paar. 2017. The first thorough side-channel hardware trojan. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 755–780.

[25] Dor Fledel and Avishai Wool. 2018. Sliding-window correlation attacks against encryption devices with an unstable clock. In *International Conference on Selected Areas in Cryptography*. Springer, 193–215.

[26] Austin W Fritzke. 2012. Obfuscating against side-channel power analysis using hiding techniques for aes. (2012).

[27] Naoki Fujieda, Masaaki Takeda, and Shuichi Ichikawa. 2019. An analysis of DCM-based true random number generator. *IEEE Transactions on Circuits and Systems II: Express Briefs* 67, 6 (2019), 1109–1113.

[28] Fatemeh Ganji and Shahin Tajik. 2022. Physically Unclonable Functions and AI. In *Security and Artificial Intelligence*. Springer, 85–106.

[29] Fatemeh Ganji, Shahin Tajik, Pascal Stauss, Jean-Pierre Seifert, Mark Tehranipoor, and Domenic Forte. 2020. Rock'n'roll PUFs: Crafting Provably Secure PUFs from Less Secure Ones (Extended Version). *Journal of Cryptographic Engineering* (2020). https://doi.org/10.1007/s13389-020-00226-7

[30] Blaise Gassend, Dwaine Clarke, Marten Van Dijk, and Srinivas Devadas. 2002. Silicon physical random functions. In *Proceedings of the 9th ACM conference on Computer and communications security*. 148–160.

[31] Richard Gilmore, Neil Hanley, and Maire O'Neill. 2015. Neural network based attack on a masked implementation of AES. In *Intrl. Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 106–111.

[32] Louis Goubin and Ange Martinelli. 2011. Protecting AES with Shamir's secret sharing scheme. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 79–94.

[33] Louis Goubin and Jacques Patarin. 1999. DES and differential power analysis the "Duplication" method. In *Intrl. Workshop on Cryptographic Hardware and Embedded Systems*. Springer, Berlin, Heidelberg, 158–172.

[34] Joe Grand. 2014. Printed circuit board deconstruction techniques. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)*.

[35] Glenn Greenwald. 2014. *No place to hide: Edward Snowden, the NSA, and the US surveillance state.* Macmillan.

[36] Vincent Grosso, Emmanuel Prouff, and François-Xavier Standaert. 2014. Efficient masked S-boxes processing–a step forward–. In *International Conference on Cryptology in Africa*. Springer, 251–266.

[37] Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, Tarik Graba, and Yves Mathieu. 2008. Evaluation of power-constant dual-rail logic as a protection of cryptographic applications in FPGAs. In *2008 Second International Conference on Secure System Integration and Reliability Improvement*. IEEE, 16–23.

[38] Tim Güneysu and Amir Moradi. 2011. Generic side-channel countermeasures for reconfigurable devices. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 33–48.

[39] Wei He, Eduardo de la Torre, and Teresa Riesgo. 2011. A precharge-absorbed DPL logic for reducing early propagation effects on FPGA implementations. In *2011 International Conference on Reconfigurable Computing and FPGAs*. IEEE, 217–222.

[40] Wei He, Andrés Otero, Eduardo de la Torre, and Teresa Riesgo. 2012. Automatic generation of identical routing pairs for FPGA implemented DPL logic. In *2012 International Conference on Reconfigurable Computing and FPGAs*. IEEE, 1–6.

[41] Clemens Helfmeier, Christian Boit, Dmitry Nedospasov, and Jean-Pierre Seifert. 2013. Cloning physically unclonable functions. In *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*. IEEE, 1–6.

[42] Clemens Helfmeier, Dmitry Nedospasov, Christopher Tarnovsky, Jan Starbug Krissler, Christian Boit, and Jean-Pierre Seifert. 2013. Breaking and entering through the silicon. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 733–744.

[43] Christoph Herbst, Elisabeth Oswald, and Stefan Mangard. 2006. An AES smart card implementation resistant to power analysis attacks. In *International conference on applied cryptography and network security*. Springer, 239–252.

[44] Benjamin Hettwer. 2021. *Deep learning-enhanced side-channel analysis of cryptographic implementations.* Ph.D. Dissertation. Ruhr University Bochum, Germany.

[45] Benjamin Hettwer, Kallyan Das, Sebastien Leger, Stefan Gehrer, and Tim Güneysu. 2020. Lightweight side-channel protection using dynamic clock randomization. In *2020 30th International Conference on Field-Programmable Logic and Applications (FPL)*. IEEE, 200–207.

[46] Benjamin Hettwer, Stefan Gehrer, and Tim Güneysu. 2020. Applications of machine learning techniques in side-channel attacks: a survey. *Journal of Cryptographic Engineering* 10, 2 (2020), 135–162.

[47] Benjamin Hettwer, Johannes Petersen, Stefan Gehrer, Heike Neumann, and Tim Güneysu. 2019. Securing cryptographic circuits by exploiting implementation diversity and partial reconfiguration on FPGAs. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 260–263.

[48] Annelie Heuser and Michael Zohner. 2012. Intelligent Machine Homicide. In *Intrl. Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 249–264.

[49] Gabriel Hospodar, Benedikt Gierlichs, Elke De Mulder, Ingrid Verbauwhede, and Joos Vandewalle. 2011. Machine learning in side-channel analysis: a first study. *Journal of Cryptographic Engineering* 1, 4 (2011), 293.

[50] Darshana Jayasinghe, Aleksandar Ignjatovic, and Sri Parameswaran. 2019. RFTC: Runtime frequency tuning countermeasure using FPGA dynamic reconfiguration to mitigate power analysis attacks. In *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.

[51] Anju P Johnson, Rajat Subhra Chakraborty, and Debdeep Mukhopadyay. 2016. An improved DCM-based tunable true random number generator for Xilinx FPGA. *IEEE Transactions on Circuits and Systems II: Express Briefs* 64, 4 (2016), 452–456.

[52] Najeh Kamoun, Lilian Bossuet, and Adel Ghazel. 2009. Correlated power noise generator as a low cost DPA countermeasures to secure hardware AES cipher. In *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*. IEEE, 1–6.

[53] Jens-Peter Kaps and Rajesh Velegalati. 2010. DPA resistant AES on FPGA using partial DDL. In *2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*. IEEE, 273–280.

[54] Nadir Khan, Benjamin Hettwer, and Jürgen Becker. 2021. Moving Target and Implementation Diversity Based Countermeasures Against Side-Channel Attacks. In *International Symposium on Applied Reconfigurable Computing*. Springer, 188–202.

[55] Jaehun Kim, Stjepan Picek, Annelie Heuser, Shivam Bhasin, and Alan Hanjalic. 2019. Make Some Noise. Unleashing the Power of Convolutional Neural Networks for Profiled Side-channel Analysis. *IACR Trans. on Cryptographic Hardware and Embedded Systems* (2019), 148–179.

[56] Paul Kocher, Joshua Jaffe, and Benjamin Jun. 1999. Differential power analysis. In *Annual international cryptology conference*. Springer, 388–397.

[57] Paul C Kocher. 1996. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Annual International Cryptology Conference*. Springer, 104–113.

[58] Thilo Krachenfels, Fatemeh Ganji, Amir Moradi, Shahin Tajik, and Jean-Pierre Seifert. 2021. Real-world snapshots vs. theory: Questioning the t-probing security model. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1955–1971.

[59] Thilo Krachenfels, Tuba Kiyan, Shahin Tajik, and Jean-Pierre Seifert. 2021. Automatic Extraction of Secrets from the Transistor Jungle using Laser-Assisted Side-Channel Attacks. In *30th USENIX Security Symposium (USENIX Security 21)*. 627–644.

[60] Jonas Krautter, Dennis Gnad, and Mehdi Tahoori. 2020. CPAmap: on the complexity of secure FPGA virtualization, multi-tenancy, and physical design. *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2020), 121–146.

[61] Jonas Krautter, Dennis RE Gnad, Falk Schellenberg, Amir Moradi, and Mehdi B Tahoori. 2019. Active fences against voltage-based side channels in multi-tenant FPGAs. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 1–8.

[62] Liran Lerman, Romain Poussier, Gianluca Bontempi, Olivier Markowitch, and François-Xavier Standaert. 2015. Template Attacks vs. Machine Learning Revisited (and the Curse of Dimensionality in Side-Channel Analysis). In *Intrl. Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 20–33.

[63] Itamar Levi, Davide Bellizia, David Bol, and François-Xavier Standaert. 2020. Ask less, get more: Side-channel signal hiding, revisited. *IEEE Transactions on Circuits and Systems I: Regular Papers* 67, 12 (2020), 4904–4917.

[64] Po-Chun Liu, Hsie-Chia Chang, and Chen-Yi Lee. 2010. A low overhead DPA countermeasure circuit based on ring oscillators. *IEEE Transactions on Circuits and Systems II: Express Briefs* 57, 7 (2010), 546–550.

[65] Heiko Lohrke, Shahin Tajik, Christian Boit, and Jean-Pierre Seifert. 2016. No place to hide: Contactless probing of secret data on FPGAs. In *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 147–167.

[66] Heiko Lohrke, Shahin Tajik, Thilo Krachenfels, Christian Boit, and Jean-Pierre Seifert. 2018. Key extraction using thermal laser stimulation: A case study on xilinx ultrascale fpgas. *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2018), 573–595.

[67] Victor Lomné, Philippe Maurine, Lionel Torres, Michel Robert, Rafael Soares, and Ney Calazans. 2009. Evaluation on FPGA of triple rail logic robustness against DPA and DEMA. In *2009 Design, Automation & Test in Europe Conference & Exhibition*. IEEE, 634–639.

[68] Yingxi Lu, Maire P O'Neill, and John V McCanny. 2008. FPGA implementation and analysis of random delay insertion countermeasure against DPA. In *2008 International Conference on Field-Programmable Technology*. IEEE, 201–208.

[69] Roel Maes. 2013. *Physically Unclonable Functions: Constructions, Properties and Applications*. Springer Berlin Heidelberg.

[70] Nele Mentens. 2017. Hiding side-channel leakage through hardware randomization: A comprehensive overview. In *2017 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*. IEEE, 269–272.

[71] Nele Mentens, Benedikt Gierlichs, and Ingrid Verbauwhede. 2008. Power and fault analysis resistance in hardware through dynamic reconfiguration. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 346–362.

[72] Amir Moradi and Vincent Immler. 2014. Early propagation and imbalanced routing, how to diminish in FPGAs. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 598–615.

[73] Amir Moradi and Alexander Wild. 2015. Assessment of hiding the higher-order leakages in hardware. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 453–474.

[74] Maxime Nassar, Shivam Bhasin, Jean-Luc Danger, Guillaume Duc, and Sylvain Guilley. 2010. BCDL: A high speed balanced DPL for FPGA with global precharge and no early evaluation. In *2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010)*. IEEE, 849–854.

[75] Hamed Okhravi, Thomas Hobson, David Bigelow, and William Streilein. 2013. Finding focus in the blur of moving-target techniques. *IEEE Security & Privacy* 12, 2 (2013), 16–26.

[76] Dag Arne Osvik, Adi Shamir, and Eran Tromer. 2006. Cache attacks and countermeasures: the case of AES. In *Cryptographers' track at the RSA conference*. Springer, 1–20.

[77] Bryan Parno, Jonathan M McCune, and Adrian Perrig. 2011. *Bootstrapping trust in modern computers*. Springer Science & Business Media.

[78] Satwik Patnaik, Nikhil Rangarajan, Johann Knechtel, Ozgur Sinanoglu, and Shaloo Rakheja. 2018. Advancing hardware security using polymorphic and stochastic spin-hall effect devices. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 97–102.

[79] Jeffrey Pawlick, Edward Colbert, and Quanyan Zhu. 2019. A Game-Theoretic Taxonomy and Survey of Defensive Deception for Cybersecurity and Privacy. *ACM Comput. Surv.* 52, 4, Article 82 (aug 2019), 28 pages. https://doi.org/10.1145/3337772

[80] Stjepan Picek, Annelie Heuser, Alan Jovic, Shivam Bhasin, and Francesco Regazzoni. 2019. The Curse of Class Imbalance and Conflicting Metrics with Machine Learning for Side-channel Evaluations. *IACR Trans. on Cryptographic Hardware and Embedded Systems* 2019, 1 (2019), 1–29.

[81] Stjepan Picek, Guilherme Perin, Luca Mariot, Lichao Wu, and Lejla Batina. 2021. SoK: Deep Learning-based Physical Side-channel Analysis. *IACR Cryptol. ePrint Arch., 2021/1092* (2021).

[82] Thomas Popp, Mario Kirschbaum, Thomas Zefferer, and Stefan Mangard. 2007. Evaluation of the masked logic style MDPL on a prototype chip. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 81–94.

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

[84] Romain Poussier, Qian Guo, François-Xavier Standaert, Claude Carlet, and Sylvain Guilley. 2017. Connecting and improving direct sum masking and inner product masking. In *International Conference on Smart Card Research and Advanced Applications*. Springer, 123–141.

[85] Emmanuel Prouff and Thomas Roche. 2011. Higher-order glitches free implementation of the AES using secure multi-party computation protocols. In *Intrl. Workshop on Cryptographic Hardware and Embedded Systems*. Springer, Berlin, 63–78.

[86] M Tanjidur Rahman, Shahin Tajik, M Sazadur Rahman, Mark Tehranipoor, and Navid Asadizanjani. 2020. The key is left under the mat: On the inappropriate security assumption of logic locking schemes. In *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 262–272.

[87] Mark Randolph and William Diehl. 2020. Power side-channel attack analysis: A review of 20 years of study for the layman. *Cryptography* 4, 2 (2020), 15.

[88] Nikhil Rangarajan, Satwik Patnaik, Johann Knechtel, Ramesh Karri, Ozgur Sinanoglu, and Shaloo Rakheja. 2020. Opening the doors to dynamic camouflaging: Harnessing the power of polymorphic devices. *IEEE Transactions on Emerging Topics in Computing* (2020).

[89] Prasanna Ravi, Shivam Bhasin, Jakub Breier, and Anupam Chattopadhyay. 2018. Ppap and ippap: Pll-based protection against physical attacks. In *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 620–625.

[90] Unai Rioja, Servio Paguada, Lejla Batina, and Igor Armendariz. 2021. The uncertainty of side-channel analysis: A way to leverage from heuristics. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 17, 3 (2021), 1–27.

[91] Matthieu Rivain, Emmanuel Prouff, and Julien Doget. 2009. Higher-order masking and shuffling for software implementations of block ciphers. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 171–188.

[92] J Robertson and M Riley. 2018. The Big Hack: How China Used a Tiny Chip to Infiltrate US Companies. Bloomberg. (2018).

[93] Durga Prasad Sahoo, Debdeep Mukhopadhyay, Rajat Subhra Chakraborty, and Phuong Ha Nguyen. 2017. A multiplexer-based arbiter PUF composition with enhanced reliability and security. *IEEE Trans. Comput.* 67, 3 (2017), 403–417.

[94] Pascal Sasdrich, Amir Moradi, and Tim Güneysu. 2017. Hiding higher-order side-channel leakage. In *Cryptographers' Track at the RSA Conference*. Springer, 131–146.

[95] Pascal Sasdrich, Amir Moradi, Oliver Mischke, and Tim Güneysu. 2015. Achieving side-channel protection with dynamic logic reconfiguration on modern

FPGAs. In *2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 130–136.

[96] Laurent Sauvage, Maxime Nassar, Sylvain Guilley, Florent Flament, Jean-Luc Danger, and Yves Mathieu. 2009. DPL on stratix II FPGA: What to expect?. In *2009 International Conference on Reconfigurable Computing and FPGAs*. IEEE, 243–248.

[97] Falk Schellenberg, Dennis RE Gnad, Amir Moradi, and Mehdi B Tahoori. 2021. An inside job: Remote power analysis attacks on FPGAs. *IEEE Design & Test* 38, 3 (2021), 58–66.

[98] Oliver Schimmel, Paul Duplys, Eberhard Boehl, Jan Hayek, Robert Bosch, and Wolfgang Rosenstiel. 2010. Correlation power analysis in frequency domain. In *COSADE 2010 First International Workshop on Constructive SideChannel Analysis and Secure Design*.

[99] Werner Schindler, Kerstin Lemke, and Christof Paar. 2005. A Stochastic Model for Differential Side Channel Cryptanalysis. In *Intrl. Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 30–46.

[100] Qihang Shi, Navid Asadizanjani, Domenic Forte, and Mark M Tehranipoor. 2016. A layout-driven framework to assess vulnerability of ICs to microprobing attacks. In *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 155–160.

[101] Alexander Spenke, Ralph Breithaupt, and Rainer Plaga. 2016. An arbiter PUF secured by remote random reconfigurations of an FPGA. In *International Conference on Trust and Trustworthy Computing*. Springer, 140–158.

[102] Michael Steil. 2005. 17 mistakes Microsoft made in the Xbox security system. In *22nd Chaos Communication Congr.*

[103] Mario Stipčević and Çetin Kaya Koç. 2014. True random number generators. In *Open Problems in Mathematics and Computational Science*. Springer, 275–315.

[104] Adrian Stoica, RS Zebulum, Xin Guo, Didier Keymeulen, MI Ferguson, and Vu Duong. 2004. Taking evolutionary circuit design from experimentation to implementation: Some useful techniques and a silicon demonstration. *IEE Proceedings-Computers and Digital Techniques* 151, 4 (2004), 295–300.

[105] Adrian Stoica, Ricardo Zebulum, and Didier Keymeulen. 2001. Polymorphic Electronics. In *ICES2001 4th International Conference on Evolvable Systems: From Biology to Hardware*. 291–301. https://doi.org/10.1007/3-540-45443-8_26

[106] Daisuke Suzuki and Minoru Saeki. 2006. Security evaluation of DPA countermeasures using dual-rail pre-charge logic style. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 255–269.

[107] Daisuke Suzuki, Minoru Saeki, and Tetsuya Ichikawa. 2004. Random switching logic: A countermeasure against DPA based on transition probability. *Cryptology ePrint Archive* (2004).

[108] Daisuke Suzuki, Minoru Saeki, and Tetsuya Ichikawa. 2005. DPA leakage models for CMOS logic circuits. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 366–382.

[109] Shahin Tajik, Enrico Dietz, Sven Frohmann, Jean-Pierre Seifert, Dmitry Nedospasov, Clemens Helfmeier, Christian Boit, and Helmar Dittrich. 2014. Physical characterization of arbiter PUFs. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 493–509.

[110] Shahin Tajik, Heiko Lohrke, Fatemeh Ganji, Jean-Pierre Seifert, and Christian Boit. 2015. Laser fault attack on physically unclonable functions. In *2015 workshop on fault diagnosis and tolerance in cryptography (FDTC)*. IEEE, 85–96.

[111] Shahin Tajik, Heiko Lohrke, Jean-Pierre Seifert, and Christian Boit. 2017. On the power of optical contactless probing: Attacking bitstream encryption of FPGAs. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1661–1674.

[112] Shahin Tajik, Dmitry Nedospasov, Clemens Helfmeier, Jean-Pierre Seifert, and Christian Boit. 2014. Emission analysis of hardware implementations. In *2014 17th Euromicro Conference on Digital System Design*. IEEE, 528–534.

[113] Mohammad Tehranipoor and Farinaz Koushanfar. 2010. A survey of hardware trojan taxonomy and detection. *IEEE design & test of computers* 27, 1 (2010), 10–25.

[114] Mark (Mohammad) Tehranipoor, Ujjwal Guin, and Domenic Forte. 2015. *Counterfeit Integrated Circuits: Detection and Avoidance*. Springer International Publishing.

[115] Kris Tiri, Moonmoon Akmal, and Ingrid Verbauwhede. 2002. A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards. In *Proceedings of the 28th European solid-state circuits conference*. IEEE, 403–406.

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

[117] Randy Torrance and Dick James. 2009. The state-of-the-art in IC reverse engineering. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 363–381.

[118] Jasper GJ van Woudenberg, Marc F Witteman, and Bram Bakker. 2011. Improving differential power analysis by elastic alignment. In *Cryptographers' Track at the RSA Conference*. Springer, 104–119.

[119] Michal Varchola and Milos Drutarovsky. 2010. New high entropy element for FPGA based true random number generators. In *International workshop on cryptographic hardware and embedded systems*. Springer, 351–365.

[120] Ingrid Verbauwhede and Patrick Schaumont. 2007. Design methods for security and trust. In *2007 Design, Automation & Test in Europe Conference & Exhibition*. IEEE, 1–6.

[121] Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. 2012. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 740–757.

[122] Mladen A. Vouk, Annie I. Antón, and Jim Yuill. 2006. Defensive computer-security deception operations: processes, principles and techniques.

[123] Cliff Wang and Zhuo Lu. 2018. Cyber Deception: Overview and the Road Ahead. *IEEE Security & Privacy* 16, 2 (2018), 80–85. https://doi.org/10.1109/MSP.2018.1870866

[124] Huanyu Wang, Domenic Forte, Mark M Tehranipoor, and Qihang Shi. 2017. Probing attacks on integrated circuits: Challenges and research opportunities. *IEEE Design & Test* 34, 5 (2017), 63–71.

[125] Tian Wang, Xiaoxin Cui, Dunshan Yu, Omid Aramoon, Timothy Dunlap, Gang Qu, and Xiaole Cui. 2018. Polymorphic gate based IC watermarking techniques. In *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 90–96.

[126] Bryan C Ward, Steven R Gomez, Richard Skowyra, David Bigelow, Jason Martin, James Landry, and Hamed Okhravi. 2018. *Survey of cyber moving targets second edition*. Technical Report. MIT Lincoln Laboratory Lexington United States.

[127] Alexander Wild, Amir Moradi, and Tim Güneysu. 2017. GliFreD: Glitch-free duplication towards power-equalized circuits on FPGAs. *IEEE Trans. Comput.* 67, 3 (2017), 375–387.

[128] Jeanette Wing, Douglas Maughan, Patricia Muoio, and Carl Landwehr. 2010. Toward A Federal Cybersecurity Research Agenda: Three Game-Changing Themes. [Online] https://www.nitrd.gov/documents/cybersecurity/documents/NITRDCybersecurityRDThemes20100519.pdf [Accessed: Sept.9, 2022]. (2010).

[129] Lichao Wu, Guilherme Perin, and Stjepan Picek. 2020. I Choose You: Automated Hyperparameter Tuning for Deep Learning-based Side-channel Analysis. *IACR Cryptol. ePrint Arch.* 2020 (2020), 1293.

[130] Yuan Yao, Pantea Kiaei, Richa Singh, Shahin Tajik, and Patrick Schaumont. 2021. Programmable ro (pro): A multipurpose countermeasure against side-channel and fault injection attack. *arXiv preprint arXiv:2106.13784* (2021).

[131] Yuan Yao, Mo Yang, Conor Patrick, Bilgiday Yuce, and Patrick Schaumont. 2018. Fault-assisted side-channel analysis of masked implementations. In *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 57–64.

[132] Sang-Kyung Yoo, Deniz Karakoyunlu, Berk Birand, and Berk Sunar. 2010. Improving the robustness of ring oscillator TRNGs. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* 3, 2 (2010), 1–30.

[133] Pengyuan Yu and Patrick Schaumont. 2007. Secure FPGA circuits using controlled placement and routing. In *Proceedings of the 5th IEEE/ACM international conference on Hardware/software codesign and system synthesis*. 45–50.

[134] Xiao Mei Zeng, Qing Liu, Jing Yun Tay, and Chee Lip Gan. 2022. Selective Staining on Non-Volatile Memory Cells for Data Retrieval. *IEEE Transactions on Information Forensics and Security* 17 (2022), 1884–1892.

[135] Zhiming Zhang, Laurent Njilla, Charles Kamhoua, Kevin Kwiat, and Qiaoyan Yu. 2018. Securing FPGA-based obsolete component replacement for legacy systems. In *2018 19th International Symposium on Quality Electronic Design (ISQED)*. 401–406. https://doi.org/10.1109/ISQED.2018.8357320

[136] Zhiming Zhang, Qiaoyan Yu, Laurent Njilla, and Charles Kamhoua. 2018. FPGA-oriented moving target defense against security threats from malicious FPGA tools. In *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. 163–166. https://doi.org/10.1109/HST.2018.8383907