PowerScout: Security-Oriented Power Delivery Network Modeling for Side-Channel Vulnerability Analysis

Huifeng Zhu, *Student Member, IEEE,* Xiaolong Guo, *Member, IEEE,* Yier Jin, *Senior Member, IEEE,* and Xuan Zhang, *Member, IEEE*

Abstract—The growing complexity of modern electronic systems leads to the design of more sophisticated power delivery networks (PDNs). Similar to other system-level shared hardware resources, the on-board PDN unintentionally introduces side channels across design layers and voltage domains which are not explicitly specified in the functional specification. Recent works have demonstrated that the exploitation of the side channel can compromise the system security such as information leakage and fault injection. In this work, we systematically investigate the PDN-based side channel as well as potential countermeasures. To facilitate this goal, we develop PowerScout, a security-oriented PDN simulation framework that unifies the modeling of different PDN-based side-channel attacks. PowerScout performs a fast nodal analysis of complex PDNs at the system level to quantitatively evaluate the severity of side-channel vulnerabilities.

With the support of PowerScout, for the first time, we validate PDN side-channel attacks in the literature via simulation. Furthermore, we are able to quantitatively measure the security impact of PDN parameters and configurations. For example, towards information leakage, removing near-chip capacitors can increase intra-chip information leakage by a maximum of 23.23dB at mid-frequency range and inter-chip leakage by an average of 31.68dB at mid- and high-frequency range. Similarly, the optimal toggling frequency and duty cycle are derived to achieve fault injection attacks with higher success rate and more precise control. In addition, the vulnerabilities are evaluated when hiding-based countermeasures are implemented. Based on the evaluation, we can understand the optimal defense configuration and explore the trade-off between information leakage mitigation and power supply stability.

Index Terms—Power delivery network, power side channel, modeling framework, information leakage, fault injection.

1 Introduction

The power delivery network (PDN) is an indispensable component central to the correct operation of any electronic systems, as each functional unit of the system requires the delivery of a stable supply voltage and sufficient power. To satisfy the exponential demand for computing power, modern electronic systems are becoming increasingly complex, as is the sophistication of the PDNs in these systems. As a result, modern PDNs can supply multiple voltage domains and satisfy their distinctive requirements, such as supply voltage level, maximum load current, and voltage noise margin for supply reliability. For example, IBM's new generation 24-core POWER9 processor has ten different input supply voltages and supports hundreds of voltage domains [1]. Moreover, modern computing platforms often integrate several different modules such as CPUs, GPUs, FPGAs, and DRAMs on the same motherboard, requiring a shared hierarchical PDN to facilitate the distribution of supply voltage among the modules across the chip, package, and PCB levels. A sample PDN is illustrated in Figure 1 where the Xilinx Kintex-7 FPGA board [2] is used as an example.

Nonetheless, as a shared resource, PDNs create many pathways for unintended interactions and expose the system to various side-channel attacks [3], [4], [5], [6], [7]. Recent works have shown that many such vulnerabilities can be exploited remotely, making them especially potent security threats to modern electronic devices with ubiquitous connectivity [8], [9], [10]. For example, in infor-

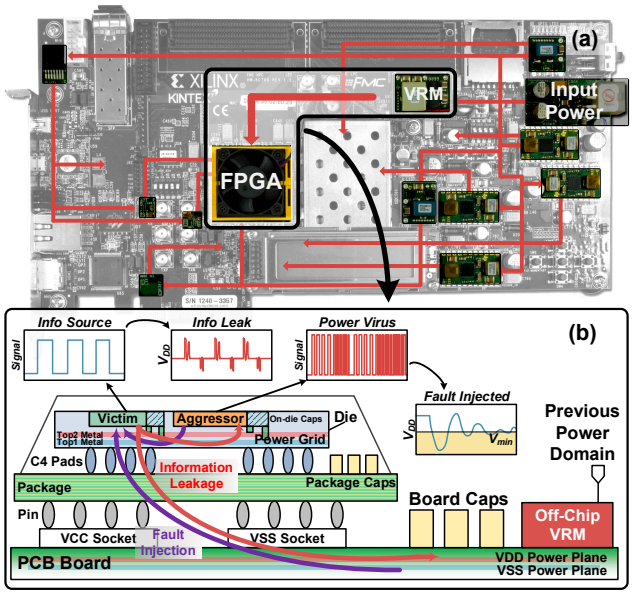

Figure 1: (a) The Xilinx Kintex-7 FPGA board PDN subsystem comprises of hierarchical VRMs (the boxes) and passive networks (the red lines), creating multiple voltage domains. (b) Single-stage PDN schematic and PDN-based side-channel attacks such as information leakage and fault injection.

- *Huifeng Zhu and Xuan Zhang (xuan.zhang@wustl.edu) are with the Department of Electrical and System Engineering, Washington University in St.Louis, MO, 63130, USA.*
- *Xiaolong Guo (guoxiaolong@k-state.edu) is with the Department of Electrical and Computer Engineering, Kansas State University, KS, 66506, USA.*
- *Yier Jin (yier.jin@ece.ufl.edu) is with the Department of Electrical and Computer Engineering, University of Florida, FL, 32611, USA.*

Manuscript received MMMM DD, YYYY; revised MMMM DD, YYYY.

mation leakage attacks, hackers can implement malicious voltmeters on FPGAs to steal sensitive information without physical access to the target systems [11], [12], [8]. PDN-based side channels can also be utilized to induce supply glitches (e.g., by implementing a power virus) in victim modules for denial-of-service (DoS) attacks [9] or differential fault analysis (DFA) [10] on cloud FPGAs. Along the emerging threats, countermeasures relying on information hiding such as active fence [13], are proposed to mitigate information leakage by introducing extra noise to PDN. However, these ad hoc experimental approaches, although useful in providing proof-of-concept demonstrations of certain PDN vulnerabilities, do not offer systematic and quantifiable guidance to discover new vulnerabilities or evaluate system resilience. As a consequence, it is urgently needed for a security-oriented modeling framework to accurately capture PDN behaviors that may lead to security vulnerabilities across multiple design layers and/or voltage domains.

Previously investigated PDN modeling and simulation tools mainly aim to estimate PDN characteristics (e.g., PowerSoC [14]), explore cross-voltage-domain PDN design space (e.g., Ivory [15]), and optimize PDN configurations (e.g., VoltSpot [16]). Existing tools tend to focus on the trade-off between performance, efficiency, and supply noise. They lack essential capabilities to perform specific side-channel vulnerability analysis. In this work, we propose PowerScout [17]—a unified PDN modeling framework that is able to perform a thorough side-channel vulnerability analysis by simulating a complete PDN system across multiple design layers (i.e., chip, package, board) and voltage domains[1].

Instead of developing new circuit-level models for PDN, our methodology focuses on attaining a balance between security interpretability and simulation accuracy by using frequency-domain analysis for vulnerability exploration and transient simulation for security validation. Thus, unlike previous coarse lumped PDN models [18], [19], [20], we are able to build a precise PDN model to quickly perform cross-domain nodal analysis, capable of characterizing information leakage between arbitrary nodes in the system. We abstract critical components at multiple layers of a PDN, and provide a voltage regulator module (VRM) model to capture bidirectional voltage domain interactions and three load models to serve different PDN-based side-channel attack scenarios and analysis objectives. The main contributions of this paper are listed as follows:

- We present a unified security-oriented PDN modeling framework named PowerScout. It can perform efficient evaluation of PDN side-channel vulnerabilities and systematic attack space exploration to guide secure PDN design and effective defense strategies.
- PowerScout can correctly predict the information leakage strength associated with PDN parameters and configurations. Our information leakage case study reveals $23.23dB$ and $31.68dB$ increase of intra- and inter-chip leakage from the removal of a near-chip capacitor, corroborating previous experimental results.
- We systematically explore the attack space of fault injection to identify the effective region with linear sensitivity to toggling frequency and duty cycle.
- We are able to evaluate the hiding-based countermeasures to understand the optimal configuration for information leakage mitigation. A sweet-spot balance between defense performance and power supply stability can be identified via PowerScout.

The rest of the paper is organized in the following way. Section 2 provides a background about PDN structure and emerging PDN-based side-channel attacks. We detail the PowerScout modeling framework in Section 3. The evaluation of information leakage attack and fault injection attack are introduced in Section 4 and Section 5, respectively. In Section 6, the optimal configuration and trade-off of the countermeasure relying on the active fence are explored. Finally, we discuss the related works in Section 7 and conclude the paper in Section 8.

2 Background

2.1 Power Delivery Network

Power Delivery Network (PDN) is an essential subsystem in modern electronic systems. One of this main roles of PDN is to keep the voltage across the chip pads stable [16]. Figure 1 (a) and (b) show a simplified PDN across multiple layers, from the board to the chip. It contains board-level VRMs, interconnects from the VRMs to the pads, on-chip power grids to distribute power locally to the die, and decoupling capacitors along various stages of the PDN. In a system, there are many devices with different voltage supply and power distribution requirements, hence multiple voltage domains are created, each with its own VRM to drive local supply voltages [15]. These VRMs form a tree structure where the upper node have a higher voltage. Between the hierarchical VRMs and chips is the board-level passive distribution network containing PCB wire lines, PCB planes, and board-level decoupling capacitors. Through the package-level sockets, pins, and C4 pads, power is supplied to the microelectronic chip, where a multilayer metal mesh forms the power grid that locally delivers power to each module inside the chip [18], [14], [21]. Decoupling capacitors are implemented on both the package and die to further mitigate supply noise.

2.2 PDN-Based Side-Channel Attacks

Emerging PDN-based security threats can be categorized into two major classes, i.e., information leakage and fault injection. These attacks utilize the intrinsic characteristics of the circuit-level behavior of the PDN, which can be captured by the supply voltage fluctuation (V_{drop}):

$$V_{drop} = IR + L\frac{di}{dt} \tag{1}$$

where the voltage drop comes from two sources: IR-drop due to the static current consumption of the modules and Ldi/dt-noise from dynamic current caused by switching activities. V_{drop} can propagate to other modules connected to the same PDN. Information leakage mainly utilizes the Ldi/dt-noise, and fault injection attacks can utilize both IR-drop and Ldi/dt-noise to generate power glitches.

1. We have open sourced the tool with the source code available at: https://github.com/xz-group/PowerScout

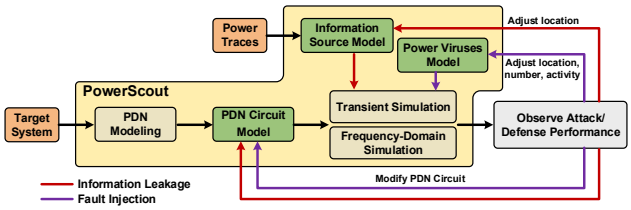

Figure 2: The workflow of PDN-based side-channel vulnerability analysis using PowerScout.

Figure 1 (b) shows the mechanisms of the two attacks. Information leakage exploits the deterministic relationship between the switching activities of digital circuits and their dynamic currents. The induced supply voltage fluctuations can further propagate to other modules connected to the same PDN. The same supply pattern occurs if the device performs the same operations while processing the same data. By recording this pattern (i.e., power/voltage traces) using on-chip or off-chip monitors, attackers can infer the information processed by the module. Recent works suggest implementing malicious on-chip voltmeters, such as ring oscillators (ROs) or time-to-digital converters (TDCs), to perform remote side-channel analysis in multi-tenant FPGAs. The power traces can be sampled by recording the oscillation frequency [11], [22] or signal propagation depth [12], [8]. Similarly, the PDNs can also be used as a medium for covert channel communications. Attackers may implement dedicated oscillating cells (e.g., LFSR [23]) as transmitters to generate information-modulated currents. The receivers can be modules that are sensitive to supply voltages.

Fault injection takes advantage of extreme supply fluctuations to violate the timing constraints and thus induce faults. A system crash can be achieved if the violations exist in a sufficient number of critical paths. The attackers normally create out-of-tolerance and/or cautiously controlled voltage drops by manipulating power-hungry blocks known as power viruses. For multi-tenant FPGAs, the attackers implement RO-based power viruses (ROPVs) and use toggling signals to control their activities [10], [24]. The oscillated RO can cause a large instant current draw when starting operating, thus inducing a voltage drop. As shown in [25], [26], PDN plays an important role in such power side-channel attacks. First, it is the fundamental medium of voltage fluctuations. Second, it intrinsically determines the performance of the attacks. A PDN design with side-channel vulnerability will compromise the security of the whole system. However, to the best of our knowledge, an effective framework supporting a systematic analysis of PDN vulnerability remains elusive.

3 Security-Oriented PDN Modeling

In this section, PowerScout framework and PDN modeling methodology are introduced. Here we assume the users/host (including both adversaries and defenders) of multi-tenant FPGA systems will use PowerScout. Threat model of previous work [11], [22], [12], [8] is adopted here, where users can implement modules on FPGAs and have physical access to the PCB to modify the PDN circuit. Adversaries can implement malicious on-chip voltmeters to perform information leakage attack (Section 4), or implement ROPVs to launch fault injection attacks (Section 5). Their purpose is to find the optimal settings (e.g., the locations of the malicious voltmeters) or PDN configurations (e.g., remove some on-board discrete capacitors) to maximize attack performance. On the other hand, the defenders will leverage ROPVs to implement an active fence to reduce the signal-noise-ration (SNR) of information leakage on the untrusted party's side (Section 6). In this paper, we aim at demonstrating that PDN fundamentally impacts the propagation of voltage drops among two points of the system, which directly affects both attack and defense performance. To achieve powerful attacks or defenses, users need to consider the properties of PDN and can use PowerScout to speedily explore the attack space or the defense space. PowerScout may also be used to conduct other explorations (e.g., design PDNs that are resistant to PDN-based attacks). But investigating such capabilities is out of the scope of this paper.

Figure 2 shows the workflow of PDN-based side-channel vulnerabilities analysis using PowerScout. We try to accelerate the loops shown in this figure, which is previously completed by many trial-and-error with time-consuming hardware experiments. To use PowerScout, users first need to model the PDN circuit of a target system (i.e., estimate the R, L, C values in Figure 3(b)). In our framework, individual components such as VRMs, discrete capacitors, and chips are separately modeled. Then these sub-models are connected according to the schematic of the system. We assume that the users will have access to both the schematic and bill-of-materials (BOM) of the target system to extract the topology and consisted components of the PDN circuit. The modeling methodology is detailed in Section 3.2.

Based on the PDN circuit model, two types of simulation can be performed: transient simulation and frequency-domain simulation. The former is for validating specific setup in the time domain, while the latter reveals the intrinsic properties of PDN. As will be shown in the following sections, insights into such properties facilitate finding a more powerful attack. For information leakage attacks, users also need to provide example power traces for transient simulation. Since this work focuses on analyzing PDN and is not for specific victim modules, their resistance against attacks is not considered. Users will adjust the attack/defense settings or PDN configurations based on the observed attack/defense performance (e.g., number of needed power traces in information leakage attack or maximum supply voltage drop in fault injection attack) and re-run the simulations. Note that for PDN configuration modifications, users modify the topology of the PDN circuit (i.e., add, remove, or replace sub-models) instead of tuning detailed parameters (e.g., R, L, C values) of the models.

3.1 PowerScout Framework

The diagram of PowerScout is shown in Figure 3 (a). PowerScout contains three main parts: the parameter panel, the PDN generator, and the vulnerability analyzer. Users define the PDN topology in the Python script (see Figure 4) and input PDN model parameters such as RLC values and the number of power grids of the chip when calling modules inside PowerScout. Users can also leverage the parameter

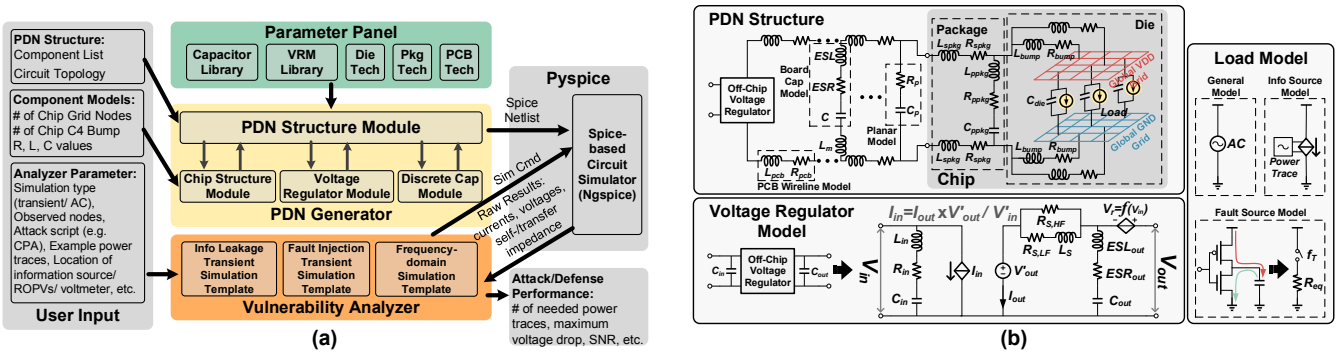

Figure 3: (a) The system diagram of the proposed PowerScout framework. (b) Security-oriented modeling of different PDN components.

panel, which contains the PDN parameters we abstracted from electronic component datasheets and technology libraries. Given these inputs, the developed PDN generator will automatically generate a full-system PDN netlist that specifies the complex hierarchical network. Meanwhile, users will call the simulation templates in the vulnerability analyzer. These templates are aimed to ease the simulation process by automating the configuration of the simulator and parsing the results, such that only hyperparameters are required. For example, when simulating the information leakage attacks using transient simulation, the users will define the locations of the information source and malicious voltmeter, input the example power traces, and provide an attack script (e.g., the script of correlation power analysis (CPA)). The information leakage transient simulation template will iterate the simulation process for each power trace. Based on the raw results (i.e., voltages and currents of every node of the PDN) reported by the simulator, results of interest will be extracted, followed by calling the attack script, calculating the number of needed power traces, and plotting the results (e.g., Figure 5 (b)).

In PowerScout, the induced voltage fluctuation $v(t)$ is computed by invoking the SPICE-level simulator, which performs a numerical nodal analysis that can be expressed in a simplified form.

$$v(t) = \int_0^t [Ce^{A(t-\tau)}B] \times i(\tau)d\tau \tag{2}$$

where $i(\tau)$ is the current consumption of the module; A, B, and C are the state-space matrices of the PDN. The values of the matrices depend on parameters and the topology of the PDN, as well as positions of the current source and observation point. With the PDN model, time-domain results can be obtained by computing the convolution between the state-space matrices and the current waveform. However, solving such a high-order differential equation is time-consuming. To efficiently explore the vulnerabilities in a large design space, in PowerScout, the PDN is evaluated in the frequency domain, and Equation 2 can be rewritten as:

$$v(t) = \mathcal{F}^{-1}[Z(f) \times I(f)] \tag{3}$$

where $\mathcal{F}^{-1}$ is the inverse Fourier operator, and $Z(f)$ and $I(f)$ are the spectra of the PDN impedance and current consumption, respectively. It is widely accepted that the PDN can be viewed as a linear time-invariant (LTI) system and inverse Fourier transform can be directly applied. Since $\mathcal{F}^{-1}$ is a linear operator, $Z(f)$ can influence the induced voltage fluctuation in a straightforward manner and serve as the quantitative metric for evaluating the vulnerability to information leakage or fault injection. $Z(f)$ can usually be obtained easily since AC analysis is supported by most SPICE engines, which allows frequency analysis for PDN vulnerability evaluation.

3.2 PDN Model Construction

The PDN models of PowerScout are composed of three parts: passive RLC network model, active voltage regulator model, and load model. The trade-off between security and accuracy is achieved by adapting distributed models while facilitating the AC analysis for fast nodal vulnerability evaluation.

Passive RLC Network Model: The structure of the PDN model is shown in Figure 3 (b). The model aims to fully reflect the power supply paths for critical devices with sufficient details. The board-level supply wireline is modeled as an inductor and a resistor, whose parameters depend on its length, width, and metal material characteristics. The PCB planes use the planar model with a lumped capacitor and resistor, since the distributed effect is minimal at this scale. Such parameters can be accurately estimated using industrial PCB design tools [27]. For the board-level capacitors, we model the characteristics of each capacitor. The frequency response of a single real capacitor is a band-pass filter instead of an ideal low-pass filter due to the parasitic effects [28]. The capacitor is thus modeled as the equivalent series inductor (ESL), equivalent series resistor (ESR), and an ideal capacitor. The values of ESL, ESR, and C can be obtained from [29]. Later in Section 4.3, we will show the necessity of individually modeling each capacitor by demonstrating the role of near-chip capacitors.

For the chip-level PDN, we use the widely accepted package and die models [18] and estimate values based on network analyzer measurements of chip-level PDN impedance. The package is modeled as an RLC network, and the C4 bumps are modeled as parallel RL pairs that connect the grid to the package. The on-chip grid (i.e., the die model) is represented as an RL network. The on-chip capacitance is evenly distributed between the VDD

and GND grids. The PDN structure and parameters have complex influence on A, B, and C of Equation 2. Overall, higher capacitance leads to a decrease in $Z(f)$ while higher resistance and inductance have the opposite effects.

Active Voltage Regulator Model: In previous works and industrial models, VRMs are typically modeled as a fixed voltage source [15], [18] or a fixed voltage regulator in series connected to the equivalent inductor, capacitor, and resistor [30], [27]. But this kind of model is not suitable for security-oriented PDN modeling since it ignores the interactions between different voltage domains. In PowerScout, we model the bidirectional interactions of different VRM topologies, including low-dropout regulators (LDOs), buck converters, and switched capacitor converters. The VRM model, shown in Figure 3 (b), contains a voltage source V_{out} and an RLC network. ESL_{OUT}, ESR_{OUT}, and C_{OUT} are based on the off-chip decoupling capacitor recommended in the datasheet; $R_{S,LF}$ is determined by the load regulation; and $R_{S,HF}$ and L_S are set to match the load transient response of the VRM. A dependent voltage source V_{PSRR} is used to model the influence of the input voltage fluctuations on the output. The frequency response of V_{PSRR} matches the reverse of the power supply ripple rejection specified in the datasheet. To capture the reverse influence by the output on the input, we observe that the output side of the VRM indirectly affects the input side by changing the current through the previous stage of the PDN. Therefore, we use a dependent current source ($I_{IN} = I_{OUT}V_{OUT}/V_{IN}$). For the input side RLC network, the values of ESL_{IN}, ESR_{IN}, and C_{IN} are also taken from the datasheet.

Analysis-dependent Load Models: In PowerScout, we provide three different load models that are suitable for different attack types and analysis objectives. As shown in Figure 3 (b), the *general model* is an ideal AC current source and is used for generalized side-channel vulnerability analysis. Using this model, frequency-domain analysis is performed, which can expose potential attacks, as shown in Equation 3. These attacks may need a large amount of time-domain experiments to successfully exploit the vulnerability.

For dedicated information leakage evaluation or validation of specific vulnerability via *information source model*, we use a time-varying current source to model the changing power consumption of the information source in a transient simulation, which is based on Equation 2. The current source takes a waveform file as input, which is generated by the power traces from other simulators, such as architectural simulators (e.g., GEM5 [31], Sniper [32], and McPAT [33]), or FPGA simulators (e.g., Xilinx ISE). The waveform captures the dynamic information leakage from the victim module in the time domain. The amplitude of the waveform is calculated according to the estimated static and dynamic power consumption and the supply voltage.

In fault injection attack, the behavior of the power-hungry modules can be modeled as switching capacitors, which is derived from the classic power consumption model ($P = \alpha_{0\to1}C_L V_{dd}^2 f_{clk}$) of the digital CMOS circuit [34]. The capacitor is the sum of the load capacitance of the malicious module. The current sink (i.e., the logic switch) is controlled by the toggling signal. When the time constant ($\tau = R_{PDN}C$) of switching capacitor is much smaller than toggling signal cycle, the *fault source model* can be further abstracted to a switched resistor, where the switch represents the toggling signal f_T. In Section 5, we will validate the abstraction by showing the consistence between simulation and measurement results of multi-tenant FPGA fault injection attacks that use ROPVs, where ROPVs are widely used to create power glitches.

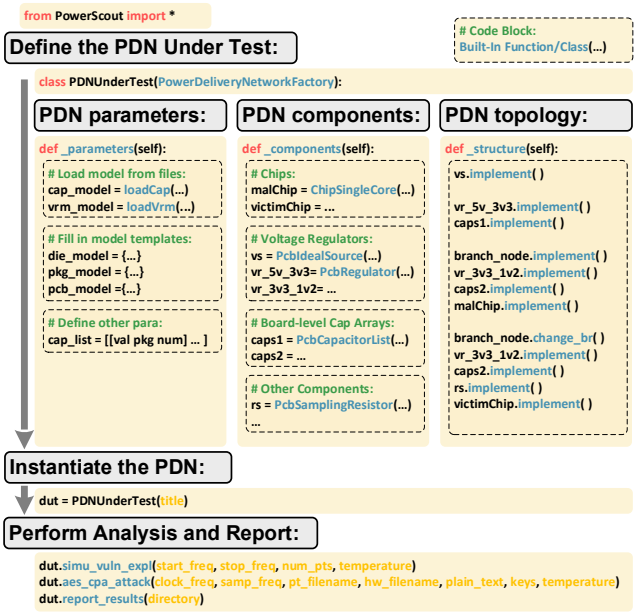

Figure 4: A code snippet of PowerScout. This code block can generate a SPICE netlist and then perform simulations. Users can easily modify the configuration of PDN and regenerate the netlist.

3.3 Simulation Infrastructure

PowerScout is built upon the open-sourced SPICE simulator Ngspice [35] and the Python package Pyspice [36]. Pyspice provides an API for Ngspice so that PowerScout can be configured in Python, which provides a convenient interface for future extension of high-level power management behaviors, such as dynamic voltage and frequency scaling (DVFS) [4], [3], [37], power budget controlling [5], etc.

Figure 4 shows an example code snippet of PowerScout, where an example PDN is built and simulated. The users first define the model parameters in *parameter()* function. PowerScout enables generating a complex PDN netlist with only a few user input parameters, such as the number of power grids and C4 pads and the loads' types and locations. The user can also assign the board-level capacitor array, and load technology parameters from the supplier model files. In *component()* function, the basic components of PDN are defined, where the PowerScout built-in module library is called to generate the SPICE netlist blocks.

The topology of PDN is then determined in the *structure()* function. The example PDN in Figure 4 has four voltage domains: the whole system is powered by an ideal 5V voltage source; this global supply voltage is converted to 3.3V to power two voltage regulators; each regulator outputs 1.2V to supply the corresponding chip. For each

voltage domain, there is a board-level decoupling capacitor array. A malicious sampling resistor is inserted besides the victim chip. Inputting a small Python code block can generate SPICE netlist with a few hundreds of lines, and it is easy for users to modify the configuration of PDN and regenerate the netlist, so that the PDN with different topologies can be efficiently evaluated. The generated SPICE netlist is simulated by Ngspice using the vulnerability analyzer. PowerScout uses transient simulation for dedicated information leakage or fault injection attacks, and uses frequency-domain simulation for nodal vulnerability analysis.

4 Information Leakage Attack Evaluation

In this section, we will demonstrate that PowerScout can predict the PDN vulnerability leading to information leakage attacks. Aided by the PowerScout, we provide insights on how to exploit PDN design parameters to maximize (or minimize) the information leakage. Our experiments focus on side-channel analysis attacks. Note that PowerScout is also suitable for other information leakage attacks, including covert channel communications.

4.1 Attack Primer

The traditional approach of PDN-based side-channel analysis is to measure the voltage of the sampling resistor inserted to the power supply rail of the victim chip. The authors in [12], [8], the perform both intra- and inter-chip remote power analysis attacks on the SAKURA-G board without external measurements. The board contains two Spartan-6 FPGAs on the same board. The authors implement a 128-bit advanced encryption standard (AES) module on one of the FPGAs and implement TDCs on either the same or the other FPGA as malicious on-chip voltmeters to measure the fluctuations of the power supply. The AES module runs at 24MHz and is based on a 32-bit datapath without side-channel protection. The authors first illustrate the remote intra-chip CPA attacks when the voltmeter is implemented on the same victim FPGA. They also successfully perform CPA attacks when the voltmeter is on the other FPGA, showing the vulnerabilities of inter-chip side-channel analysis.

4.2 PowerScout Configuration

To analyze information leakage using PowerScout, we build a PDN model configured with parameters extracted from SAKURA-G board, which is suitable for evaluating both intra- and inter-chip information leakage. The structure of PDN is based on the schematic of the SAKURA-G board, including hierarchical VRMs and the two FPGAs. For each capacitor, the model parameters are extracted from the component datasheets [29]. The parameters of the PCB, package, and die model [18] are listed in Table 1.

The vulnerabilities of information leakage attacks are systemically evaluated using PowerScout under multiple PDN configurations. To predict the performance of the side-channel analysis attacks, the information source model is implemented on the victim chip. The traces of the information source model are generated by Xilinx ISE power estimation, where we set the simulation interval adequately small to approximately represent the transient power consumption. The noise level of the PDN from the information source to the observation point is simulated, and a voltage source with Gaussian noise is accordingly implemented at the observation point. We record power traces from both intra- and inter-chip observation points and perform CPA attacks. In the vulnerability analysis, the general model is used. We perform AC analysis and observe the information strength, i.e., $Z(f)$ in Equation 3, on each node of the PDN. The information strength is defined as the amplitude of the voltage fluctuations induced by the unit information source current. A higher information strength means information is more easily leaked at the same noise level. Note that noise is not included in the general analysis since we focus on the worst case for the defender side, i.e., the fewest power traces needed in CPA attacks. Still, users have the flexibility to insert noise given the application scenarios.

Validation for the information leakage attack prediction by comparing prediction results with real-world experiments from prior work [12], [8] is presented in Figure 5. In this simulation, we first design an AES module in Xilinx ISE and set the target platform as Sakura-G board. Thus using the Xilinx ISE's power estimation function, we can capture the time-dependent power consumption waveform of the AES module in the clock-cycle resolution. We input power traces are inputted to the information source model and run the transient simulation in PowerScout. There are two observation points: one is located at the same chip as the AES module for intra-chip attacks, and the other point is on the FPGA chip without an AES module for inter-chip attacks. The upper panels show intra-chip CPA attack results from PowerScout and corresponding experiment measurements. During CPA attack, the correlation coefficient is iteratively computed between the power traces and the modeled power consumption. As the number of power traces increases, the correlation coefficient of the correct key guess can be distinguished from other guesses. After multiple tests, we find that removing the capacitors near FPGAs can significantly reduce the number of needed traces. This configuration is similar to [8], and the results shown in Figure 5 (a) and (b) are consistent. The bottom panels show comparative inter-chip CPA attack performances. Besides removing the near-chip capacitors, we find that more information will be leaked if we short the voltage regulators of the two FPGAs, where each FPGA is originally supplied by an individual voltage regulator. Comparing the results of experiment measurements in Figure 5 (c) with similar scenarios [12], PowerScout can also precisely predict the performance of information leakage attacks across several domains. Although the absolute values of the needed power traces of the two attacks are different due to the parameters setup, the relative values validate the developed PowerScout framework.

4.3 PowerScout Results and Discussion

Near-Chip Capacitors: The authors [12], [8] remove the near-chip capacitors without detailed explanation on how the removal of capacitors will impact the experimental results. Our PowerScout clearly reveals the reason. The values

Table 1: PDN Model Parameters for the Attack and Countermeasure Evaluations.

Parameter	SAKURA-G Board			ML605 Board			Radiona ULX3S Board		
	R	L	C	R	L	C	R	L	C
Z_{pcb}	$0.58m\Omega$	$0.09nH$	—	$58\mu\Omega$	$91.7pH$	—	$0.58m\Omega$	$0.09nH$	—
Z_{spkg}	$3.3m\Omega$	$0.5nH$	—	$0.55m\Omega$	$0.06nH$	—	$23m\Omega$	$0.37\mu H$	—
Z_{ppkg}	$1.8m\Omega$	$28pH$	$270nF$ $172.3nF$	$0.1m\Omega$	$2.8pH$	$52\mu F$	$1.1m\Omega$	$0.28nH$	$30nF$
Z_{bump}	$10m\Omega$	$0.32nH$	—	$20m\Omega$	$36pH$	—	$50m\Omega$	$4.5nH$	—
Z_{die}	$3m\Omega$	$2.91fH$	$5.3nF$	$25m\Omega$	$2.91fH$	$10nF$	$76m\Omega$	$18pH$	$0.1nF$

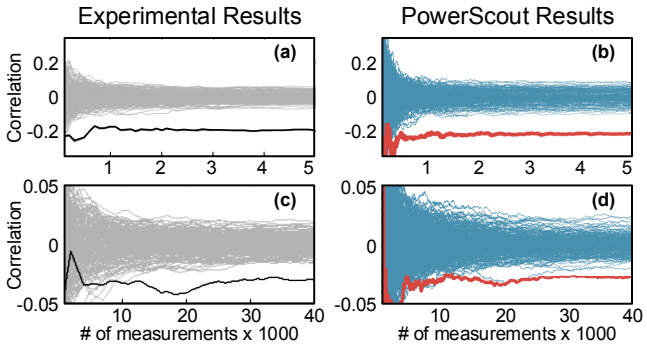

Figure 5: The experimental results for (a) intra-chip and (c) inter-chip CPA attacks [12], [8], and (b) (d) corresponding results from PowerScout.

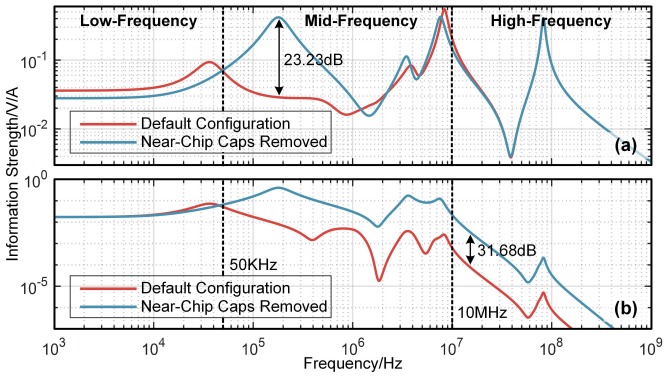

Figure 6: The information strength of (a) intra-chip (both attacker and victim are on the same chip) and (b) inter-chip (attacker and victim are on two different chips) information leakage under two configurations (i.e., with and without near-chip capacitors removed).

of board-level capacitors cover a wide range and can be split into two groups: distant large capacitors and near-chip small capacitors. As mentioned before, removing the near-chip capacitors can significantly increase the information leakage. Figure 6 compares the information strengths of the two PDN configurations. The upper part shows that the changes in intra-chip information leakage, where the information strength at mid-frequency increases as much as 23.23dB when near-chip capacitors are removed. From Equation 3, this removal can increase the induced voltage fluctuation for a given information source, and thus increase the information leakage at this frequency. However, due to the C4 bump parasitic inductance, near-chip capacitors have relatively small effects at the high frequency (see the overlapped plots at high-frequency range in Fig. 6(a)). For

Table 2: The information strength after passing through the voltage regulators between different domains.

Voltage Domain	V_{DD}	12V	5V	3.3V	1.2V
PDN Branch	B_{vict}	—	-15.2dB	-11.2dB	**0dB**
	B_{agg1}	[1]	-21.3dB	-101.9dB	
	B_{agg2}	[2]	-25.4dB	-87.0dB	-165.0dB

[1] ⌊: B_{vict} and B_{agg1} share the same 5V→3.3V voltage regulator.
[2] ⌊: B_{vict} and B_{agg2} come from one 12V→5V voltage regulator.

inter-chip information leakage, as shown in Figure 6 (b), near-chip capacitors significantly increase the information strength at both mid and high frequencies by an average of 31.68dB. Thus, near-chip capacitors play an important role in information leakage, although they account for only a small portion of the gross capacitance.

Cross-Domain Leakage: In [12], a small bridge shorts the power rails so that the core voltage of the main FPGA is provided by the same power supply as for the auxiliary FPGA. The authors of [12] claimed that this configuration resembles more typical industrial boards and did not provide an analysis on how this modification would affect the attack. Again, our PowerScout framework provides the reason of such a setting. The information leakage among multiple domains is presented in Table 2, where a PDN with three branches (B_{vict}, B_{agg1}, and B_{agg2}) and four supply voltage levels (12V, 5V, 3.3V, and 1.2V) is built. A voltage regulator is inserted between the adjacent voltage domains of one branch. B_{vict} and B_{agg1} share the same 5V→3.3V voltage regulator, while B_{vict} and B_{agg2} come from one 12V→5V voltage regulator. For other voltage domains, there are no direct connections. The information source is located at $B_{vict,1.2V}$, with an information strength of 0dB. Although the information decays significantly after passing through the voltage regulators, it still can leak through multiple voltage domains. It would be harder to detect the information if the observation point is structurally far from the information source, e.g. $B_{aggr2,1.2V}$ compared to $B_{aggr1,1.2V}$. For better attack performance, attackers need to reduce the distance from the source. This is achieved effectively in inter-chip attacks by directly connecting the power supply of two chips. The leakage increases by 57.9dB when the two FPGA chips share the same voltage regulator.

5 Fault Injection Attack Evaluation

In this section, we will evaluate fault injection attacks that use ROPVs. Rather than performing time-consuming experiments to evaluate the fault injection attacks, PowerScout

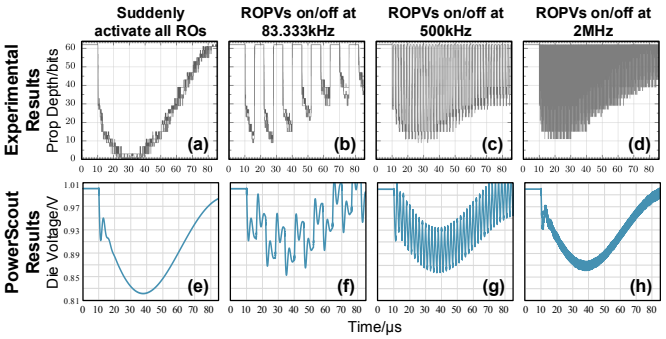

Figure 7: The experimental results of fault injection attacks [9], where the propagating depth is linear proportional to voltage drop (a)-(d), and corresponding results generated by PowerScout (e)-(h).

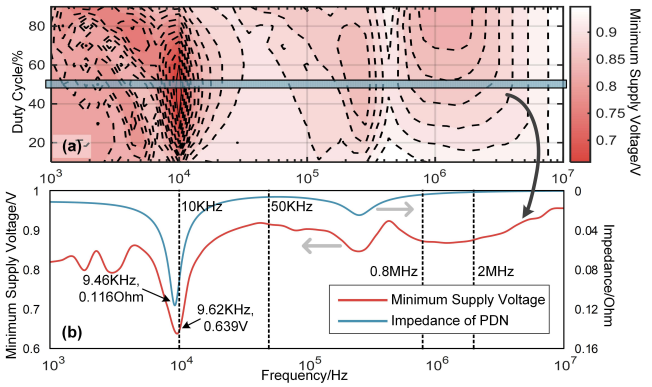

Figure 8: (a) The minimum die voltage under different toggling frequencies and duty cycles, and (b) the PDN impedance compared to the minimum die voltage at 50% duty cycle.

allows comprehensive and efficient attack space exploration via simulation. Our findings not only are consistent with those at previous works [9], [10], [24], but also provide better interpretability.

5.1 Attack Primer

In current multi-tenant FPGA fault injection attacks, the adversaries often implement ROPVs which are first introduced in [9], where the authors implement 18720 ROPVs (12.4% LUTs used) on an ML605 board and conducted several experiments to investigate the performance of attacks with different ROPVs toggling frequencies. Later, a more precise control of fault injection using ROPVs is investigated. For example, FPGAHammer [10] controls fault injection by changing the toggling frequency and duty cycle. An automated calibration algorithm is developed to iteratively tune these two parameters according to the results. Follow-up work also discusses the precise injection of faults by independently controlling two groups of ROPVs [24]. The ROPVs are first toggled with a period of fast-changing signals. Then the first group is kept active and the second group is disabled. After a specific delay, these two groups switch. In this way, attackers can induce a controllable period of time of stable voltage drop, which is sufficient for fault injection without crashing the system.

5.2 PowerScout Configuration

To systematically explore the attack space of fault injection attacks, we generate a PDN model using PowerScout and perform extensive experiments with different attack parameters. The structure of the model is based on the ML605 FPGA board schematic. For simplicity, we build only one stage of the supply voltage domain. We use the general load model for vulnerability analysis and use the fault source model to perform transient simulation. Since the oscillation frequency of ROPVs is usually much higher than the frequency of the toggling signal, its current consumption can thus be viewed as constant to increase the simulation speed without much accuracy loss. The detailed parameters of the generated PDN model are listed in Table 1. We believe that both the toggling frequency and duty cycle can affect the induced voltage drop. Therefore, we first simulate the attack in PowerScout while sweeping the toggling frequency from 1kHz to 10MHz and varying the duty cycle ranging from 10% to 90%. For each configuration, the minimum supply voltages when the fluctuations become periodic are recorded. Later, two groups of ROPVs with different control timings are evaluated to design the methodology of precise fault injection.

5.3 PowerScout Simulation Results

The consistency of the fault injection results between PowerScout and experiment measurements [9] is shown in Figure 7, where we run the transient simulation when the ROPVs are toggled at different frequencies, and observe the supply voltage drop on the FPGA chip. The upper row shows the induced voltage fluctuations versus time when the toggling signals are one pulse, (83.3kHz,50%), (500kHz,50%), and (2MHz,50%). Compared to the experimental results that are also based on the ML605 board, even though the experimental results contain more glitches due to the oscillation of ROPV and measurement noise, it is clear that with the same toggling signals, the supply waveform envelopes match between the experiment and PowerScout simulation.

The attack space exploration for a single group of ROPVs is presented in Figure 8, where the heat map shows the minimum die voltage under different toggling frequencies and duty cycles. Generally, the toggling frequency and duty cycle do not have linear influences on the voltage drop. However, when the toggling frequency ranges between 10KHz and 50KHz, the fault injection performance is linear with the toggling frequency. For some regions (e.g., around 0.8MHz and 2MHz), the voltage drop is proportional to the duty cycle. Moreover, there exists the most efficient toggling frequency which can induce the maximum voltage drop. At this frequency, the duty cycle does not have much influence on the fault injection performance. Figure 8 (b) shows the minimum supply voltage versus toggling frequency when the duty cycle is 50%, and also shows the simulated PDN impedance. The resonant frequency of the PDN impedance is almost the same as the most efficient toggling frequency, so that by using the resonant frequency and corresponding impedance (9.46KHz,0.116Ohm) from the vulnerability analysis, we can effectively predict the maximum fault

Algorithm 1 Optimal toggling timing exploration

1: Power virus toggling signal $T_{PV} \leftarrow$ enable
2: **while** power supply voltage sample V_i **do**
3: **if** $\frac{V_i+V_{i-2}}{2} < V_{i-1}$ **and** $\frac{V_{i-1}+V_{i-3}}{2} > V_{i-2}$ **and** T_{PV} **is** enable **then**
4: $T_{PV} \leftarrow$ disable, Wait Δt
5: **end if**
6: **if** $\frac{V_i+V_{i-2}}{2} > V_{i-1}$ **and** $\frac{V_{i-1}+V_{i-3}}{2} < V_{i-2}$ **and** T_{PV} **is** disable **then**
7: $T_{PV} \leftarrow$ enable, Wait Δt
8: **end if**
9: **end while**

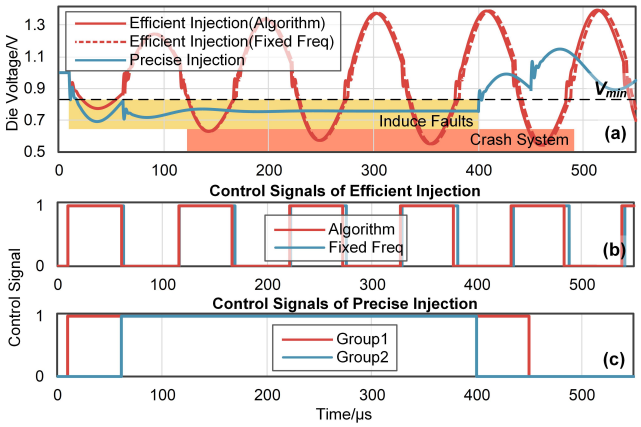

Figure 9: (a) The simulated supply waveforms under different fault injection configurations; the corresponding control signal timing for (b) efficient fault injection and (c) precise control with two activation groups.

injection performance (9.62KHz,0.639V). Our findings are consistent with the results from previous works [9], [10], [24]. Furthermore, the PowerScout framework provides a key insight for more efficient exploration of the attack space.

Efficient Fault Injection: With the most efficient toggling signal $(9.46KHz, 50\%)$, we find that the ROPVs are toggled when the supply voltage recovers at the maximum speed. Based on this finding, we can perform the most effective fault injection without modeling the PDN or knowing the resonant frequency. The procedure is shown in Algorithm 1. One can continuously monitor the voltage fluctuations with an on-chip voltmeter and calculate the voltage change rates of the past four sample points in real time. When the maximum or minimum voltage change rates are detected, i.e., the absolute value of the dynamic current di/dt achieves its maximum, the status of ROPVs will be flipped. The validation of the algorithm is illustrated in Figure 9. The timings of the toggling signals generated by the algorithm and the fixed resonant frequency are almost the same (see Figure 9 (b)), as well as the induced voltage fluctuations (the red lines in Figure9 (a)).

Precise Fault Injection Control: In some cases, instead of crashing the system, attackers intend to inject controllable faults. During our exploration, we find several ways to precisely control fault injection. Besides, using the number of ROPVs or the duty cycle of the toggling signal, enabling two groups of independently controlled ROPVs can create a stable voltage drop for a controllable time duration. Figure 9 (c) illustrates the control methodology and the performance is shown as the blue line in Figure 9 (a). One group is first activated, but the other group is not activated until the supply voltage recovery speed achieves the maximum, which is the same as the timing found by Algorithm 1. When all groups are enabled, the generated voltage drop becomes stable and is proportional to the number of ROPVs. The duration of the voltage drop is manipulated by the control signal. At the end of the attack, one group is deactivated first, and then the other one is deactivated after a delay to prevent the voltage fluctuations from inducing more faults. Note that the numbers of ROPVs of the two groups should be the same.

6 Active Fence-based Information Leakage Mitigation Evaluation

We have validated the accuracy of the PowerScout framework; in this section, we will illustrate that in addition to evaluating PDN vulnerabilities, PowerScout can efficiently and comprehensively explore the countermeasure performance under different configurations. Specifically, we will systemically evaluate an emerging technique named active fence [13], which utilizes ROPVs as a noise source to mitigate the information leakage. Furthermore, we show that a trade-off between the defense performance and power supply stability can be identified with the help of PowerScout.

6.1 Defense Primer

There are two main strategies for mitigating information leakage: masking- and hiding-based countermeasures. The former is implemented on the algorithmic level and aims to reduce the correlation between sensitive data and side-channel information. Hiding-based solutions focus on the electrical level and aim at reducing the information seen by the adversaries, including either decreasing the information strength at the source end or adding extra noise to the leaked information. In recent work [13], an active fence-based hiding countermeasure is implemented in a multi-tenant FPGA on an Radiona ULX3S board [38] to mitigate intra-chip information leakage. The authors implement 672 ROPVs as an active fence between the victim AES module and the malicious TDC-based voltmeter. The AES and voltmeter are operated at 12.5MHz and 100MHz, respectively. Other than injecting faults, these ROPVs are used as a noise source to decrease the SNR. The number of activated ROPVs during each clock cycle is controlled by a random number generator, thus inducing random voltage fluctuations on the PDN. The authors perform the intra-chip CPA attacks under different configurations and illustrate that with an active fence implemented, two orders of magnitude more power traces are needed for a successful attack. The main advantages of active fences are the generality and simplicity.

6.2 PowerScout Configuration

To comprehensively evaluate the defense performance and to explore the optimal active fence configuration, a PDN model of the Radiona ULX3S board is generated based

on its schematic. Similar to Section 5, we build only one stage of the supply voltage domain for simplicity. The die-level PDN is abstracted with 20-by-20 nodes and 3-by-3 C4 bumps for both power supply and ground. The C4 bumps are evenly distributed on the die. We adapt the general load model representing both information and noise sources for defense performance analysis. To calculate the supply voltage ripple, we use the modified fault source model in the transient simulation. The amplitude of the current draw of the fault source model is a random number sequence. The random number sequence follows the uniform distribution according to [13]. The generated current has a flat spectrum at frequencies far lower than the sampling frequency. Thus, the noise source can be viewed as white noise and the transient simulation can be consistent with the performance evaluation using the general load model. The detailed parameters of the PDN model are listed in Table 1.

6.3 PowerScout Results and Discussion

We have validated the PowerScout framework in Section 4 and 5 by comparing the simulation results and hardware measurements. In this section, we focus on systemically analyzing the intra-chip information leakage under different configurations including various information sources as well as noise source locations (see Figure 10). The X-axis and Y-axis of Figure 10 are the numbers of die power grids in X and Y directions, respectively. The contours and the colored fill-in represent the signal-to-noise ratio (SNR) of the leaked information. Compared with using information strength to evaluate the vulnerability, the SNR is suitable for capturing the information leakage with various levels of noise. With lower SNR, the side-channel analysis attacks are more difficult to succeed. SNRs are computed by considering the information strength at the clock frequency of AES module (i.e., 12.5MHz) and its decayed harmonics (25MHz, 37.5MHz, 50MHz, etc.)[2]. The information and noise at each node are normalized to the information source, and the amplitude of the noise floor is set to be 0.001, thus the SNR of the information source is $60dB$. In the experiments, the victim is assumed to possess the resources within $x+y \leq 21$, and the attacker owns the other half die area. We ensure that the SNR patterns are consistent with previous work [39] but our work provides more insights.

Information Source Placement Strategy: Figure 10 (a)-(c) illustrate that the location of the information source affects the intra-chip information leakage. In these subfigures, the locations of the information sources (i.e., green stars in the figures) are set to be $(5, 5)$, $(7, 7)$, and $(10, 10)$, respectively. As the information source moves from the corner to the center of the chip, the SNR of leaked information is increasing. The worst case (i.e., the highest SNR) is achieved when the information source is located at $(10, 10)$. At this point, the victim module can draw current from most power grid nodes of the die, thus the information can be leaked to the whole chip. Note that the contours are not in a regular pattern due to the existence of the C4 bumps connecting the die and the package. For the nodes connected to C4 bumps (e.g., $(16, 16)$), the SNRs are relative low since these nodes can draw extra current from the package, reducing the voltage fluctuations. We find that for multi-tenant FPGAs, the sensitive modules should be placed at the corner of the chip to avoid information leakage. Meanwhile, we notice that without information leakage mitigation techniques, the overall SNR of leaked information is high. For example, in Figure 10 (a), although the information source is located at the corner, the point with the lowest SNR still achieves $57.7dB$.

Effective and Efficient Active Fencing: Figure 10 (d)-(i) show that implementing an active fence can effectively reduce the SNR of leaked information, and different fencing configurations can affect the efficiency of the countermeasure. In these subfigures, the location of the information source is set to be $(5, 5)$ and the red triangles are the noise source, with bigger triangles indicating higher noise amplitudes. In Figure 10 (d), by implementing a noise source at $(5, 8)$ with 0.001 amplitude (i.e., $(5, 8, 0.001)$), the SNRs seen by the attackers are decreased by $5.1dB$. In addition, in Figure 10 (e) where the noise source is implemented at $(5, 10, 0.001)$. Compared with Figure 10 (d), the configuration of Figure 10 (e) has better information leakage mitigation with an overall lower SNR. We conclude that to have a better countermeasure performance, the noise sources need to be placed at a distance from the information source, which makes noise sources equivalently closer to the potential malicious sensors and more powerfully decreases SNR on the attacker side. Note that in this evaluation, the amplitudes of noise sources are set to be small for better graphing. The SNR contour patterns hold with higher amplitude. As shown in Figure 10 (f), a noise source is implemented at $(5, 10, 0.002)$, which causes a similar SNR pattern (except values) compared with Figure 10 (e).

In Figure 10 (g)-(i), instead of placing one noise source, we explore more efficient configurations with multiple groups of noise sources. The sum of the amplitude of noise sources in each subfigure is 0.002, the same as Figure 10 (f). In Figure 10 (g) the noise sources are at $(4, 10, 0.001)$ and $(10, 4, 0.001)$, respectively. Comparing with Figure 10 (g) and Figure 10 (f), we find splitting one noise source with high amplitude into two groups can increase the fencing performance. In Figure 10 (h), the noise sources are moved to $(5, 10, 0.001)$ and $(10, 5, 0.001)$, decreasing the distance between the two noise sources. It can be seen that the fencing performance is further increased, indicating that the noise sources can be placed in the center area of the chip with proper distance to achieve good performance. After several iterations, we find an outstanding active fence configuration. As shown in Figure 10 (i), the noise source is spitted into three groups: $(9, 10, 0.0005)$, $(10, 9, 0.0005)$ and $(10, 10, 0.001)$. In this configuration, we utilize all three strategies found in previous explorations: (1) implement multi-group noise sources; (2) keep a proper distance between the noise source and the information source; and (3) place the noise sources in the central part of chip. The resulting active fence configuration can effectively and efficiently reduce information leaking on the adversarial side (i.e., die area $x + y > 21$).

2. The decay factor is calculated as $V = V_0 \times (f/f_0)^2$, where V_0 is the strength of information source, $f_0 = 12.5MHz$, and f is the frequency of harmonics.

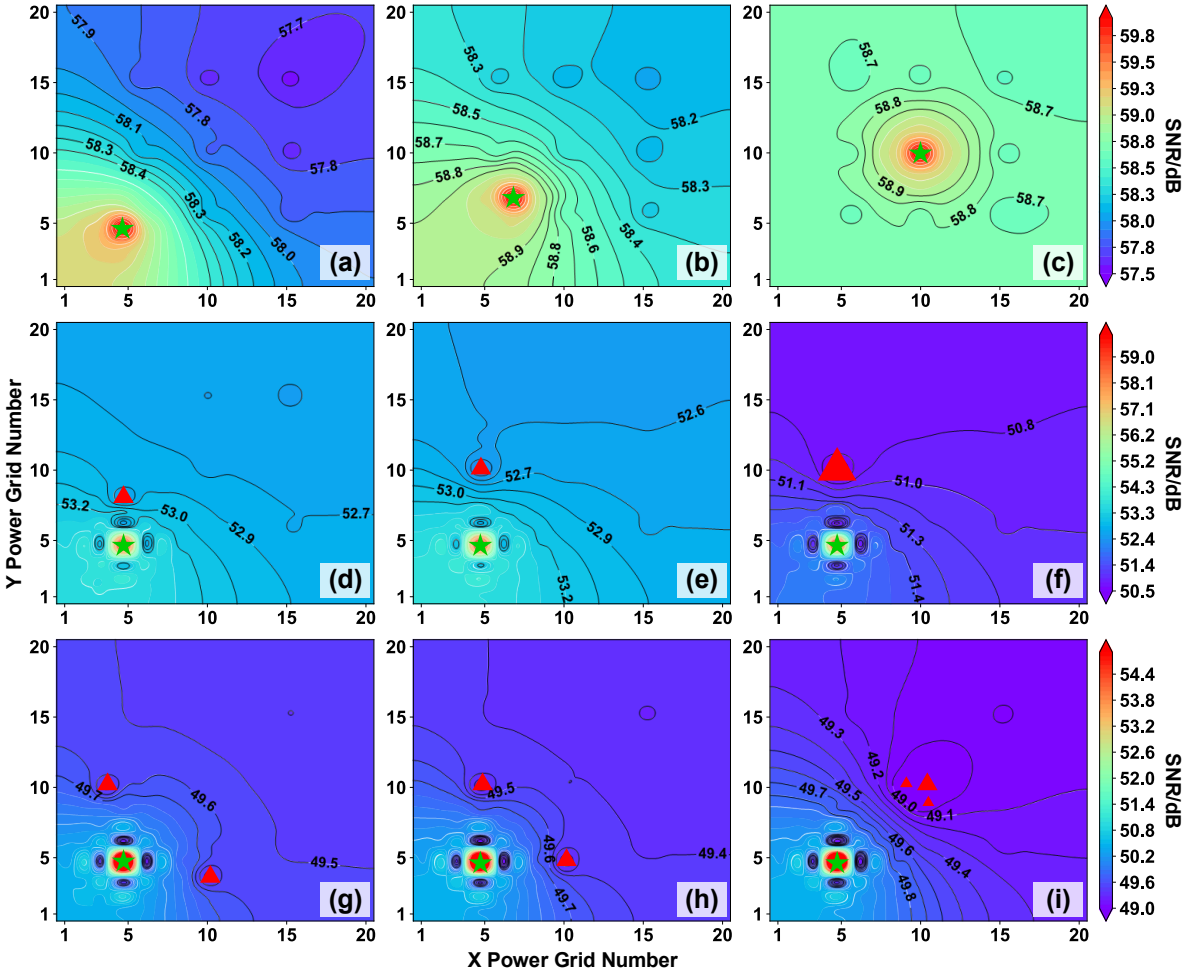

Figure 10: The evaluation of SNRs under different information and noise sources configurations. Lower SNRs indicate less leaked information. The green stars are the locations of information source, and the red triangles are the locations of RO.

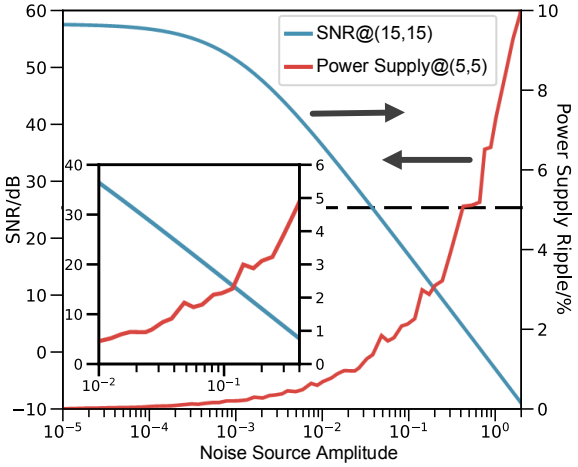

Figure 11: The trade-off between the defense performance (blue line) and power supply stability (red line).

Trade-Off between Defense Performance and Power Supply Stability: As a side effect of active fence-based countermeasures, the induced power supply noise inevitably increases the power supply ripple, deteriorating the power supply stability. A subtle balance needs to be achieved to avoid injecting faults when mitigating information leakage. Figure 11 shows this trade-off simulated by PowerScout, where the X-axis is the amplitude of the noise source in the frequency domain[3]. The blue line (corresponding to the left Y-axis) is the SNR seen by attackers at $(15, 15)$, and the red line (corresponding to the right Y-axis) is the power supply ripple. The power supply ripple is calculated as the percentage of the maximum voltage drop compared to the DC supply voltage. The common tolerance of supply noise ripple in commercial products is around $\pm 5\%$. From the figure, with higher noise source amplitude, the SNR decreases at the cost of higher power supply ripple. The SNR can be decreased by more than $50dB$ by setting the amplitude of noise at 0.4 which is the maximum available amplitude. Note that in real scenarios, power supply ripples are also contributed from other modules. Using PowerScout, we can achieve the trade-off according to the available tolerance and set the proper noise source amplitude to attain better defense performance while maintaining the required power supply stability.

3. The induced noise ripple is calculated in transient simulation. Note that the amplitude of noise source $i(t)$ in time domain is different from its power spectrum density $S_i(f)$ in frequency domain. The relationship is calculated by the Fourier transform of the auto-correlation of the $i(t)$, as: $S_i(f) = \int_{-\infty}^{+\infty} E[i(t)i(t+\tau)]e^{-j2\pi f\tau}d\tau$, where $E[\cdot]$ is the expectation operator.

7 Related Works

Table 3 lists the feature comparison between PowerScout and related works including PDN modeling and simulation framework, and simulation methods. Ivory [15] and PowerSoC [14] are two PDN design space exploration frameworks for the applications with both on-chip and off-chip VRMs. They can analyze the area overhead, efficiency, transient voltage drop, and stability for a given PDN configuration. VoltSpot [16] is a pre-RTL PDN model framework for optimizing chip I/O pads configuration analysis. It embeds fine-grained chip-level PDN models and supports transient supply voltage noise simulation when combining architecture-level tools GEM5 and McPAT. Later, an extended version is proposed [41] with the capability of simulating voltage-stacked PDN in multi-layer 3D IC. Since most frameworks [14], [15], [16] are not security-oriented, the authors in [26], [40] propose simulation methods aiming at PDN-based side-channel analysis attacks. The comparison between simulation and hardware experimental results validates the proposed model. However, the authors do not provide a framework for further analyzing the impact of PDN design on power side channel vulnerabilities.

8 Conclusion

In this paper, we present a security-oriented PDN modeling framework named PowerScout. Focusing on the vulnerability of PDN itself, we enable cross-domain nodal analysis by providing a precise and unified PDN model. Different from previous frameworks, the developed PowerScout enables full system-level simulation by bidirectional VRM model. Moreover, by considering the effects of both distributed on-board capacitors and the on-chip power grid, PowerScout achieves high accuracy in its simulation of the PDN. Multiple PDN side-channel vulnerability simulations are demonstrated with the proposed analysis-dependent load models. Having a user-friendly interface, PowerScout can easily generate complex PDNs and perform thorough attack space exploration. In addition to transient simulation, PowerScout can perform fast system-level nodal vulnerability analysis via frequency-domain simulations. Programmed in Python, PowerScout also has good compatibility with other frameworks. We show that PowerScout can successfully predict the performances of both information leakage attacks and fault injection attacks. Hiding-based countermeasures for information leakage can also be evaluated to increase system resistance.

In the future, architectural power efficiency strategies, such as DVFS [42], thermal controlling [43], instruction throttling [44], etc., can be modeled in PowerScout to facilitate cross-layer (i.e., circuit, architecture, software) side channel vulnerability evaluation. Security-aware power delivery system design can be explored at the early stage of system designing. PowerScout can also be applied to strengthen traditional test methods (e.g., test vector leakage assessment (TVLA) [45]).

Acknowledgments

Portions of this work were funded by National Science Foundation (NSF) Award CNS-1739643, Semiconductor Research Corporation (Task No. 2810.003 through UT Dallas' Texas Analog Center of Excellence), DARPA and the Office of Advanced Scientific Computing Research, US Department of Energy. The views expressed in the paper are the opinions of the authors and do not represent the official positions of DARPA, Pacific Northwest National Laboratory, the US Department of Energy, nor the US Government. Pacific Northwest National Laboratory is operated by Battelle for US Department of Energy under contract DE-AC05-76RL01830.

References

[1] C. Gonzalez *et al.*, "The 24-core power9 processor with adaptive clocking, 25-gb/s accelerator links, and 16-gb/s pcie gen4," *IEEE Journal of Solid-State Circuits*, 2017.

[2] Xilinx, "Kc705 evaluation board for the kintex-7 fpga - user guide," https://www.xilinx.com/support/documentation/boards_and_kits/kc705/ug810_KC705_Eval_Bd.pdf, 2019, accessed July 10, 2019.

[3] A. Tang, S. Sethumadhavan, and S. Stolfo, "{CLKSCREW}: exposing the perils of security-oblivious energy management," in *26th {USENIX} Security Symposium ({USENIX} Security 17)*, 2017, pp. 1057–1074.

[4] L. Bossuet *et al.*, "Dvfs as a security failure of trustzone-enabled heterogeneous soc," in *25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2018.

[5] S. K. Khatamifard, L. Wang, A. Das, S. Kose, and U. R. Karpuzcu, "Powert channels: A novel class of covert communicationexploiting power management vulnerabilities," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2019, pp. 291–303.

[6] I. Levi, D. Bellizia, D. Bol, and F.-X. Standaert, "Ask less, get more: Side-channel signal hiding, revisited," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 4904–4917, 2020.

[7] F. S. Hossain, M. Shintani, M. Inoue, and A. Orailoglu, "Variation-aware hardware trojan detection through power side-channel," in *2018 IEEE International Test Conference (ITC)*. IEEE, 2018, pp. 1–10.

[8] F. Schellenberg *et al.*, "An inside job: Remote power analysis attacks on fpgas," in *IEEE Design, Automation & Test in Europe Conference & Exhibition*, 2018.

[9] D. R. Gnad *et al.*, "Voltage drop-based fault attacks on fpgas using valid bitstreams," in *Field Programmable Logic and Applications*, 2017.

[10] J. Krautter *et al.*, "Fpgahammer:remote voltage fault attacks on shared fpgas,suitable for dfa on aes," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018.

[11] M. Zhao *et al.*, "Fpga-based remote power side-channel attacks," in *IEEE Symposium on Security and Privacy (SP)*, 2018.

[12] F. Schellenberg *et al.*, "Remote inter-chip power analysis side-channel attacks at board-level," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2018.

[13] 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.

[14] X. Wang *et al.*, "An analytical study of power delivery systems for many-core processors using on-chip and off-chip voltage regulators," *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2015.

[15] A. Zou *et al.*, "Ivory: Early-stage design space exploration tool for integrated voltage regulators," in *ACM Proceedings of the 54th Annual Design Automation Conference*, 2017.

[16] R. Zhang *et al.*, "Architecture implications of pads as a scarce resource," in *IEEE International Symposium on Computer Architecture*, 2014.

[17] H. Zhu, X. Guo, Y. Jin, and X. Zhang, "Powerscout: A security-oriented power delivery network modeling framework for cross-domain side-channel analysis," in *2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. IEEE, 2020, pp. 1–6.

[18] M. S. Gupta *et al.*, "Understanding voltage variations in chip multiprocessors using a distributed power-delivery network," in *IEEE Proceedings of the conference on Design, automation and test in Europe*, 2007.

Table 3: Feature comparison between PowerScout and related works.

	Ivory[15]	PowerSoC[14]	VoltSpot[16]	[40]	[26]	**PowerScout (This work)**
Security-Oriented	No	No	No	Yes	Yes	**Yes**
Programming Language	C++	C++	C	N/A[1]	N/A[1]	**Python**
Modeling Level	PCB-Level[2]	PCB-Level[2]	Chip-Level	PCB-Level[2]	Chip-Level	**System-Level**
PCB PDN Model Type	Lumped	Lumped	-	Lumped	-	**Distributed**
Chip PDN Model Type	Distributed	Distributed	Distributed	Lumped	Distributed	**Distributed**
Transient Simulation	Yes	Yes	Yes	Yes	Yes	**Yes**
Frequency-Domain Simulation	Yes	No	No	No	No	**Yes**
Analysis Type	N/A	N/A	N/A	Information Leakage	Information Leakage	**Information Leakage Fault Injection Attack Nodal Vulnerability Evaluation Hiding-based Countermeasure**

[1] This work provides a simulation method instead of a framework.
[2] It contains both the PCB-level modeling with one-stage voltage domain and the chip-level modeling.

[19] J. Leng, Y. Zu, and V. J. Reddi, "Gpu voltage noise: Characterization and hierarchical smoothing of spatial and temporal voltage noise interference in gpu architectures," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2015, pp. 161–173.

[20] J. Leng, T. Hetherington, A. ElTantawy, S. Gilani, N. S. Kim, T. M. Aamodt, and V. J. Reddi, "Gpuwattch: Enabling energy optimizations in gpgpus," *ACM SIGARCH Computer Architecture News*, vol. 41, no. 3, pp. 487–498, 2013.

[21] A. Zou, J. Leng, X. He, Y. Zu, C. D. Gill, V. J. Reddi, and X. Zhang, "Voltage-stacked gpus: a control theory driven cross-layer solution for practical voltage stacking in gpus," in *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2018, pp. 390–402.

[22] I. Giechaskiel, K. B. Rasmussen, and K. Eguro, "Leaky wires: Information leakage and covert communication between fpga long wires," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018, pp. 15–27.

[23] S. Kutzner *et al.*, "Trojanus:an ultra-lightweight side-channel leakage generator for fpgas," in *IEEE International Conference on Field-Programmable Technology(FPT)*, 2013.

[24] D. Mahmoud *et al.*, "Timing violation induced faults in multi-tenant fpgas," in *Design,Automation Test in EuropeConferenceExhibition*, 2019.

[25] X. Wang *et al.*, "Role of power grid in side channel attack and power-grid-aware secure design," in *50th Design Automation Conference*, 2013.

[26] J. Yang *et al.*, "Power supply noise aware evaluation framework for side channel attacks and countermeasures," in *IEEE International Conference on Field-Programmable Technology (FPT)*, 2014.

[27] Intel, "Using the altera pdn tool to optimize your power delivery network design," https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/an/an750.pdf, 2015, accessed April 30, 2020.

[28] S. Zhao *et al.*, "Frequency-domain power delivery network self-characterization in fpgas for improved system reliability," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 11, pp. 8915–8924, 2018.

[29] Murata, "Simsuring capacitor models," https://ds.murata.co.jp/simsurfing/mlcc.html?lcid=en-us, 2019, accessed July 10, 2019.

[30] M. Kar *et al.*, "Reducing power side-channel information leakage of aes engines using fully integrated inductive voltage regulator," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 8, 2018.

[31] N. Binkert *et al.*, "The gem5 simulator," *ACM SIGARCH computer architecture news*, vol. 39, no. 2, pp. 1–7, 2011.

[32] T. E. Carlson *et al.*, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations," in *International Conference for High PerfoSrmance Computing, Networking, Storage and Analysis (SC)*, Nov. 2011, pp. 52:1–52:12.

[33] S. Li *et al.*, "Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2009, pp. 469–480.

[34] A. P. Chandrakasan *et al.*, "Minimizing power consumption in digital cmos circuits," *Proceedings of the IEEE*, pp. 498–523, 1995.

[35] H. Vogt *et al.*, "Ngspice users manual version 31," http://ngspice.sourceforge.net/docs/ngspice-31-manual.pdf, 2019.

[36] F. Salvaire, "Pyspice," https://pyspice.fabrice-salvaire.fr, 2019.

[37] P. Qiu, D. Wang, Y. Lyu, and G. Qu, "Voltjockey: Breaching trustzone by software-controlled voltage manipulation over multi-core frequencies," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 195–209.

[38] Radiona, "Ulx3s: A powerful ecp5 board for open source fpga development," https://radiona.org/ulx3s/, accessed June, 2021.

[39] G. Provelengios, D. Holcomb, and R. Tessier, "Characterizing power distribution attacks in multi-user fpga environments," in *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2019, pp. 194–201.

[40] A. Tsukioka *et al.*, "A fast side-channel leakage simulation technique based on ic chip power modeling," *IEEE Letters on Electromagnetic Compatibility Practice and Applications*, 2020.

[41] R. Zhang *et al.*, "A cross-layer design exploration of charge-recycled power-delivery in many-layer 3d-ic," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 2015, pp. 1–6.

[42] K. Murdock, D. Oswald, F. D. Garcia, J. Van Bulck, D. Gruss, and F. Piessens, "Plundervolt: Software-based fault injection attacks against intel sgx," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 1466–1482.

[43] R. J. Masti *et al.*, "Thermal covert channels on multi-core platforms," in *24th {USENIX} Security Symposium ({USENIX} Security 15)*, 2015, pp. 865–880.

[44] J. Haj-Yahya, J. S. Kim, A. G. Yaglikci, I. Puddu, L. Orosa, J. G. Luna, M. Alser, and O. Mutlu, "Ichannels: Exploiting current management mechanisms to create covert channels in modern processors," *arXiv preprint arXiv:2106.05050*, 2021.

[45] 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.

Huifeng Zhu (S'19) received the M.S. degree from the Department of Electrical and System Engineering, Washington University in St.Louis (WUSTL), USA, in 2020, where he is currently pursuing the Ph.D. degree. He received the B.Eng. degree from Beihang University (BUAA), China, in 2017. His current research interests include hardware security and power management. He was a recipient of the Best Paper Award at DATE 2019 and AsianHost 2020, and Best Paper Nomination at ASP-DAC 2021.

Xiaolong Guo (S'14-M'20) is an assistant professor in the Department of Electrical and Computer Engineering at Kansas State University (KSU). He received his PhD degree in Electrical and Computer Engineering from University of Florida (UF) in 2019. His research focuses on detecting hardware or computer vulnerabilities using formal verification and program analysis. He has been recognized with Best Paper Awards at the 2020 AsianHost and the 2019 DATE, and Best Paper Candidate at ASP-DAC 2021.

Yier Jin (M'12-SM'19) is an Associate Professor and IoT Term Professor in the Department of Electrical and Computer Engineering (ECE) in the University of Florida (UF). He received his PhD degree in Electrical Engineering in 2012 from Yale University after he got the B.S. and M.S. degrees in Electrical Engineering from Zhejiang University, China, in 2005 and 2007, respectively. His research focuses on the areas of hardware security, embedded systems design and security, trusted hardware intellectual property (IP) cores and hardware-software co-design for modern computing systems. Dr. Jin is a recipient of the DoE Early CAREER Award in 2016 and ONR Young Investigator Award in 2019. He received Best Paper Award at DAC'15, ASP-DAC'16, HOST'17, ACM TODAES'18, GLSVLSI'18, DATE'19, and AsianHOST'20. He is also the IEEE Council on Electronic Design Automation (CEDA) Distinguished Lecturer.

Xuan Zhang (S'08–M'15) is an Associate Professor in the Department of Electrical and Systems Engineering at Washington University in St. Louis. Before joining Washington University, she was a Postdoctoral Fellow at Harvard University. She received her B. Eng. degree from Tsinghua University in China, and her MS and Ph.D. degrees from Cornell University. Her research interests include hardware/software co-design for efficient machine learning and artificial intelligence, adaptive power and resource management for autonomous systems, and hardware security primitives in analog and mixed-signal domain. Dr. Zhang is the recipient of NSF CAREER Award in 2020, AsianHOST Best Paper Award in 2020, DATE Best Paper Award in 2019, and ISLPED Design Contest Award in 2013, and her work has also been nominated for Best Paper Awards at ASP-DAC 2021, DATE 2019 and DAC 2017.