Information Leakage through Physical Layer Supply Voltage Coupling Vulnerability

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

***Abstract*—Power side-channel attacks are widely known for extracting information from data processed within a device while assuming that an attacker has physical access or the ability to modify the device. In this paper, we introduce a novel side-channel vulnerability that leaks data-dependent power variations through physical layer supply voltage coupling (PSVC). Unlike traditional power side-channel attacks, the proposed vulnerability allows an adversary to mount an attack and extract information without modifying the device. Additionally, unlike existing power-based remote attacks on FPGAs, the PSVC vulnerability applies to both on-chip and on-board attacks. We assess the effectiveness of the PSVC vulnerability through three case studies, demonstrating several end-to-end attacks on general-purpose microcontrollers with varying adversary capabilities. These case studies provide evidence for the existence of the PSVC vulnerability, its applicability to on-chip as well as on-board side-channel attacks, and how it can eliminate the need for physical access to the target device, making it applicable to any off-the-shelf hardware. Our experiments also reveal that designing devices to operate at the lowest operational voltage significantly reduces the risk of PSVC side-channel vulnerability.**

***Index Terms*—Side-channel vulnerability, supply voltage coupling, security, information leakage.**

I. Introduction

Side-channel leakage represents a significant security concern with the diversity of computing devices used for security-sensitive applications. An adversary can exploit various side-channels, including power consumption [1], electromagnetic (EM) emanation [2], impedance variations [3], and silicon substrate coupling [4] as well as channels created by malicious implants (e.g., hardware Trojans [5]). For example, the covert avenue for information extraction can occur when the power consumption of a device fluctuates during its operation, indirectly revealing valuable insights about the data and cryptographic processes that are being executed. In essence, an adversary can deduce the information processed by a device by closely monitoring and analyzing its power consumption characteristics.

To conduct traditional side-channel attacks, several requirements must be met. First and foremost, the adversary requires physical access to the target device, which may be a microcontroller unit (MCU), an integrated circuit (IC), an embedded system, or even a server. This physical access allows the adversary to monitor the power consumption in real-time with minor modifications to the existing hardware. However, if current sensors are already available in the device, the adversary can directly probe into the sensors. Figure 1 illustrates a typical power side-channel attack setup, where the adversary probes across the current sensor (shunt resistor R_S).

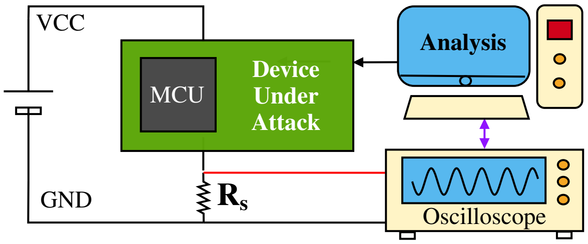


Fig. 1: Overview of a power-based side-channel attack. The current flow through the device is measured using a shunt resistor ($\mathbf{R_S}$) connected in series to the device, which translates to the power consumption of the device. The power traces provide insights into the ongoing computation in the device.

Then by manipulating the inputs to the design, the adversary is able to correlate the computation with the observed power trace of the device. Usually, the initial proof-of-concept (PoC) is constructed offline with a spare device with the same specification as the original device that the adversary is planning to attack. The final attack can be launched based on the PoC on the actual target device and the computations that happened in the device can be deduced in real-time during execution. In the multi-tenant FPGA domain, adversaries exploit voltage fluctuations from victim circuits by using power monitoring circuits such as Time-to-Digital Converters (TDCs) and Ring Oscillators (ROs). This approach enables them to extract sensitive information from a victim sharing the FPGA, with modifications to their own design. However, these attacks have been limited to on-chip multi-tenant FPGAs.

In this paper, we introduce a new side-channel that exposes data-dependent power variations through physical layer supply voltage coupling (PSVC). Unlike traditional power side-channel attacks, our method enables an adversary to extract on-chip or on-board victim information without requiring modifications to the device. We evaluate the effectiveness of the PSVC vulnerability through three case studies, each reflecting diverse adversary capabilities. These studies illustrate the presence of the PSVC vulnerability in general-purpose microcontrollers, its ability to propagate across different voltage domains, and its potential as a remote attack vector.

A. Threat Model

We assume three levels of adversary capabilities as ***level-1***, ***level-2*** and ***level-3***, as illustrated by Table I. With the *level-1* capability, the adversary is able to modify the device under attack (e.g. install a shunt resister, modify the firmware, etc.). Attack models relying on side-channel leakage via power consumption typically fall under *level-1* category. However, *level-1* category capabilities may not be practical in most scenarios. Adversaries with ***level-2*** capabilities cannot make modifications to the hardware but are able to probe into the power supply lines of the victim device. Finally, the most constrained adversary is the ***level-3***, where they don't have modification ability (*level-1*) or physical access (*level-2*). Instead in the case of ***level-3***, an adversary has access to the wireless communication of the victim device, such as Wi-Fi and Bluetooth. We assume that the adversary is situated within the wireless communication range of the victim device.

TABLE I: Adversary levels based on their ability to access the victim device. To launch a side-channel attack using PSVC, the minimum required adversary capability level is ***level-3***.

Level	**Adversary Capabilities**
1	Able to modify the victim device
2	Has physical access, cannot modify
3	Wireless access within a certain range

B. Physical Side-Channel Evaluation

There are a wide variety of methods for design-time as well as run-time side-channel analysis to reveal secrets. The goal of the design time techniques, such as test vector leakage assessment [6]–[10], is to detect potential power side-channel leaky designs at the early stages of the design cycle. A vast majority of run-time methods analyze variations in power, current, or path delay to evaluate information leakage. However, these methods are not applicable in many scenarios since they rely on physical access (e.g., to probe power lines) or even minor modifications (e.g., inserting a shunt register) of the device. There are approaches for remote attacks that exploit various side-channels, such as electromagnetic emanation [11], [12], radio frequency emissions [4], and video-based cryptanalysis [13]. However, these methods evaluate computation inside the target MCU itself or data transmissions within chip interconnects. In contrast, our proposed PSVC vulnerability evaluation methodology explores information leakage across voltage domains as well as through wireless carrier signals without modification of the device.

C. Research Contributions

We introduce a new dimension of physical side-channel vulnerability based on PSVC that can be exploited to launch a wide variety of attacks, including on-chip attacks, on-board attacks, and fully remote attacks. Due to the drastic difference between PSVC from other side-channel sources, it requires the development of new techniques for PSVC side-channel extraction as well as utilization of PSVC for information leakage. Specifically, this paper makes the following contributions.

- We introduce a new and effective vulnerability for information leakage, referred as physical layer supply voltage coupling (PSVC).
- We propose an efficient technique to isolate the PSVC signature of the computation from the actual noise and a methodology for evaluating information leakage through PSVC vulnerability.
- We perform end-to-end attacks exploiting PSVC vulnerability on two off-the-shelf victim devices supporting different instruction-set architectures.
- We show that PSVC signature propagates between voltage domains, and illustrate an on-board attack on a victim MCU from an IC that shares the same power supply.
- We show that PSVC signature can propagate with wireless carrier signals, and illustrate a successful end-to-end attack over Bluetooth transmission.
- We establish the fidelity of the PSVC vulnerability with respect to voltage variations in the power supply.
- We show that operating an MCU at the lowest operational voltage is a better choice in terms of security since it reduces the risk of PSVC side-channel vulnerability.

To the best of our knowledge, our work is the first effort to discover the PSVC as a side-channel source in the embedded systems domain to extract data that is being processed inside the device under attack on off-the-shelf hardware.

D. Paper Organization

The remainder of this paper is organized as follows. Section II provides background on various side-channel sources and surveys related efforts on side-channel attacks. Section III demonstrates our methodology for the evaluation of side-channel leakage through physical layer supply voltage coupling. Section IV performs end-to-end attacks on several configurations of off-the-shelf hardware components and shows the effectiveness of the proposed side-channel attack. In Section V, we discuss the applicability and limitations of our proposed work. Finally, Section VI concludes the paper.

II. Background and Related Work

In this section, we first discuss different physical side-channel sources used to perform attacks. Next, we survey related efforts on side-channel attacks.

A. Physical Side-Channel Sources

There are different types of side-channels that can leak information about the internal operation of a device. Specifically, we start the discussion with the most popular power side-channels [13] followed by recently introduced silicon substrate coupling [4], [14]. Finally, we introduce supply voltage coupling, which we will be using in this paper as the side-channel leakage source.

Side-channels due to power consumption: The power consumption varies in response to the changing logic states and data processing during computation in a device. These fluctuations in power create distinctive patterns that can be analyzed to infer information about the operations being executed. Figure 1 shows a typical setup for performing power side-channel attacks. Device power consumption is assessed by employing a shunt resistor (denoted as R_S), while simultaneously adjusting the device's inputs. This process helps establish the relationship between computational activity and power usage. Once the correlation is figured out, an adversary can observe power variations when a similar device is deployed and the adversary is able to deduce sensitive data, such as cryptographic keys or plaintext, without directly accessing the internal memory or processes of the target device.

Side-channels due to silicon substrate coupling: In the context of electronic systems, silicon substrate coupling refers to the phenomenon where energy or signals traverse between different components and traces within the same silicon substrate, whether by magnetic fields inducing voltage, electric fields enabling energy transfer, or through direct connections. This phenomenon can lead to undesirable electromagnetic interference and radio-frequency interference, which an adversary can use to recover sensitive information about the operations being computed inside the device [4], [14]. Designers try to make use of the effects of coupling to optimize the performance and reliability of devices.

Side-channels due to supply voltage coupling: In this paper, we introduce the tolerable noise that is coupled with the power signature of the dominant component of the device as **physical layer supply voltage coupling** (PSVC) and utilize it as the physical side-channel source to leak information from the power dominant component. In order to launch an attack by exploiting PSVC vulnerability, the minimum adversary capabilities must be at ***level-3***. PSVC adversary capabilities are similar to the attack methods that rely on side-channel leakage due to silicon substrate coupling. It is worth noting that silicon substrate coupling-based methods have the ability to extract data that is only in transit within chip interconnects or data processed within the wireless transmitter itself.

To the best of our knowledge, the concept of using physical layer supply voltage coupling as a side-channel on off-the-shelf hardware has not been explored in the existing literature.

B. Related Work

A vast majority of research on side-channel attacks relies on power side-channel analysis that measures a device's power consumption by monitoring variations in its current consumption during internal computations. Since the focus of this paper is on side-channel vulnerabilities that extend beyond the direct measurement of current consumption, in this section, we survey methods that exploit radio frequency emissions as well as visual feedback to uncover potential security risks and avenues for information leakage.

A side-channel attack based on the standard power LED's brightness and the color variations is proposed in [13]. In practice, power LEDs are typically connected to the embedded device, serving to offer a visual indication to the user that the device is powered and operational. The underlying concept is that minor power variations introduced during the computation will be reflected in the LED and can be captured visually. Therefore, the authors have used a video-based cryptanalysis to recover the secret information that was processed inside the embedded system.

The side-channel attack based on silicon substrate coupling is illustrated in [4]. The authors illustrate the effect of silicon substrate coupling leaking the information computed in mixed-signal chips. While this specific attack is constrained to chips housed on the same die, the authors demonstrate their ability to extract confidential data from the digital processor by analyzing the frequency variations of an analog radio frequency device that coexists on the same silicon substrate.

Danieli et al. [15] propose a simple power analysis technique to retrieve data from peripheral communication such as serial communication (UART), JTAG, communication to memory, and flash drives. It can only extract information traveling through interconnects and unable to retrieve any information from the processing cores. The author's primary focus is on extracting information from on-board signals, both with and without a galvanic connection to the TX antenna chip. In other words, the attack is capable of retrieving data traveling through the same die chip interconnects and unable to retrieve information that is being processed in the processing core.

A covert channel that uses malware to modulate secret data and transmit via a power supply unit of a desktop computer was proposed in [16]. The authors utilized the high-frequency voltage ripples generated by the power factor correction of the power supply and used the shared power grid for propagation of the secret. This covert channel is a deliberate communication mechanism that requires an infected transmitter. Furthermore, [16] exploits the shared power grid, which features a different power network topology compared to on-chip and inter-chip power distribution networks as the communication channel between transmitter and receiver.

FPGA-based side-channel attacks that rely on a hardware Trojan introduced during the manufacturing process are proposed in [14], [17]. FPGA Power Distribution Network (PDN) is a complex and unique network of power supply connections and pathways within an FPGA, responsible for delivering regulated power to gate arrays while ensuring signal integrity and minimizing electromagnetic interference. During the manufacturing process of the PDN, a malicious hardware Trojan needs to be inserted to perform the side-channel attack and retrieve the information that was being computed inside the FPGA. Recent studies have revealed the vulnerabilities of multi-tenant FPGAs to remote power side-channel attacks [18], [19], [19], [20]. Attackers can exploit shared resources to leak sensitive information by monitoring power consumption patterns, allowing them to infer cryptographic keys and other critical data from co-located circuits without physical access. Authors had to implement custom power monitoring circuits such as TDCs [18], [19] and ROs [19], [20] to extract the power consumption details required for these attacks.

The existing methods have three significant drawbacks. First, they often necessitate physical or visual proximity to the target device under attack. Second, these approaches frequently demand the introduction of malicious hardware modifications, such as shunt resistors [21] or hardware Trojans [22]–[24]. Lastly, remote methods that do not rely on hardware alterations (or insider help) struggle to recover data that is actively being computed within the other components of the device under attack. In Table II, we compare the state-of-the-art methods with the PSVC vulnerability. Unlike previous approaches, our method leverages off-the-shelf devices and utilize their existing architectures to extract confidential information, eliminating the need for any hardware modifications.

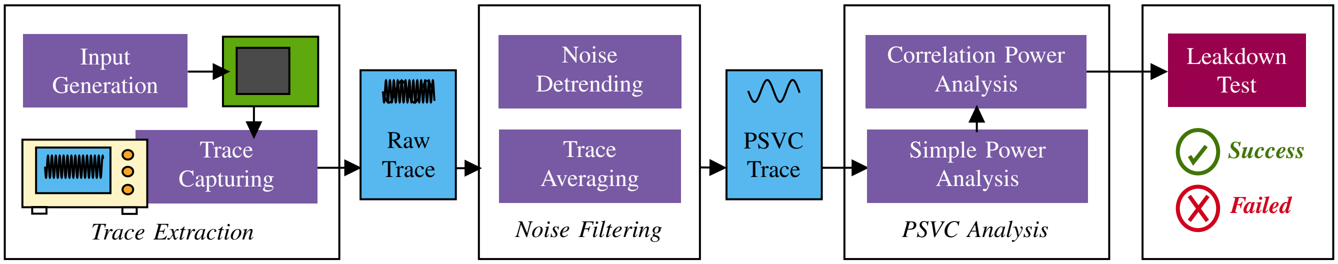


Fig. 2: Overview of our side-channel analysis framework utilizing physical layer supply voltage coupling (PSVC). This framework consists of four major tasks: trace extraction from the victim device, noise filtering to extract PSVC traces, PSVC trace analysis, and leakdown test to determine whether the device under attack is revealing any secret.

TABLE II: Summary of comparison between our proposed PSVC vulnerability and related efforts. PSVC vulnerability propagates through multiple voltage domains (VDs).

	Screaming Channel [4]	**Inter-FPGA [14]**	**EM [2]**	**PSVC (Proposed)**
Scope	On-chip	Same VD	On-chip	Multiple VD
Applicability	Single Die	FPGA	Any	Any
Invasiveness	None	Trojan	None	None
Trace Collect.	Remote	Internal	Remote	Remote

III. Evaluation of PSVC Vulnerability for Information Leakage

In this section, we present our methodology to evaluate hardware devices against side-channel leakage due to physical layer supply voltage coupling (PSVC). Figure 2 provides an overview of the proposed methodology that consists of four major tasks. The first task generates inputs, feeds them into the device under attack, and captures raw traces. The second task filters the noise to isolate the PSVC signature from traces obtained by running the test vectors on the device under attack. The third task performs simple power analysis as well as correlation power analysis of the collected PSVC traces. The fourth task performs leakdown test to determine the success rate of the attack. We first outline the problem formulation. Next, we describe each of the four tasks in detail.

A. Problem Formulation

During the design phase, every integrated circuit (IC) is engineered with specific tolerances for both input voltage and noise variations, ensuring that the IC's functionality remains unaffected within these defined ranges. The manufacturer will specify these values in their application notes (datasheet) with recommended capacitor configurations and layout design at the IC supply voltage points. This information assists printed circuit board (PCB) designers in effectively incorporating these values into their designs. The practical reason behind the coupling between the supply voltage noise and the data being processed can be explained as follows. Power is hierarchically distributed and undergoes voltage regulation at multiple stages, supplying various chips on the board. Inside the chip, a mesh-like network powers individual transistors, with integrated impedance comprising resistance (R), capacitance (C), and inductance (L) components, either by design or as parasitic elements [25].

Figure 3 illustrates a simplified diagram of a device with two MCU chips sharing the same power supply. Here MCU$_1$ is running a more power-intensive application (APP) and MCU$_2$ is running a general computation. During MCU operation, transistors switch based on the processed data, resulting in varying power consumption ($P = V \times$I), which leads to fluctuations in current. These current changes, described by the equation $\Delta V = L \times \frac{dI}{dt} + IR$, which induce voltage fluctuations on the power rails due to the internal impedance. These voltage fluctuations are supposed to be mitigated by the smoothing capacitors recommended by the manufacturer [1]. However, when designers use these manufacturer-specified decoupling capacitor configurations, still a tolerable amount of noise will escape. In the case of the example system in Figure 3, the power signature of APP running on MCU$_1$ will be present in the power rails as noise. This tolerable noise does not affect the operation of MCU$_2$ and other components present in the system. As this noise is still coupled with the power signature of the APP running on MCU$_1$, MCU$_2$ will be able to monitor the power signature of MCU$_1$ and able to deduce what is happening inside the APP. Information leakage that happens due to the above phenomena can be categorized under the scope of PSVC vulnerability. Figure 2 illustrates the four main steps involved in evaluating PSVC vulnerability and the following four sections describe each step in detail.

B. Trace Extraction from Victim Device

The first step for trace extraction is to start with a victim device that we want to evaluate against PSVC side-channel leakage. Before obtaining the PSVC trace from the selected device, we need to generate known input values to be fed into the application program (APP in Figure 3) that we intend to recover secrets. For ease of illustration, let us assume that the APP is an Advanced Encryption Standard (AES) cryptographic algorithm from the AESLib [26] library with AES-128-bit electronic codebook (ECB) implementation. We first describe the generation of input patterns. Next, we discuss the collection of PSVC traces.

Randomized Input Generation: Our goal is to formulate inputs in a manner that amplifies the correlation between the PSVC trace and the input. Ensuring an even distribution across the keyspace is crucial in this process because if a correlation exists between the key and the PSVC trace, uniformly manipulating the keyspace can accentuate the trace

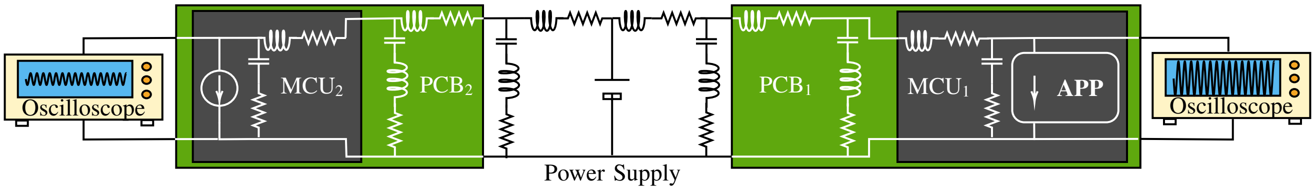


Fig. 3: Power distribution system model at different hierarchical levels of hardware components. On-chip current demand propagates through the power rails into the board-level power rails. Here the power signature of MCU$_1$ when running an application program (APP) will be coupled with the power rails of the entire system. This will affect other components that share the same power rails and MCU$_2$ will see a power signature of MCU$_1$ running the APP.

data, thereby highlighting the correlation. In the case of AES, we fix the key and randomize the plaintext that is used for encryption. Once the input data is written into the internal memory of the device, we proceed to the next step.

Trace Collection: In this step, the objective is to capture a PSVC trace from the victim's device. Note that the PSVC trace is embedded into the noise on the power rails. Therefore, PSVC carrying noisy trace can be collected in two ways:

- Use an oscilloscope to probe directly into victim device power rails (*VCC* and *GND*).
- Using Software Defined Radio (SDR) to capture the carrier signal of the device under attack.

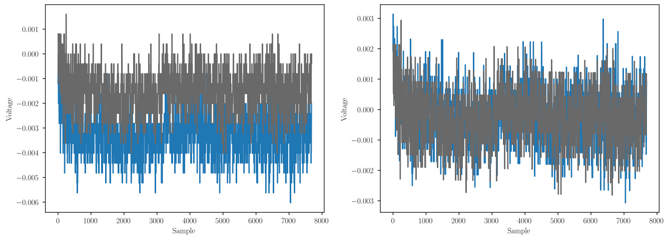
The first method requires physical access to the device or its power supply (e.g., wall charger or battery) to probe directly into the power lines. The next method is applicable if the device under attack consists of a radio transmitter (e.g., Bluetooth or WiFi). If automated trace alignment is required, a trigger pin can be assigned from the hardware device to trigger the trace-capturing device. However, this step is optional since there are signal processing tools that can perform the required alignment on collected trace data.

C. Noise Filtering and PSVC Extraction

The trace extracted from the device is dominated by the noise introduced by different components in the circuit. We need to isolate the PSVC trace from the noise to evaluate the effect of information leakage. For this purpose, we use two noise filtering techniques: noise detrending and trace averaging.

Noise Detrending: Unwanted trends often appear in captured traces due to various sources of noise. These noise sources may include switching noise from power supplies, environmental interference, and noise from the measurement equipment itself. Such noise introduces inconsistencies in the traces, leading to variations in signal amplitude (y-axis) across individual traces, which can hinder accurate analysis. To isolate the PSVC trace, we follow a two-step process. First, we perform linear detrending, which is the process of removing linear trends from a signal to isolate the meaningful fluctuations. The linear trend in signals is modeled using a straight-line equation through least-squares regression. The trend is then subtracted from the original signal, leaving a detrended signal that focuses on the variations of interest.

Finally, we use low-pass filtering to remove high-frequency noise, preserving the inherent signal frequencies of the hidden PSVC signature. The cutoff frequency of the filter was set to the third harmonic frequency of the target device's clock frequency. Figure 4 illustrates an instance of two collected traces from the AES ECB mode, before (Figure 4a) and after (Figure 4b) performing the noise detrending. It can be observed that the variations in the trace due to noise are removed after performing the detrending operation.



(a) Before detrending operation (b) After detrending operation

Fig. 4: Noise removal with detrending operation on traces collected over AES block cipher.

Trace Averaging: In the process of highlighting the PSVC trace from the collected trace data, we need to consider the Signal to Noise Ratio (SNR). Here the signal is the effect of PSVC. Our objective is to improve the signal (effect of PSVC) while reducing the effect of noise. In other words, the higher the SNR ratio, the better we isolate the PSVC signature from the noise. To improve the SNR ratio, we perform trace averaging. The number of traces considered for the averaging depends on the specific scenario and configurations of the device under attack. Therefore, we define the $Avg(N)$ function in Equation 1 to denote the trace averaging function where $\bar{X}(t)$ is the average trace at time t, N is the total number of traces, $X_n(t)$ is the value of the n^{th} trace at time t. The summation runs through all N traces, and t represents a specific time point.

$$\bar{X}(t) = \frac{1}{N}\sum_{n=1}^{N} X_n(t)$$
$$Avg(N) = \{\bar{X}(0), \bar{X}(1), \ldots, \bar{X}(T)\} \tag{1}$$

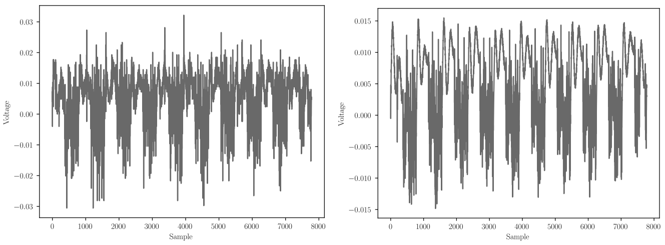
Table III presents the variation of the SNR with respect to different numbers of averaged traces of $Avg(N)$ of AES block cipher running on an *Arduino nano* victim device. When the number of traces (N) increases, the SNR improves significantly. Figure 5 presents an instance of trace average on the AES block cipher. It can be observed that compared to one trace ($Avg(1)$) in Figure 5a, an average of $N = 10$ traces

($Avg(10)$) is superior with respect to SNR in Figure 5b with eliminated random DC shifts caused by the power supply.

$Avg(N)$	1	2	5	10
SNR (dB)	5.06	8.61	14.24	21.93

TABLE III: Improvement of SNR of the PSVC signal with the increase in number of traces (N) used for trace averaging, $Avg(N)$.

After executing noise detrending and subsequently implementing trace averaging, we successfully isolated the trace signal pertinent to the PSVC signature of the target device. The next step involves analyzing this refined signal to assess whether the captured PSVC trace has the potential to leak confidential information from the application (APP in Figure 3) running on the victim device or not.



(a) Before averaging ($Avg(1)$) (b) After averaging ($Avg(10)$)

Fig. 5: Improving the SNR with trace averaging on traces collected over AES block cipher on *BlackPill*. Trace averaging is able to improve SNR by eliminating random DC shifts caused by the power supply.

D. PSVC Trace Analysis

At this stage, we have an isolated signal trace that encodes the PSVC signature of the device. Next, we need an effective way to process this signal to find a correlation between the input secret values given to the tasks and the PSVC signature. In order to perform this, we explore two techniques: simple power analysis and correlation power analysis.

Simple Power Analysis (SPA): The intention of SPA is that if the device leaks information from the side-channel environment, it might be directly visible on the signature traces. This is a very simple technique that is performed by doing visual inspections on the side-channel traces. During the inspection of PSVC traces, we pay attention to the algorithm of the application program (APP) running on the victim device. If the implementation has obvious drawbacks such as imbalance branch statements or specific hardware components that created specific PSVC signatures during certain operations, their effects can be visually observed on the PSVC trace itself. Note that simple power analysis may not work for all the devices as we illustrate in Section IV.

Correlation Power Analysis (CPA): When the observation based on SPA does not yield any meaningful results, we can move towards a more experimental method of correlation power analysis. CPA aims to establish a correlation between measured side-channel signature and expected side-channel signature [27]. CPA has the ability to reveal correlations even if noise is not completely filtered from the trace. CPA takes two inputs, expected value and observed value, and produces the output of correlation value between the observed and expected value. The expected value can be calculated in the following two ways from the randomized input values generated in Section III-B.

- *Hamming Weight Model*: PSVC signature is correlated with the number of 1's in the selected data, that is being computed in the device.
- *Hamming Distance Model*: Consecutive PSVC signatures are correlated with the number of transitions from $1 \rightarrow 0$ and $0 \rightarrow 1$ in each bit of the two consecutive selected data.

We utilize the most simple and straightforward Hamming weight model for the correlation power analysis. Then we compute the Pearson's correlation coefficient r_{xy} between the expected value x and observed value y from Equation 2. Here, $\bar{x}$ and $\bar{y}$ are average of samples x and y, respectively.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2}$$

We extract an array of values from each trace at time t, representing the observed values y. The expected value array x comprises Hamming weight values for each plaintext corresponding to a guessed key byte value. Subsequently, we compute r_{xy} for every time point and each key byte value and store the absolute maximum correlation value for each key byte value in an array denoted as r. The array r reflects the correlation between the guessed secret key values assumed to be used in the application (APP) of the victim device and the PSVC power trace. The next step involves classifying these correlation values to determine whether the device under test is indeed leaking information through the PSVC side-channel.

Example 1 (CPA): *Suppose* $p_1 = [0.01, 0.2, 0.05, 0.1]$, $p_2 = [0.02, 0.1, 0.03, 0.2]$ *and* $p_3 = [0.01, 0.2, 0.04, 0.2]$ *are three traces observed after trace extraction and noise filtering stages. Assume that* p_1*,* p_2*, and* p_3 *are captured for a fixed secret key and three different plain texts* `0x02`*,* `0x05` *and* `0x01`*, respectively. Let us guess the secret key as* `0x03` *and choose selected data as the xor output between plain text and the guessed key to calculate the Hamming weight. Note that, in practice, the selected data can vary depending on the attack. Here, we choose the xor output for the ease of illustration. Then, for* $t = 0$*, we get* $x = [0.01, 0.02, 0.01]$*,* $y = [1, 6, 2]$*, and* $r_{xy} = 0.98$*. For* $t = 1$*,* $x = [0.2, 0.1, 0.2]$*,* $y = [1, 6, 2]$*, and* $r_{xy} = -0.98$*. Likewise, after calculating* r_{xy} *for all time sampling points for three different key guesses (*`0x01`*,* `0x02` *and* `0x03`*), we get* $r = [0.69, 0.99, 0.98]$. ■

E. Leakdown Test

It is important to clarify that a higher absolute correlation value does not guarantee the accuracy of the key guess; it simply makes it more likely. Therefore, we implement the final step as the leakdown test where we perform key guess validation with correlation thresholding and conclude whether the attack is a "success" or "failed". We define a distance value as shown in Equation 3, denoted as d, which measures the difference between the highest correlation value and the average correlation values in r.

$$d = max(r) - \frac{1}{256}\sum_{j=1}^{256} r[j] \tag{3}$$

If the d value for a guess key value is greater than a predefined threshold λ, we can say that the guessed key byte value is a correct key guess. This signifies that the attack is successful. However, it is important to strike a balance with the threshold, as setting it too low can result in false positives (accepting incorrect key guesses) while setting it too high can lead to false negatives (rejecting correct key guesses). The value of the threshold is based on empirical experiments and involves some trial and error. The threshold value is directly affected by the characteristics of the device under attack and the quality of the trace-capturing device.

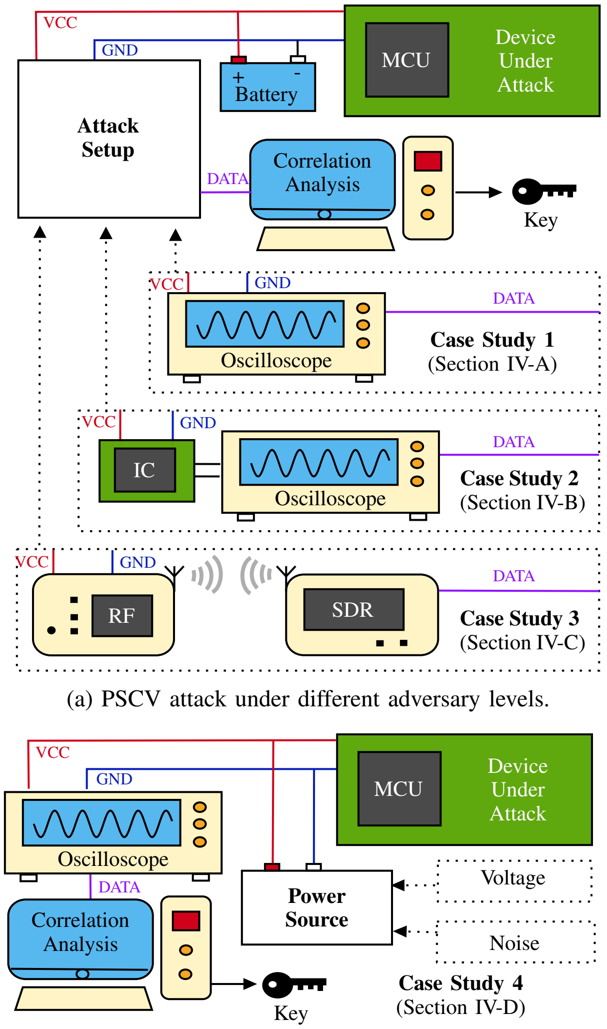
Example 2 (Leakdown Test): *Distance values for guessed key in Example 1 are as follows: for key guess* `0x01`, $d = -0.20$*; for key guess* `0x02`, $d = 0.10$*; and for key guess* `0x03`, $d = 0.09$. *If we define* $\lambda = 0.095$ *based on historical data, we can conclude that* `0x02` *is the secret key.* ■

IV. Experiments

To demonstrate the applicability and effectiveness of information leakage through physical layer voltage coupling (PSVC), we have performed four case studies utilizing two hardware boards and two trace collection methods.

Four Case Studies: The first three case studies explore the possibility of launching a PSVC side-channel attack on three different configurations. We have selected three different configurations of devices with increasing complexity as three case study scenarios (Section IV-A to Section IV-C). Figure 6a shows the overview of the device configurations used in the first three case studies. Case Study 1 directly connects the oscilloscope inputs to the power lines (VCC and GND) and the oscilloscope output (DATA) to the computer for PSVC trace (correlation) analysis. In contrast, Case Study 2 connects the oscilloscope to an IC, which is connected to the device under attack. In other words, Case Study 2 represents the scenario in Figure 3. Note that Case Study 3 is the ultimate attack where the adversary is in ***level-3*** and only needs wireless proximity to the device. All these three case studies will have three things in common: the power supply, victim MCU, and the PSVC evaluation framework as discussed in Section III. The fourth case study evaluates the effect of the PSVC side-channel attack under different power supply configurations in Section IV-D. The different power supply configurations used for Case Study 4 are illustrated in Figure 6b.

Two Hardware Boards: To demonstrate the applicability of our framework, we have used two types of devices with different instruction-set architectures in each of the configurations: (i) *Arduino nano* development board with *Atmel ATmega328* microprocessor based on *RISC AVR* architecture, and (ii) *BlackPill* development board with *STM32F401* microprocessor based on *ARM* architecture. These two hardware boards are shown in Figure 8.



(a) PSCV attack under different adversary levels.

(b) Effect of power supply voltage on PSVC attack.

Fig. 6: Experimental setup for four case studies. Case Study 1 assumes direct access to the victim device, while Case Study 2 indirectly (via another IC) connects to the victim device. Case Study 3 is the ultimate end-to-end attack that can be launched using PSVC, where the adversary is in most constrained ***level-3***. Case Study 4 explores the effect of power supply voltage on PSVC vulnerability.

Two Trace Collection Methods: For the experiments, PSVC traces were collected using the following two devices. From both trace capturing devices, the data were collected from *Matlab R2020a* API. To process the traces collected in each case study, we have developed scripts in *Python* and *Matlab*. All the algorithms for correlation power analysis were written in Python. Experiments were conducted in a system with 16Gb RAM on an AMD Ryzen 7 processor.

- *Keysight DSOX1102G* [28] oscilloscope controlled through the Virtual Instrument Software Architecture (VISA) protocol. This trace collection scheme is used for Case Study 1, Case Study 2, and Case Study 4.
- HackRF [29] with CubicSDR [30] software-defined radio (SDR) module using radio frequency (RF) signal captur-

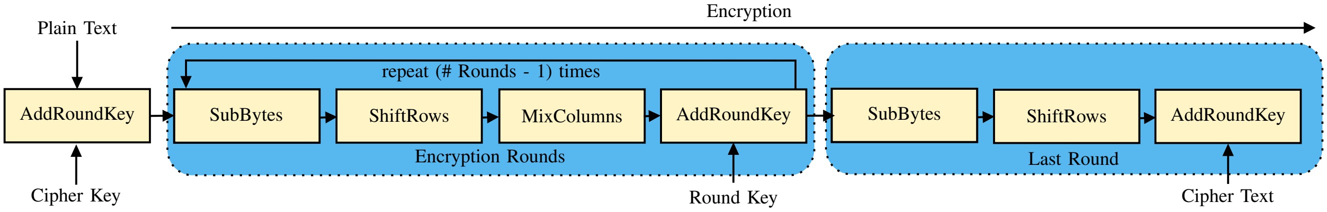


Fig. 7: Overview of AES encryption algorithm that consists of a series of well-defined steps, including substitution, permutation, and mixing operations, which are known as the SubBytes, ShiftRows, and MixColumns transformations. The AES encryption process consists of multiple rounds. If a device is leaking information via PSVC vulnerability, these individual rounds and their internal steps become visible on the PSVC trace.

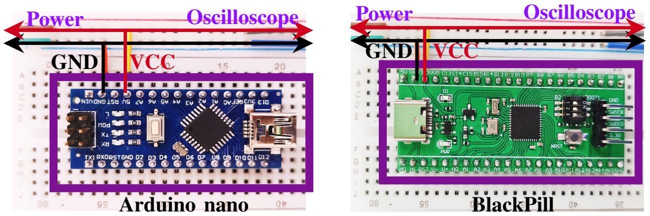
ing ability. This trace collection mechanism is used for Case Study 3.

Application Program (APP): We have selected the Advanced Encryption Standard (AES) as the application (APP) running on the MCU for our case studies. We use AES since it is designed to operate on fixed-size data blocks (typically 128 bits), and its internal operations consist of a series of well-defined steps, including substitution, permutation, and mixing operations, which are known as the SubBytes, ShiftRows, and MixColumns transformations as illustrated in Figure 7. AES employs a key schedule to generate round keys for each encryption round followed by an AddRoundKey operation. The encryption process consists of multiple rounds (typically 10, 12, or 14 rounds, depending on the key size 128, 192, and 256 bits, respectively), wherein the data undergoes transformations via these operations. If a device is leaking information via PSVC vulnerability, these individual rounds and their internal steps become visible on the PSVC trace as we demonstrate through four case studies.

A. Case Study 1: Exploit PSVC Vulnerability to Mount an Attack with Power Rail Probing

In this case study, we perform an end-to-end attack exploiting the PSVC vulnerability on the victim device when probe access is available to launch the attack. We set the adversary capability as ***level-2*** as defined in the threat model (Section I-A). Here the adversary has physical access to the victim's device but is not allowed to do any modifications to the device. First, we explain the experimental setup used for this case study. Then we perform an end-to-end attack on *Arduino nano* and *BlackPill* development boards and demonstrate the attack capabilities of PSVC-based side-channel attacks with power rail probing.

1) Experimental Setup: In this case study, we utilize SPA followed by CPA to analyze AES encryption. Typically, side-channel attacks focus on modeling power patterns from the first round of computation of AES. The first round provides valuable insights into how plaintext and key bytes are combined. We focus on the SubBytes operation, chosen for its susceptibility to side-channel attacks. The SubBytes's vulnerability stems from its data-dependent and key-dependent transformations. During this operation, each data byte undergoes substitution with another byte based on a fixed lookup table. As these substitutions occur, the power consumption discloses information about the processed bytes and the encryption key. The inherent data and key dependencies within the SubBytes operation result in observable patterns and correlations in the side-channel data. We leverage this effect to illustrate the PSVC side-channel leakage in this case study.



(a) RISC based *Arduino nano* (b) ARM based *BackPill*

Fig. 8: Two attack setups used in Case Study 1 with two different development boards with two different instruction set architectures of RISC and ARM.

Figure 8 illustrates the two hardware setups used in this case study, these devices are connected to a noisy power supply, which serves as the worst-case power source (lower SNR ratio) for the device under attack. Inside the MCU of the device under attack, we execute an AES encryption repeatedly. In this case study, we have performed two sets of experiments, using two different MCUs: the *Arduino nano* development board with an *Atmel ATmega328* MCU (Figure 8a) and the *BlackPill* development board with an *STM32F401* MCU (Figure 8b). With ***level-2*** adversary capabilities, we probe the power inputs to the MCU (development board). Subsequently, we capture sufficient traces using an oscilloscope to perform CPA. After the detrend operation on captured traces based on the steps discussed in Section III-C, we take an average of 10 traces for the *Arduino nano* ($Avg(10)$) and 10 traces for the *BlackPill* ($Avg(10)$) to find better traces. Then, CPA is applied to these filtered traces.

2) Results: Figure 10 illustrates the SPA of traces captured during the experiment. Figures 10a presents the PSVC signature of AES-128 on *ATmega328* MCU and Figure 10b presents the PSVC signature of *STM32F401* MCU. Since AES-128 computation consists of ten encryption rounds, the traces exhibit ten distinct power signatures corresponding to each encryption round. As illustrated in Figure 9, we can clearly distinguish individual functions within the AES algorithm for *STM32F401*. Since the traces captured for *ATmega328* exhibit more noise compared to *STM32F401*, we conducted CPA for *ATmega328* to simulate the worst-case scenario. The

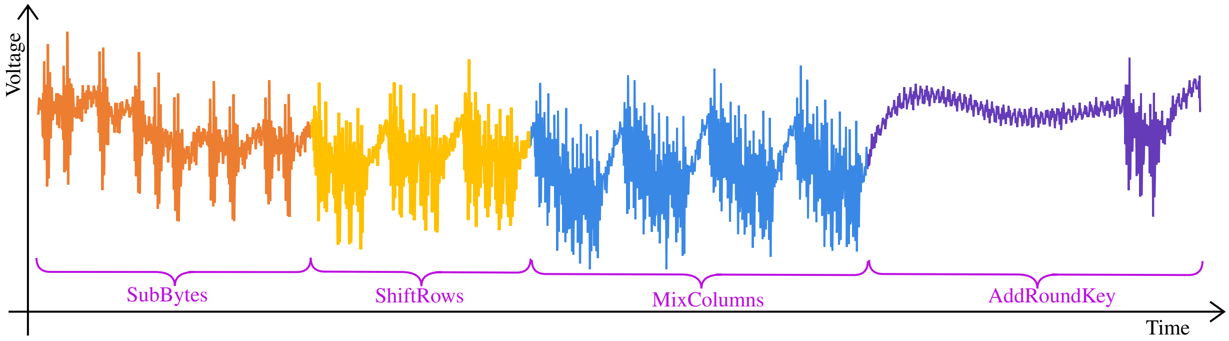
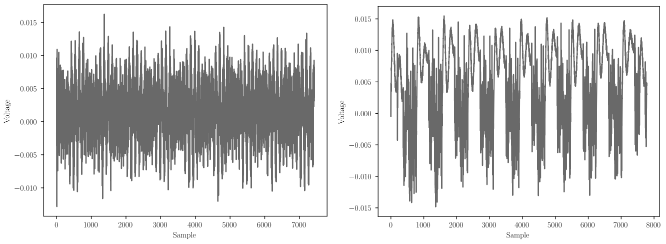


Fig. 9: Simple power analysis of a PSVC trace for one round of AES-128 captured from *STM32F401* MCU in *BlackPill* development board. Here all operations illustrated within the encryption round of Figure 7 can be clearly observed.



(a) PSVC trace on ATmega328 (b) PSVC trace on STM32F401

Fig. 10: Simple power analysis of PSVC traces for AES-128 on two different MCUs demonstrates the PSVC vulnerability. The traces include the signature of ten AES rounds.

correlation coefficient results with a varying number of averaged traces are displayed in Figure 11, which highlights two key findings. First, they provide evidence of PSVC coupling and the side-channel vulnerability of PSVC and demonstrate the ability to mount a key recovery attack using the PSVC vulnerability. Second, they demonstrate that as the number of traces increases, the success rate of the CPA attack also increases, which is useful in the next case studies with complex scenarios and more noise.

We have evaluated the effectiveness of PSVC against the state-of-the-art EM side-channel attack, conducted on the same Arduino Nano board running the same AES library following the same flow shown in Figure 2. EM analysis was chosen for this comparison because, like PSVC, it does not require any modifications to the device. Table IV presents the comparison of the number of recovered key bytes with different numbers of traces to highlight the advantages of our framework. We use Measurements to Disclosure (MTD) as the evaluation metric. The number of measurements required to reveal the correct key is a commonly used metrics to assess the success of an attack. As shown in the table, the PSVC-based attack requires 20 times fewer traces compared to the EM-based attack to disclose the full correct key.

Figure 12 compares PSVC and EM-based side-channel attacks, showing the correlation coefficient versus the number of traces for recovering key byte 0. This provides visual evidence supporting the results presented in Table IV. Specifically, for key byte 0, the PSVC-based attack was able to extract the key using approximately 400 averaged traces, whereas the EM-based attack required around 19,000 averaged traces to extract the same key byte. This better performance of PSVC compared to the EM-based attack can largely be attributed to differences in the SNR, as outlined in Table V. The results show that the PSVC traces exhibit a higher SNR value than the EM traces, indicating a stronger correlation between PSVC signals and the device's computational activity. In contrast, EM signals are more susceptible to noise mainly caused by environmental interference, which can degrade the quality of the extracted trace and reduce the attack's effectiveness.

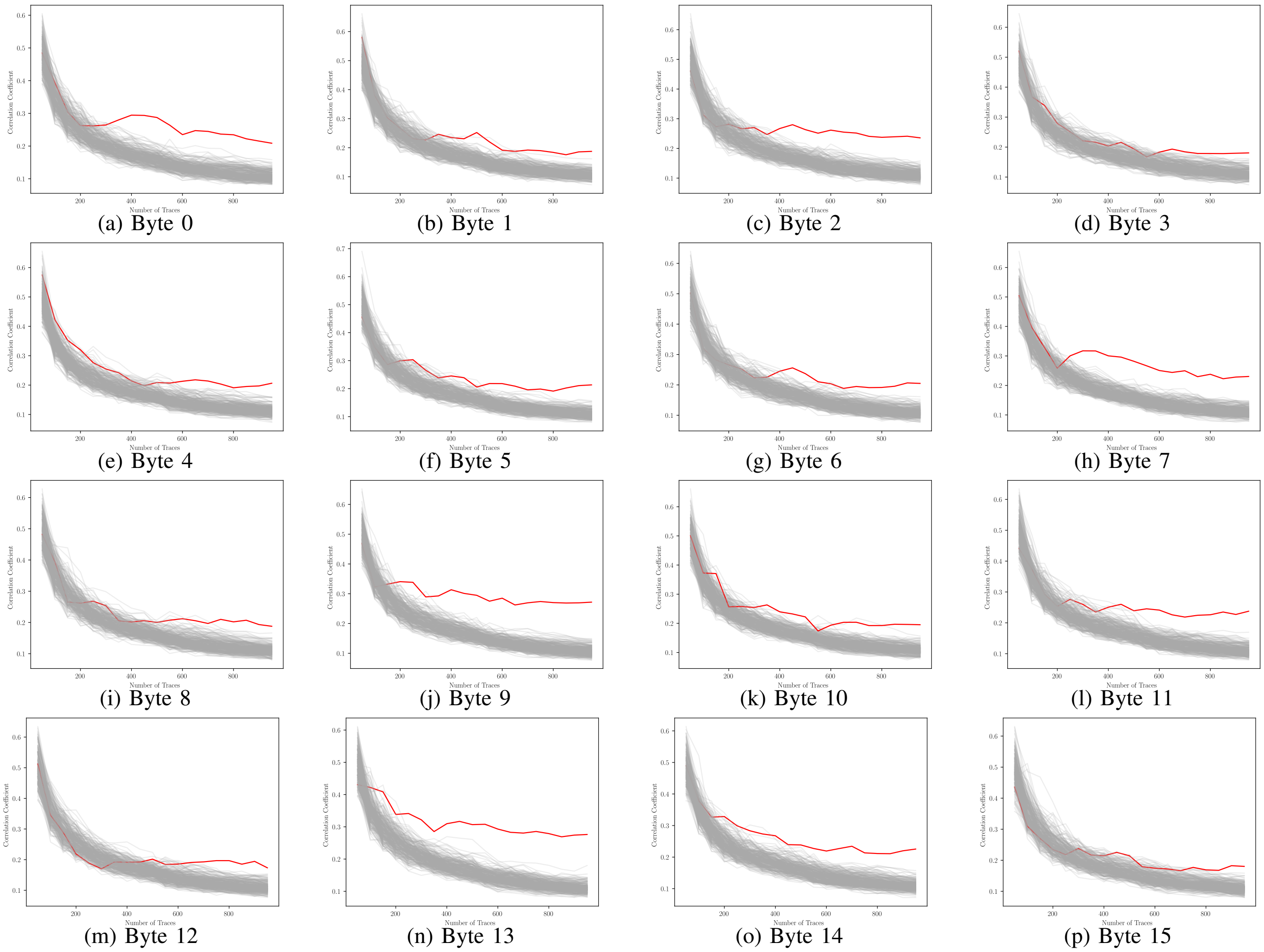
TABLE IV: Number of extracted key bytes by PSVC and EM-based attacks for different number of traces. Both attacks were conducted on the same Arduino Nano board running the same AES library. The framework shown in Figure 2 was followed for both attacks, except for the trace acquisition process.

No. of Averaged Traces		**1000**	**5000**	**10000**	**20000**
Leakage Source	***PSVC***	16	16	16	16
	EM	0	0	5	16

TABLE V: Comparison of Signal-to-Noise Ratio (SNR) for PSVC and EM traces.

Type	PSVC	EM
SNR (dB)	21.93	15.72

In terms of resources, our case study with PSVC requires only an oscilloscope with a reasonable sampling rate. The EM side-channel attack required specialized equipment, including an EM probe (H-Field 170 mm Long, 10 mm Loop probe from the TEKBOX EMC probe set) and signal amplifiers (TEKBOX 40dB, 2 MHz–6 GHz wideband amplifier), making it more resource-intensive to perform a successful attack. The EM probe was positioned at a location where the measured radiation waveform's peak voltage was maximal. The time required for a successful attack using PSVC is approximately 1 hour, compared to 23 hours for the EM side-channel attack. Both approaches involve three main phases: trace collection, trace preprocessing, and analysis (or information extraction). The EM attack required significantly longer trace collection time (22 hours) due to its higher MTD, while PSVC achieved the same in just 1 hour. For preprocessing and analysis, the EM attack took about 50 minutes, whereas PSVC required only 5 minutes. Thus, our proposed method is efficient, achieves similar results in significantly less time (20X).



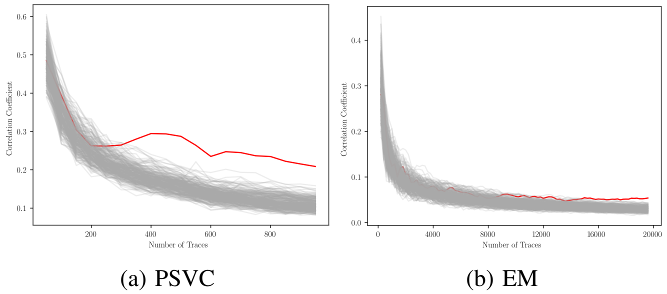
(a) Byte 0 (b) Byte 1 (c) Byte 2 (d) Byte 3

(e) Byte 4 (f) Byte 5 (g) Byte 6 (h) Byte 7

(i) Byte 8 (j) Byte 9 (k) Byte 10 (l) Byte 11

(m) Byte 12 (n) Byte 13 (o) Byte 14 (p) Byte 15

Fig. 11: Correlation coefficient (Y-axis) for byte 0 to byte 15 of an AES 128-bit key with 256 different key values across varying numbers of averaged traces (X-axis). Note that there is a noticeable difference between the correct key byte (indicated by the red line) and incorrect key bytes which are identified during the leakdown test.



(a) PSVC (b) EM

Fig. 12: Comparison of the correlation coefficients (Y-axis) for key byte 0 of the AES 128-bit key, evaluated by varying numbers of averaged traces (X-axis) for PSVC and EM-based attacks. The correct key byte is marked with a red line.

B. Case Study 2: Exploit PSVC Vulnerability to Mount an Attack using Voltage Regulator

We have created this case study to illustrate that the effect of PSVC can be reflected via secondary ICs that share the same power source. In this case study, the victim device and the attack-launching device share the same power source. For this case study, we set the adversary capability as ***level-2*** as defined in the threat model (Section I-A). Here the adversary has physical access to the attack-launching IC but is not allowed to do any modifications to the attack-launching IC or to the victim MCU. First, we explain the experimental setup used for this case study. Then we perform an end-to-end attack to show the possibility of launching a PSVC-based side-channel attack via a neighboring IC that shares the same power source.

1) Experimental Setup: Figure 13 shows the experimental setup for Case Study 2. Here, we have utilized the same power supply that we used in Case Study 1. For this experiment, we opted for *STM32F401* MCU as the victim device, and the 3.3V Low Dropout voltage regulator of the *Arduino nano* board as the attack-launching IC. We selected the Voltage Regulator Module (VRM) as the attack-launching IC in this experiment since VRMs are available in almost all devices to convert higher supply voltage into lower levels required by various electronic components [31]. The attack methodology is similar to Case Study 1, except instead of probing victim device power rails (input voltage domain), we take measurements by probing the VRM output (regulated voltage domain).

2) Results: Figure 14 illustrates the comparison between the SPA of captured traces for AES-128 on the *STM32F401* through direct probing (Figure 14a) of the power rails versus

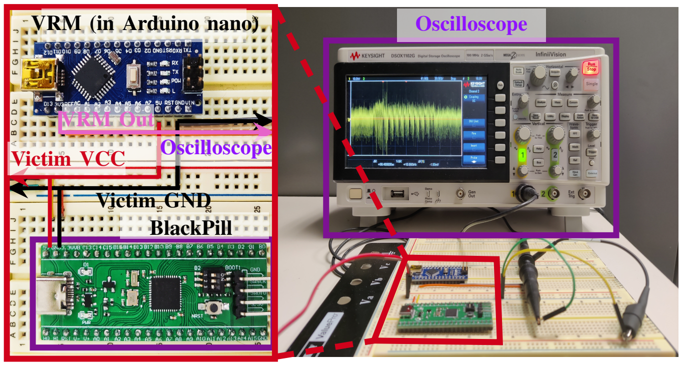
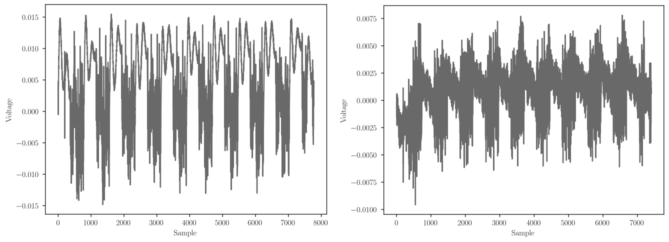


Fig. 13: Attack setup in Case Study 2. Both the victim IC and the attack-launching IC share the same power source. Oscilloscope probes are connected to the VRM output.

traces captured through the $3.3V$ output of the *Arduino nano* via the VRM (Figure 14b). Similar to Figure 14a, ten rounds of AES-128 implementation are clearly visible in Figure 14b. However, the noise level of the captured traces is elevated due to interference from various components.



(a) PSVC with direct probing (b) PSVC with VRM probing

Fig. 14: Comparison between PSVC traces of *STM32F401* collected by direct probing into the power rails vs PSVC trace collected from the $3.3V$ VRM output of the *Arduino nano*. Although in Figure 14b, the trace is collected in another voltage domain of the same circuit, the PSVC signature is still preserved compared to the original domain (Figure 14a).

In this case study, 12,000 averaged traces were needed for full key recovery using CPA. This 12-fold increase in MTD compared to Case Study 1 is caused by the high noise levels present in the captured traces. However, our findings in this case study reveal that PSVC can propagate through VRMs from one voltage domain to another within an electronic device. Although the primary responsibility of VRMs is to isolate voltage domains and prevent the propagation of voltage fluctuations between them, the PSVC side-channel vulnerability exploits the tolerable noise margin accepted by the VRM, allowing the PSVC signature to propagate across voltage domains. This finding represents a significant advancement in the PSVC-based attack compared to prior works such as [14], which extract leaky information across components within the same voltage domain. Moreover, PSVC does not require any type of insider assistance (hardware Trojan) to perform the attack, as opposed to the method described in [14]. This makes this case study broadly applicable to any electronic circuit with multiple voltage domains, regardless of the underlying hardware platform.

C. *Case Study 3: Exploit PSVC Vulnerability to Mount an End-to-End Wireless Attack*

In this section, we perform a completely remote end-to-end attack proposed in [4] to exploit the PSVC vulnerability on the victim device which consists of a wireless Bluetooth module. For this case study, we set the adversary capability as ***level-3*** as defined in the threat model (Section I-A). Here the adversary is within the wireless range of the victim's device. First, we outline the experimental setup. Next, we perform an end-to-end remote attack and demonstrate the attack capabilities of PSVC-based side-channel attack when the adversary does not have physical access to the device under attack.

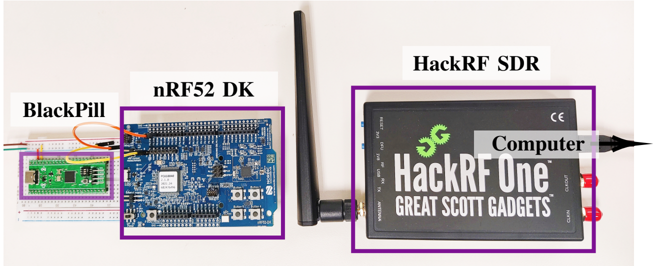
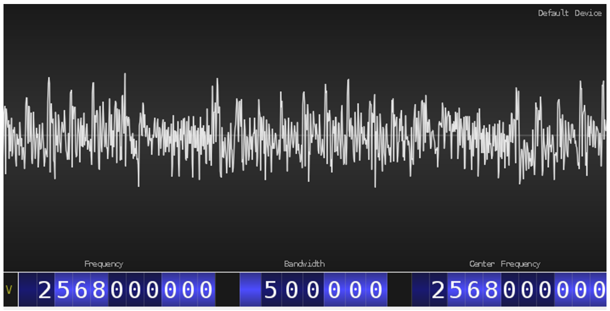


Fig. 15: Attack setup used in the Case Study 3. Here the PSVC traces are collected from the HackRF [29] SDR.

1) *Experimental Setup:* Figure 15 presents the experimental setup we have used for Case Study 3. Here the device under attack consists of two physically separated development boards that share the same power supply. We have the *BlackPill* development board which runs the cryptographic application (APP) and the *nRF52-DK* development board consists of a programmable *nRF52832* Bluetooth transceiver that runs a generic Bluetooth transmission. In this experiment, we utilized the radio test example, which provides support for nRF52-DK, and made modifications to enable the transmission of a high-power continuous carrier signal at 2.4 GHz. Both *BlackPill* and the *nRF52-DK* boards were powered using the same power supply. During the experiment, we maintained a distance of approximately 50 cm between the two boards to minimize electromagnetic coupling. As a result, the sole connection between the *BlackPill* and *nRF52-DK* was through the shared power lines, ensuring that digital noise propagation in the setup occurs exclusively via the PSVC. Instead of probing the power rails with the oscilloscope, in this experiment, we use the HackRF [29] software-defined radio module to capture the Bluetooth transmission of the *nRF52-DK* development board via CubicSDR application. We placed the HackRF at a small distance (10 cm) from the *nRF52-DK*. After capturing the RF signal, we convert the frequency domain data into the time domain and apply the steps discussed in Section III. A trigger mechanism inspired by [4] is used to separate the APP trace segment and align the extracted traces. Specifically, we manually inspected the captured signal's frequency response to identify a component that appears exclusively just before the AES-128 execution, which is then used as the trigger.

2) *Results:* Figure 16 displays the SPA of the captured trace obtained using HackRF and CubicSDR. In comparison to the SPA of previous case studies, this SPA exhibits more noise.

Nonetheless, we can still recover the signature of ten rounds of AES-128 within the trace.



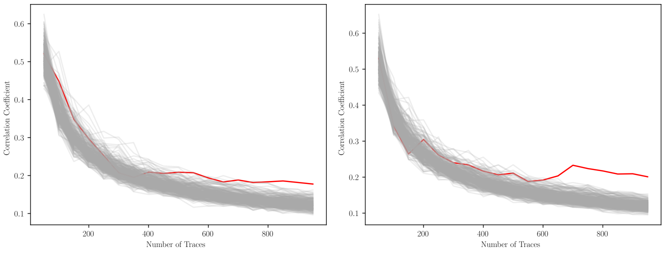
Frequency 2568000000

Bandwidth 500000

Center Frequency 2568000000

Fig. 16: The signature of PSVC leakage of the *BlackPill* development board observed through CubicSDR [30] application. The trace capture was performed with a HackRF [29] SDR tuned to a harmonic of the *STM32F401* clock frequency which receives the amplitude-modulated (AM) Bluetooth carrier signal. Here, we can identify the rounds of AES-128. Note that it may not be as clear as the SPA of previous case studies.

Figure 17 presents leakdown test results after performing CPA on two key bytes (Byte 0 in Figure 17a and Byte 15 in Figure 17b) of end-to-end remote attack on *BlackPill* development board via a Bluetooth carrier signal.



(a) Byte 0

(b) Byte 15

Fig. 17: Results of the key bytes recovery after performing the leakdown test on the end-to-end attack launched on *BlackPill* development board via the Bluetooth carrier signal of the *nRF52 DK* development board. Traces are captured using the HackRF SDR module and recorded by the CubicSDR application followed by filtering with $Avg(300)$.

Even though this case study performed the attack proposed in [4], the attack in [4] relies on the silicon substrate coupling between digital-to-analog domains in the same integrated circuit, whereas PSVC enables digital-to-analog coupling between different ICs. In fact, PSVC provides a stronger attack surface by demonstrating digital-to-analog coupling between two ICs on two different boards. This demonstrates that digital noise propagation to the analog domain can occur through the PSVC as well.

This leads to the conclusion that if the PSVC signature is strong enough, it can propagate from the victim MCU to the attack IC (in this case, RF). Therefore, the results of our experiments from Case Study 1, Case Study 2, and Case Study 3 provide evidence of the following: (1) the existence of PSVC, (2) the potential of PSVC as a vulnerability, (3) the ability to propagate from one voltage domain to another using the same power source, and (4) the ability to propagate from the digital domain (MCU) to the analog domain (RF).

D. Case Study 4: Effect of Supply Voltage Range on PSVC Attack

In this section, we explore the effect of the supply voltage range of the power source on the recoverability of secret data from the application (APP) with a PSVC side-channel attack. For this, we change the quality of the power supply with different voltage levels supported by the device under attack. In this section, we first outline the experimental setup. Next, we present the results about the behavior of PSVC side-channel leakage with the variation of input voltage.

1) Experimental Setup: We use the *Arduino nano* development board with the *Atmega328* MCU and power it with a variable power supply. *Atmega328* MCU has a rated input voltage range between $+1.8V$ to $+5.5V$. For this case study, we use the same setup of Case Study 1 for different input voltages of $3.0V, 4.0V$, and $5.0V$.

2) Results: Figure 18 illustrates the leakdown test results after performing CPA for input voltages of $5.0V, 4.0V$ and $3.0V$ in Figure 18a, Figure 18b and Figure 18c, respectively. Figure 19 presents the attack success rate after performing the experiment for three selected input voltage values. The behavior illustrated in Figure 19 can be primarily attributed to variations in the signal-to-noise ratio (SNR) with the input supply voltage. Our observations revealed a decline in the SNR of the captured trace as the input voltage decreases, as demonstrated in Table VI. This reduction in SNR can be mainly attributed to the reduced strength of power fluctuations at lower input voltages. It is important to highlight that all input voltage values used in our experiments remained within the specified operating voltage range of the *ATmega 328* MCU.

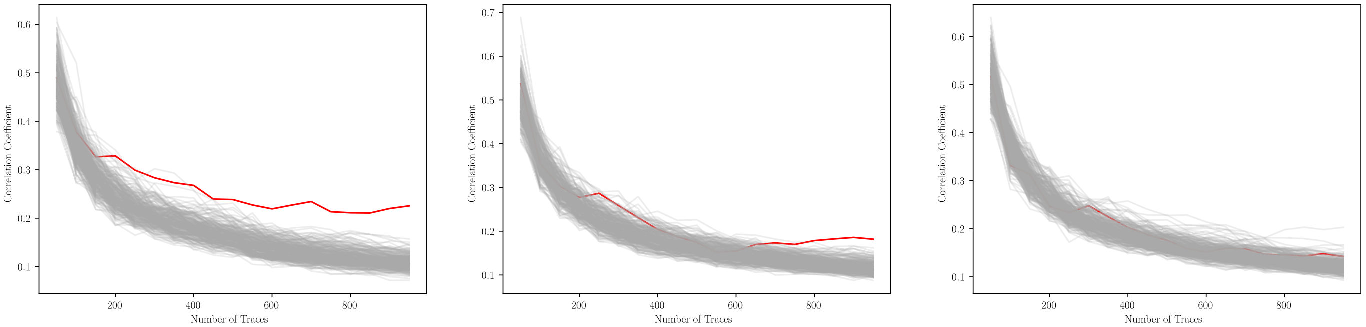
This experiment highlights a significant fact. Our results demonstrate that operating an MCU at its minimum input voltage is a preferable choice for enhancing security, as it is more challenging to attack the MCU when the input voltage is near the lower bound of the specified operating voltage.

TABLE VI: SNR of captured traces with different input voltages (compared to the power trace captured using the conventional method, as shown in Figure 1)

Vin (V)	3	4	5
SNR (dB)	-9.77	-6.53	-5.95

V. Applicability and Limitations

Our proposed framework enables designers to evaluate off-the-shelf MCU boards before incorporating them into their security projects. Different board configurations with different computation tasks will yield different results. Therefore, it is important to evaluate the candidate development boards for the specific purpose. As demonstrated in Section IV, PSVC vulnerability is exploitable as an attack in a wide variety of scenarios, including the scenario when the adversary does not have physical access to the device. We have shown the effectiveness of our framework on two hardware designs supporting different instruction-set architectures.



(a) Input voltage = $5.0V$

(b) Input voltage = $4.0V$

(c) Input voltage = $3.0V$

Fig. 18: Correlation coefficient for example key byte of an AES 128-bit key with 256 different key values across varying numbers of averaged traces for different input voltages. The correct key byte is indicated by the red line.

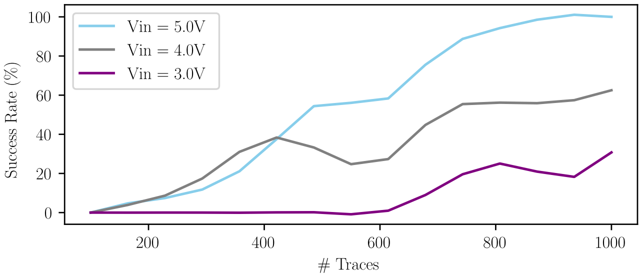


Fig. 19: SCA success rate with different input voltages. It can be observed when the device is operating at a higher input voltage ($5V$), the attack success rate is higher and the number of traces to perform an attack is less compared to the same device operating at a lower input voltage ($3V$).

Our evaluation of PSVC vulnerability has revealed that the manufacturers are focused on functionality, and security is overlooked. The smoothing capacitor configurations that we have observed on these devices are capable of handling tolerable noise margins that the MCU can handle. However, it does not completely mitigate the noise, making it vulnerable to PSVC leakage.

A promising avenue to defend against PSVC leakage is to develop the PCB in-house instead of using third-party development boards. The most critical thing would be designing the power supply rails such that they can mitigate the PSVC vulnerability. A designer can explore the following directions to mitigate the vulnerability.

- Use of discrete decoupling capacitor between the power supply lines (*VCC* and *GND*) [32]–[34] is expected to reduce the input voltage variations.
- Combinations of bypass capacitors and ferrite beads [35] is expected to reduce the variation in supply voltage.
- Designers can also use resistive edge termination [36] to reduce the input voltage variations. This can be achieved by inserting a resistor in parallel, connected between the signal and ground, with a value selected to create an effective parallel resistance when combined with the termination resistor.
- Designing the PCB to work with the lowest possible voltage will reduce the SNR and the risk of PSVC vulnerability as shown in Section IV-D.

Even after a complete redesign of the PCB, it is important to use our analysis framework since manufacturing tolerances of capacitors and other components may still induce enough resistance, inductance, and capacitance to leak information via PSVC side-channel, even on a perfectly calibrated design.

The success of our attack relies on the extraction of high-quality PSVC signals. In order to achieve a good success rate using our attack methodology, it requires high-quality measurement devices that can capture the computation of the device under attack with enough sampling rate. In the experiments, the *BlackPill* development board was the device that was operating at the highest frequency of $84MHz$. We were able to use both the Keysight DSOX1102G oscilloscope and the HackRF SDR to sample the traces from $84MHz$ and perform an end-to-end attack exploiting PSVC vulnerability. However, performing the attack on devices with higher frequencies requires high-quality measuring devices that can sample the signals at adequate frequency rates.

VI. Conclusion

In this paper, we introduced a novel vulnerability that can leak sensitive information through physical layer supply voltage coupling (PSVC) on off-the-shelf devices used in embedded systems. We proposed a methodology to evaluate different device configurations against PSVC vulnerability. We conducted four different case study experiments, where the first three case studies were focused on launching end-to-end attacks exploiting the PSVC vulnerability with different adversary capabilities and different device configurations. Experimental results revealed that PSVC vulnerability is present in off-the-shelf hardware components, and can be exploited to mount attacks in multiple ways including an extreme remote attack where the adversary does not need physical access to the device under attack. In the final case study, we performed an end-to-end attack while changing the supply voltage in the manufacturer-specified range. This experiment revealed that the effect of PSVC vulnerability in different voltage levels is affected by the signal-to-noise ratio and when the device is operating at the lowest specified operational voltage, performing an attack using the PSVC vulnerability requires much more effort than performing an attack in the highest voltage level. We also discussed potential methods that designers can follow to reduce the effect of PSVC leakage.

Acknowledgments

This work was partially supported by the grant from the National Science Foundation (CCF-1908131).

References

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

[2] M. G. Kuhn and R. J. Anderson, "Soft tempest: Hidden data transmission using electromagnetic emanations," in *Information Hiding: Second International Workshop*. Springer, 1998, pp. 124–142.

[3] S. K. Monfared, T. Mosavirik, and S. Tajik, "Leakyohm: Secret bits extraction using impedance analysis," in *ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 1675–1689.

[4] G. Camurati, S. Poeplau, M. Muench, T. Hayes, and A. Francillon, "Screaming channels: When electromagnetic side channels meet radio transceivers," in *ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 163–177.

[5] Y. Lyu and P. Mishra, "Maxsense: Side-channel sensitivity maximization for trojan detection using statistical test patterns," *ACM Transactions on Design Automation of Electronic Systems*, vol. 26, no. 3, 2021.

[6] 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.

[7] 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.

[8] T. Zhang, J. Park, M. Tehranipoor, and F. Farahmandi, "PSC-TG: RTL power side-channel leakage assessment with test pattern generation," in *ACM/IEEE Design Automation Conference (DAC)*, 2021, pp. 709–714.

[9] 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.

[10] 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 *IEEE VLSI Test Symposium*, 2019, pp. 1–6.

[11] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The em side—channel (s)," in *International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*. Springer, 2003, pp. 29–45.

[12] J. Longo, E. De Mulder, D. Page, and M. Tunstall, "Soc it to em: electromagnetic side-channel attacks on a complex system-on-chip," in *International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*. Springer, 2015, pp. 620–640.

[13] B. Nassi, E. Iluz, O. Cohen, O. Vayner, D. Nassi, B. Zadov, and Y. Elovici, "Video-based cryptanalysis: Extracting cryptographic keys from video footage of a device's power led," *Cryptology ePrint Archive*, 2023.

[14] F. Schellenberg, D. R. Gnad, A. Moradi, and M. B. Tahoori, "Remote inter-chip power analysis side-channel attacks at board-level," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2018, pp. 1–7.

[15] E. Danieli, M. Goldzweig, M. Avital, and I. Levi, "Revealing the secrets of radio-enabled embedded systems: on extraction of raw information from any on-board signal through rf," *Cryptology ePrint Archive*, 2023.

[16] Z. Shao, M. A. Islam, and S. Ren, "Your noise, my signal: Exploiting switching noise for stealthy data exfiltration from desktop computers," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 4, no. 1, pp. 1–39, 2020.

[17] 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.

[18] 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, no. 2, pp. 357–370, 2021.

[19] 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)*. IEEE, 2022, pp. 1–8.

[20] 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.

[21] C. Flynn, "I, for one, welcome our new power analysis overlords," *Presented at Black Hat USA*, vol. 2018, 2018.

[22] J. Robertson and M. Riley. (2018) The big hack: How china used a tiny chip to infiltrate us companies. [Online]. Available: https://rb.gy/lqsaxa

[23] T. Hudson, "Modchips of the state." 2019, the 35th Chaos Communication Congress. [Online]. Available: https://fahrplan.events.ccc.de/congress/2018/Fahrplan/events/9597.html

[24] O. Shwartz, A. Cohen, A. Shabtai, and Y. Oren, "Inner conflict: How smart device components can cause harm," *Computers & Security*, vol. 89, p. 101665, 2020.

[25] K. Arabi, R. Saleh, and X. Meng, "Power supply noise in socs: Metrics, management, and measurement," *IEEE Design & Test of Computers*, vol. 24, no. 3, pp. 236–244, 2007.

[26] D. Landman and B. Hackerspace. (2015) Arduino aeslib. [Online]. Available: https://github.com/DavyLandman/AESLib

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

[28] Keysight, "Keysight dsox1102g user manual," 2023. [Online]. Available: https://www.keysight.com/th/en/assets/7018-05520/data-sheets/5992-1965.pdf

[29] G. Scott, "Hackrf's documentation," 2023. [Online]. Available: https://hackrf.readthedocs.io/en/latest/index.html

[30] C. Cliffe, "Cubicsdr documentation," 2023. [Online]. Available: https://cubicsdr.com/

[31] H. Zhu, "Security analysis of hidden analog-domain vulnerabilities in digital electronic systems: A deep dive into power delivery networks," Ph.D. dissertation, Washington University in St. Louis, 2023.

[32] A. Weiler and A. Pakosta, "High-speed layout guidelines," *Texas Instruments*, pp. 1999–2007, 2006.

[33] I. Novak, "Comparison of power distribution network design methods: bypass capacitor selection based on time domain and frequency domain performances," *TF-MP3 at DesignCon*, pp. 6–9, 2006.

[34] O. Tereshchenko, F. J. K. Buesink, and F. B. J. Leferink, "Power isolation strategies to reduce pdn noise," in *Asia-Pacific International Symposium on Electromagnetic Compatibility (APEMC)*, 2013.

[35] B. R. Archambeault and J. Drewniak, *PCB design for real-world EMI control*. Springer Science & Business Media, 2013, vol. 696.

[36] I. Novak, "Reducing simultaneous switching noise and emi on ground/power planes by dissipative edge termination," *IEEE Transactions on Advanced Packaging*, vol. 22, no. 3, pp. 274–283, 1999.

Sahan Sanjaya is a second-year 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 encompass side-channel attacks, hardware security, pre-silicon validation, and post-silicon validation.

Aruna Jayasena is a Ph.D student in the Department of Computer & Information Science & Engineering at the University of Florida. He received his B.S. in the Department of Computer Science and Engineering at the University of Moratuwa, Sri Lanka, in 2019. His research focuses on systems security, hardware-firmware co-validation, test generation, trusted execution, side-channel analysis, and system-on-chip debug.

Prabhat Mishra is a Professor in the Department of Computer and Information Science and Engineering at the University of Florida. His research interests include embedded and cyber-physical systems, hardware security and trust, and energy-aware computing. He currently serves as an Associate Editor of ACM Transactions on Design Automation of Electronic Systems and ACM Transactions on Embedded Computing Systems. He is an IEEE Fellow, an AAAS Fellow, and an ACM Distinguished Scientist.