Application-Specific Power Side-Channel Attacks and Countermeasures: A Survey

Sahan Sanjaya, *Student Member, IEEE,* Aruna Jayasena, *Member, IEEE,* and Prabhat Mishra, *Fellow, IEEE*

***Abstract*—Side-channel attacks try to extract secret information from a system by analyzing different side-channel signatures, such as power consumption, electromagnetic emanation, thermal dissipation, acoustics, time, etc. Power-based side-channel attack is one of the most prominent side-channel attacks in cybersecurity, which rely on data-dependent power variations in a system to extract sensitive information. While there are related surveys, they primarily focus on power side-channel attacks on cryptographic implementations. In recent years, power side-channel attacks have been explored in diverse application domains, including key extraction from cryptographic implementations, reverse engineering of machine learning models, user behavior data exploitation, and instruction-level disassembly. In this paper, we provide a comprehensive survey of power side-channel attacks and their countermeasures in different application domains. Specifically, this survey aims to classify recent power side-channel attacks and provide a comprehensive comparison based on application-specific considerations.**

***Index Terms*—Power Side-Channel, Cryptographic Implementations, Machine Learning, User Behavior Data, Disassembler, Survey, Side-Channel Attacks, Hardware Security, Countermeasures**

I. Introduction

With the rapid advancement of technology, electronic devices have become an integral part of everyday life. From simple personal devices, like smartphones and wearables, to complex systems in critical infrastructure, these devices process and store a significant amount of sensitive information. As a result, the security of electronic devices is of critical importance, specially in the face of an increasingly sophisticated threat landscape. While traditional cryptographic measures are designed to protect data during transmission and storage, they often overlook the potential risks posed by physical side channels, which can leak information unintentionally through observable side-channels, such as timing, electromagnetic emanation, power consumption, acoustics, etc.

Power side-channel (PSC) attacks have gained significant attention due to their ability to exploit power consumption patterns to retrieve sensitive information from different application implementations. As illustrated by Figure 1, PSC attack represents the most researched side-channel attack type between 2015 and 2020, accounting for 36.5% of the published work. This highlights the prominent interest among researchers in exploiting vulnerabilities related to power consumption in electronic devices. Since the introduction of Differential Power Analysis (DPA) by Kocher et al. [1], PSC attacks have become a major concern, particularly with the expanding use of cryptographic algorithms in consumer devices. Figure 2 illustrates a specific attack scenario utilizing correlation power analysis (CPA) on AES cryptographic implementation running on a CPU-based platform. The attack flow involves the collection of power traces, statistical analysis, and correlation with hypothesized power traces to successfully recover the secret key. Also, designers integrate countermeasure methods into the both platform and application to prevent future attacks.

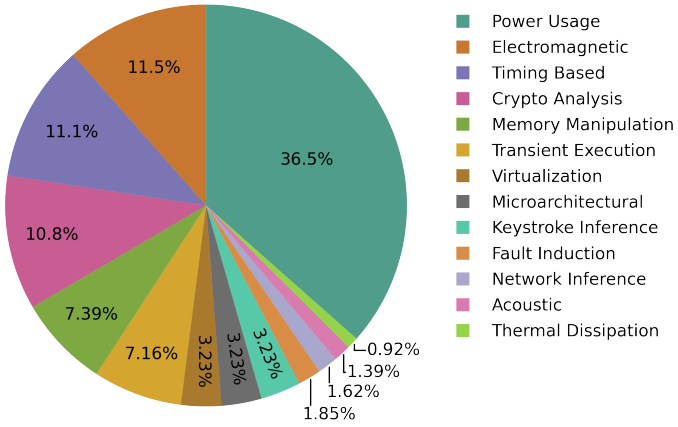

Fig. 1. Most common side-channel attack types based on published results [2]. Power side-channel represents the most researched attack in the literature.

Recent works demonstrate that PSC attacks are successful in extracting a wide variety of information, including secret keys from cryptographic implementations, confidential inputs used in machine-learning models, proprietary machine-learning model architectures, user behavior data, and proprietary instruction-level code. In this paper, we present a survey on application-specific PSC attacks and their countermeasures.

A. Major Differences with Existing Surveys

There are several promising surveys related to PSC attacks [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. However, these surveys are either outdated or limited in scope. For example, several surveys do not consider research efforts from the last decade [3, 4, 5]. Most of the recent surveys focus exclusively on PSC attacks targeting cryptographic implementations [4, 6, 7, 8, 9, 10, 11, 12, 13]. While there are recent surveys that cover side-channel attacks for the extraction of neural network architecture [24, 25] and code execution monitoring [26], they are not focused on power side-channel attacks.

There are also surveys focused on the target implementation platform, including Internet of Things (IoT) devices [6, 14],

S. Sanjaya and P. Mishra are with the Department of Computer & Information Science & Engineering, University of Florida, USA. Email: {ssanjaya, prabhat}@ufl.edu. A. Jayasena is with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, USA. Email: arunaj@umass.edu

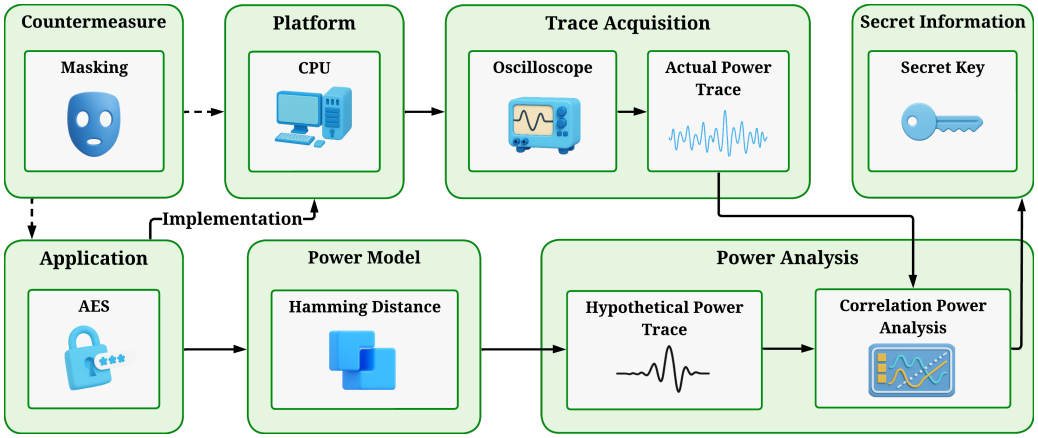

Fig. 2. Correlation power analysis (CPA) attack flow, illustrating the targeted cryptographic algorithm (AES), the implementation platform (CPU) for power trace collection, the trace acquisition method (oscilloscope), the power model (Hamming distance) to develop hypothetical traces, and the statistical analysis method (CPA) to correlate actual and hypothetical traces to extract the secret key. The figure also highlights the use of countermeasures to mitigate attacks.

field-programmable gate arrays (FPGA)[7, 15], graphical processing units (GPU)[16], mobile devices [17], and intelligent sensors [18]. There are recent surveys exclusively focused on power analysis techniques, such as machine learning [10, 19, 20], deep learning [21, 22, 23], and profiled methods [7]. However, these surveys are focused on either hardware platforms or power analysis methods. As a result, they do not capture the interactions between a wide variety of applications, diverse trace collection interfaces, different power models, various hardware platforms, and diverse analysis methods. Specifically, this survey makes the following key contributions:

- **Multi-dimensional taxonomy:** We propose a novel six-axis taxonomy that classifies power side-channel (PSC) research based on (i) target application, (ii) implementation platform, (iii) probing interface, (iv) power model, (v) analysis method, and (vi) countermeasure granularity. This taxonomy provides a structured view of the research landscape that unifies previously fragmented literature.
- **Cross-domain coverage:** Unlike prior surveys that focus on a single application domain (e.g., cryptography), we systematically analyze PSC attacks across a wide variety of applications, including machine learning model extraction, user-behavior monitoring, and instruction disassembly, offering the most comprehensive review across various target applications.
- **Platform-specific insights:** We categorize attacks across a diverse range of targeted platforms, including general-purpose processors (e.g., CPUs), reconfigurable hardware (e.g., FPGAs, soft cores), application-specific hardware (e.g., ASICs, GPUs, TPUs), embedded systems (e.g., MCUs, IoT devices), secure hardware platforms (e.g., SGX), and other peripherals (e.g., USB controllers, printers). Our classification highlights platform-specific leakage sources and challenges associated with each platform.
- **Discussion on power trace collection interfaces:** This category encompasses both conventional physical interfaces used in local or remote hardware-based attacks and emerging software-based interfaces such as Intel's RAPL that exploit built-in performance monitoring frameworks. These interfaces facilitate the acquisition of power traces from the targeted platforms.
- **Comparative analysis of power models and analysis methods:** We examine how different power consumption models (e.g., Hamming weight, switching distance) and analysis methods (e.g., template attacks, correlation power analysis) are tailored to different applications.
- **Countermeasure classification and evaluation:** We classify countermeasures based on their abstraction levels, such as algorithmic, logic, and circuit, and evaluate their applicability across domains.
- **Literature coverage mapping:** We visualize the density of prior research across the taxonomy dimensions to expose underexplored areas and identify promising directions for future work.

By integrating these contributions into a single coherent framework, this survey not only bridges gaps in existing literature but also serves as a practical reference for researchers and practitioners seeking to understand, evaluate, or develop PSC attacks and defenses across multiple domains.

B. Survey Outline

This survey seeks to answer the following three fundamental questions:

- How are power traces collected from various *target applications* implemented on different *target platforms* using diverse *probing interfaces*? This question addresses three key categorization axes: **Target Application**, **Target Platform**, and **Probing Interface**.
- What are the side-channel *analysis methods* used to extract sensitive information from the target application using the collected power traces and specific *power models*, which are derived based on the characteristics of the application and platform? This question corresponds to the **Power Model** and **Analysis Method** axes.
- How can we defend against these attacks with *countermeasures*, the final categorization axis: **Countermeasure**.

Each axis reflects a distinct property relevant to power side-channel vulnerability and analysis:

- **Target Application** denotes the computation being attacked, ranging from cryptographic key extraction to neural network inference or instruction monitoring.

- **Target Platform** denotes the execution environment, such as general-purpose processors (e.g., CPUs), reconfigurable hardware (e.g., FPGAs, soft cores), application-specific hardware (e.g., ASICs, GPUs, TPUs), embedded systems (e.g., MCUs, IoT devices), secure hardware platforms (e.g., SGX), and other peripherals (e.g., USB controllers, printers). Each platform reflects distinct leakage characteristics and constraints on observability.
- **Probing Interface** captures the adversary's access model, whether leakage is obtained via physical probing (e.g., oscilloscopes), software-based (e.g., utilizing software interface to observe hardware performance counters), or remote channels that convert power variations into indirectly observable metrics (e.g., timing differences). The choice of interface significantly influences the attacker's capabilities and the required setup complexity.
- **Power Model** defines the abstraction that relates internal transitions to observable power, such as Hamming weight or switching activity.
- **Analysis Method** categorizes the effort to interpret the trace, distinguishing profiled and non-profiled analysis.
- **Countermeasure** captures the abstraction level at which defenses are applied, such as algorithmic (e.g., masking), logic-level (e.g., dual-rail encoding), or circuit-level (e.g., noise injection).

The remainder of this section details each axis and explains how the subsequent sections of the survey map onto the taxonomy as follows. Section II surveys various leakage models, including Hamming weight, Hamming distance, switching distance, and custom models. Section III surveys both profiled and non-profiled PSC analysis methods. Next, we survey various side-channel attacks based on the target applications, including cryptographic implementations (Section IV), user behavior data (Section V), assembly instructions (Section VI), and machine learning models (Section VII). We survey application-specific countermeasures in Section VIII. Finally, Section IX concludes this survey.

II. Power Side-Channel Models

In order to perform a PSC attack, it is essential to understand the internal operations of the hardware. Since almost all attacks target devices using semiconductor components, it is necessary to discuss the power consumption model in semiconductor devices. Complementary Metal-Oxide-Semiconductor (CMOS) technology is the predominant technology for constructing application-specific integrated circuits (ASICs), including central processing units (CPUs), field programmable gate arrays (FPGAs), graphics processing units (GPUs), memory chips, and other digital logic circuits. CMOS is widely used due to low power, high noise immunity, and high logic density.

CMOS power consumption can be primarily divided into dynamic and static components. Dynamic power consumption occurs when the circuit transitions between different states, specifically when transistors switch from off (logic 0) to on (logic 1) or vice versa. This form of power consumption can be expressed by Equation 1. Where α represents the activity factor (indicating the fraction of time the transistors are switching), C_L denotes the load capacitance, V_{DD} is the supply voltage, and f is the switching frequency.

$$P_{\text{dynamic}} = \alpha C_L V_{DD}^2 f \tag{1}$$

On the other hand, static power consumption arises from leakage currents even when transistors are not switching. This leakage power illustrated by Equation 2 has become more significant in advanced technology nodes. In Equation 2, I_{leakage} represents the leakage current.

$$P_{\text{static}} = I_{\text{leakage}} V_{DD} \tag{2}$$

In the earlier days of CMOS technology, dynamic power consumption was the dominant component because the switching activity and load capacitance significantly contributed to the overall power usage. However, as technology has scaled down to nanometer dimensions, the leakage currents have increased due to thinner gate oxides and higher electric fields, making static power consumption comparable to dynamic power consumption in many domains, such as embedded devices with large memory. Additionally, with the increase in the number of transistors per chip, the aggregate leakage current has become substantial, elevating the importance of addressing static power in modern CMOS designs.

In order to perform a successful PSC attack, the discussed parameters should be computed from the device under attack. The computation of these parameters varies depending on whether the attack is performed on pre-silicon designs or post-silicon (real hardware) implementations.

During pre-silicon information leakage assessment, these values are estimated from simulation results [27, 28, 29]. The pre-silicon designs are simulated with appropriate inputs, obtaining value change dump (VCD) files that capture the behavior of internal signals. By analyzing signal transitions across various power model combinations, the expected power signature of the design is extracted, allowing for an early evaluation of potential PSC leakages before the device's physical implementation. These estimation models are particularly useful for emerging technologies, such as negative capacitance field-effect transistors (NCFETs), which aim to overcome limitations in traditional MOSFETs. Although NCFETs are not yet at the production level, their security applicability can be evaluated using pre-silicon leakage analysis methods [30]. While pre-silicon analysis enables early detection of potential leakages, it relies on simulated switching activity and abstract power models that often overlook real-world effects such as parasitics, IR drop, and process variation. Consequently, its accuracy is generally lower than post-silicon analysis, which captures actual physical leakage but requires fabricated hardware and measurement effort.

For post-silicon power side-channel attacks, the researchers utilize abstract versions of power models that are easier to model, easier to compute, and still provide accurate results [31]. Before the actual power side-channel analysis, the expected power consumption is computed based on these power models. Unlike real power traces, the abstract versions of the power models are computationally efficient. These abstract models have been used to generate the hypothetical power traces [32, 33, 34, 35, 36] to perform statistical analysis

with actual power traces. However, the use of abstract power models may fail to capture subtle fine-grained microarchitectural behaviors, introducing trade-offs between analysis simplicity and modeling accuracy. While, post-silicon analysis benefits from high fidelity due to real power measurements, it requires access to fabricated hardware, precise instrumentation, and careful noise handling, which can be costly and time-consuming.

While this section discusses the power models used in PSC attacks, Section III explains how these power models are utilized in analysis methods. Figure 3 shows the four popular power models for side-channel analysis in the literature. Among these, Hamming weight, Hamming distance, and custom combinations of these two models are widely used.

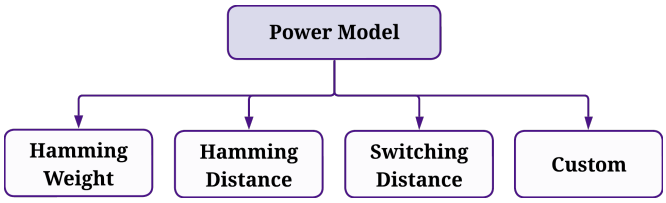

Fig. 3. Overview of various power models used for power side-channel analysis, including Hamming weight, Hamming distance, switching distance, and custom combinations and models.

A. Hamming Weight

The Hamming weight (HWT) represents the number of "1" bits in a binary number [37]. This metric is directly related to the power consumption of the device, as each "1" bit contributes to the static power dissipation [38]. The formula to compute Hamming weight is illustrated by Equation 3, where D_i is the i-th bit of the data byte D.

$$HWT(D) = \sum_{i=0}^{n-1} D_i \tag{3}$$

For example, the Hamming weight of the binary number 10110010 is 4. This model has been successfully applied in attacks on ASIC, CPU, and FPGA implementations of CMOS devices [39, 40, 41, 42, 43, 44].

B. Hamming Distance

The Hamming distance (HD) measures the number of bit positions at which two binary numbers differ, providing insight into the power consumed during state transitions. This power model was introduced by Brier et al. [45] in a PSC attack. The Hamming distance is calculated using the formula shown in Equation 4, where $D1_i$ and $D2_i$ are the i-th bits of data bytes $D1$ and $D2$, respectively, and $\oplus$ denotes the XOR operation.

$$HD(D1, D2) = \sum_{i=0}^{n-1} (D1_i \oplus D2_i) \tag{4}$$

For instance, the Hamming distance between 10110010 and 11011001 is 5. The Hamming distance model was successfully used by various PSC attacks [33, 34, 46, 47, 48, 49, 50].

C. Switching Distance

Peeters et al. introduced a refined power model based on the fact that transitions from 0 to 1 and from 1 to 0 consume different power [51]. This model aims to provide a more accurate representation of leakage by distinguishing between the power consumed during different state transitions in CMOS gates, compared to the Hamming distance model. The switching distance power consumption model is detailed in Table I, where transitions from 0 to 1 and from 1 to 0 are assigned specific power values to reflect their differing consumption patterns. Here, d is the normalized difference in transition leakages, as defined in Equation 5, where $P_{0\to1}$ is the probability of a 0 to 1 transition, and $P_{1\to0}$ is the probability of a 1 to 0 transition.

$$d = (P_{0\to1} - P_{1\to0}) / P_{0\to1} \tag{5}$$

Peeters et al. [51] and Tran et al. [52] used this model to perform attacks on RISC-based processors, while there were promising efforts of applying this model to perform attacks on ASICs and FPGAs [53, 54, 55].

TABLE I
SWITCHING DISTANCE POWER CONSUMPTION MODEL

Transition	Power
$0 \to 0$	0
$0 \to 1$	1
$1 \to 0$	$1 - d$
$1 \to 1$	0

D. Custom Power Models

Besides the above-mentioned power models, there are also custom power models in the literature based on the target application and implemented hardware. The stochastic power model is such a custom model, which was introduced to enhance the efficiency of differential side-channel analysis against block ciphers by approximating the leakage function within a low-dimensional vector space, thereby avoiding the need for full high-dimensional profiling as required in classical template attacks [56]. The stochastic model learns a compact set of parameters using measurements from only a single profiling key, enabled by the equal-images property that keeps the leakage structure invariant across keys. The model expresses each observed power sample as

$$I_t(x, k) = h_t(x, k) + R_t,$$

where the deterministic component $h_t(x, k)$ captures key- and data-dependent leakage, while the random component R_t represents key- and data-independent noise. For the AES implementation on an 8-bit microcontroller, the deterministic leakage is approximated by a linear combination of S-box output bit functions,

$$\hat{h}_t^*(\phi(x, k)) = b_{0,t} + \sum_{i=1}^{8} b_{i,t}\, g_i(\phi(x, k)), \quad \phi(x, k) = x \oplus k,$$

where each g_i extracts one bit of the S-box output and the coefficients $b_{i,t}$ are obtained via least-squares fitting during

profiling. The noise term is modeled as a multivariate Gaussian with an empirically estimated covariance matrix.

Xiang et al. proposed another custom power model for Deep Neural Networks (DNNs) [57]. The model calculates the power consumption for each layer of the neural network, considering convolutional, pooling, fully connected, and activation layers. Specifically, Equation 6 illustrates the power consumption of a convolutional layer, where C is the number of input channels, L and W are the input dimensions, S is the stride, N is the number of kernels, F is the kernel size, and p_m and p_a are the average power consumptions for multiplication and addition operations, respectively.

$$P_{conv}(C, L, W, S, N, F) = p_m \frac{LWNCF^2}{S^2} + p_a \frac{LWNCF^2}{S^2} \quad (6)$$

The power consumption for the pooling layer is given in Equation 7, where p_c is the average comparison operation power consumption.

$$P_{pl}(C, L, W, S, F) = p_c \frac{CLWF^2}{S^2} \quad (7)$$

For the fully connected layer, the power consumption is modeled as shown in Equation 8, where X and Y are the number of input and output neurons, respectively.

$$P_{fc}(X, Y) = p_m XY + p_a XY \quad (8)$$

Equation 9 illustrates the formula for the activation function, where p_{ac} is the power consumed by one operation in the activation function and α is the operational coefficient determined by the specific type of activation function.

$$P_{ac}(C, L, W) = p_{ac} \alpha CLW \quad (9)$$

To further refine the power model, the authors introduced a Parameter Sparsity Model [57], accounting for the sparsity in the DNN parameters due to pruning techniques. The sparsity is represented by the coefficients λ_1 and λ_2, modifying the power consumption as shown in Equation 10 and Equation 11. The overall power consumption model is thus a combination of these equations, allowing for accurate prediction of the power consumption of DNNs based on their architecture and operational parameters.

$$P'_{\text{conv}}(C, L, W, S, N, F) = \lambda_1 \cdot P_{\text{conv}}(C, L, W, S, N, F) \quad (10)$$

$$P'_{\text{fc}}(X, Y) = \lambda_2 \cdot P_{\text{fc}}(X, Y) \quad (11)$$

Another custom power model was proposed by Wang et al. to perform the 'Hertzbleed' attack [58]. Even though the authors have primarily considered the Hamming weight and Hamming distance models to explain the power consumption of modern Intel and AMD x86 CPUs, they further explored an additive model combining both Hamming weight and Hamming distance as the power model. This approach also highlights the non-linearity of the Hamming weight model, especially concerning the bit position of '1's, where the leakage is more pronounced when there are '1's in the most significant byte than in the least significant byte.

Another custom power model was proposed by Kogler et al. in the Collide+Power attack [59]. The authors emphasized that simple HWT or HD models fail to capture the leakage arising from interactions between attacker-controlled and victim cache line values. To address this, they introduced a composite power model incorporating both the HD and the HWT of the attacker-controlled value (G) and secret victim value (V) of different security domains, formally modeled as:

$$P_{cache} \approx a_0 \cdot \mathsf{hd}(G, V) + w_0 \cdot \mathsf{hwt}(G) + w_1 \cdot \mathsf{hwt}(V) + w, \quad (12)$$

where $\mathsf{hd}(G, V)$ denotes the Hamming distance between G and V, $\mathsf{hwt}(\cdot)$ represents the HWT, and w captures independent noise components [59]. This model reflects the observation that power consumption during cache line replacement is predominantly influenced by bit transitions (captured by HD), but is also affected by the HWT of the involved values.

In summary, power modeling plays a significant role in shaping the effectiveness of PSC attacks. While traditional models such as HWT and HD offer foundational insights into digital switching activity, recent works demonstrate that these models alone are often insufficient to capture the nuanced leakage behaviors of modern computing platforms. Consequently, researchers have proposed hybrid and architecture-specific models that better reflect practical leakage scenarios, including nonlinear effects, bit position dependencies, and inter-value interactions. These evolving models not only enhance attack success rates but also guide the design of more robust countermeasures. In the subsequent section, we delve into the diverse set of analysis techniques that leverage these models to extract sensitive information from power traces.

III. Power Side-Channel Analysis Methods

Figure 4 shows an overview of PSC analysis methods that can be broadly divided into two categories (profiled and non-profiled) based on the method used to extract information from the power traces. Both types of attacks rely on collecting power traces based on the power consumption of the platform during the execution of the target applications. By analyzing these power traces, attackers can infer secret information due to the correlation between the power consumption and the operations being performed or data being used in the platform.

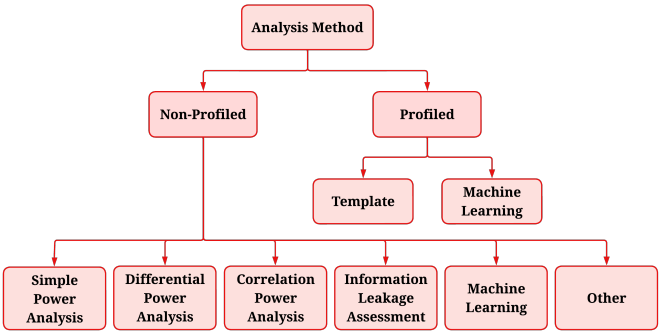

Fig. 4. Power side-channel analysis can be viewed as profiled or non-profiled based on the method used to extract information from the power traces.

As shown in Figure 5, a typical physical power trace acquisition setup involves the following components: a target device (e.g., ASIC, CPU, FPGA, GPU), a power measurement tool (such as an oscilloscope or a dedicated power analysis tool), and a computer to control the device and collect the power traces. The device executes the targeted application while the

power measurement tool records its power consumption. These power traces are then analyzed using statistical or machine learning (ML) techniques to extract the secret information.

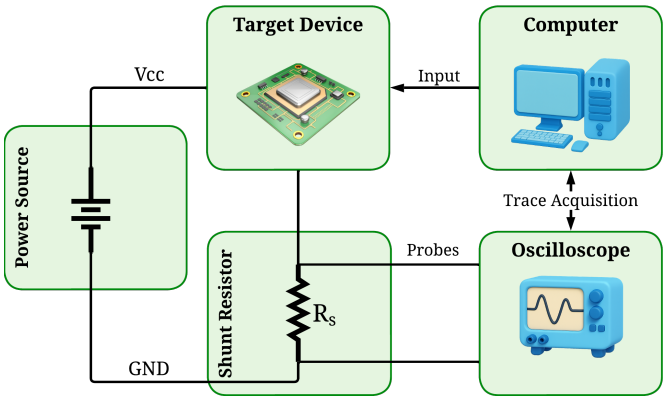

Fig. 5. A typical PSC attack trace acquisition setup includes some mechanism (e.g., oscilloscope) to gather power traces from the device under attack (e.g., a CPU running the target application) as well as a controller to control the device and collect traces for further analysis (e.g., a computer) [60].

The power consumption (P) of the device is given by Equation 13, where V is the supply voltage and I is the current.

$$P = IV \tag{13}$$

Since V is a known value by measuring I we can calculate P. However, to accurately measure P, a shunt resistor is often placed in series with the power supply line of the targeted device. Equation 14 illustrates the voltage drop (V_{drop}) across this resistor R_s.

$$V_{drop} = IR_s \tag{14}$$

Therefore V_{drop} across the R_s is proportional to I. By measuring the voltage drop across the shunt resistor and knowing the resistance, the power consumption can be calculated.

Beyond traditional physical power trace acquisition using oscilloscopes and shunt resistors, recent advancements have enabled PSC attacks via remote and software-accessible interfaces. Remote PSC attacks leverage implementations such as time-to-digital converters (TDCs), which convert power variations into another observable metric, such as time, to infer power consumption without requiring direct physical access to the victim implementation. Software-based PSC attacks, an emerging class of PSC techniques, such as PLATYPUS [50], exploit privileged or publicly accessible hardware performance monitoring frameworks. Notable examples include Intel's Running Average Power Limit (RAPL) interface and NVIDIA's NVML interface, both of which provide energy telemetry at fine granularity. These approaches broaden the threat model by eliminating the need for physical proximity, thereby increasing the practicality and scalability of PSC attacks. The remainder of this section presents two major approaches for PSC analysis that are applied to the power traces collected using the acquisition techniques discussed above. We begin by exploring non-profiled side-channel analysis methods, which operate without prior knowledge of the target's leakage characteristics. Next, we examine profiled side-channel analysis methods, which leverage a trained leakage model obtained from a similar device to improve attack efficiency and accuracy.

A. Non-Profiled Side-channel Analysis Methods

Non-profiled side-channel analysis attacks are a category of power analysis attacks that do not rely on a pre-established profile of the target device's behavior. These attacks focus on exploiting the power consumption patterns of targeted devices during their operation to extract secret information such as cryptographic keys. The procedure typically involves collecting power consumption traces while the device is executing targeted applications, followed by statistical analysis to identify correlations between the power consumption data and the secret information. Common types of non-profiled side-channel analysis include simple power analysis (SPA), differential power analysis (DPA), and correlation power analysis (CPA). SPA involves visually inspecting power traces to identify patterns directly related to the operations performed. DPA uses statistical methods to compare multiple power traces to filter out noise and highlight the secret information. CPA, a more advanced form of DPA, uses correlation coefficients to match the observed power consumption with hypothetical power consumption models.

The primary advantage of non-profiled side-channel analysis attacks is that they do not require prior knowledge or access to the target device, making them applicable to a wide range of devices. In this section, we describe five popular approaches for side-channel analysis: (i) simple power analysis, (ii) differential power analysis, (iii) correlation power analysis, (iv) information leakage assessment, and (v) machine learning.

*1) **Simple Power Analysis**:* In the late 1990s, Kocher et al. introduced simple power analysis (SPA) [1]. SPA involves the direct observation and interpretation of power consumption patterns of a target device while it performs operations using a single or a few inputs on the target application. SPA does not rely on statistical methods but instead focuses on identifying distinct power signatures associated with specific operations. During a SPA attack, the attacker captures the power traces of the target device and visually inspects these traces to identify patterns and distinguish between different operations, such as key loading, encryption, and decryption. If the target application is a cryptographic algorithm, the secret key information can sometimes be inferred directly from these visual patterns, particularly if the device performs cryptographic operations in a manner that leaks information through distinct power variations. For instance, the execution of cryptographic algorithms like DES [1] can exhibit power consumption variations depending on the value of the key bits being processed. By carefully analyzing these variations, an attacker can deduce the key bit by bit.

Existing literature has shown that the standard unsecured hardware implementation of the quantum-secure encryption scheme based on the Binary-Ring-Learning-with-Errors (B-RLWE) problem is particularly vulnerable to SPA due to its use of binary coefficients [32]. During polynomial multiplication, the hardware generates partial products based on the binary values of the secret key bits. If a key bit is '1', the corresponding coefficients are added to the intermediate sum, leading to detectable variations in power consumption. Aysu et al. illustrated that this phenomenon makes it possible to

recover the secret key with a few power measurements by observing these differences in power traces [32].

2) **Differential Power Analysis:** Differential power analysis (DPA), introduced by Kocher et al. [1], is a more sophisticated technique than SPA. In DPA, the attacker collects a large number of power traces while the target device processes different known inputs. DPA focuses on the statistical analysis of differences in power consumption to identify secret keys. During a DPA attack on a cryptographic algorithm, the attacker first makes hypotheses about the key bits and computes the expected intermediate values of the cryptographic algorithm. The attacker then partitions the power traces into different sets based on the hypothetical intermediate values. By comparing the average power consumption of these sets, the attacker identifies the presence of significant differences that correlate with specific key bits.

The first step of a DPA attack is to execute the targeted algorithm n times to capture T_n traces. The output ciphertext is also captured and labeled with C_i corresponding to its ith trace. The selection function $S(C_i, K_m)$ is defined, where K_m is the key guess. The selection function determines whether a specific bit of the intermediate value (derived from the key guess and ciphertext) is 0 or 1, thereby classifying each trace into one of two groups. The differential trace T_Δ for the guess K_m is determined by Equation 15, where $T_\Delta[j]$ is the differential trace at point j. In Equation 15, $T_i[j]$ is the power consumption at time offset j in the i-th trace. The first term in the equation represents the average power consumption when the selection function is 1, and the second term represents the average power consumption when the selection function is 0. The differential trace T_Δ highlights the differences in power consumption that correlate with the correct key guess. The guesses for K_m that produce the largest spikes in the differential trace T_Δ are considered to be the most likely candidates for the correct value.

$$T_\Delta[j] = \frac{\sum_{i=1}^{n} S(C_i, K_m) \cdot T_i[j]}{\sum_{i=1}^{n} S(C_i, K_m)} - \frac{\sum_{i=1}^{n} (1 - S(C_i, K_m)) \cdot T_i[j]}{\sum_{i=1}^{n} (1 - S(C_i, K_m))} \tag{15}$$

An example of a successful DPA attack is demonstrated on a hardware implementation of B-RLWE [32]. By analyzing the differences in power consumption during polynomial multiplication, they were able to recover the secret key. This attack illustrates the effectiveness of DPA in breaking quantum-secure cryptographic implementations.

3) **Correlation Power Analysis:** Brier et al. introduced correlation power analysis (CPA) [45]. In CPA, the attacker collects multiple power traces while the target device processes different known inputs. The attacker then hypothesizes intermediate values of the cryptographic algorithm based on possible key guesses and uses power models to develop hypothetical power traces. At a specific time, the actual power traces are statistically correlated with the hypothetical power traces using Pearson correlation. The Pearson correlation coefficient, r, is calculated by Equation 16, where X_i and Y_i are the actual and hypothetical power trace values, respectively, $\bar{X}$ and $\bar{Y}$ are the means of these values, and n is the number of samples. The key hypothesis that results in the highest correlation with the actual traces is deemed the correct guess.

$$r = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n} (X_i - \bar{X})^2 \sum_{i=1}^{n} (Y_i - \bar{Y})^2}} \tag{16}$$

In recent years, several PSC attacks using CPA have been performed. An application of CPA on an AES implementation running on a state-of-the-art GPU was demonstrated in [33]. The authors successfully extracted the secret key by analyzing the power consumption patterns of the GPU during AES encryption operations. They developed a Hamming distance-based power model targeting registers used in the last round of AES. Benhadjyoussef et al. [34] proposed an attack scenario combining both CPA and differential fault analysis. Here, the last round of an AES-128 encryption implementation on an FPGA was selected as the target.

Measurement to Disclosure (MTD) is an indication of the number of power traces required to extract the secret information. Munny and Hu evaluated the relationship between the attacker's probing effect in the system and MTD, the time to collect the MTD number of traces, and the time for a successful CPA to extract the full key [35]. They used an AES-128 implementation on an XMEGA board and targeted the output of the S-Box while using Hamming distance as the power model in their CPA. The results showed that with a sufficiently high resistive probing effect, the MTD is reduced, and the time for MTD collection and the time for the CPA are almost linear with MTD.

Another systematic evaluation of PSC attacks using CPA on AES implementations was explored by [36]. The authors outlined how to conduct attacks under different restriction levels shown in Figure 6: the most restrictive "red box" with no additional information, the "black box" with output observability, the "gray box" with input repeatability and output observability, the "white box" with input controllability and output observability, and the "golden box" where both the key and input are controllable and the output is observable. A static PSC attack using CPA on an AES-128 implementation in an FPGA environment was proposed in [42]. The S-box outputs in the final round are the targeted intermediate values, and the Hamming weight power model was used to generate hypothetical power trace values. The CPA attack successfully recovered the entire AES key with 1,500 traces.

Mozipo and Acken [61] explored the application of CPA attacks on selected lightweight ciphers (LWC), specifically targeting Ascon, Sparkle, PHOTON-Beetle, GIFT-COFB, Grain, Romulus, and TinyJambu. These algorithms are the seven out of ten finalists from the NIST lightweight cryptography competition, which are designed for resource-constrained IoT devices. The authors detailed the attack methodology, including the targeted intermediate values and the necessary conditions for successful CPA [61]. The findings demonstrated that increasing the number of collected power traces can significantly improve the effectiveness of CPA, thereby highlighting the ongoing risk posed by such attacks even in advanced lightweight cryptographic implementations.

Jayasena et al. proposed a pre-silicon vulnerability evaluation framework for SoC designs using CPA [62]. The authors demonstrated a vulnerability (EvilCS) associated with

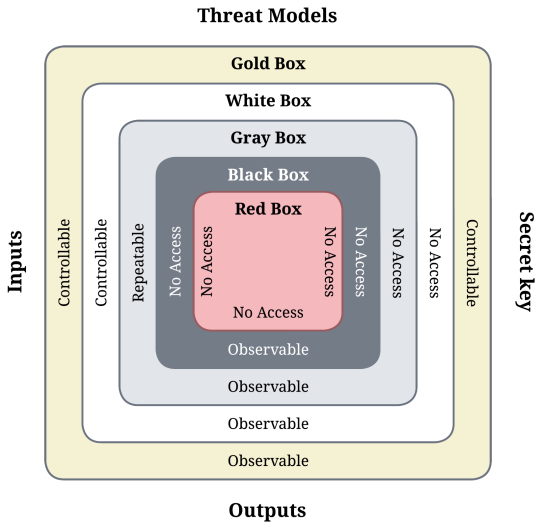

Fig. 6. Different levels of PSC attacks based on model restrictions, ranging from the most flexible Golden Box to the most restrictive Red Box [36].

context switching of the operating system kernel which leaks data associated with user applications. This framework was later extended for pre-silicon CPA analysis of cryptographic instruction set extensions (CISE) [63]. The authors proposed a technique to generate firmware images to activate the functional units implementing the CISE components and used the estimated power model to evaluate the leakage of the cryptographic secrets. The authors demonstrated the cryptographic leakage on RISC-V XCRYPTO extension [63].

Higher-order CPA is one of the promising options for attacking target implementations with countermeasures. Moos et al. demonstrated the use of CPA and third-order CPA to exploit both static and dynamic PSC leakage of the PRESENT-80 cipher implemented on a 150 nm CMOS prototype chip [43]. The third-order CPA, collision-based Moments Correlating DPA (MCDPA), is used to attack the PRESENT-80 block cipher with a first-order countermeasure threshold masking.

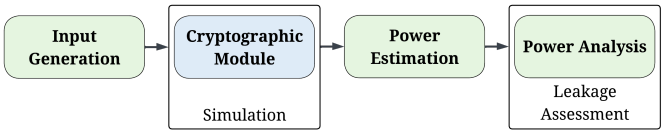

Fig. 7. Overview of test vector leakage assessment (TVLA) framework.

4) **Information Leakage Assessment:** While SPA, DPA, and CPA are focused on extracting secrets such as key recovery attacks, information leakage assessment focuses on quantifying the amount of information leakage from an implementation. Test Vector Leakage Assessment (TVLA) is a widely used statistical testing methodology used for information leakage assessment [64]. Test Vector Leakage Assessment (TVLA) is a widely adopted statistical method for assessing information leakage [64]. Its primary objective is to detect the presence of side-channel vulnerabilities using a conformance-style testing approach. This allows designers to identify and mitigate leakage early in the development lifecycle, ideally before fabrication for hardware designs and before deployment for software or firmware. TVLA is typically categorized into two types: non-specific and specific. The non-specific (fixed vs. random) leakage assessment technique was proposed to detect information leakage dependent on input data with the power signature. This method statistically compares trace differences between fixed input data and random input data. The specific leakage assessment method was proposed targeting specific intermediate values of the implementation.

Figure 7 illustrates the steps involved during information leakage assessment with TVLA. Once the type of assessment is determined, the necessary inputs need to be generated and simulated to obtain the expected power signatures using the steps outlined in Section II. Once the expected power traces are collected, they have to be analyzed for information leakage. Recent efforts have commonly used statistical techniques such as Welch's t-test, Pearson's chi-squared test (χ^2-test), and the Kullback–Leibler (KL) divergence metric [65].

The KL divergence (KLD) metric was utilized for evaluating hardware implementations of symmetric key cryptography implementations [27, 28]. KL divergence measures the divergence of one probability distribution from a second, expected probability distribution. Given two probability distributions, P and Q, defined over the same probability space, the KL divergence from Q to P in the context of continuous probability distributions is shown by Equation 17. Where $p(x)$ and $q(x)$ are the probability density functions of P and Q, respectively.

$$D_{KL}(P\|Q) = \int p(x) \log \frac{p(x)}{q(x)}\, dx \tag{17}$$

The RTL-PSC [27] framework employed a combined KL divergence and success rate metric to evaluate information leakage based on simulation-derived power profiles. This approach facilitated early identification and mitigation of side-channel leakage, proving to be both efficient and cost-effective in securing cryptographic implementations on Galois-Field (GF) and Look-up Table (LUT)-based AES designs.

Pundir et al. assessed several symmetric and post-quantum cryptographic algorithms using both KL divergence and Welch's t-test [28]. A metric was introduced where a high KL divergence indicates distinguishable power leakage probability distributions for two different keys, suggesting that an adversary could easily correlate power consumption between the keys. While this method was noted for its speed, it was less accurate. The pre-silicon assessment was validated through the post-silicon TVLA using the fixed vs. random test and Welch's t-test. The authors used Welch's t-test to determine the difference between the means of two populations that do not assume equal variances.

Equation 18 illustrates parameters used by Welch's t-test, where $\bar{X}_1$ and $\bar{X}_2$ are the sample means, s_1^2 and s_2^2 are the sample standard deviations, and n_1 and n_2 are the sample sizes of the two groups, respectively. In the context of TVLA, a t-value above 4.5 was used to indicate potential information leakage through side-channel attacks.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{18}$$

Welch's t-test was employed for a detailed simulation-based evaluation of how different sampling intervals impact the detection of PSC leakage in AES circuits on an FPGA [66]. Power traces were simulated to assess how the sampling rate affects the visibility of leakage and the effectiveness of side-channel attacks. Welch's t-test was also used to evaluate how memory-based high-level synthesis (HLS) optimizations affect the security of cryptographic implementations [46]. PSC leakage was assessed under different memory-based HLS optimizations by converting HLS implementations into RTL designs and mapping them to FPGA.

Jayasena et al. proposed a TVLA technique to evaluate hardware implementations of asymmetric key cryptographic systems [29]. A major limitation of applying methods used by symmetric key TVLA techniques was the leakage assessment techniques lost the notion of timing during the analysis. In order to solve this, authors have used Bonferroni correction [67] combined with Welch's t-test. This method allowed the authors to partition the power traces along the time axis and perform analysis on bit-serialized algorithms such as Elliptic Curve Cryptography (ECC) and RSA.

Liu et al. has explained an example of a use case of TVLA in the post-silicon phase, as it involves evaluating the system's side-channel characteristics after hardware fabrication through experiments on physical systems [31]. Here, TVLA is used to assess potential side-channel leakage from an AES-NI-based AES implementation due to frequency throttling activity. The TVLA methodology utilizes t-scores from Welch's t-test to compare different data sets and identify statistically distinguishable behaviors, suggesting the presence of side-channel leakage.

In addition to employing TVLA for leakage assessment, promising efforts were made to utilize graph neural networks for analyzing pre-silicon PSC leakages [48]. In this framework, the RTL design was converted into control and data flow graphs (CDFG) to detect vulnerable hardware modules and code lines. The framework was tested on several symmetric and post-quantum cryptographic hardware implementations, highlighting the effectiveness compared to statistical methods.

Since TVLA operates as a conformance-style test and is a strong candidate for use in FIPS certification [68], it cannot quantify the extent of side-channel vulnerability in the same way as evaluation metrics used in Common Criteria (CC) [69] certification, such as success rate (SR) and guessing entropy (GE). To address this gap between conformance-based (TVLA) and evaluation-based (CC) testing, Roy et al. introduced a framework that links the two domains by using TVLA to estimate the signal-to-noise ratio (SNR), which can then be used to compute the expected SR [70]. More recently, [71] introduced Bounded Information (BI), a finite-sample leakage certification framework that provides confidence-based upper bounds on the recovery success rate of a specified family of side-channel attackers, further supporting quantitative security evaluation beyond traditional leakage detection methods.

Besides the wide use of TVLA for information leakage assessment, [72] highlights several limitations of the fixed vs. random Welch's t-test. Specifically, Standaert demonstrates that this test is unable to reliably detect leakages in implementations protected with higher-order side-channel countermeasures, such as masking. In an d-th order masking scheme, the sensitive variable is split into $d+1$ shares, and leakage only appears through joint combinations of these shares. As noted in [72], when the noise level increases, the number of traces required to detect such higher-order leakage grows exponentially with the number of shares.

5) **Machine Learning (N-ML):** Timon introduced Differential Deep Learning Analysis (DDLA), a method that applies deep learning to non-profiled PSC attacks by leveraging convolutional neural networks (CNNs) [73]. DDLA uses CNNs to handle de-synchronized power traces and extract secret keys from cryptographic devices such as the ChipWhisperer-Lite. The approach includes metrics based on sensitivity analysis to identify the secret key and points of interest like leakage points and mask locations in the traces, making it effective against masked implementations without needing prior knowledge of the masking techniques.

Building on the foundational work of DDLA, Ahmed et al. presented optimization techniques to enhance the efficiency of these deep learning methodologies [74]. By employing parallel neural network structures and shared layers, this study reduced the time and memory required for training multiple models, thus speeding up the attack process. These optimizations were validated using ASCAD and ChipWhisperer-Lite datasets, showing improved performance and accuracy in extracting secret keys from power traces. They also applied deep learning methods, particularly CNNs, to PSC analysis, automating the process of signal re-alignment and noise reduction [75].

6) **Other:** In recent literature, several PSC attacks have emerged that utilize analysis methods not covered by traditional non-profiled categories. Wang et al. [58] introduced a novel attack interface that exploits dynamic voltage and frequency scaling (DVFS) in modern Intel and AMD x86 CPUs, effectively transforming a PSC attack into a remote timing attack. In this approach, the attacker uses the median of a probability distribution to cluster observed trace values, which are then leveraged to recover the secret key from a SIKE implementation. In an extension of this work, Wang et al. [76] applied a similar probability-distribution-based classification to exploit constant-time implementations of ECDSA and Classic McEliece, and also demonstrated a pixel-stealing side-channel attack.

Yu et al. utilized the same DVFS-based vulnerability explored in [58] to attack the Number Theoretic Transform in a lattice-based key-encapsulation mechanism. By comparing the mean frequency values, they were able to determine whether the unknown input contained only one or two non-zero coefficients [77]. Hot Pixels proposed by Taneja et al. read software-accessible internal sensors on Arm SoCs and GPUs while running crafted workloads, showing instruction- and operand-dependent variations in the power telemetry without any external probes [78]. The authors induced heavy, color-dependent computation via an SVG-filter stack and then read out the throttling as timing, enabling pixel stealing, history sniffing, and even website fingerprinting without privileged access using a distribution-based classification.

Sanjaya et al. [79] proposed a cryptographic key recovery attack on SIKE and AES by exploiting residual power signatures generated during context switches, specifically when the built-in sleep function induces power spikes. A probability-distribution-based classification method was again employed to infer secret key bit values in SIKE and byte values in AES.

Wei et al. introduced a non-traditional background-detection analysis on CNN accelerators that exploits the observation that when data inside the convolution unit remains unchanged between cycles, internal transitions are limited and the per-cycle power stays low [47]. Foreground and background have different power signatures because changing convolution windows over foreground pixels trigger more internal transitions, while repeated identical background pixels result in fewer transitions and lower dynamic power. A per-cycle power histogram is constructed, and a threshold is chosen at the maximal drop in cycle counts to distinguish background cycles from foreground activity. Cycles below this threshold are then mapped back to pixel coordinates, producing a binary background mask that reveals the foreground outline of the input image [47].

Moini et al. utilized a similar analysis method used in [47] to demonstrate a PSC attack on binarized neural network (BNN) accelerators in multi-tenant FPGA environments by leveraging on-chip time-to-digital converters (TDCs) to sense per-cycle voltage fluctuations [80]. This attack targeted the first convolution layer of the BNN, where foreground and background pixels create distinct power signatures due to differences in switching activity. The adversary constructed a histogram of voltage magnitudes and used a threshold to separate foreground from background pixels. This process enabled the reconstruction of input images, with denoising techniques further enhancing clarity [80].

Zhang et al. exploited the fact that ReLU implementations consume different amounts of power depending on whether the input is positive or negative, leading to two separable energy distributions observable through Intel's RAPL interface [81]. By repeatedly executing ReLU operations and constructing kernel density estimates of their power consumption, the attacker identifies the branch direction of each neuron and reconstructs the activation pattern of the network. With these activation patterns, they search for special input pairs, called flip pairs, that differ only in the activation of a single target neuron, enabling precise localization of the switching point. Comparing the input gradients of flip pairs yields direct equations for the associated weights, which are solved layer by layer, while the biases are subsequently derived from input-output relations once the weights are known. This density-distribution based analysis turns coarse RAPL power readings into a reliable signal for recovering the complete parameters of DL models.

In [82], the authors captured synchronized multi-axis oscilloscope traces (X, Y, Z, E) from the printer's stepper motor. The traces are low-pass filtered with a Butterworth filter, and peaks are detected with axis-specific height/prominence thresholds and logic to reject simultaneous or near-duplicate peaks. An algorithm was implemented to learn each axis's firing order and direction reversals, segment the timeline into valid/invalid regions, and apply a heuristic edit step (peak insertions/deletions) to enforce motor-sequence validity before resuming detection. Finally, a state machine integrated the corrected peaks to estimate head position and extrusion state over time, producing a millimeter-scaled 3D point cloud of the toolpath that reconstructs the printed model.

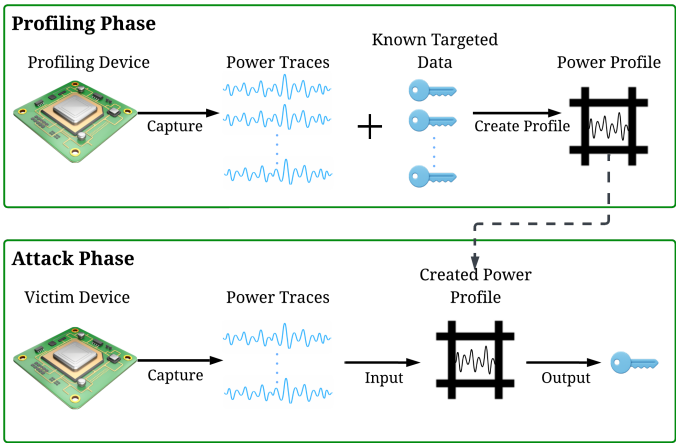

Fig. 8. An overview of profiled power side-channel analysis that consists of two phases: the profiling phase and the attack phase.

B. Profiled Side-Channel Analysis

Profiled side-channel analysis attacks leverage a detailed profile of the target device's power consumption behavior, typically obtained through a similar device under controlled conditions. This type of attack is considered more powerful and effective because it uses the pre-established profile to analyze the power traces of the actual target device. Figure 8 shows the two major phases for profiled side-channel analysis: the profiling phase and the attack phase. During the profiling phase, a detailed model of the device's power consumption patterns is created, often using statistical or machine learning techniques. During the attack phase, the power traces from the victim device is analyzed to extract the sensitive information. Two prominent types of profiled side-channel analysis are template attacks and ML-based attacks.

1) **Template-based Attacks:** Template-based attacks were introduced by Chari et al. [83]. They rely on constructing precise statistical models of a device's power consumption for each possible value of a secret intermediate, typically using multivariate Gaussian distributions whose mean and covariance matrices are learned during the profiling phase. A key component of template attacks is the identification of points of interest (POIs), which are the time samples most correlated with the secret-dependent leakage. During the attack phase, the attacker compares the observed power traces with the precomputed templates to determine the most likely value of the secret intermediate.

A stochastic-model-based attack enhances the classical DPA framework by replacing simple leakage predictions with a statistically learned approximation of the deterministic leakage and a shared noise model [56, 84]. During profiling, the attacker fits a low-dimensional model of how intermediate values leak and learns the noise distribution from traces collected

under a single key, enabling more accurate hypothesis testing during the attack phase. By combining an approximate deterministic leakage model with a shared noise model, the attack with a stochastic model achieves key-extraction performance close to template attacks while requiring orders of magnitude fewer traces during profiling, making it suitable when profiling measurements are limited or costly.

Wei et al. proposed a template attack for active adversaries, where a "power template" is constructed by profiling the relationship between power consumption and pixel values across multiple kernels in an input image reconstruction attack [47]. In the profiling stage, each cycle's associated pixel set Px_j and its corresponding power feature vector

$$\rho_j = (\rho_{0,j}, \ldots, \rho_{|K|,j}),$$

where $|K|$ denotes the number of kernels used, are collected to form a template:

$$PT = \{(Px_j : \rho_j) \mid j \in \text{cycles}\}.$$

The profiling phase is concluded by taking the union set of every input image template as the final power template. The image reconstruction in the attack phase selects consistent candidates across multiple cycles by minimizing the sum of the variance of pixel values for each position:

$$\min_{\text{Sel}} \sum_{x \in I} \text{var}\{pv(x) \mid pv(x) \in P_{x_{c_t}},\, c_t \in C_x\}.$$

,where I is the set of pixel positions, $pv(x)$ denotes a candidate pixel value for position x, $P_{x_{c_t}}$ is the set of possible values derived from cycle c_t, and C_x is the set of all cycles contributing information about pixel x. This variance-minimization ensures consistent reconstruction of each pixel, reducing uncertainty and enabling more accurate recovery of fine image details compared to simple threshold-based background detection.

Nešković et al. also demonstrated a template attack within their SystemC-based evaluation framework for AI accelerators [49]. In the profiling phase, the adversary assumes full control of the accelerator and builds templates by modeling the data-dependent power and noise distributions of each processing element, using chosen weights to characterize leakage behavior. During the attack phase, these templates are applied to traces from unknown weights, and candidate values are assigned by comparing measured power vectors against the pre-computed templates. This method enables recovery of all targeted weights with only a small number of traces and shows that template attacks remain effective even under noisy conditions.

2) **Machine Learning based (P-ML) Attacks:** ML–based attacks use ML models to classify power traces by learning discriminative patterns from profiling data. Commonly used models include Support Vector Machines (SVM) [85], Random Forests (RF) [86], neural networks [87], and deep neural networks (DNN) [88]. These models are trained on labeled power traces from a device similar to the target, learning to distinguish between different key values based on subtle variations in the power consumption patterns. In profiled side-channel analysis, inputs, outputs, and secret data are controlled and known during the profiling phase, which helps create accurate models for the attack phase. The key advantage of profiled side-channel analysis attacks is their high success rate and efficiency due to the detailed profiling, making them particularly effective against well-studied and high-value targets.

DeepTheft [89] was an ML-based attack designed to extract DNN model architecture, operating in two phases: offline and online. In the offline phase, a set of meta-models was trained using a dataset of energy traces from various neural network architectures. These traces were collected through sampling via the powercap interface on a ML model similar to the victim's, with each sampling point labeled by its corresponding layer type. In the online phase, the target model was queried to capture its energy trace, and a two-step attack was implemented. First, a pre-trained meta-model from the offline phase segmented the energy trace to identify the network structure. Then, using the segmented points, multiple pre-trained meta-models and domain knowledge were applied to infer layer-wise hyperparameters.

A CNN was used to perform a black-box attack against the masking countermeasure implemented in AES for the DPAcontestV4 [75]. X-DeepSCA [90] introduced the concept of multi-device training, which uses traces from multiple devices and optimizes hyperparameters for a 256-class DNN. By training the DNN model with data from four different devices, X-DeepSCA achieved an accuracy of over 99.9% across all four test devices, enabling effective single-trace attacks on an AES implementation.

The study conducted by authors Ghandali et al. focused on a new approach to profiled PSC attacks, employing a Twin Support Vector Machine (K-TSVM) with a deep kernel method [91]. This approach enhanced the effectiveness of power analysis by pre-training with Restricted Boltzmann Machine methods and fine-tuning using gradient descent. The attack method was evaluated against both masked and unmasked hardware implementations of AES-128.

An attack involved in creating a database of power consumption templates for various DNN layer and hyper-parameter configurations by profiling the power traces of known DNN models was proposed in [92]. Once the templates are established, the attacker matches the captured power traces against these templates to identify the corresponding DNN layers and their hyper-parameters. By using this template matching technique, the attack effectively reconstructed the sequence of layers and hyper-parameters of DNN models like MLP, AlexNet, and VGG16, achieving high accuracy.

The profiling phase in the attack proposed in [93] involved training on a known neural network configuration using RF classifiers. The authors used the tsfresh library for feature selection and extraction, obtaining a comprehensive set of features from time-series data. The classifiers were trained to identify folding parameters and neuron counts from power traces by using the top 10 features and top 50 features extracted from the tsfresh library, respectively. Once the model is trained, it is deployed on a target FPGA, where it accurately matches the power traces to the pre-learned profiles, successfully extracting the folding parameters and neuron counts.

A conditional Generative Adversarial Network (cGAN) was

TABLE II
SUMMARY OF POWER ANALYSIS METHODS.

Method	Figures of Merit	Advantages	Disadvantages / Trade-offs
SPA	Visual distinctness	Zero statistical overhead. Fast visual feedback.	Fragile to noise & simple countermeasures. Limited to implementations with strong leakage.
DPA	Difference of means	Effective with large traces; Extract information even from noisy power traces. No precise models required.	Needs multiple trace groups. Sensitive to trace alignment. Susceptible to higher-order countermeasures
CPA	Correlation coefficient, Success rate, Guess entropy, MTD	Generic. Widely used in cryptographic evaluations. Efficient.	Assumes an accurate power model. Requires sufficient power traces.
Leakage Assessment	t-value, KL-Divergence	Key-independent leakage detection. Useful for verification.	Cannot directly recover the key. Accuracy depends on trace count and alignment. False-Positive scenarios
Non-profiled ML	Accuracy	Learns statistical patterns; does not require a power model	Lower accuracy than DL; needs manual feature extraction
Non-profiled Other	Cluster distance, BCDC	Suited for remote attacks	Attack-specific heuristics. Limited generalizability.
Profiled ML	Recall	High success in noisy conditions. Robust to desynchronization and masking	Requires labeled traces. Computationally expensive to train.

used to perform a PSC attack in [94]. The generator in the cGAN aims to reconstruct private input data from the power leakage measurements of the compute-in-memory system. It takes the preprocessed power trace data as input and tries to produce a realistic reconstruction of the input data. The discriminator, on the other hand, attempts to distinguish between the real power traces and the generated ones. The training of the cGAN involves the generator trying to mislead the discriminator, which helps refine the data reconstruction to achieve high-quality results even in the presence of noise. In the attack phase, the trained cGAN model was used to recover the user's private data using newly acquired power traces.

TransNet [95] introduces a novel profiled side-channel analysis method based on a Transformer Network (TN) architecture that is both shift-invariant to trace misalignment and effective against masked implementations. Traditional profiled attacks often rely on well-aligned traces and tend to degrade in performance under high desynchronization. To overcome this limitation, TransNet incorporates a local embedding layer using one-dimensional convolution and a Transformer encoder, enabling the model to capture long-range dependencies and extract shift-invariant features. This architecture allows the network to autonomously learn leakage patterns without requiring manual preprocessing or desynchronization countermeasures. TransNet demonstrates strong performance in profiled attack scenarios where conventional alignment-based techniques struggle, making it a compelling alternative to existing CNN-based profiled attacks.

Building upon TransNet, EstraNet [96] introduces a computationally efficient TN-based architecture for profiled side-channel attacks. One of the primary challenges addressed by Hajra et al. is the scalability issue associated with long power traces in [95]. The proposed method achieves linear memory and time complexity with respect to trace length, while preserving the core principle of shift invariance to handle trace misalignments. EstraNet incorporates a novel attention mechanism that operates with linear computational overhead and introduces a new layer-centering normalization technique to overcome the limitations of standard normalization methods such as batch normalization and layer normalization. Similar to TransNet, EstraNet is trained on labeled power traces, and its learned representations are used to predict key-dependent features in unseen traces. Its computational efficiency and high classification accuracy make EstraNet a practical and effective solution for real-world profiled side-channel attacks targeting embedded cryptographic implementations.

C. Summary of Analysis and Insights

In summary, this section presented a comprehensive overview of the major PSC analysis techniques, categorized into non-profiled and profiled methods. Non-profiled attacks, such as SPA, DPA, and CPA, rely on statistical techniques without prior leakage characterization, making them practical but often less efficient under noisy or desynchronized conditions. In contrast, profiled attacks leverage supervised learning with labeled traces to create templates on device-specific leakage, offering significantly higher accuracy, particularly with the advent of deep learning-based models, such as CNNs and transformer architectures. The comparative summary in Table II highlights the advantages, disadvantages, and key performance metrics of each method. These foundational analysis techniques serve as the core analytical engines behind a broad spectrum of application-specific PSC attacks. In this paper, we explore how these methods are adapted across distinct application domains, each presenting unique leakage characteristics, platform constraints, and attacker capabilities.

The success of PSC attacks is not determined by the analysis technique alone, but by the alignment among the leakage source, attacker access model, power model, and trace quality. Classical non-profiled techniques remain effective when the leakage can be accurately modeled using assumptions such as HWT or HD, while profiled attacks are more suitable when the leakage is nonlinear, noisy, misaligned, or affected by hiding countermeasures. Stronger attack performance often comes at

TABLE III
SUMMARY OF POWER SIDE-CHANNEL ATTACKS TARGETING HARDWARE CRYPTOGRAPHIC IMPLEMENTATIONS. THE TABLE INCLUDES DETAILS ABOUT THE PROBING INTERFACE, SOFTWARE (SW), REMOTE (RM), AND PHYSICAL (PH), AS WELL AS THE TARGET PLATFORM, CRYPTOGRAPHIC ALGORITHM, ANALYSIS METHOD (NON-PROFILED: SPA, DPA, CPA, TVLA, N-ML; PROFILED: P-ML), THE POWER MODEL, NUMBER OF POWER TRACES USED (K AND M REPRESENT 1000 AND MILLION, RESPECTIVELY), AND ATTACK SUCCESS RATE OR ACCURACY.

Reference	Probing Interface	Target Platform	Target Algorithm	Analysis Method	Power Model	Number of Traces	Success Rate/ Accuracy
Jayasankaran et al. [97]	RM	FPGA	AES	CPA	HD	8100	100%
Schellenberg et al. [98]	RM	FPGA	AES	CPA	N/A	5000	100%
Gnad et al. [99]	RM	FPGA	AES	CPA	HWT	100K	50%
Gravellier et al. [100]	RM	FPGA	AES	CPA	HD	100K	100%
Ramesh et al. [101]	RM	FPGA	AES	DPA	N/A	2.66M	100%
O'Flynn and Dewar [102]	RM	MCU TEE	AES	CPA	HWT	160M	100%
Knechtel et al. [30]	PH	ASIC	AES	CPA	HD	2000	99.9%
Moos et al. [40]	PH	ASIC	PRESENT-80	CPA	HWT	5M	100%
Moos et al. [43]	PH	ASIC	PRESENT-80	CPA	HWT	4.9M	100%
Srivastava et al. [48]	PH	ASIC FPGA	AES RSA PRESENT SABER CRYSTALS-KYBER	TVLA	HD	N/A	94.49%
He et al. [27]	PH	ASIC FPGA	AES	TVLA	N/A	1000	N/A
Pundir et al. [28]	PH	ASIC FPGA	AES PRESENT SABER	TVLA	HD HWT	10K	N/A
Benhadjyoussef et al. [34]	PH	FPGA	AES	CPA	HD	20K	100%
Ghandali et al. [91]	PH	FPGA	AES	P-ML	N/A	60K	>99%
Dumitru et al. [42]	PH	FPGA	AES	CPA	HWT	1500	100%
Miura et al. [66]	PH	FPGA	AES	TVLA	N/A	30	N/A
Zhang et al. [46]	PH	FPGA	AES PRESENT	TVLA CPA	HD	1100	100%
Aysu et al. [32]	PH	FPGA	Binary Ring-LWE	SPA DPA	N/A	80K	100%
Ahmadi et al. [103]	PH	FPGA	RSA	SPA	N/A	500	100%
Iyer et al. [36]	PH	FPGA ASIC	AES	CPA	HD	100K	100%
Kamucheka et al. [104]	PH	FPGA	PQC	TVLA	N/A	4000	N/A
Benhadjyoussef et al. [41]	PH	FPGA	AES	CPA	HWT	6000	100%
Mozipo and Acken [61]	PH	IoT	LWC	CPA	HD	∼10K	100%
Das and Sen [90]	PH	MCU	AES	P-ML	N/A	∼1000	99.9%

the cost of stricter assumptions, such as access to a profiling device, large labeled datasets, or high computational effort for training and tuning. Therefore, no single PSC analysis method universally dominates, instead, the practical threat depends on the target platform, the available measurement interface, the amount of exploitable leakage, and the adversary's capability to collect and process traces.

IV. PSC ATTACKS ON CRYPTOGRAPHIC DESIGNS

Among all the applications targeted by PSC attacks, cryptographic applications are the most prominent targets due to the severe damage that can result from compromising these systems. Since the SPA and DPA attacks conducted by Kocher et al. on DES implementations in 1999, extensive research has been undertaken to bypass or weaken the confidentiality and privacy provided by cryptographic algorithms [1]. These cryptographic applications can be further categorized into two major subcategories based on their implementation era. These are pre-quantum cryptographic algorithms, which include well-established algorithms such as AES, RSA, and SHA, and post-quantum cryptographic algorithms, which are emerging and aim to provide more secure encryption schemes in the quantum computing era.

The National Institute of Standards and Technology (NIST) announced a competition aimed at finding encryption methods capable of resisting attacks from future quantum computers in 2016 [110]. The initial candidates were grouped into two categories based on the algorithm: key-encapsulation mechanisms (KEM) and public-key encryption (PKE), and digital signatures (DS). In 2017, sixty-nine candidates were selected from the submissions for the first round [111]. Then, twenty-six algorithms advanced to the second round, and only fifteen of those from the second round were selected for the third round in 2020 [112]. One algorithm (CRYSTALS-Kyber) from the KEM/PKE category and three algorithms (CRYSTALS-Dilithium, Falcon, SPHINCS+) from the DS category were selected for standardization, in 2022 [113]. Another four KEM/PKE algorithms (BIKE, Classic McEliece, HQC, SIKE) were advanced to the fourth round of the competition for further evaluation. The selected post-quantum cryp-

TABLE IV
SUMMARY OF POWER SIDE-CHANNEL ATTACKS TARGETING SOFTWARE CRYPTOGRAPHIC IMPLEMENTATIONS. THE TABLE INCLUDES DETAILS ABOUT THE PROBING INTERFACE, SOFTWARE (SW), REMOTE (RM), AND PHYSICAL (PH), AS WELL AS THE TARGET PLATFORM, CRYPTOGRAPHIC ALGORITHM, ANALYSIS METHOD (NON-PROFILED: SPA, DPA, CPA, TVLA, N-ML; PROFILED: P-ML), AND THE POWER MODEL, NUMBER OF POWER TRACES USED (K AND M REPRESENT 1000 AND MILLION, RESPECTIVELY), AND ATTACK SUCCESS RATE OR ACCURACY.

Reference	Probing Interface	Target Platform	Target Algorithm	Analysis Method	Power Model	Number of Traces	Success Rate/ Accuracy
Liu et al. [31]	SW	CPU	AES	CPA TVLA	Custom	8M	100%
Lipp et al. [105]	SW	CPU	KASLR	SPA	N/A	N/A	N/A
Lipp et al. [50]	SW	CPU TEE	AES-NI RSA KASLR	SPA CPA	HD	4M	93.75%
Wang et al. [58]	RM	CPU	SIKE	Other	Custom	364K	100%
Wang et al. [76]	RM	CPU	ECDS McEliece	Other	Custom	40M	N/A
Yu et al. [77]	RM	CPU	Kyber	Other	Custom	200K	100%
Lee and Han [106]	PH	ASIC	AES	P-ML	N/A	11K	79.71%
Sanjaya et al. [79]	PH	CPU	AES SIKE	Other	Custom	364k	100%
Ahmed et al. [75]	PH	FPGA	AES	N-ML	N/A	40K	100%
Benhadjyoussef et al. [41]	PH	MCU	AES	CPA	HWT	200	100%
Kamucheka et al. [104]	PH	MCU	PQC	TVLA	N/A	4000	N/A
Luo et al. [33]	PH	GPU	AES	CPA	HD	160K	100%
Munny and Hu [35]	PH	MCU	AES	CPA	HD	N/A	N/A
Jevtic et al. [107]	PH	MCU	AES	CPA DPA	N/A	130K	100%
Timon [73]	PH	MCU	AES	N-ML	N/A	60K	100%
Ahmed et al. [74]	PH	MCU	AES	N-ML	N/A	60K	N/A
Lee and Han [108]	PH	MCU	AES	CPA DPA	N/A	1M	99.92%
Saito et al. [109]	PH	MCU	RSA-CRT	P-ML	N/A	∼2.2M	99.69%

tography (PQC) schemes have underlying mathematical formulations that were either lattice-based (CRYSTALS-Kyber, CRYSTALS-Dilithium, Falcon) or hash-based (SPHINCS+). The selected fourth-round candidates were based on code-based (BIKE, Classic McEliece, HQC) and supersingular elliptic curve isogeny-based (SIKE) methods. However, some third-round candidates, like GeMSS were based on multivariate mathematical formulation.

A. *Software-based Attacks*

Lipp et al. exploited the Intel Running Average Power Limit (RAPL) interface to gather low-resolution power consumption data to perform the side-channel attack PLATYPUS [50]. RAPL provides an interface for managing the core's frequency and voltage, as well as tracking the power usage of the socket and memory domain. By statistically analyzing variations in power consumption, the PLATYPUS attacks were able to distinguish between different instructions, the hamming weights of operands, the hamming weight of the most significant bit of data values loaded from the cache, and the cache status of a load destination. This enabled the extraction of RSA private keys from mbedTLS, AES-NI keys from Intel SGX enclaves and the Linux kernel using CPA attacks, the breaking of kernel address-space layout randomization (KASLR), and the establishment of covert channels without requiring physical access to the target device.

A frequency throttling side-channel attack that shares a conceptual foundation with the PLATYPUS attack was presented in [31]. However, while PLATYPUS relied on direct power consumption telemetry, accessible via RAPL to perform a CPA, the frequency throttling side-channel attack leveraged the indirect timing information caused by power limit-induced frequency adjustments in the CPU. These adjustments, triggered when power consumption exceeds predefined thresholds, cause variations in execution time, which can then be correlated with the processed data. The authors were able to successfully retrieve the complete key bytes from both kernel space and user space with a constant-cycle implementation of the AES algorithm, which utilized AES-NI instructions, by analyzing encryption execution times.

The Prefetch+Power [105] attack is a novel side-channel technique that exploits the power consumption patterns associated with the execution of prefetch instructions on AMD CPUs. By leveraging the RAPL interface, the attack measured the power consumption of prefetch instructions to leak information about the page-table level at which the page-table walk is aborted. This method was particularly effective for breaking KASLR by detecting differences in power consumption when accessing different kernel memory locations. The Prefetch+Power attack has the significant advantage of not requiring high-resolution timers, relying instead on the power consumption measurements provided by the RAPL interface, which can be accessed from unprivileged user space.

B. *Physical Attacks*

In the early days of PSC attacks, attackers primarily focused on dynamic power. However, as we explained earlier, the total

power consumption of a chip is the sum of both dynamic and static power. One of the main reasons for the initial negligence of static power is its relatively smaller contribution compared to dynamic power. However, with advancements in semiconductor manufacturing processes, the transistor density of a chip has increased exponentially, creating a new target for PSC attackers to exploit static power as a side channel.

Unlike dynamic power-based physical side-channel attacks, performing a physical static power-based side-channel attack presents two primary challenges. The first challenge is the relatively small voltage drop across the shunt resistor, and the second is sensitivity to temperature variations. Moos et al. presented a measurement setup to carry out static PSC attacks by addressing the above challenges [40]. Additionally, Moos et al. evaluated the effects of temperature, measurement interval, and supply voltage on static PSC attacks, concluding that by increasing the operating temperature and increasing the measurement interval, an attacker can force a device to leak more information [40].

Similarly, Moos et al. focused on comparing static and dynamic PSC analysis of an ASIC implementation targeting a 150 nm CMOS prototype chip implementing the PRESENT-80 lightweight block cipher [43]. The authors successfully performed CPA and third-order CPA, collision-based Moments Correlating DPA (MCDPA), using the HWT power model on the F-box output for both dynamic and static power traces.

A conventional CPA-based PSC attack on an AES implementation on an 8051 MCU has been conducted [34]. In this attack, the authors used the HW power model for the AES first-round S-box output values. The authors claim that AES software implementations are more vulnerable to PSC attacks than AES hardware implementations.

Das and Sen conducted a deep learning (DL)-based profiled attack by developing a cross-device side-channel attack on cryptographic implementations in edge devices [90]. The proposed approach involved training a single deep neural network model using traces from multiple devices, combined with an optimal selection of hyperparameters for the 256-class DNN.

A profiled PSC attack on an 8-bit AVR microcontroller implementing a masked version of AES-128 was shown in [91]. By leveraging the DL-based K-TSVM approach, the attack exploits the power consumption patterns of the microcontroller to extract cryptographic keys.

The residual vulnerabilities of lightweight cryptographic algorithms (LWC) were examined in [61]. Despite the absence of integrated countermeasures, these LWCs showed vulnerabilities in their power consumption profiles, which can be exploited to leak sensitive information using CPA attacks. For ciphers like Ascon, Sparkle, and PHOTON-Beetle, the study considered chosen ciphertext attacks. On the other hand, the vulnerabilities of GIFT-COFB, Grain, Romulus, and TinyJambu were assessed using the HD power model.

Interesting vulnerabilities were found in several experiments where the countermeasure exposes new vulnerabilities [106, 108, 109]. The study [108] showed how attackers can leverage a vulnerability in AES implementations to distinguish between real and dummy operations when using the random insertion of dummy operations and shuffling as software countermeasures. By analyzing power traces with the Bounded Collision Detection Criterion (BCDC), the dummy operations were distinguished from real ones with high accuracy regardless of the dummy operation type. The authors explained that this distinction allows attackers to perform side-channel attacks with reduced complexity, significantly lowering the number of traces required for a successful attack.

Lee and Han [106] improved the above attack on AES software implementation that utilized dummy-load and shuffle-based countermeasures. They created a multi-label classification CNN to identify dummy S-box operations and subsequently performed a power analysis, such as DPA. Their results showed that the CNN-based detection method surpassed the accuracy of the approach proposed by Lee and Han [108]. The same dummy-load vulnerability was utilized by Saito et al. [109] to perform a DL-based single-trace PSC attack targeting the RSA–CRT (Chinese Remainder Theorem) implementation with windowed exponentiation. The primary focus was on open-source cryptographic libraries such as Gnu MP, which employ a dummy-load scheme as a countermeasure against side-channel attacks. The attack distinguished between true and dummy loads of the multiplicand using a neural network. It converted the 2^w-classification problem of finding the windowed exponentiation where the temporal window value is from 0 to 2^{w-1} into a $2^w \times 2$ classification problem.

The vulnerability exploited by Sanjaya et al. [60] analyzed power variations that propagate through shared voltage domains, allowing data extraction without direct access to the processor where the cryptographic operation occurs. Although this vulnerability was initially exploited in shared FPGAs, the paper showed that this vulnerability can be found in other hardware, including general-purpose MCUs in embedded systems. This novel physical layer supply voltage coupling (PSVC) vulnerability leaks sensitive data without requiring physical modifications to the embedded device, demonstrating PSVC as a viable attack vector for extracting information, even remotely. The authors also explained that this vulnerability can be used to perform attacks in both on-chip and on-board devices by demonstrating several SPA and CPA attacks on AES with different experimental setups.

An analysis of side-channel leakage of post-quantum algorithms was performed by Kamucheka et al. [104], where three NIST PQC algorithms selected for standardization (CRYSTALS-KYBER, CRYSTALS DILITHIUM, and FALCON) and two third-round finalists (SABER, NTRU) were analyzed using TVLA. Software implementations of these algorithms, which can be found in the "pqm4" library, were implemented on an STM32F4 Discovery evaluation kit. The authors also proposed a platform for trace collection from software implementations of PQC algorithms.

SleepWalk exposes a sleep-induced context-switch power spike as a single-sample leakage source: reverse-engineering the user-space call chain shows that sleep ultimately invokes the kernel scheduler, and these calls yield the most prominent, attacker-friendly spikes [79]. Unlike full-trace attacks, the method records only the spike amplitude, eliminating trace alignment and external triggers while an off-the-shelf Raspberry Pi 4 (Cortex-A72) with oscilloscope capture is used

to automate the collection of peak values for analysis [79]. Sanjaya et al. showed that, by placing the sleep immediately after the Montgomery three-point ladder in SIKE, yields a clear mean-peak separation due to reduced register HWT when anomalous zeros occur; a practical variant triggers after decapsulation and amplifies residual effects via repeated decaps to recover the key [79]. For AES-128, a chosen-plaintext attack forces HWT 0 at selected final-round bytes and, after large-batch amplification before sleep, the mean-peak differences recover most round-10 key bytes, reducing brute-force effort while using only one point per trace [79].

FPGA-based pre-quantum cryptographic implementations are well-explored attack targets in the PSC domain. For example, Benhadjyoussef et al. attacked an AES-128 implementation on a Side-channel Attack Standard Evaluation Board (SASEBO) using CPA [34]. The authors used the last round S-box values as the target intermediate values. In a different setup, the attack proposed by Ghandali et al. targeted a Virtex-5 FPGA embedded on a SASEBO-GII board, evaluating both masked and unmasked implementations of AES-128 [91]. The deep K-TSVM model was employed to analyze power traces from the FPGA, effectively distinguishing between the masked and unmasked states. This version of the attack underscores the robustness of the deep learning model in profiling and identifying vulnerabilities in more complex hardware platforms, highlighting its capability to bypass both types of cryptographic protections.

A static PSC attack targeting cryptographic implementations on FPGA hardware was proposed [42]. Specifically, it evaluated the effectiveness of a CPA attack on an AES-128 implementation within an FPGA. Kamucheka et al. presented a TVLA analysis of PCQ implementation on an FPGA [104]. They implemented CRYSTALS-Kyber, the selected lattice-based KEM/PKE PQC scheme for standardization. The authors demonstrated that there are some leakages at specific locations based on their TVLA results.

A hardware implementation of a quantum-secure encryption scheme based on the B-RLWE problem was introduced in [32]. The research detailed the vulnerabilities of the B-RLWE polynomial multiplication process, where partial product generation and intermediate sum updates create exploitable power leakage. During polynomial multiplication, each bit of the secret key polynomial determines whether the corresponding coefficients of the input polynomial are added to the intermediate sum. This step-by-step accumulation of products leads to discernible patterns in power consumption, which can be exploited by SPA and DPA attacks. The SPA attack leveraged the partial product generation and binary nature of B-RLWE coefficients to recover secret keys with minimal measurements. The DPA attack, on the other hand, utilized the values of intermediate sum updates in the B-RLWE implementation.

Performing PSC attacks on GPU implementations is a relatively difficult task compared to performing attacks on conventional CPUs due to the parallel processing nature of GPUs. Therefore, Luo et al. proposed a method for capturing power traces, sampling power traces, and processing power traces to perform a CPA and extract the key from a CUDA AES implementation on an NVIDIA TESLA GPU [33]. The

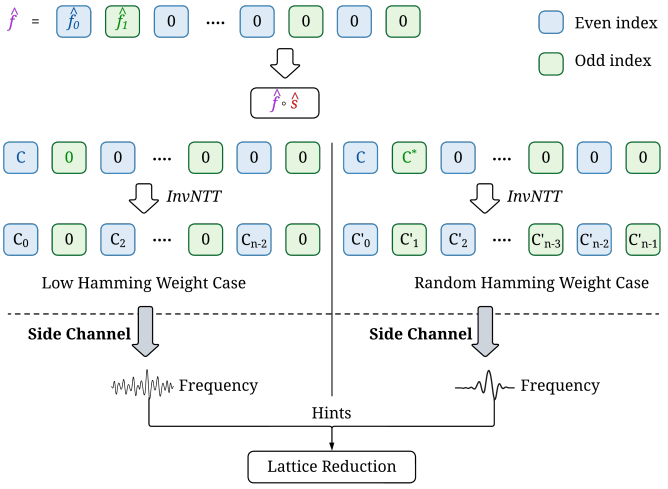

Fig. 9. Attack idea of the DVFS-based frequency side-channel on lattice-based KEMs in [77]. The approach relies on the observation that when carefully crafted inputs $\hat{f}$ are applied to the inverse Number Theoretic Transform $(invNTT)$ function, the resulting transformed outputs $g = InvNTT(\hat{f} \circ \hat{s})$ exhibit different Hamming weights. Here, $\hat{s}$ denotes the secret key in the NTT domain, and only the last two coefficients of $\hat{f}$ are non-zero: $\hat{f} = (\hat{f}_0, \hat{f}_1, 0, \cdots, 0)$. By exploiting this frequency-dependent timing leakage, the attacker can infer the HWT of $\hat{s}$ by controlling the $\hat{f}$.

results highlight the potential risks in deploying cryptographic algorithms on GPUs without proper countermeasures, as the power consumption data can reveal encryption keys.

C. *Remote Attacks*

The 'Hertzbleed' [58] attack exploits DVFS in modern Intel and AMD x86 CPUs to turn a PSC attack into a remote timing attack. This attack targeted cryptographic software and leveraged the data-dependent frequency adjustments caused by DVFS to create a remote timing attack vector. The Hertzbleed attack was theoretically more powerful than traditional software side channels and has a coarse granularity that makes it challenging to exploit. However, through sophisticated, well-crafted inputs that amplify single key-bit guesses into thousands of high- or low-power operations, the authors were able to achieve full key extraction remotely from two production-ready SIKE implementations. This attack is particularly concerning as it undermines the foundations of constant-time programming, which has been a critical defense against timing attacks. Wang et al. expanded on the Hertzbleed attack, demonstrating its applicability beyond the SIKE cryptosystem to include other cryptosystems such as ECDSA and Classic McEliece running on modern x86 CPUs [76]. The authors illustrated how even highly optimized constant-time cryptographic implementations can be vulnerable when subjected to Hertzbleed attacks.

Yu et al. investigated the side-channel leakage induced by DVFS on lattice-based cryptographic schemes, focusing on the Number Theoretic Transform (NTT) operations within CPA-secure Kyber and CCA-secure NTTRU [77]. Similar to the [58], this work demonstrated how frequency scaling effects could be exploited to infer information about modular arithmetic operations central to lattice-based KEMs. As shown in Figure 9, the attack leverages the fact that DVFS-induced

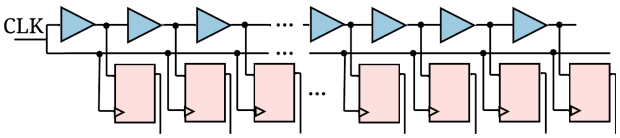

Fig. 10. Time-to-digital circuit [97].

frequency variations correlate with the HWT of intermediate NTT coefficients. By carefully analyzing execution-time variations correlated with DVFS-induced frequency adjustments, the authors were able to recover hints about intermediate NTT coefficients. This attack highlighted that even post-quantum cryptographic primitives, when implemented on DVFS-enabled CPUs, are vulnerable to software-based PSC attacks. This work extends the threat model beyond classical public-key cryptosystems, underscoring the need for hardware- and software-level countermeasures.

Every remote PSC attack on FPGAs consists of a power monitoring logic design, such as time-to-digital converters (TDC), ring oscillators (RO), and analog-to-digital converters (ADC). Figure 10 shows an example design of a TDC. Figure 11 depicts three different variants of ROs. These monitors are used to measure the voltage variations caused by cryptographic operations. Such attacks are very common in cloud FPGA environments, also referred to as multi-tenant setups, where the same FPGA is shared among several users.

Another remote PSC attack was introduced by Ramesh et al., which requires no physical access to the target board [101]. They embedded an RO sensor array to measure on-chip voltage variations and align per-byte oscillator counts with the cycles when each ciphertext byte is produced [101]. Using the ciphertext and the inverse AES S-box, they built last-round key hypotheses and correlated them with the measured RO counts to recover key bytes. This work shows that the FPGA fabric itself can serve as the sensing infrastructure, turning multi-tenant devices into self-contained side-channel benches [101].

Schellenberg et al. discussed an attack [98] that is capable of performing remote power analysis, which utilized TDC-based voltage sensors presented in [114]. The proposed attack was crafted from FPGA primitives to sense voltage fluctuations in the power distribution network (PDN). A standard CPA attack on AES-128 implementation on the side-channel evaluation platform SAKURA-G was performed using the traces collected using the proposed remote attack.

Jayasankaran et al. proposed a method to improve the sensitivity of TDC-based monitors by manually placing the TDC monitor in the FPGA [97]. In this work, they analyze the effects of junction temperature and the placement of the victim cryptographic implementation on the attack success rate. They were able to perform a CPA attack on AES-128 with a smaller number of traces using the improved sensitivity of the TDC. Also, the authors demonstrate the repeatability of the attack.

Gravellier et al. proposed high-speed RO voltage sensors that enable nanosecond-scale monitoring of supply fluctuations in multi-tenant FPGAs [100]. Using these sensors, they performed a remote CPA against a co-located 50 MHz AES hardware core and recover the key, demonstrating the first

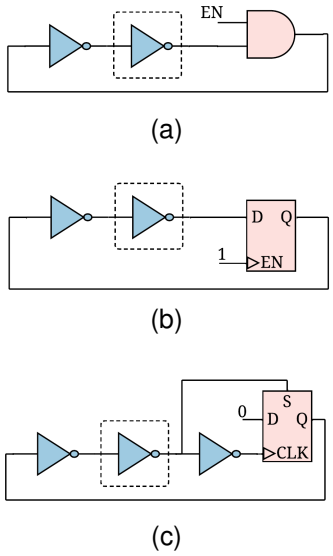

Fig. 11. Different types of ring oscillators (RO). The dashed box indicates an even count of inverters. (a) A basic RO utilizing a combinatorial loop, with each inverter represented by a single LUT in the FPGA. (b) An RO design employing latches to prevent the formation of a combinatorial loop. (c) Another RO design that uses flip-flops to prevent combinatorial loops [97].

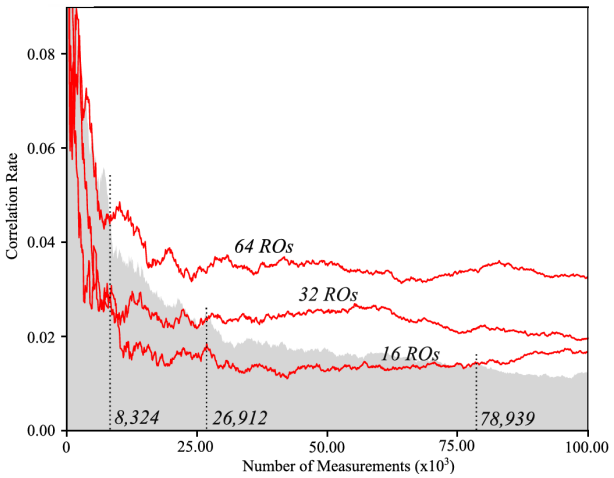

Fig. 12. CPA results obtained using 100000 traces captured with different numbers of RO-based sensors. The correlation corresponding to the correct key hypothesis is highlighted in red [100].

CPA conducted entirely with RO sensors inside the FPGA fabric [100]. They compared RO, TDC, and near-field EM setups, showing that on-chip proximity yields EM-comparable results despite lower sampling rates, while RO sensors nearly match TDC accuracy with lower area and simpler deployment [100]. As shown in Figure 12, they also showed that increasing the number of deployed RO sensors improves the trace SNR, leading to stronger correlations and a reduced MTD.

Gnad et al. showed that a tenant can mount a side-channel attack entirely from within the FPGA, undermining digital isolation in multi-tenant deployments [99]. With TDC-based on-chip sensors on a Lattice ECP5 board, they executed a CPA against an AES core and recovered roughly half of the key bytes with up to 10^5 traces gathered in under three hours, proving the feasibility of remote power analysis [99]. They further demonstrated that toggling on-chip oscillators at 1–3 MHz induces timing faults in AES and summarize

successful attacks across multiple FPGA families [99].

O'Flynn and Dewar mounted an on-device, cross-domain power side-channel attack on TrustZone-M, which is the Trusted Execution Environment (TEE) on ARM Cortex-M microcontrollers, by running non-secure code that triggers secure-world encryptions while the chip's own ADC samples the core supply voltage, eliminating any need for external probes [102]. The attacker arms the ADC immediately before invoking the secure AES routine, turning standard peripherals into a built-in oscilloscope. The primary target was the hardware AES accelerator in the secure world, and the authors further show that controlling the ADC entirely from the non-secure world still enables key-recovery analysis, demonstrating cross-domain leakage [102].

D. *Summary of Attacks and Insights*

In Table III and Table IV, we provide an overall summary of the similarities and differences between recent PSC attacks targeting cryptographic hardware and software implementations. This summary highlights the vast array of attack options used in the literature, including targeted hardware, cryptographic algorithms, analysis methods, power models, required number of traces (measurements), and attack success rate (accuracy).

Figure 13 shows that trace requirements vary by several orders of magnitude across probing interfaces and platforms. Remote and software probing-based attacks generally require more traces due to lower signal quality, while several FPGA-based remote attacks can achieve disclosure with only $10^3 - 10^5$ traces. Physical attacks show the widest spread, from hundreds to millions of traces, indicating that direct access alone does not guarantee low-cost key recovery. Overall, the data suggests that MTD is decided not only by the cryptographic algorithm, but also by the probing interface, target platform, leakage strength, and attack methodology.

Beyond the traditional focus on physical probing, recent developments in software-based PSC attacks have significantly reshaped the threat landscape for cryptographic implementations. Interfaces such as Intel's RAPL and NVIDIA's NVML enable attackers to obtain power-related telemetry without requiring physical access to the victim device, making software and remote PSC attacks considerably more practical than the classical oscilloscope-based power trace acquisition setup. Their accessibility, scalability, and ability to operate in virtualized or cloud environments broaden the adversarial model and eliminate the laboratory constraints that historically limited physical PSC attacks.

Meanwhile, the emergence of PQC algorithms has further expanded the scope of PSC research beyond classical primitives such as AES and RSA. Although PQC schemes offer strong mathematical guarantees against quantum adversaries, recent works demonstrate that their implementations still exhibit data-dependent leakage, indicating that PSC vulnerabilities may persist even in the post-quantum era. At the same time, advancements in ML, particularly profiled models capable of extracting shift-invariant features from noisy or misaligned traces, have enhanced the capabilities of modern PSC attacks by improving efficiency and noise tolerance.

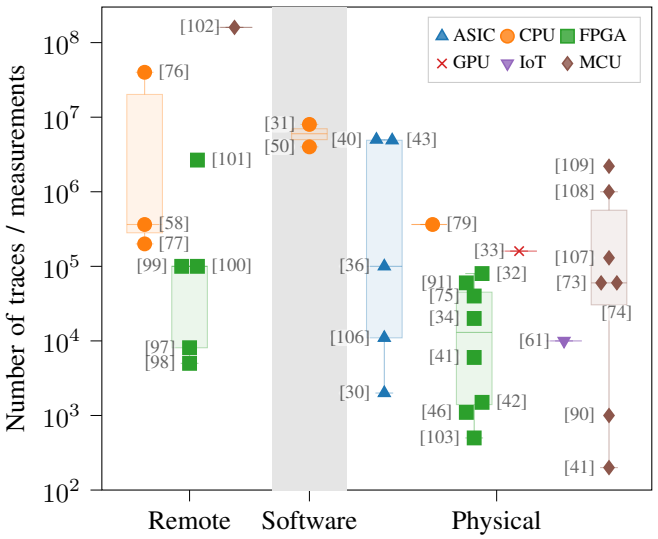

Fig. 13. Trace requirements for cryptographic PSC attacks across probing interfaces and target platforms. Points denote individual studies, while shaded box plots summarize the trace-count distribution for each platform group.

Despite these developments, the literature continues to focus predominantly on symmetric-key algorithms, while asymmetric primitives and PQC implementations remain comparatively underexplored due to their algorithmic complexity, diverse arithmetic structures, and challenging leakage characteristics. Collectively, these trends illustrate how evolving software interfaces, emerging cryptographic standards, and ML-driven analysis methods are expanding both the practicality and the reach of PSC attacks well beyond their classical boundaries.

V. User-Specific Data Privacy

Another notable category of PSC attacks targets user behavior data, such as website fingerprinting. These attacks primarily focus on mobile and embedded devices, which frequently involve user interactions. By analyzing the power consumption patterns during user activities, attackers can infer detailed user behavior and patterns. Given the nature of these attacks, most are profiled PSC attacks, where the attacker builds a profile of power consumption patterns to effectively identify and exploit user behaviors. This makes them particularly dangerous, as they can be used to track user activities, infer visited websites, and compromise user privacy without needing direct access to the data being transmitted or stored.

A. *Software-based Attacks*

Intel's RAPL interface can be exploited to conduct website fingerprinting attacks. By monitoring the fine-grained power consumption data available through RAPL, attackers can infer which websites a user is visiting. This approach relies on the distinctive energy usage patterns generated during the rendering of different web pages, which can be captured and analyzed to identify specific websites with high accuracy. Zhang et al. demonstrated that using this method, it is possible to achieve up to 99% accuracy for regular network scenarios and 81% accuracy when the Tor browser is used [115].

POWERFUL [116] demonstrated a method for fingerprinting mobile applications by analyzing the power consumption

TABLE V
SUMMARY OF POWER SIDE-CHANNEL ATTACKS TARGETING USER BEHAVIOR DATA EXTRACTION. THE TABLE INCLUDES DETAILS ABOUT THE PROBING INTERFACE: SOFTWARE (SW), REMOTE (RM), AND PHYSICAL (PH), AS WELL AS THE TARGET PLATFORM, EXTRACTED USER BEHAVIOR DATA, NUMBER OF POWER TRACES USED (K REPRESENTS 1000), AND ATTACK ACCURACY.

Reference	Probing Interface	Targeted Platform	Extracted Information	Attack Model	Number of Traces	Accuracy/ F-score
Zhang et al. [115]	SW	CPU	Website fingerprinting	KNN CNN	3700	99%
Chen et al. [116]	SW	Smartphones	Mobile app fingerprinting	C4.5 RF SVM	N/A	92.9%
Kogler et al. [59]	SW RM	CPU	Data in memory hierarchy	CPA	23K	99.8%
Taneja et al. [78]	RM	CPU GPU	Website fingerprinting Image pixel value	Other	150K	49%
Wang et al. [76]	RM	CPU	Image pixel value	Other	200K	N/A
La Cour et al. [117]	PH	Smartphones	Website fingerprinting Passcode length OLED screen activity Audio Fingerprinting	1D CNN RF	1000	95.5%
Gatlin et al. [82]	PH	3D Printers	3D model reconstruction	Other	N/A	99%
Yang et al. [118]	PH	Smartphones	Website fingerprinting	RF - WEKA	2000	75.2%
Matovu et al. [119]	PH	Smartphones	Website fingerprinting Keystroke inference Incoming call detection Camera usage detection	RF, SVM	3000	71.45%
Cronin et al. [120]	PH	Smartphones	Touchscreen activities Passcode inference	1D CNN	12K	92.8%
Yucebas and Yuksel [121]	PH	VLC systems	Binary data	SPA	800	76%
Spolaor et al. [122]	PH	USB peripherals	Peripheral type and identity	RF	20K	99%
Qin and Yue [123]	PH	Smartphones	Website fingerprinting	SVM	1000	55%

profiles on Android devices. By capturing the distinct power consumption patterns associated with different apps, the authors inferred which app a user is currently using. To collect power data, the authors implemented an Android application that periodically reads voltage and current measurements from publicly accessible system files at a frequency of 2 Hz. The collected data was processed and used to build models using three lightweight supervised classifier techniques, which can accurately identify which app was being used based on real-time power consumption data.

Kogler et al. introduced Collide+Power, a software-based PSC that exploits the mere co-location of attacker- and victim-controlled values in the CPU memory hierarchy to create exploitable, data-dependent power differences [59]. The attacker primes a target cache set with a chosen guess G, lets the victim access the secret value V, and observes that the energy used when a cache line transitions from G to V scales with the $HD(G, V)$. This leakage is observable either via power interfaces (e.g., RAPL) or indirectly through throttling-induced timing variations, enabling unprivileged attacks. They model the leakage as a linear combination of HD and HWT, and amplify the signal with a differential measurement: measure once with G and once with the masked inverse $\tilde{G} = G \oplus m$, then subtract, which cancels constant terms, doubles the HD component, improving the SNR by $\approx 8.8\times$ in practice [59]. Instantiations include MDS-Power (leaking "data-in-use" from a sibling hyperthread) and Meltdown-Power (leaking "data-at-rest" via kernel prefetch gadgets), illustrating that Collide+Power is generic, independent of a specific cryptographic algorithm, and effective with either direct power readings or timing proxies.

B. *Physical Attacks*

A profiled attack to identify web browsing activities on smartphones by analyzing power consumption during USB charging was employed in [118]. By training an RF classifier on power traces of the Alexa top 50 websites on multiple smartphones, the authors developed a model that can accurately identify websites visited by the user. The attack involved feature extraction from the frequency domain of power traces, allowing the classifier to recognize distinct power signatures associated with different websites. The attack was evaluated under different environmental conditions, such as varying battery levels, enabling or disabling the browser cache, using Wi-Fi or LTE, and background app activities, to ensure robustness.

Another profiled attack was implemented by Qin and Yue to identify websites visited on Android 7 smartphones [123]. The authors collected CPU load and frequency data while measuring power data to estimate power consumption without requiring user permission. The linear regression-based power estimation model was used to perform the attack to identify web browsing activities. By training an SVM classifier on the estimated power traces, the authors achieved high accuracy in distinguishing between different websites.

Matovu et al. explored three distinct PSC attacks on charging smartphones [119]: website inference, keystroke inference, and incoming call detection. In these scenarios, the authors leverage the power consumption data captured while

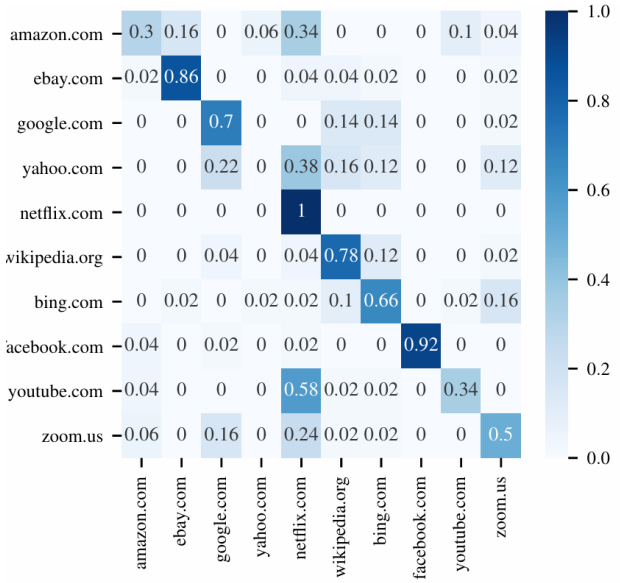

Fig. 14. Attack results when training on wired-charging traces and testing on wireless-charging traces. The vertical axis represents the true labels, while the horizontal axis represents the predicted labels [117].

the smartphone is charging to infer sensitive information. For instance, the website inference attack aims to identify which websites the user is visiting, while the keystroke inference attack attempts to determine the keys being pressed to extract passwords. The attacks are implemented on two different Samsung Galaxy phones during the charging process, in a public charging station controlled by an adversary.

The Charger-surfing attack [120] exploited PSC leakage to extract sensitive information from smartphones during charging. This attack targeted dynamic content on the smartphone touchscreen by analyzing power consumption patterns, specifically during touchscreen animations such as button presses. By capturing power traces during touchscreen activities, the authors trained neural network models to classify the location and type of animations displayed on the screen. The attack was tested under various scenarios, such as different battery levels and screen technologies, to ensure robustness and effectiveness across a wide range of scenarios.

A method to exploit PSC attacks during wireless charging, demonstrating that activities such as website loading on smartphones can be inferred by monitoring the power consumption of a Qi wireless charging transmitter was proposed [117]. The authors captured power traces and successfully identified websites loaded on iPhone 11 and Google Pixel 4 devices by analyzing the current drawn during wireless charging without needing direct device access. This was achieved using a CNN-based classifier to analyze the collected power traces and identify specific websites loaded on the device from the power consumption patterns observed during wireless charging. Figure 14 demonstrates the transferability of the profiled attack, achieving an accuracy of 49%. Specifically, a model trained on power traces acquired through a wired charger connection, which provides a higher SNR, was successfully used to classify traces acquired through a wireless charger.

PowerID was a framework proposed for demonstrating the feasibility of fingerprinting USB peripherals by analyzing their power consumption patterns [122]. The evaluation was conducted using power traces from 82 USB peripherals, covering 35 models across 8 types, including flash drives, external hard drives, WiFi adapters, Bluetooth adapters, microphones, webcams, mice, and keyboards. Using RF classifiers, the authors developed profiles that could distinguish between different models and types of USB peripherals with high accuracy. These profiles were used to analyze the power traces to accurately identify the type, model, activity, and identity of each USB peripheral. A method to reconstruct 3D-printed models by exploiting PSC leakages from Fused Deposition Modeling (FDM) 3D printers was demonstrated in [82]. By analyzing the power consumption of the printer's stepper motors, the authors developed a model that correlates specific current draw patterns with the movement of the printhead, such as reversal, start, stop, and dwell. Using this profiled data, the toolpath and subsequently the entire 3D model were accurately reconstructed. This attack showed that even with end-to-end encryption, the PSC provides sufficient information to bypass security measures and extract sensitive design data in environments like outsourced additive manufacturing (AM).

Security vulnerabilities of visible light communication (VLC) systems through PSC attacks were investigated by [121]. The authors designed a bidirectional VLC system and implemented a current sensor on the power lines from the power supply to the VLC transmitter. The attack exploited the power variation caused by the LEDs rapidly switching on and off to transmit binary data, resulting in distinct power levels. By analyzing the instantaneous power consumption during data transmission, the transmitted binary data was extracted.

C. *Remote Attacks*

Wang et al. demonstrated how the Hertzbleed attack can be adapted into a pixel-stealing attack on modern Intel and AMD x86 CPUs [76]. This attack exploits the variation in power consumption associated with rendering different pixel colors, which leads to detectable changes in CPU frequency. These fluctuations are caused by the data-dependent power usage of the integrated GPU (iGPU) when applying SVG filters to graphical data. Here, instead of relying on an approach where the CPU frequency is directly observed, the authors took advantage of how CPU frequency influences the number of JavaScript operations that can be executed within a given time frame. The proof-of-concept demonstrated in the study successfully extracted pixel values at a rate of 1 to 3 pixels per second, with a relatively low error rate, depending on the system configuration in the latest version of Google Chrome.

Taneja et al. utilized a similar approach to the Hertzbleed attack to perform a pixel-stealing attack that exploits the DVFS mechanism in integrated and discrete GPUs (iGPUs and dGPUs), as well as ARM SoCs [78]. The Hot Pixels attack was executed through a JavaScript-based approach, where an attacker applies an SVG filter to a targeted pixel in a cross-origin iframe. This attack can recover images from cross-origin iframes and extract browsing history in web browsers like Chrome and Safari, even with all side-channel countermeasures enabled. Additionally, the authors

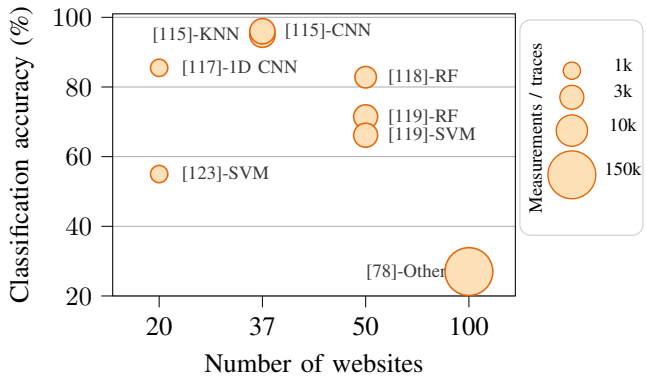

Fig. 15. Accuracy–measurement trade-off for website-fingerprinting PSC attacks. Shape indicates the number of required traces (measurements).

performed a website fingerprinting attack using Apple iGPUs. This attack leveraged the fact that different websites induce distinct patterns of computation on the GPU, leading to unique power and frequency fluctuations that can be measured via internal sensors through unprivileged software.

D. *Summary of Attacks and Insights*

Table V summarizes key information from our survey of PSC attacks targeting user behavior data extraction. The table includes details such as the probing interface, targeted hardware, extracted user behavior data, machine learning models used to perform the profiled attacks, required number of power traces, and attack accuracy (F-score). It highlights the similarities and differences among recent approaches. As the majority of the surveyed attacks on user behavior data are profiled, power model information is not included. Instead of categorizing by power analysis methods, this table focuses on preprocessing techniques and the profiled models employed in each study.

A notable trend in PSC attacks targeting user behavior data is the predominance of fingerprinting-based approaches, with website fingerprinting being the most widely explored. These attacks leverage the strong correlation between high-level software activity and power consumption patterns, enabling profiled classifiers to distinguish between websites based on their unique computational and rendering characteristics. This close alignment between fingerprinting tasks and profiled analysis methods stems from the fact that such attacks rely less on explicit power models and more on learned representations that capture subtle statistical differences in power traces.

Figure 15 shows that website-fingerprinting PSC attacks do not exhibit a simple monotonic relationship between the number of measurements, number of websites, and classification accuracy. Several attacks achieve high accuracy with relatively few measurements when the classification space is moderate, such as 20-50 websites, suggesting that strong activity-dependent power signatures can enable efficient profiling. However, the 100-website case requires substantially more measurements while achieving much lower accuracy, indicating that increasing the number of website classes significantly increases classification complexity. The results suggest that attack success is shaped not only by trace count, but also by signal quality, probing setup, device behavior, and the effectiveness of the selected classification model.

While existing works have primarily focused on website identification, similar fingerprinting principles could be extended to a broader range of behavioral contexts, such as application usage states, background service activity, or device-specific interaction patterns. In addition to website identification, surveyed studies show that PSC attacks are capable of extracting even fine-grained touchscreen-related behaviors, including touch positions, swipe patterns, and interaction timing, demonstrating that modern user interfaces expose rich and highly distinguishable side-channel leakage.

VI. Reverse Engineering of Functionality

PSC-based disassembly attacks represent a sophisticated and emerging threat in cybersecurity, leveraging the analysis of power consumption patterns to deduce the internal operations of a target device. These attacks focus on extracting assembly-level instructions by monitoring the power usage during the execution of software on a device. By analyzing variations in power consumption, attackers can identify the sequence of executed instructions, effectively "disassembling" the software without needing direct access to the code. This type of attack is concerning for embedded systems and mobile devices, where the integrity and confidentiality of the software are important. PSC-based disassembly attacks can be used to reverse-engineer proprietary algorithms, identify vulnerabilities, and even modify the behavior of the software, posing a significant risk to the security and functionality of affected devices.

A. *Software-based Attacks*

In the PLATYPUS attack [50], although the primary focus was on cryptographic applications, the attack exploited information leakage from Intel RAPL to distinguish between different instructions, the HWT of operands, HWT and the value of the most significant bit of data values loaded from the cache, and the cache status of load destinations. Figure 16 depicts the power consumption distributions of several instructions. Using these distributions, Lipp et al. [50] demonstrated instruction-level distinguishability through power analysis. Therefore, the methodology used in the PLATYPUS attack can also be used to perform instruction disassembly.

B. *Physical Attacks*

A PSC-based disassembler that identified and disassembled instruction sequences in embedded systems at an instruction-level granularity was introduced by [128, 131]. The proposed methods involved collecting power traces through an oscilloscope, transforming these traces into the time-frequency domain using continuous wavelet transform (CWT), and then applying feature selection with KL-divergence followed by dimensionality reduction using Principal Component Analysis (PCA). Machine learning classifiers, such as quadratic discriminant analysis (QDA), Linear Discriminant Analysis (LDA), SVM, and Naïve Bayes, were used to create profiles of power consumption patterns for each instruction [128]. Machine

TABLE VI
SUMMARY OF POWER SIDE-CHANNEL ATTACKS TARGETING INSTRUCTION DISASSEMBLY. THE TABLE PROVIDES DETAILS ABOUT THE PROBING INTERFACE (SOFTWARE (SW), REMOTE (RM), AND PHYSICAL (PH)), TARGET PLATFORM, TYPE OF EXTRACTED INSTRUCTION-LEVEL INFORMATION, NUMBER OF POWER TRACES USED (K AND M REPRESENT 1000 AND MILLION, RESPECTIVELY), AND ATTACK ACCURACY.

Reference	Probing Interface	Targeted Platform	Extracted Information	Attack Model	Number of Traces	Accuracy/ F-score
Lipp et al. [50]	SW	CPU	Instructions	Non-profiled Other	10K	N/A
Glamočanin et al. [124]	RM	Soft-Core CPU	Instructions	LSTM CNN MLP ResNet	110M	86.46%
Krishnankutty et al. [125]	PH	Soft-Core CPU MCU	Instructions	SVM	15K	∼99%
Fendri et al. [126]	PH	CPU	Instructions	MLP	∼56K	96.01%
Han et al. [127]	PH	MCU	Control flow	BiLSTM BiRNN HMM	150	98%
Park et al. [128]	PH	MCU	Instructions	LDA QDA SVM Naive Bayes	24K	93.03%
Vafa et al. [44]	PH	MCU	Instructions	Ensemble of Classifiers	7K	99%
Narimani et al. [129]	PH	MCU	Instructions	CNN	>1M	63.29%
Van Geest and Buhan [130]	PH	MCU	Instructions	LDA QDA MLP CNN	250K	88.2%
Bae and Ha [131]	PH	MCU	Opcode	RF KNN MLP CNN	120K	91.9%

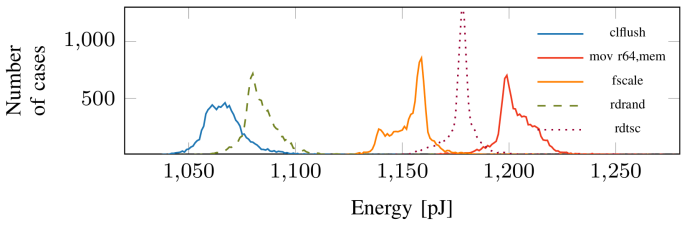

Fig. 16. Power consumption histograms of several instructions on an Intel i7-6700K desktop processor [50].

learning models, including K-nearest neighbor (KNN), RF, CNN, and Multi-Layer Perceptron (MLP), were used [131] to recover the instructions from the power traces with high accuracy. This hierarchical classification framework allowed for the identification of instructions, including register names.

Reverse engineering the instructions of the 8-bit PIC16F690 microcontroller and the 32-bit ARM Cortex-M3 processor were targeted by the authors Vafa et al. [44]. The method leveraged both dynamic and static power consumption to extract the maximum information leakage, revealing opcodes and operands of real instructions. Classification algorithms and KL feature selection methods were employed to improve the classification rate of instructions into different HW groups.

Another disassembler attack aimed to reverse-engineer the instructions executed on the ARM Cortex-M0 processor-based MCU was proposed in [130]. ARM Cortex-M0 is a 32-bit architecture commonly used in low-power IoT devices. This attack also employed a profiling approach using machine learning algorithms, including LDA, QDA, MLP, and CNN, to classify power traces associated with different instructions. The attack successfully identified instruction sequences and distinguished between instruction groups based on the power consumption patterns.

A disassembly attack aimed to reverse-engineer executed instructions by performing a template attack on power traces of the Atmel 8-bit AVR MCU was proposed [129]. The attack employed a profiling technique using CNNs to classify the power traces corresponding to different instructions executed on the AVR microcontroller. It achieved an accuracy of 98.21% on real code, significantly outperforming previous ML-based methods such as KNN and LDA.

Han et al. craft assembly-level malware that preserves program functionality while manipulating power traces so that power side-channel control-flow monitors classify it as benign [127]. Their methodology builds a substitute detection setup, partitions the payload into context-independent chunks, performs data-flow analysis to identify each chunk's live range, and uses Monte Carlo Tree Search to select injection points that maximize the detector's confidence score [127]. The attack is evaluated on ARM Cortex-M programs across multiple detector families (e.g., Bidirectional Long Short-Term Memory neural network (BiLSTM), Bidirectional Recurrent Neural Network (BiRNN), and Hidden Markov Model (HMM)). The authors conclude that such monitors are less secure than commonly assumed and recommend stronger designs, such as multi-cycle windows and fusing power with EM to withstand instruction-level, side-channel-aware injections [127].

Fendri et al. presented a comprehensive framework designed to evaluate the vulnerability of CPU designs to PSC disassem-

bly attacks at the design phase [126]. The proposed method involved generating simulated power traces from HDL models of RISC-V CPUs at the design stage and employing sparse dictionary learning (SDL) for feature extraction and dimensionality reduction. The deep learning model, specifically a multilayer perceptron (MLP), is trained to classify instructions based on these power traces. This developed framework facilitates the evaluation of vulnerabilities and countermeasures before hardware manufacturing.

Krishnankutty et al. presented an advanced technique for identifying and disassembling multi-clock cycle instruction sequences on embedded systems with a pipelined architecture using PSC analysis [125]. By observing the power variations correlated with instruction execution, the authors were able to identify instruction boundaries using a dynamic programming (DP) algorithm and classify instruction types using a fine-grained classifier with high accuracy. The study also demonstrated that by using multiple power supply pin measurements, the precision and reliability of instruction classification can be significantly improved on both FPGA-based soft processors and discrete microcontrollers.

C. *Remote Attacks*

Glamočanin et al. evaluated instruction-level code disassembly using PSC analysis on RISC-V soft-core CPUs (RISCY and PicoRV32) implemented in shared FPGA environments [124]. The focus is on multitenant FPGA scenarios where attackers can deploy on-chip TDC-based voltage sensors to capture power traces remotely. The study presented a profiled attack employing both traditional machine learning and novel deep learning techniques to classify CPU instructions based on power consumption patterns.

D. *Summary of Attacks and Insights*

PSC-based disassembly attacks exploit power variations to recover executed instructions and program behavior, posing a severe threat to software confidentiality on different platforms. As summarized in Table VI, these attacks have been demonstrated in diverse settings, ranging from software-based approaches that leverage low-level telemetry (e.g., PLATYPUS) to physical and remote setups using oscilloscope measurements or FPGA-integrated sensors. Attackers applied wide range of ML models, including CNNs, RF, SVMs, and MLPs, to classify instruction opcodes and operands with high accuracy. The surveyed works show that profiled analysis dominates this domain, with attacks successfully reverse-engineering instruction sequences across architectures, such as AVR, ARM Cortex, and RISC-V. Therefore, power model information is not included in the Table since power models are typically used only in non-profiled attacks. Instead of categorizing by power analysis methods, this table focuses on preprocessing techniques, feature extraction methods, and the profiled models employed in each study.

Figure 17 shows that PSC-based instruction disassembly attacks can achieve high recovery accuracy across a wide range of instruction-set coverage, but their measurement cost varies significantly. Physical attacks on MCU and soft-core platforms

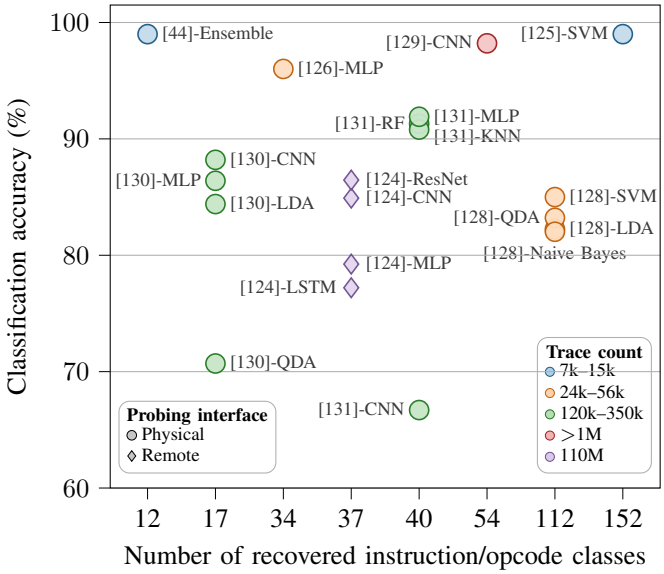

Fig. 17. Accuracy-trace trade-off for PSC-based instruction disassembly attacks. Each point represents a reported attack model configuration from Table VI, where the x-axis shows the number of recovered instruction/opcode classes, the y-axis shows the reported accuracy, color indicates the power trace budget, and shape denotes the physical versus remote probing interface.

generally report high accuracy with relatively modest trace budgets, indicating that instruction-dependent leakage is often distinguishable under controlled measurement conditions. In contrast, the remote soft-core CPU attack requires a substantially larger trace budget while recovering fewer instruction classes with lower accuracy, reflecting the weaker signal quality and higher noise of remote probing.

A closer examination of PSC-based disassembly attacks reveals several important insights regarding both their methodological trends and their practical constraints. Nearly all surveyed studies rely on physical probing, reflecting the early dominance of MCU-based evaluations where on-chip telemetry interfaces, such as RAPL, NVML, or FPGA-integrated sensors, are unavailable. This historical focus on MCUs naturally constrained the attack surface to physical measurement setups, but it also provided cleaner and more deterministic leakage signals that facilitated instruction classification.

The surveyed literature shows that profiled analysis methods overwhelmingly dominate this domain, as machine learning models are highly effective at capturing instruction-dependent variations within localized trace segments, enabling fine-grained opcode and operand recovery.

However, this strong dependence on profiled approaches and physical access highlights several limitations: these attacks have not yet been widely demonstrated on more complex platforms where multi-core interference, dynamic scheduling, cache effects, and power-management policies introduce substantial noise. Additionally, the reliance on MCU platforms leaves open questions about the generalizability of instruction-disassembly attacks to CPUs, GPUs, and heterogeneous accelerators where richer remote or software-accessible telemetry channels exist. These observations suggest that while PSC-based disassembly has matured rapidly in controlled MCU environments, its broader applicability depends on extending current methodologies beyond physical probing and adapting

TABLE VII
SUMMARY OF POWER SIDE-CHANNEL ATTACKS TARGETING MACHINE LEARNING MODELS. IT INCLUDES DETAILS ABOUT THE PROBING INTERFACE (SOFTWARE (SW), REMOTE (RM), AND PHYSICAL (PH)), TARGET PLATFORM, TARGETED MODEL TYPE, AND THE EXTRACTED MODEL.

Reference	Probing Interface	Target Platform	Target Model / Information	Preprocessing Methods	Feature Extraction Method	Attack Model
Gao et al. [89]	SW	CPU	DNN / Hyper-parameters	N/A	N/A	U-Net + BiLSTM
Ryu et al. [132]	SW	CPU	CNN / Hyper-parameters	Segmentation	Statistical	KNN Gaussian NB RF
Jha et al. [133]	SW	GPU	Compact DNN / Architecture Hyper-parameters	Average	N/A	Non-profiled Other
Arefin et al. [134]	SW	GPU	CNN / Model type	N/A	Statistical Spectral	RF XGBoost Light GBM
Zhang et al. [81]	SW	TEE	DNN / Parameters	Averaging	N/A	Non-profiled other
Moini et al. [80]	RM	FPGA	BNN / Input data	Averaging Filtering	N/A	Non-profiled other
Zhang et al. [92]	RM	FPGA	DNN / Hyper-parameters	Segmentation	tsfresh	XGBoost
Meyers et al. [93]	RM	FPGA	MLP, CNN / Folding parameters Number of neurons	N/A	tsfresh	RF
Tian et al. [135]	RM	FPGA	N/A / Accelerator operations Matrix shapes	Normalization	N/A	SPA
Huegle et al. [136]	RM	FPGA	LeNet / Input data	N/A	N/A	GAN
Nešković et al. [49]	PH	Accelerator	N/A / Parameters	N/A	N/A	CPA Template attack
Wang et al. [94]	PH	ASIC	U-Net / Input data	Weighted sum Sliding window	Power matrices	GAN
Xiang et al. [57]	PH	CPU	DNN / Model type Parameters	N/A	Statistical	SVM
Wei et al. [47]	PH	FPGA	CNN / Input data	Filtering Curve fitting Alignment	N/A	Non-profiled other Template attack
Wolf et al. [137]	PH	MCU	NN, LR / Model type Architecture Hyper-parameters	N/A	N/A	KNN

them to modern architectures with more diverse and less predictable leakage behaviors.

VII. EXTRACTING MACHINE LEARNING MODELS

In this section, we survey PSC attacks that leverage variations in device power consumption to extract sensitive information from ML models, such as input data, model parameters, and model architectures. These attacks pose significant risks, especially for ML models used in resource-limited environments, such as smartphones and IoT devices. By analyzing power usage during inference and training, attackers can potentially reverse engineer input data and uncover proprietary model details. The increasing use of ML in sensitive applications underscores the need for effective countermeasures to protect against these attacks.

A. *Software-based Attacks*

DeepTheft is a software PSC attack developed to steal DNN model architectures by leveraging the RAPL interface [89]. DeepTheft employed a hybrid learning framework combining U-Net for spatial features and BiLSTM for temporal features to dissect the energy traces captured through RAPL. This allows the attacker to recover detailed layer-wise hyperparameters, such as kernel number and output feature size, through regression analysis of the computing overhead of different layers. Another software-based PSC attack that uses the RAPL interface to capture power traces on deep learning

models [81]. The attack exploited power leakage from the ReLU activation function to recover activation patterns, which are then used to extract full DNN model parameters when the victim model was executed on an Intel SGX enclave. By applying input gradients and PSC information, the authors developed a method that significantly reduces the number of queries needed for model extraction.

The "gamma-Knife" [132] attack also used Intel RAPL to measure power consumption without physical access to the hardware to perform a software-based PSC attack to extract the neural network architecture executed on CPUs. This attack was able to identify key architectural elements such as filter sizes, the number of filters, the activation functions, the weights, and the proportion of network layers with high accuracy. This attack can be categorized as a profiled attack, where the profiling step involves identifying the family of neural networks to which the target network belongs by analyzing the power consumption patterns. The subsequent architecture search step reduces the candidate space by leveraging statistical features of power consumption data.

"DeepPeep" proposed a black-box, two-stage PSC attack that infers the architectural building blocks of compact DNNs on cloud GPUs by correlating power footprints across batch sizes [133]. Using nvidia-smi for power telemetry, the attacker first performs inter-group identification and then intra-group refinement to pinpoint modules such as fire blocks, depthwise separable convolutions, channel shuffle, dense blocks, and inception cells. The threat model assumes an untrusted multi-tenant GPU setting where only coarse runtime metrics and power/energy readings are observable, yet the modular design of compact DNNs makes their topology distinguishable. Experiments on Tesla P100 and Quadro P4000 GPUs showed that characteristic trends in metrics and kernel utilization generalize across hardware, enabling architecture recovery without physical access.

Arefin and Serwadda analyzed another software-based PSC attack on GPUs that also leveraged the built-in nvidia-smi interface (with no additional privileges) to log power draw and classify CNN architectures [134]. The attacker collected high-rate power traces from both on-premise and cloud GPUs and demonstrated that distinctive draw patterns reveal model families and variants, highlighting a practical avenue for architecture inference in multi-tenant environments. Overall, this work further clarifies the sources of DNN power footprints and suggests countermeasures such as restricting or rate-limiting access to nvidia-smi and injecting power noise in virtualized cloud environments where feasible.

B. *Physical Attacks*

Nešković et al. demonstrated that a SystemC model can be used to simulate PSC attacks on AI accelerators at the electronic system level (ESL) [49]. The authors proposed a dynamic power consumption model that accounts for switching activities during multiplication operation, addition operations, and during register access. They employed CPA and template attacks using the proposed power model to evaluate the security of AI accelerators. These simulations revealed that both CPA and template attacks can successfully extract secret parameters from AI models by analyzing power traces generated during the inference phase.

Xiang et al. demonstrated a PSC attack that dissects the structure and parameters of DNN models [57]. By analyzing power traces collected during the inference phase, the authors were able to identify different DNN models, including AlexNet, VGG, and ResNet, implemented on embedded platforms like the Raspberry Pi. The study employed power models to understand the power consumption patterns of different DNN layers, including convolutional layers, pooling layers, fully connected layers, and activation functions. The study used a machine learning classifier to analyze power features, achieving high accuracy in distinguishing model architectures and estimating parameter sparsity.

PSC analysis was used to differentiate and extract information from ML models running on MCUs [137]. Using the ChipWhisperer-Lite, the authors captured power traces during the training of neural networks and linear regression models. The analysis involved using KNN models with dynamic time warping (DTW) and traditional Euclidean distance (ED) metrics to classify power traces. The study demonstrated that power traces can reveal differences between neural network and linear regression models, between neural networks with different hidden layer sizes, and between two nearly identical linear regression models with different hyperparameters.

Wei et al. proposed an attack where both passive and active adversaries can recover input images processed by a CNN accelerator [47]. Passive adversaries eavesdrop on power consumption during the inference stage, while active adversaries build a power template to profile the relationship between power consumption and input pixels. This attack used cycle-accurate power traces using oscilloscope measurements.

Wang et al. introduced PowerGAN, a PSC attack targeting analog compute-in-memory (CIM) accelerators that aims to recover sensitive user inputs [94]. By capturing power traces from the first layer of DNN inference and feeding them into a conditional GAN, the framework successfully reconstructs high-quality input data. The study emphasizes that CIM, while promising for overcoming the memory bottleneck in digital architectures, inherits unique leakage properties that make it vulnerable to side-channel attacks. Moreover, the evaluation demonstrated that the PowerGAN method remains effective even under noisy measurement conditions.

C. *Remote Attacks*

Moini et al. presented a detailed remote PSC attack on Binarized Neural Network (BNN) accelerators running on various FPGA platforms, including those provided by cloud services like AWS F1 instances and different Xilinx FPGAs [80]. The authors successfully detected and exploited power consumption variations by using TDC-based on-chip voltage sensors to extract information about the input data processed by the BNN. The first kernel of the first convolution layer for an input image is used as the attack point. Experimental results indicate that the processing of foreground pixels shows higher dynamic power consumption compared to background pixels.

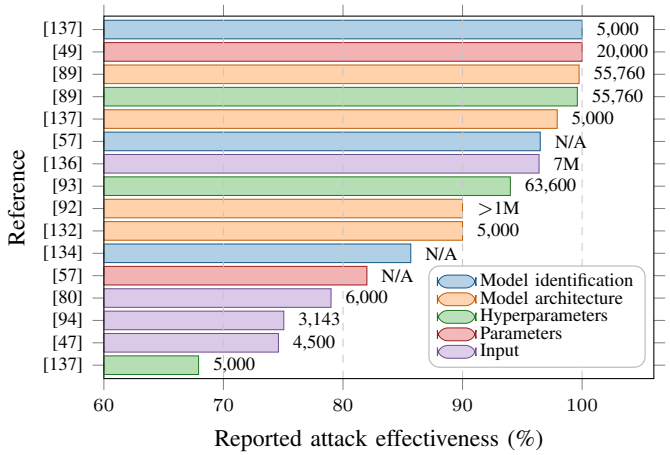

Fig. 18. Reported effectiveness of PSC attacks on ML systems across different extraction objectives. Each bar represents a attack result ranked by the primary success metric reported in the corresponding paper. Colors indicate the target objective, while annotations show the reported trace or measurement budget.

Zhang et al. performed a remote PSC attack on DNN models executed on cloud FPGA instances [92]. The attack captured voltage fluctuations to infer the structure of DNN models utilizing RO-based power sensors. The attack flow involved segmenting power traces to identify layer boundaries, classifying the type of each layer (e.g., convolutional, fully-connected, and pooling), and then predicting the hyper-parameters, such as the number of neurons, filter sizes, and strides. The evaluation of models like MLP, AlexNet, and VGG16 demonstrated that the attack achieves high accuracy in reconstructing the network architectures. Similar remote PSC attack on folded neural network implemented on FPGAs was proposed in [93]. The authors used TDCs to capture voltage fluctuations during neural network inference. By employing a profile-based attack, they first recovered the folding parameters, which denote the degree of parallelism and sequential operations in the neural network layers. This information was then used to recover the number of neurons in each layer, achieving significantly higher accuracy compared to prior works. This approach was validated on different neural networks, including MLP and CNN models, under various environmental conditions.

Tian et al. demonstrated a power trace collection method for practical remote PSC attacks on machine learning accelerators in cloud FPGAs [135]. By implementing compressed TDC RTL kernels within the cloud FPGA, the authors captured power traces to create detailed profiles of different ML operations in a realistic environment. Additionally, authors introduced an auto-triggered mechanism that enables power trace collection without requiring hard-wired communication between the attacker and the victim.

Huegle et al. proposed Power2Picture, a GAN-based framework designed to extract sensitive user inputs from DNN inference implementations on FPGAs [136]. The attack leverages on-chip TDC sensors to capture power traces during inference and then employs a generative adversarial network to reconstruct the corresponding input data. Their results show that even in resource-constrained FPGA environments, power side-channel information can be effectively transformed into input-level leakage through a learning-based approach.

D. *Summary of Attacks and Insights*

Table VII summarizes the growing body of work on PSC attacks targeting machine learning models across software, remote, and physical probing interfaces. These studies highlight how power leakage can be exploited to recover diverse forms of sensitive information, ranging from model architectures, hyperparameters, and parameters to input data. A clear trend is the use of profiled analysis pipelines, where preprocessing and feature extraction methods are coupled with statistical or learning-based classifiers, enabling high accuracy in model reconstruction. However, non-profiled attacks show that even coarse-grained telemetry, such as RAPL counters or on-chip sensors, can still yield exploitable leakage.

The rapid advancement of ML has also contributed to the evolution of PSC attacks by both enriching profiled analysis pipelines and expanding the range of viable attack targets. On the analysis side, modern ML techniques provide highly expressive models capable of learning complex leakage patterns from large trace sets, enabling profiled attacks to reconstruct architectural details, layer dimensions, and activation behaviors with greater accuracy and resilience than traditional statistical methods. At the same time, the increasing adoption of ML across domains such as healthcare, finance, autonomous systems, and personalized services has amplified the implications of PSC leakage, since extracting model parameters or input data from these systems can directly compromise privacy, intellectual property, financial assets, and safety-critical decision processes.

Another key observation is that software-based and remote PSC attacks have become the most prevalent, primarily due to widely available power-telemetry interfaces, including RAPL, NVML, and various on-chip sensors, on modern ML-capable platforms. The accessibility and scalability of these interfaces make such attacks significantly easier to mount than physical probing. However, this balance is likely to shift as edge-AI devices and ML-enabled IoT platforms become more widespread, creating new opportunities for physical PSC attacks on embedded accelerators and inference engines.

Figure 18 depicts the reported effectiveness of PSC attacks targeting ML models, including model type identification, architecture recovery, hyperparameter extraction, parameter recovery, and input reconstruction. The results show that several attacks achieve high effectiveness, often above 90%, for model-level and architecture-level recovery tasks, indicating that ML execution exposes distinguishable power signatures at coarse structural granularity. Input-recovery attacks also demonstrate practical leakage, but their reported effectiveness is lower or requires larger trace budgets, reflecting the higher difficulty of reconstructing fine-grained private data.

The surveyed works further demonstrate that the attack surface spans the full spectrum of ML deployment platforms, including CPUs, GPUs, FPGAs, ASICs, and TEEs, each exposing unique leakage properties influenced by hardware primitives, execution environments, and model implementations. Overall, Sections IV to VII illustrate that PSC attacks are not confined to cryptographic workloads but also extend to diverse applications, including user data exploitation, program

instruction disassembly, and ML model extraction.

VIII. Power Side-Channel Countermeasures

PSC attacks pose a significant threat to user privacy and confidentiality by exploiting their power consumption patterns to extract sensitive information. To mitigate these attacks, various countermeasures have been proposed in the literature. Figure 19 illustrates the three main countermeasure categories: design-specific and circuit-level (generic) countermeasures. Design-specific countermeasures can be further divided into two levels: logic level and algorithmic level. The basic idea is to decrease the Signal to Noise Ratio (SNR) by obscuring the power consumption patterns or reducing the correlation between the power consumption and the secret data being processed. This is typically achieved using two key techniques: noise injection and sensitive power signature flattening.

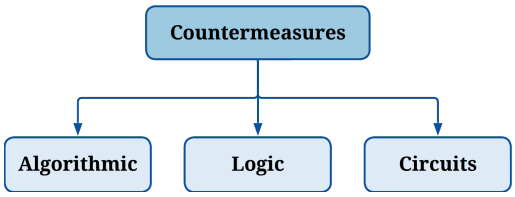

Fig. 19. Overview of countermeasures to defend against PSC attacks.

Noise injection involves adding random or structured noise to the power consumption signal to mask the variations caused by targeted operations, thereby making it more difficult for attackers to extract useful information. There are promising techniques for noise injection, including random power consumption and voltage noise generation. Random power generation utilizes additional operations to introduce noise, whereas noise is directly injected into the power supply for voltage noise generation. Sensitive power signature flattening, on the other hand, aims to flatten the power consumption profile by ensuring that the power consumption remains constant or minimally variable regardless of the data being processed. There are popular approaches for power signature flattening, such as masking, hiding, and dual-rail logic. Masking [138, 139] combines data with random values to hide its true value. Hiding [108, 140, 141] makes some changes to equalize power consumption across different operations. Dual-rail logic can balance the power consumption of each bit operation, and constant time execution ensures that operations take the same amount of time, regardless of the input data. By employing these techniques across different levels of design, it is possible to significantly enhance the resilience of cryptographic devices against PSC attacks.

Figure 20 categorizes various countermeasures in terms of design abstraction levels. We first survey design-specific countermeasures that are applicable on logic and algorithmic levels. Next, we discuss generic countermeasures that are suitable for circuit-level implementation.

A. Algorithm-Specific Countermeasures

Design-specific masking can be implemented at both the algorithmic and logic levels. Algorithmic-level masking has

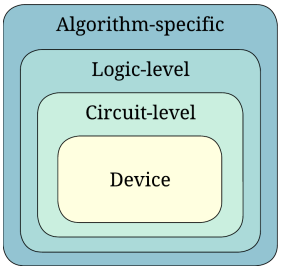

Fig. 20. Abstraction levels of countermeasures against PSC attacks. Algorithmic- and logic-level countermeasures are design-specific, targeting particular architectures or implementations, whereas circuit-level countermeasures are generic techniques applied at the transistor level.

the advantage of being easy to implement but incurs high latency for higher-order masking. The basic principle of masking is to randomly divide the secret data into multiple shares, which breaks the link between sensitive data and intermediate operations in cryptographic algorithm. If the number of shares is d+1, then the masking scheme theoretically defends against d-order side-channel attacks.

A hardware-aware algorithmic level masking was proposed for Chaskey, Simon, and AES, while improving power, performance, and area (PPA) [142]. The proposed masking method is well-suited for embedded and IoT domains, which are used in resource-constrained environments. To achieve this improved PPA, the authors implemented a security-aware hardware design with a minimal set of instructions and a simple design inspired by the SubRISC+ embedded processor architecture.

Hiding is another widely used countermeasure against PSC attacks. The goal of hiding is to obscure the sensitive data-dependent power signature with another signature. Methods like noise addition are well-known options for creating additional power signatures. Benhadjyoussef et al. used such a hiding countermeasure against CPA on an FPGA-based AES implementation [34]. Their attack target was the final round S-box values of AES, and authors employed an additional parallel circuit consisting of AddRoundKey and SubBytes stages with a random key. The power signature from these new stages were able to hide the power signature from the concurrent AddRoundKey and SubBytes stages.

Aysu et al. proposed two novel countermeasures againt SPA and DPA attacks on B-RLWE hardware implementation [32]. For SPA, the authors have introduced the "always-add-with-redundant-store" method, which normalizes power consumption by always performing additions and memory updates, regardless of the binary key value. For DPA, they implemented random initialization and masked threshold decoder. This method begins each decryption process by setting the intermediate sum to a random value, and the final step utilizes a masked threshold decoder to recover the impact of this random initialization. This significantly increased the number of measurements required for a successful DPA attack.

AIAShield [143] defense was designed to protect FPGA-based AI accelerators from PSC attacks. Adversarial attack techniques from machine learning were leveraged to generate precise noise that obfuscated unintended power consumption patterns. This approach operated in two phases. In the offline phase, a surrogate attack model was trained, and the required adversarial noise was calculated. In the online phase, the

adversarial noise was injected using ring oscillators (ROs) into the model's execution to obfuscate side-channel traces.

There are several PSC attack detection mechanisms proposed in the literature. These detection mechanisms help users take protective actions against attacks, which countermeasures alone cannot provide. Munny and Hu used a battery impedance monitoring system to detect adversarial probing [35]. Based on real-world experimental results, this detection method has a detection time of around 22 ms, which is sufficient to prevent a successful CPA-based attack.

When it comes to FPGA-based PSC attack countermeasures, some techniques used in the ASIC domain are ineffective due to the transistor-level inaccessibility in FPGAs. Therefore, new methods should be developed to overcome these challenges and defend against PSC attacks on FPGAs. As discussed earlier, FPGA-based remote PSC attacks use power monitors to collect data-dependent power traces in a shared resource environment. Recent works show that designers can use the same technique to monitor their own power fluctuations, and if information leakage is detected, they can enable an obfuscation circuit in the design. Most of the time, this obfuscation circuit is a noise generation circuit. To monitor power fluctuations, Glamočanin et al. used a TDC-based power sensor [144], while Ahmadi et al. used an RO-based power monitor [103]. TDC-based sensors offer high accuracy and high resolution at the cost of high area overhead. In contrast, RO-based sensors provide a lightweight power sensor solution to defend against remote PSC attacks on FPGAs.

Deviating from traditional methods, a combination of TDC and RO was used to implement a countermeasure against multi-tenant FPGA-based remote PSC attacks [145]. The proposed countermeasure works as an active fence where the fence is made of ROs and the activation sensor is made of TDC. These fences are strategically placed around the victim cryptographic module to inject controlled noise into the power grid, thereby obfuscating the voltage fluctuations caused by the cryptographic operations. The countermeasure aims to equalize the power consumption seen by potential adversaries, significantly reducing the signal-to-noise ratio and making side-channel attacks much more difficult.

Wang et al. suggested algorithmic-level measures such as disabling Turbo Boost or hardware P-state management, which force the CPU to operate at fixed base frequencies, thereby eliminating DVFS-induced leakage. While effective, these countermeasures introduce significant performance penalties and may not hold under custom power-limit configurations. For cryptographic libraries like SIKE, additional defenses include input validation to prevent attacker-crafted ciphertexts from triggering anomalous zero values, as deployed by Cloudflare and Microsoft after disclosure [58].

The PLATYPUS attack [50] highlighted how Intel's RAPL interface can be abused for software-based PSC attacks and stresses the need for both system-level and algorithmic defenses. Lipp et al. recommended restricting or removing unprivileged access to RAPL counters, lowering the resolution of energy measurements, and preventing precise execution control primitives such as zero-stepping in SGX. They also note that traditional software hardening techniques like masking or constant-time coding can reduce but not fully eliminate leakage when privileged access is possible. Intel's mitigation included microcode and driver updates that degrade RAPL fidelity and enforce stricter access controls, though residual risks remain for enclave security.

B. Logic-Level Countermeasures

A logic-level countermeasure for static PSC attacks was proposed in [42]. They proposed a technique called Borrowed Time, an on-chip countermeasure that detects stopped clock signals. This countermeasure eliminates the main assumption in most static PSC attacks, which is the ability to stop the clock over a reasonable period. Results showed that the proposed countermeasure has been able to increase the measurement to disclosure (MTD).

Padmini and Ravindra proposed CALPAN, a logic-level countermeasure to protect cryptographic hardware against PSC attacks [146]. The approach enhances Delay-based Dual-Rail Precharge Logic (DDPL) by introducing two techniques: (i) sleep transistors to manage static leakage currents, and (ii) normalization to balance the leakage across input patterns. While the sleep transistor method increases overall power consumption, it reduces the data dependency of leakage by about 20%. The normalized DDPL technique achieves superior results, lowering power consumption by 30% and reducing the coefficient of deviation (CoD) by 89.4%, making leakage traces nearly data-independent. These results demonstrate that CALPAN significantly strengthens resistance against PSC without prohibitive design overheads [146].

Sengupta et al. evaluated the security of logic locking (LL) techniques against PSC attacks and proposed provably secure strategies that resist both SAT and DPA attacks [147]. It reveals that conventional LL methods like Random Logic Locking (RLL) and Fault-based Logic Locking (FLL) are vulnerable, but resilience improves through key-aliasing, where multiple incorrect keys produce indistinguishable leakage, hampering DPA effectiveness. The authors formally established that LL schemes resilient to SAT attacks are inherently resistant to DPA as well, validating this with experiments on the Stripped Functionality Logic Locking (SFLL) approach. By concentrating key gates within logic cones and leveraging aliasing effects, SFLL achieves strong provable resistance to power analysis while still mitigating piracy and overbuilding threats.

A Standard-Cell Delay-based Dual-Rail Precharge Logic (SC-DDPL) was proposed in [39] as a logic-level countermeasure against static power side-channel attacks. SC-DDPL builds on the Time Enclosed Logic (TEL) protocol, which encodes data in the relative timing of dual-rail transitions rather than their steady-state values, thereby eliminating exploitable static power leakage when the clock is halted [39]. The authors implemented SC-DDPL on an FPGA with a PRESENT crypto-core and show that it achieves superior security metrics, including a lower SNR ratio, reduced mutual information, and no exploitable leakage under t-test evaluation.

C. Circuit-Level Countermeasures

Circuit-level noise injection is used as a generic, application-independent countermeasure. Prior works have de-

veloped noise injection designs to suppress data-dependent power signatures by embedding the targeted application hardware in a signature attenuating hardware (SAH) [148, 149]. Das and Sen proposed a similar approach, but specifically targeting the current [90]. Their method uses a stable constant current source to maintain a supply current independent of the targeted application current consumption.

Parrilla et al. introduced a novel countermeasure against SPA attacks for FPGA-based IoT devices with the Xored High Consumption Module (XHCM) [140]. XHCM operates by generating controlled power consumption peaks with varying amplitudes, which effectively obscure the power consumption patterns of the target device. The authors developed a Multi-Level XHCM (ML-XHCM) that allows generating different power consumption levels with minimal area overhead, enhancing the security of cryptographic operations on FPGAs by adding crafted power noise in both the amplitude and time domains. Jevtic et al. presented a generic circuit-level hiding method that involves modulating the power supply voltage to obscure the power consumption patterns exploited by attackers [107]. By dynamically varying the power supply, the correlation between power consumption and targeted operations is reduced, making it significantly harder for attackers to perform DPA and CPA attacks.

Mozipo and Acken investigated three generic circuit-level countermeasures to reduce the efficacy of remote and local PSC attacks [150]. Integrated voltage regulators (IVRs) are shown to be effective in scrambling the power traces in local attacks by introducing non-linear transformations that obscure the correlation between the IC's internal current and the external measurements. However, IVRs are less effective against remote attacks. On-package decoupling capacitors (OPDs), while providing some level of filtering, only marginally increase the difficulty of remote attacks. Voltage noise injection, which involves introducing noise into the IC power delivery network, significantly increases the MTD by up to 37 times. The study concludes that noise injection is the most effective countermeasure against remote and local PSC attacks.

Transistors are the fundamental building blocks of any semiconductor device. Therefore, to mitigate PSC attacks, existing works focused on security at the transistor level as well. Knechtel et al. presented such a promising technology named NCFET, that aims to replace traditional MOSFETs [30]. The authors demonstrated that NCFET-based circuits exhibit greater resistance to CPA attacks due to the significant impact of negative capacitance on switching (dynamic) power. LOCK&ROLL is a mitigation technique proposed using reconfigurable devices and logic locking to obscure power consumption patterns to counter DL-based PSC attacks [151]. The proposed Symmetrical MRAM-based LUT (SyM-LUT) showed resilience to machine learning-assisted PSC attacks by achieving nearly zero power variation at the output, thereby reducing the feasibility of such attacks. These countermeasures are suitable to defend against diverse PSC attacks.

D. Summary of Countermeasures and Insights

Across the algorithmic, logic, and circuit levels, countermeasures for the PSC share a unifying objective: reducing the exploitable correlation between power consumption and sensitive data. Algorithmic and logic-level defenses predominantly rely on techniques such as masking, hiding, noise generation, and dual-rail or timing-equalized logic to suppress distinguishable leakage. These strategies often introduce performance or area overheads, yet they remain highly adaptable to diverse architectures, including block ciphers, lattice-based cryptography, and AI accelerators. Notably, emerging defenses leverage adversarial machine learning, dynamic obfuscation circuits, and hybrid sensor-driven activation mechanisms, demonstrating a shift toward adaptive and context-aware security rather than static protection. There is increasing focus on threat detection because practical attacks often rely on long data-collection periods, allowing detection mechanisms to interrupt or stop malicious activity before it succeeds.

At the lower abstraction levels, circuit and transistor-level countermeasures aim to structurally reshape the power delivery and switching characteristics of hardware, yielding more fundamental suppression of leaked information. Noise injection, IVRs, on-package decoupling, and power-supply modulation have shown varying effectiveness against both local and remote attacks, with voltage noise injection consistently emerging as the most robust protection. Novel technologies such as NCFETs and reconfigurable MRAM-based LUTs highlight a trend toward hardware primitives explicitly designed with side-channel resilience in mind, rather than retrofitted security. These observations collectively suggest that future PSC-resistant systems will benefit most from multilayered designs, where adaptive algorithmic defenses, leakage-equalizing logic styles, and noise-modulating circuit techniques operate synergistically to reduce the SNR while preserving system performance and usability.

IX. Conclusion and Future Directions

Power side-channel (PSC) attacks have demonstrated their ability to extract secret information across application domains. In the late 1990s, most PSC attacks targeted the extraction of secret keys from cryptographic implementations. In recent years, there are a wide variety of PSC attacks on diverse application domains, including key extraction from cryptographic implementations, reverse engineering of machine learning models, derivation of user-specific behavior data, and identification of assembly instructions from power traces. In this survey, we have provided a comprehensive overview of these attacks and their countermeasures in different application domains. Specifically, this survey covers PSC attacks and countermeasures across a wide variety of application domains with different power models and analysis methods as well as diverse hardware implementations.

Based on a thorough analysis of recent PSC attacks across various hardware types and proximity settings, several promising directions for future research have emerged. The following discussion is derived from the insights gathered through this survey in each application domain.

A. Cryptographic Implementation

Despite decades of research on power-based PSC attacks against cryptographic implementations, several gaps remain.

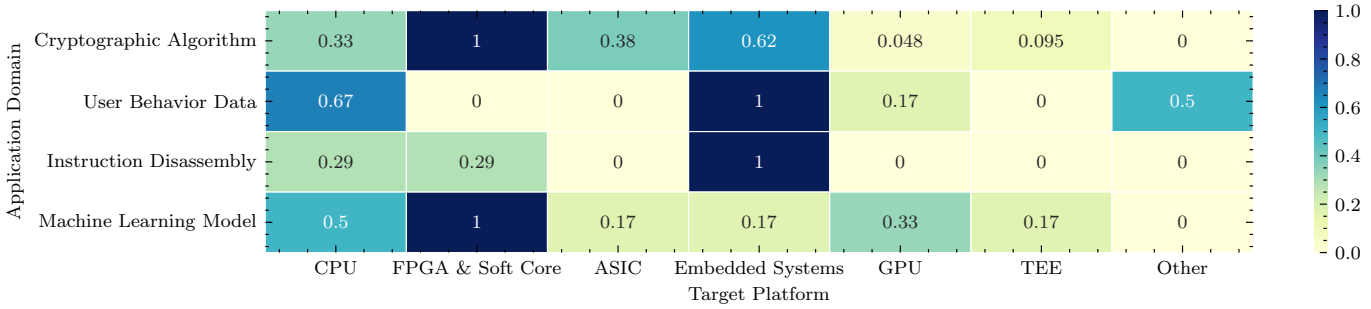

Fig. 21. Heatmap of application versus platform coverage for PSC attacks. It shows relative emphasis per application domain, highlighting platform bias within each application domain. For example, there are a lot of research efforts on *instruction disassembly* using *embedded systems*, however, there are limited efforts on *instruction disassembly* using *GPU* architectures.

As shown in Table III and Table IV, most prior works have focused on AES and RSA, while lightweight ciphers and PQC schemes have only recently been evaluated. As illustrated in Figure 22, the research trajectory has gradually shifted from classical ciphers, such as DES and AES, toward LWC designed for resource-constrained IoT devices, and more recently to PQC candidates such as CRYSTALS-Kyber, CRYSTALS-Dilithium, and Falcon. Our survey shows that even these NIST-selected PQC algorithms exhibit exploitable leakages under power analysis, raising concerns that algorithmic strength alone is insufficient without robust implementation-level countermeasures. Furthermore, there remains a significant research gap in evaluating PSC vulnerabilities in elliptic curve–based cryptographic algorithms, such as ECDH and ECDSA.

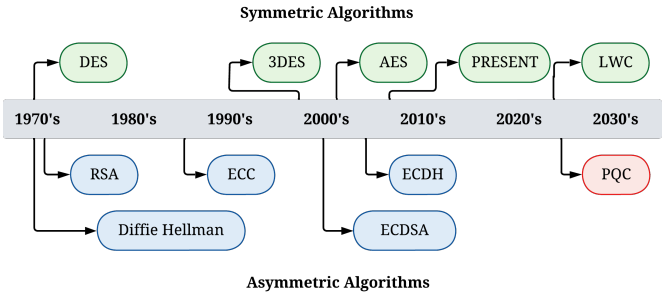

Fig. 22. Timeline of PSC attacks targeting cryptographic algorithms, showing progression from DES and AES towards lightweight cryptography (LWC) and post-quantum cryptography (PQC).

As shown in the first row of Figure 21, while FPGA and embedded platforms have been studied extensively, TEE-based cryptographic implementations remain underexplored, despite their widespread use in security-critical applications. Another gap arises in the countermeasure space: masking and hiding continue to dominate, but attackers have repeatedly demonstrated ways to circumvent these defenses (e.g., distinguishing dummy operations using machine learning). This highlights a pressing need for adaptive and composable countermeasures that can withstand evolving attack methodologies.

Looking forward, several directions are promising. First, standardized leakage assessment frameworks are needed for PQC algorithms, similar to what has been achieved for AES through TVLA, but extended to account for diverse algorithmic structures and hardware platforms. Second, there is a strong opportunity in leveraging formal security validation tools (pre-silicon and post-silicon) to proactively identify vulnerabilities before deployment. Finally, the community should explore AI-assisted adaptive countermeasures that can monitor leakage in real time and dynamically inject obfuscation, particularly relevant for shared-resource platforms.

B. User Behavior Data

PSC attacks targeting user behavior data have revealed the surprising extent to which seemingly low-level power fluctuations can expose high-level user activities such as website visits, keystroke inputs, touchscreen gestures, and even peripheral identities. As shown in Table V, most works have concentrated on profiled attacks where adversaries build detailed power-consumption signatures to achieve accurate classification of websites or applications. While these attacks demonstrate feasibility across smartphones, USB-powered devices, and wireless charging setups, they are still limited in scope. The majority of research focuses on website fingerprinting, leaving other user interactions, such as continuous voice recognition, health-monitoring wearables, or mixed-reality platforms, largely unexplored. Moreover, most of the evaluated setups assume stable laboratory conditions, whereas real-world scenarios involve far noisier environments with heterogeneous devices, background activity, and dynamic charging states.

Future directions call for broadening the attack surface as well as advancing defensive design. First, new studies should evaluate PSC leakage across emerging user-centric platforms, including augmented/virtual reality devices, voice assistants, and biometric sensors, to understand the extent of private data exposure. Second, researchers should explore cross-device generalization, where a profile built on one smartphone or wearable can be reliably transferred to another, as this would directly impact the scalability of real-world adversaries. Finally, current countermeasures, such as power noise injection or charging obfuscation, remain reactive. There is strong potential for adaptive and context-aware defenses that dynamically detect abnormal leakage patterns and mitigate them in real time, providing stronger user privacy guarantees in everyday environments.

C. Instruction Disassembly

PSC-based instruction disassembly attacks highlight how power variations can leak fine-grained program execution de-

tails, enabling recovery of opcodes, operands, and control flow. Table VI shows that most existing works rely on profiled machine learning or deep learning classifiers to map power traces to instruction classes, achieving high accuracy across MCUs, ARM processors, and FPGA soft-cores. However, these studies typically focus on isolated or controlled environments, where synchronization and profiling are feasible. Two notable gaps emerge from these studies. First, the resilience of these attacks in realistic, noisy, multi-threaded, or obfuscated execution environments remains largely unexplored. Second, as shown in the third row of Figure 21, there is limited research on ASIC-based implementations. Additionally, there is an opportunity to develop proactive design-time evaluation frameworks that integrate leakage assessment into the hardware/software co-design cycle, enabling countermeasures before deployment.

D. Machine Learning Applications

PSC attacks on ML models demonstrate that power consumption can reveal model parameters, architectures, and even user input data. As presented in Table VII, our survey highlights that most attacks have focused on extracting neural network structures and hyperparameters from CPUs, FPGAs, and embedded platforms, often using profiled frameworks to classify power traces. While these works prove feasibility, they also expose critical gaps. First, current efforts primarily target inference-time leakage, whereas training-phase vulnerabilities, which could reveal optimizer states or gradient updates remain underexplored. Second, as depicted in the fourth row in Figure 21, the majority of evaluated platforms are CPUs and FPGAs, leaving GPUs, edge TPUs, and custom accelerators insufficiently studied despite their widespread deployment in modern AI pipelines. Third, existing countermeasures are often borrowed from cryptography and lack adaptation to ML workloads, where model dynamics and data-dependent sparsity create unique leakage patterns.

Future research should broaden the attack surface to include training-time operations and diverse accelerator architectures, while also developing ML-specific leakage models that capture sparsity, pruning, and quantization effects. There is a strong potential in creating cross-platform attack frameworks capable of generalizing across hardware families, which would mirror realistic adversarial capabilities in cloud and federated learning environments. On the defense side, adaptive countermeasures that combine hardware-level obfuscation with algorithm-aware noise injection or randomized execution strategies can provide robust protection. Finally, proactive assessment tools that integrate side-channel leakage analysis into the ML accelerator design flow are needed to ensure secure deployment of AI in sensitive domains such as healthcare, finance, and defense.

E. Conclusion

This survey systematically organized power side-channel attacks using a taxonomy that spans power models, analysis methods, targeted applications, probing interface, target platform, and countermeasures. Our study shows that while classical power models, such as Hamming weight and Hamming distance, remain dominant, customized models are emerging for specialized domains, such as lightweight cryptography and machine learning. Similarly, analysis methods have evolved from simple and differential power analysis toward machine learning and deep learning–based profiled attacks, significantly broadening the attack surface. At the application level, PSC attacks have advanced from classical cryptographic key extraction to user behavior profiling, instruction disassembly, and model extraction in AI accelerators. Finally, countermeasures exist across algorithmic, logic, and circuit levels, yet our taxonomy reveals a persistent gap between proposed defenses and the adaptability of evolving attacks.

Building on this taxonomy, several future research directions are evident. First, cross-platform analysis across CPUs, GPUs, FPGAs, and heterogeneous accelerators must be prioritized to understand leakage transferability. Second, standardized leakage assessment frameworks are needed for post-quantum cryptographic algorithms and machine learning workloads. Third, adaptive runtime countermeasures that dynamically respond to leakage conditions offer a promising alternative to static design-time protections.

While our survey has focused on classical and emerging application domains, it is important to recognize that quantum computing may also become a prominent target for PSC attacks in the future [152, 153]. Early indications suggest that qubit state preparation, measurement operations, and error correction cycles could produce exploitable power or thermal signatures. Although no systematic studies currently exist, extending PSC research toward quantum devices will be critical to ensure that platforms envisioned to secure the post-quantum era are themselves robust against physical leakage.

REFERENCES

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

[2] A. Johnson, "Side channel attacks and mitigations 2015-2020: A taxonomy of published work," in *European Conference on Cyber Warfare and Security*, 2021, pp. 482–XII.

[3] O.-X. Standaert, E. Peeters, G. Rouvroy, and J.-J. Quisquater, "An overview of power analysis attacks against field programmable gate arrays," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 383–394, 2006.

[4] L. Zhang, L. Vega, and M. Taylor, "Power side channels in security ics: hardware countermeasures," *arXiv preprint arXiv:1605.00681*, 2016.

[5] L. Yan, Y. Guo, X. Chen, and H. Mei, "A study on power side channels on mobile devices," in *ACM International Conference Proceeding Series*, 2015, pp. 30–38.

[6] C. Liptak, S. Mal-Sarkar, and S. A. Kumar, "Power analysis side channel attacks and countermeasures for the internet of things," in *IEEE Physical Assurance and Inspection of Electronics, PAINE*, 2022.

[7] A. Hasnain, Y. Asfia, and S. G. Khawaja, "Power profiling-based side-channel attacks on fpga and countermeasures: A survey," in *2022 2nd International Conference on Digital Futures and Transformative Technologies, ICoDT2 2022*, 2022.

[8] M. Taouil, A. Aljuffri, and S. Hamdioui, "Power side channel attacks: Where are we standing?" in *Proceedings - 2021 16th International Conference on Design and Technology of Integrated Systems in Nanoscale Era, DTIS 2021*, 2021.

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

[10] A. Hasnain, Y. Asfia, and S. G. Khawaja, "Role of machine learning in power analysis based side channel attacks on fpga," in *International Conference on Robotics and Automation in Industry (ICRAI)*, 2023.

[11] M. C. Martínez-Rodríguez, I. M. Delgado-Lozano, and B. B. Brumley, "Sok: Remote power analysis," in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, 2021, pp. 1–12.

[12] P. Socha, V. Miškovskỳ, and M. Novotnỳ, "A comprehensive survey on the non-invasive passive side-channel analysis," *Sensors*, vol. 22, no. 21, p. 8096, 2022.

[13] P. Ravi, A. Chattopadhyay, J. P. D'Anvers, and A. Baksi, "Side-channel and fault-injection attacks over lattice-based post-quantum schemes (kyber, dilithium): Survey and new results," *ACM Transactions on Embedded Computing Systems*, vol. 23, no. 2, pp. 1–54, 2024.

[14] M. Devi and A. Majumder, "Side-channel attack in internet of things: A survey," in *Applications of Internet of Things: Proceedings of ICCCIOT 2020*. Springer, 2021, pp. 213–222.

[15] S. S. Mirzargar and M. Stojilovic, "Physical side-channel attacks and covert communication on fpgas: A survey," in *Proceedings - 29th International Conference on Field-Programmable Logic and Applications, FPL 2019*, 9 2019, pp. 202–210.

[16] T. Kim and Y. Shin, "Gpu side-channel attacks are everywhere: A survey," in *2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, 2020, pp. 1–4.

[17] R. Spreitzer, V. Moonsamy, T. Korak, and S. Mangard, "Systematic classification of side-channel attacks: A case study for mobile devices," *IEEE communications surveys & tutorials*, vol. 20, no. 1, 2017.

[18] S. R. Shanmugham and S. Paramasivam, "Survey on power analysis attacks and its impact on intelligent sensor networks," *IET Wireless Sensor Systems*, vol. 8, no. 6, pp. 295–304, 2018.

[19] A. Ghimire, V. V. Baligodugula, and F. Amsaad, "Power analysis side-channel attacks on same and cross-device settings: A survey of machine learning techniques," in *IFIP International Internet of Things Conference*. Springer, 2023, pp. 357–367.

[20] B. Hettwer, S. Gehrer, and T. Güneysu, "Applications of machine learning techniques in side-channel attacks: a survey," *Journal of Cryptographic Engineering*, vol. 10, no. 2, pp. 135–162, 2020.

[21] S. Picek, G. Perin, L. Mariot, L. Wu, and L. Batina, "Sok: Deep learning-based physical side-channel analysis," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–35, 2023.

[22] M. Panoff, H. Yu, H. Shan, and Y. Jin, "A review and comparison of ai-enhanced side channel analysis," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 18, no. 3, 2022.

[23] R. Maheswari and M. Krishnamurthy, "Profiling and non-profiling key retrieval attacks in programmable object interfaces using deep learning cryptanalytic techniques: A survey," in *International Conference on Device Intelligence, Computing and Communication Technologies*, 2024, pp. 1–5.

[24] H. Chabanne, J.-L. Danger, L. Guiga, and U. Kühne, "Side channel attacks for architecture extraction of neural networks," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 1, pp. 3–16, 2021.

[25] M. Méndez Real and R. Salvador, "Physical side-channel attacks on embedded neural networks: A survey," *Applied Sciences*, vol. 11, no. 15, p. 6790, 2021.

[26] Y. Han, I. Christoudis, K. I. Diamantaras, S. Zonouz, and A. Petropulu, "Side-channel-based code-execution monitoring systems: a survey," *IEEE Signal Processing Magazine*, vol. 36, no. 2, pp. 22–35, 2019.

[27] M. He, J. Park, A. Nahiyan, A. Vassilev, Y. Jin, and M. Tehranipoor, "Rtl-psc: Automated power side-channel leakage assessment at register-transfer level," in *2019 IEEE 37th VLSI Test Symposium (VTS)*. IEEE, 2019, pp. 1–6.

[28] N. Pundir, J. Park, F. Farahmandi, and M. Tehranipoor, "Power side-channel leakage assessment framework at register-transfer level," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2022.

[29] A. Jayasena, E. Andrews, and P. Mishra, "Tvla*: Test vector leakage assessment on hardware implementations of asymmetric cryptography algorithms," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2023.

[30] J. Knechtel, S. Patnaik, M. Nabeel, M. Ashraf, Y. S. Chauhan, J. Henkel, O. Sinanoglu, and H. Amrouch, "Power side-channel attacks in negative capacitance transistor," *IEEE Micro*, vol. 40, 11 2020.

[31] C. Liu, A. Chakraborty, N. Chawla, and N. Roggel, "Frequency throttling side-channel attack," in *ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 1977–1991.

[32] A. Aysu, M. Orshansky, and M. Tiwari, "Binary ring-lwe hardware with power side-channel countermeasures," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 1253–1258.

[33] C. Luo, Y. Fei, P. Luo, S. Mukherjee, and D. Kaeli, "Side-channel power analysis of a gpu aes implementation," in *2015 33rd IEEE International Conference on Computer Design (ICCD)*. IEEE, 2015.

[34] N. Benhadjyoussef, M. Karmani, and M. Machhout, "Power-based side channel analysis and fault injection: Hacking techniques and combined countermeasure," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, 2021.

[35] R. Munny and J. Hu, "Power side-channel attack detection through battery impedance monitoring," in *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2021-May, 2021.

[36] V. Iyer, M. Wang, J. Kulkarni, and A. E. Yilmaz, "A systematic evaluation of em and power side-channel analysis attacks on aes implementations," in *Proceedings - 2021 IEEE International Conference on Intelligence and Security Informatics, ISI 2021*, 2021.

[37] T. S. Messerges, "Using second-order power analysis to attack dpa resistant software," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2000, pp. 238–251.

[38] M. Tehranipoor and C. Wang, *Introduction to hardware security and trust*. Springer Science & Business Media, 2011.

[39] D. Bellizia, R. D. Sala, and G. Scotti, "Sc-ddpl as a countermeasure against static power side-channel attacks," *Cryptography*, vol. 5, 2021.

[40] 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, pp. 376–389, 2 2020.

[41] N. Benhadjyoussef, M. Karmani, and M. Machhout, "Power-based side-channel analysis against aes implementations: Evaluation and comparison," *IJCSNS International Journal of Computer Science and Network Security*, vol. 21, 2021.

[42] R. Dumitru, A. Wabnitz, and Y. Yarom, "On borrowed time–preventing static power side-channel analysis," *preprint arXiv:2307.09001*, 2023.

[43] T. Moos, A. Moradi, and B. Richter, "Static power side-channel analysis of a threshold implementation prototype chip," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017.

[44] S. Vafa, M. Masoumi, and A. Amini, "An efficient profiling attack to real codes of pic16f690 and arm cortex-m3," *IEEE Access*, vol. 8, pp. 222 520–222 532, 2020.

[45] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2004, pp. 16–29.

[46] L. Zhang, D. Mu, W. Hu, Y. Tai, J. Blackstone, and R. Kastner, "Memory-based high-level synthesis optimizations security exploration on the power side-channel," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, 2019.

[47] L. Wei, B. Luo, Y. Li, Y. Liu, and Q. Xu, "I know what you see: Power side-channel attack on convolutional neural network accelerators," in *ACM International Conference Proceeding Series*. Association for Computing Machinery, 12 2018, pp. 393–406.

[48] A. Srivastava, S. Das, N. Choudhury, R. Psiakis, P. H. Silva, D. Pal, and K. Basu, "Scar: Power side-channel analysis at rtl level," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2024.

[49] A. Nešković, S. Mulhem, A. Treff, R. Buchty, T. Eisenbarth, and M. Berekovic, "Systemc model of power side-channel attacks against ai accelerators: Superstition or not?" in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE, 2023.

[50] M. Lipp, A. Kogler, D. Oswald, M. Schwarz, C. Easdon, C. Canella, and D. Gruss, "Platypus: Software-based power side-channel attacks on x86," in *IEEE Symposium on Security and Privacy (SP)*, 2021.

[51] E. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Power and electromagnetic analysis: Improved model, consequences and comparisons," *Integration*, vol. 40, no. 1, pp. 52–60, 2007.

[52] T.-H. Tran, B.-A. Dao, T.-T. Hoang, V.-P. Hoang, and C.-K. Pham, "Transition factors of power consumption models for cpa attacks on cryptographic risc-v soc," *IEEE Transactions on Computers*, vol. 72, no. 9, pp. 2689–2700, 2023.

[53] H. Liu, G. Qian, S. Goto, and Y. Tsunoo, "Aes key recovery based on switching distance model," in *2010 Third International Symposium on Electronic Commerce and Security*. IEEE, 2010, pp. 218–222.

[54] H. Mestiri, N. Benhadjyoussef, M. Machhout, and R. Tourki, "A comparative study of power consumption models for cpa attack," *International Journal of Computer Network and Information Security*, vol. 5, no. 3, p. 25, 2013.

[55] W. Cao, F. Huang, M. Zheng, and H. Hu, "Attacking fpga-based dual complementary aes implementation using hd and sd models," in *2020 16th International Conference on Computational Intelligence and Security (CIS)*. IEEE, 2020, pp. 278–282.

[56] W. Schindler, K. Lemke, and C. Paar, "A stochastic model for differential side channel cryptanalysis," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2005.

[57] Y. Xiang, Z. Chen, Z. Chen, Z. Fang, H. Hao, J. Chen, Y. Liu, Z. Wu, Q. Xuan, and X. Yang, "Open dnn box by power side-channel attack," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, pp. 2717–2721, 11 2020.

[58] Y. Wang, R. Paccagnella, E. T. He, H. Shacham, C. W. Fletcher, and

D. Kohlbrenner, "Hertzbleed: Turning power {Side-Channel} attacks into remote timing attacks on x86," in *USENIX Security Symposium*, 2022, pp. 679–697.

[59] A. Kogler, J. Juffinger, L. Giner, L. Gerlach, M. Schwarzl, M. Schwarz, D. Gruss, and S. Mangard, "{Collide+ Power}: Leaking inaccessible data with software-based power side channels," in *USENIX Security Symposium*, 2023, pp. 7285–7302.

[60] S. Sanjaya, A. Jayasena, and P. Mishra, "Information leakage through physical layer supply voltage coupling vulnerability," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2025.

[61] A. T. Mozipo and J. M. Acken, "Residual vulnerabilities to power side channel attacks of lightweight ciphers cryptography competition finalists," pp. 75–88, 7 2023.

[62] A. Jayasena, R. Bachmann, and P. Mishra, "Evilcs: An evaluation of information leakage through context switching on security enclaves," in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2024, pp. 1–6.

[63] A. Jayasena *et al.*, "CISELEAKS: Information leakage assessment of cryptographic instruction set extension prototypes," *IEEE Transactions on Information Forensics and Security*, 2025.

[64] B. J. Gilbert Goodwill, J. Jaffe, P. Rohatgi *et al.*, "A testing methodology for side-channel resistance validation," in *NIST non-invasive attack testing workshop*, vol. 7, 2011, pp. 115–136.

[65] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[66] Y. Miura, H. Nishikawa, X. Kong, and H. Tomiyama, "Simulation-based analysis of power side-channel leakage at different sampling intervals," in *Proceedings - 2023 11th International Symposium on Computing and Networking Workshops, CANDARW 2023*, 2023, pp. 364–365.

[67] E. Weisstein, "Bonferroni correction," *https://mathworld.wolfram.com/*, 2004.

[68] "Fips 140-3 security requirements for cryptographic modules," https://csrc.nist.gov/pubs/fips/140-3/, accessed: 2025-11-20.

[69] "The common criteria," https://www.commoncriteriaportal.org/, accessed: 2025-11-20.

[70] D. B. Roy, S. Bhasin, S. Guilley, A. Heuser, S. Patranabis, and D. Mukhopadhyay, "Cc meets fips: A hybrid test methodology for first order side channel analysis," *IEEE Transactions on Computers*, vol. 68, no. 3, pp. 347–361, 2018.

[71] K. Pratihar, N. Mishra, and D. Mukhopadhyay, "Bounded information: Pac certification of multivariate side-channel traces," *Cryptology ePrint Archive*, 2026.

[72] F.-X. Standaert, "How (not) to use welch's t-test in side-channel security evaluations," in *International conference on smart card research and advanced applications*. Springer, 2018, pp. 65–79.

[73] B. Timon, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 107–131, 2019.

[74] A. A. Ahmed, M. K. Hasan, N. S. Nafi, A. H. Aman, S. Islam, and M. S. Nahi, "Optimization technique for deep learning methodology on power side channel attacks," in *2023 33rd International Telecommunication Networks and Applications Conference, ITNAC 2023*, 2023.

[75] A. A. Ahmed, R. A. Salim, and M. K. Hasan, "Deep learning method for power side-channel analysis on chip leakages," *Elektronika ir Elektrotechnika*, vol. 29, pp. 50–57, 2023.

[76] Y. Wang, R. Paccagnella, A. Wandke, Z. Gang, G. Garrett-Grossman, C. W. Fletcher, D. Kohlbrenner, and H. Shacham, "Dvfs frequently leaks secrets: Hertzbleed attacks beyond sike, cryptography, and cpu-only data," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 2306–2320.

[77] T. Yu, C. Cheng, Z. Yang, Y. Wang, Y. Pan, and J. Weng, "Hints from hertz: Dynamic frequency scaling side-channel analysis of number theoretic transform in lattice-based kems," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2024, no. 3, pp. 200–223, 2024.

[78] H. Taneja, J. Kim, J. J. Xu, S. Van Schaik, D. Genkin, and Y. Yarom, "Hot pixels: Frequency, power, and temperature attacks on {GPUs} and arm {SoCs}," in *USENIX Security Symposium*, 2023, pp. 6275–6292.

[79] S. Sanjaya, A. Jayasena, and P. Mishra, "Sleepwalk: Exploiting context switching and residual power for physical side-channel attacks," *IEEE Transactions on Information Forensics and Security*, 2026.

[80] S. Moini, S. Tian, D. Holcomb, J. Szefer, and R. Tessier, "Power side-channel attacks on bnn accelerators in remote fpgas," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, pp. 357–370, 6 2021.

[81] X. Zhang, A. A. Ding, and Y. Fei, "Deep-learning model extraction through software-based power side-channel," in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, 2023.

[82] J. Gatlin, S. Belikovetsky, Y. Elovici, A. Skjellum, J. Lubell, P. Witherell, and M. Yampolskiy, "Encryption is futile: Reconstructing 3d-printed models using the power side-channel," in *ACM International Conference Proceeding Series*. Association for Computing Machinery, 10 2021, pp. 135–147.

[83] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *International workshop on cryptographic hardware and embedded systems*. Springer, 2002, pp. 13–28.

[84] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks: Revealing the secrets of smart cards*. Springer, 2007.

[85] A. Heuser and M. Zohner, "Intelligent machine homicide: Breaking cryptographic devices using support vector machines," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 2012, pp. 249–264.

[86] L. Lerman, G. Bontempi, and O. Markowitch, "A machine learning approach against a masked aes: Reaching the limit of side-channel attacks with a learning model," *Journal of Cryptographic Engineering*, vol. 5, no. 2, pp. 123–139, 2015.

[87] R. Gilmore, N. Hanley, and M. O'Neill, "Neural network based attack on a masked implementation of aes," in *2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2015, pp. 106–111.

[88] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *International Conference on Security, Privacy, and Applied Cryptography Engineering*. Springer, 2016, pp. 3–26.

[89] Y. Gao, H. Qiu, Z. Zhang, B. Wang, H. Ma, A. Abuadbba, M. Xue, A. Fu, and S. Nepal, "Deeptheft: Stealing dnn model architectures through power side channel," *arXiv preprint arXiv:2309.11894*, 2023.

[90] D. Das and S. Sen, "Electromagnetic and power side-channel analysis: Advanced attacks and low-overhead generic countermeasures through white-box approach," *Cryptography*, vol. 4, pp. 1–19, 12 2020.

[91] S. Ghandali, S. Ghandali, and S. Tehranipoor, "Deep k-tsvm: A novel profiled power side-channel attack on aes-128," *IEEE Access*, vol. 9, pp. 136 448–136 458, 2021.

[92] Y. Zhang, R. Yasaei, H. Chen, Z. Li, and M. A. Al Faruque, "Stealing neural network structure through remote fpga side-channel analysis," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4377–4388, 2021.

[93] V. Meyers, D. Gnad, and M. Tahoori, "Reverse engineering neural network folding with remote fpga power analysis," in *2022 IEEE 30th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2022, pp. 1–10.

[94] Z. Wang, Y. Wu, Y. Park, S. Yoo, X. Wang, J. K. Eshraghian, and W. D. Lu, "Powergan: A machine learning approach for power side-channel attack on compute-in-memory accelerators," *Advanced Intelligent Systems*, vol. 5, no. 12, p. 2300313, 2023.

[95] S. Hajra, S. Saha, M. Alam, and D. Mukhopadhyay, "Transnet: Shift invariant transformer network for side channel analysis," in *International Conference on Cryptology in Africa*. Springer, 2022, pp. 371–396.

[96] S. Hajra, S. Chowdhury, and D. Mukhopadhyay, "Estranet: An efficient shift-invariant transformer network for side-channel analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2024, no. 1, pp. 336–374, 2024.

[97] N. G. Jayasankaran, H. Guo, S. Patnaik, Jeyavijayan, Rajendran, and J. Hu, "Securing cloud fpgas against power side-channel attacks: A case study on iterative aes," 7 2023. [Online]. Available: http://arxiv.org/abs/2307.02569

[98] F. Schellenberg, D. R. Gnad, A. Moradi, and M. B. Tahoori, "An inside job: Remote power analysis attacks on fpgas," *IEEE Design & Test*, vol. 38, no. 3, pp. 58–66, 2021.

[99] D. R. Gnad, J. Krautter, M. B. Tahoori, F. Schellenberg, and A. Moradi, "Remote electrical-level security threats to multi-tenant fpgas," *IEEE Design & Test*, vol. 37, no. 2, pp. 111–119, 2020.

[100] J. Gravellier, J.-M. Dutertre, Y. Teglia, and P. Loubet-Moundi, "High-speed ring oscillator based sensors for remote side-channel attacks on fpgas," in *2019 International conference on ReConFigurable computing and FPGAs (ReConFig)*. IEEE, 2019, pp. 1–8.

[101] C. Ramesh, S. B. Patil, S. N. Dhanuskodi, G. Provelengios, S. Pillement, D. Holcomb, and R. Tessier, "Fpga side channel attacks without physical access," in *2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2018, pp. 45–52.

[102] C. O'Flynn and A. Dewar, "On-device power analysis across hard-

ware security domains.: Stop hitting yourself." *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 126–153, 2019.
[103] M. M. Ahmadi, F. Khalid, R. Vaidya, F. Kriebel, A. Steininger, and M. Shafique, "Shield: An adaptive and lightweight defense against the remote power side-channel attacks on multi-tenant fpgas," 3 2023. [Online]. Available: http://arxiv.org/abs/2303.06486
[104] T. Kamucheka, M. Fahr, T. Teague, A. Nelson, D. Andrews, and M. Huang, "Power-based side channel attack analysis on pqc algorithms," *Cryptology ePrint Archive*, 2021.
[105] M. Lipp, D. Gruss, and M. Schwarz, "{AMD} prefetch attacks through power and time," in *USENIX Security Symposium*, 2022, pp. 643–660.
[106] J. Lee and D.-G. Han, "Dlddo: deep learning to detect dummy operations," in *International Conference on Information Security Applications*. Springer, 2020, pp. 73–85.
[107] R. Jevtic, P. Perez-Tirador, C. Cabezaolias, P. Carnero, and G. Caffarena, "Side-channel attack countermeasure based on power supply modulation," in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 618–622.
[108] J. Lee and D.-G. Han, "Security analysis on dummy based side-channel countermeasures—case study: Aes with dummy and shuffling," *Applied Soft Computing*, vol. 93, p. 106352, 2020.
[109] K. Saito, A. Ito, R. Ueno, and N. Homma, "One truth prevails: A deep-learning based single-trace power analysis on rsa–crt with windowed exponentiation," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 490–526, 2022.
[110] D. Moody, "Let's get ready to rumble. the nist pqc "competition"," in *Proc. of First PQC Standardization Conference*, 2018, pp. 11–13.
[111] G. Alagic, G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, Y.-K. Liu, C. Miller, D. Moody, R. Peralta *et al.*, "Status report on the first round of the nist post-quantum cryptography standardization process," 2019.
[112] G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, J. Kelsey, Y.-K. Liu, C. Miller, D. Moody, R. Peralta *et al.*, "Status report on the second round of the nist post-quantum cryptography standardization process," *US Department of Commerce, NIST*, vol. 2, p. 69, 2020.
[113] G. Alagic, G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, Y.-K. Liu, C. Miller *et al.*, "Status report on the third round of the nist post-quantum cryptography standardization process," 2022.
[114] K. M. Zick, M. Srivastav, W. Zhang, and M. French, "Sensing nanosecond-scale voltage attacks and natural transients in fpgas," in *ACM international symposium on Field programmable gate arrays*, 2013, pp. 101–104.
[115] Z. Zhang, S. Liang, F. Yao, and X. Gao, "Red alert for power leakage: Exploiting intel rapl-induced side channels," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 2021, pp. 162–175.
[116] Y. Chen, X. Jin, J. Sun, R. Zhang, and Y. Zhang, "Powerful: Mobile app fingerprinting via power analysis," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.
[117] A. S. La Cour, K. K. Afridi, and G. E. Suh, "Wireless charging power side-channel attacks," in *ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 651–665.
[118] Q. Yang, P. Gasti, G. Zhou, A. Farajidavar, and K. S. Balagani, "On inferring browsing activity on smartphones via usb power analysis side-channel," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1056–1066, 2016.
[119] R. Matovu, A. Serwadda, A. V. Bilbao, and I. Griswold-Steiner, "Defensive charging: Mitigating power side-channel attacks on charging smartphones," in *CODASPY 2020 - Proceedings of the 10th ACM Conference on Data and Application Security and Privacy*. Association for Computing Machinery, Inc, 3 2020, pp. 179–190.
[120] P. Cronin, X. Gao, C. Yang, and H. Wang, "{Charger-Surfing}: Exploiting a power line {Side-Channel} for smartphone information leakage," in *USENIX Security Symposium*, 2021, pp. 681–698.
[121] D. Yucebas and H. Yuksel, "Power analysis based side-channel attack on visible light communication," *Physical Communication*, vol. 31, pp. 196–202, 12 2018.
[122] R. Spolaor, H. Liu, F. Turrin, M. Conti, and X. Cheng, "Plug and power: Fingerprinting usb powered peripherals via power side-channel," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.
[123] Y. Qin and C. Yue, "Website fingerprinting by power estimation based side-channel attacks on android 7," in *International Conference On Trust, Security And Privacy In Computing And Communications/International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2018, pp. 1030–1039.
[124] O. Glamočanin, S. Shrivastava, J. Yao, N. Ardo, M. Payer, and M. Stojilović, "Instruction-level power side-channel leakage evaluation of soft-core cpus on shared fpgas," *Journal of Hardware and Systems Security*, vol. 7, no. 2, pp. 72–99, 2023.
[125] D. Krishnankutty, Z. Li, R. Robucci, N. Banerjee, and C. Patel, "Instruction sequence identification and disassembly using power supply side-channel analysis," *IEEE Transactions on Computers*, vol. 69, no. 11, pp. 1639–1653, 2020.
[126] H. Fendri, M. Macchetti, J. Perrine, and M. Stojilović, "A deep-learning approach to side-channel based cpu disassembly at design time," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 670–675.
[127] Y. Han, M. Chan, Z. Aref, N. O. Tippenhauer, and S. Zonouz, "Hiding in plain sight? on the efficacy of power side Channel-Based control flow monitoring," in *USENIX Security Symposium*, 2022, pp. 661–678.
[128] J. Park, X. Xu, Y. Jin, D. Forte, and M. Tehranipoor, "Power-based side-channel instruction-level disassembler," in *Proceedings - Design Automation Conference*, vol. Part F137710, 6 2018.
[129] P. Narimani, M. A. Akhaee, and S. A. Habibi, "Side-channel based disassembler for avr micro-controllers using convolutional neural networks," in *2021 18th International ISC Conference on Information Security and Cryptology (ISCISC)*. IEEE, 2021, pp. 75–80.
[130] J. Van Geest and I. Buhan, "A side-channel based disassembler for the arm-cortex m0," in *International Conference on Applied Cryptography and Network Security*. Springer, 2022, pp. 183–199.
[131] D. Bae and J. Ha, "Implementation of disassembler on microcontroller using side-channel power consumption leakage," *Sensors*, vol. 22, no. 15, p. 5900, 2022.
[132] D. Ryu, Y. Kim, and J. Hur, "gamma-knife: Extracting neural network architecture through software-based power side-channel," *IEEE Transactions on Dependable and Secure Computing*, 2023.
[133] N. K. Jha, S. Mittal, B. Kumar, and G. Mattela, "Deeppeep: Exploiting design ramifications to decipher the architecture of compact dnns," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 17, no. 1, pp. 1–25, 2020.
[134] S. E. Arefin and A. Serwadda, "Dissecting the origins of the power side-channel in deep neural networks," in *2024 IEEE World AI IoT Congress (AIIoT)*. IEEE, 2024, pp. 414–420.
[135] S. Tian, S. Moini, D. Holcomb, R. Tessier, and J. Szefer, "A practical remote power attack on machine learning accelerators in cloud fpgas," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2023, pp. 1–6.
[136] L. Huegle, M. Gotthard, V. Meyers, J. Krautter, D. R. Gnad, and M. B. Tahoori, "Power2picture: Using generative cnns for input recovery of neural network accelerators through power side-channels on fpgas," in *2023 IEEE 31st Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2023, pp. 155–161.
[137] S. Wolf, H. Hu, R. Cooley, and M. Borowczak, "Stealing machine learning parameters via side channel power attacks," in *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI*, vol. 2021-July. IEEE Computer Society, 7 2021, pp. 242–247.
[138] M. Asfand Hafeez, M. Mazyad Hazzazi, H. Tariq, A. Aljaedi, A. Javed, and A. R. Alharbi, "A low-overhead countermeasure against differential power analysis for aes block cipher," *Applied Sciences*, vol. 11, no. 21, p. 10314, 2021.
[139] J. Bhandari, M. Nabeel, L. Mankali, O. Sinanoglu, R. Karri, and J. Knechtel, "Lightweight masking against static power side-channel attacks," *arXiv preprint arXiv:2402.03196*, 2024.
[140] L. Parrilla, A. García, E. Castillo, S. Rodríguez-Bolívar, and J. A. López-Villanueva, "Time-and amplitude-controlled power noise generator against spa attacks for fpga-based iot devices," *Journal of Low Power Electronics and Applications*, vol. 12, no. 3, p. 48, 2022.
[141] A. Alipour, A. Papadimitriou, V. Beroulle, E. Aerabi, and D. Hély, "On the performance of non-profiled differential deep learning attacks against an aes encryption algorithm protected using a correlated noise generation based hiding countermeasure," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 614–617.
[142] M. Yang, T. Ahmed, S. Inagaki, K. Sakiyama, Y. Li, and Y. Hara-Azumi, "Hardware/software cooperative design against power side-channel attacks on iot devices," *IEEE Internet of Things Journal*, 2024.
[143] X. Yan, C. H. Chang, and T. Zhang, "Defense against ml-based power side-channel attacks on dnn accelerators with adversarial attacks," *arXiv preprint arXiv:2312.04035*, 2023.
[144] O. Glamočanin, L. Coulon, F. Regazzoni, and M. Stojilović, "Are cloud fpgas really vulnerable to power analysis attacks?" in *2020 Design,*

Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2020, pp. 1007–1010.

[145] J. Krautter, D. R. Gnad, F. Schellenberg, A. Moradi, and M. B. Tahoori, "Active fences against voltage-based side channels in multi-tenant fpgas," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–8.

[146] C. Padmini and J. Ravindra, "Calpan: Countermeasure against leakage power analysis attack by normalized ddpl," in *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*. IEEE, 2016, pp. 1–7.

[147] A. Sengupta, B. Mazumdar, M. Yasin, and O. Sinanoglu, "Logic locking with provable security against power analysis attacks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 4, pp. 766–778, 2019.

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

[149] ——, "Asni: Attenuated signature noise injection for low-overhead power side-channel attack immunity," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 10, pp. 3300–3311, 2018.

[150] A. T. Mozipo and J. M. Acken, "Analysis of countermeasures against remote and local power side channel attacks using correlation power analysis," *IEEE Transactions on Dependable and Secure Computing*, 2024.

[151] G. Kolhe, T. Sheaves, K. I. Gubbi, S. Salehi, S. Rafatirad, P. D. Manoj, A. Sasan, and H. Homayoun, "Lock&roll: Deep-learning power side-channel attack mitigation using emerging reconfigurable devices and logic locking," in *Proceedings - Design Automation Conference*, 7 2022, pp. 85–90.

[152] C. Xu, F. Erata, and J. Szefer, "Exploration of power side-channel vulnerabilities in quantum computer controllers," in *ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 579–593.

[153] F. Erata, C. Xu, R. Piskac, and J. Szefer, "Quantum circuit reconstruction from power side-channel attacks on quantum computer controllers," *arXiv preprint arXiv:2401.15869*, 2024.



Sahan Sanjaya is a Ph.D. student in the Department of Computer & Information Science & Engineering at the University of Florida. In 2022, he completed his B.Sc. in the Department of Electronic and Telecommunication Engineering at the University of Moratuwa, Sri Lanka. His research interests include side-channel attacks, hardware security, pre-silicon validation, and quantum computing.



Aruna Jayasena is an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Massachusetts, Amherst. He received his Ph.D. from the University of Florida in 2025. His research focuses on systems security and secure computing architectures. His work lies at the intersection of Computer and Electrical Engineering, with a focus on heterogeneous system design, applied cryptography, trusted execution, and hardware-firmware co-validation.



Prabhat Mishra is a Professor in the Department of Computer and Information Science and Engineering at the University of Florida. He received his Ph.D. in Computer Science from the University of California at Irvine in 2004. His research interests include hardware security and trust, hardware validation and debug, and embedded systems. He currently serves as an Associate Editor of IEEE Transactions on VLSI Systems and ACM Transactions on Embedded Computing Systems. He is an IEEE Fellow, an AAAS Fellow, and an ACM Distinguished Scientist.