Survey of Challenges in Implementation-Security Metrics for Side-Channel and Fault-Injection Attacks

Arsalan Malik*, Sahan Sanjaya‡, Harshvadan Mihir*, Ashley Kurian*, Aruna Jayasena†, Prabhat Mishra‡, Aydin Aysu*
*Department of Electrical and Computer Engineering, North Carolina State University
{aamalik3,akurian,hmihir,aaysu}@ncsu.edu
†Department of Computer Science and Engineering, University of Tennessee - Chattanooga
{aruna}@tennessee.edu
‡Department of Computer and Information Science and Engineering, University of Florida
{ssanjaya,prabhat}@ufl.edu

***Abstract*—Objective metrics for performance, power, and area are well established in digital hardware design. By contrast, metrics for implementation security, particularly for quantifying resistance to physical attacks, remain less unified and are often reported under varying assumptions, datasets, and threat models. This article surveys metrics used for two important classes of physical implementation attacks: side-channel analysis and fault-injection attacks. Rather than simply enumerating prior work, we organize existing metrics according to evaluation stage, including pre-silicon and post-silicon analysis, security objective, including leakage detection, exploitability estimation, localization, and countermeasure assessment, and attack model. This organization highlights four recurring challenges: (i) limited comparability across studies, (ii) weak support for early-stage design evaluation, (iii) ambiguity in interpreting attack-independent versus attack-based metrics, and (iv) less mature reporting practices for fault-injection attack evaluation. By examining side-channel analysis and fault-injection attack metrics together, this article identifies areas where current evaluation practices remain inconsistent or incomplete. Our goal is not to introduce another new metric, but to distill open research questions and lay a principled foundation upon which the community can build reliable and comparable security metrics.**

***Index Terms*—Side-channel analysis, fault injection attacks, hardware security, metrics.**

I. Introduction

Objective metrics drive progress. In cryptographic engineering, we routinely compare algorithmic speed in clock cycles, area cost in gate equivalents (or slice equivalents), and energy consumption in joules. Yet when the focus shifts to *implementation security*—how well real silicon resists physical attacks—quantitative discipline fades. Designers and security engineers still rely on ad-hoc plots, binary pass/fail labels, or anecdotal "works-for-us" claims, a weakness already highlighted in the broader security-metrics literature [21].

Among physical implementation attacks, SCA [7] and fault-injection attacks (FIA) [5, 29] are two of the most widely studied classes, particularly from the perspective of empirical security evaluation. They are therefore the focus of this survey, *not* because they exhaust the space of hardware-security threats, but because they are the attack families for which comparative metrics, leakage tests, and countermeasure-evaluation methodologies are most developed.

SCA deals with the extraction of secret data from unintentional physical emanations such as power, electromagnetic (EM) radiation, physical layer supply voltage coupling, sound, or photonic emissions. By contrast, FIA is the deliberate induction of transient faults to corrupt computation. SCA and FIA are now prime threats to cryptographic and artificial intelligence (AI) hardware alike. SCA metrics research is not barren. Test vector leakage assessment (TVLA) has risen as a de facto metric in academic works that provides a quick statistical gate [7], while mutual-information, guessing-entropy, and success-rate, among others, curves plot an attacker's learning process. Hettwer *et al.* recently catalogued some of these tools, and attempts have been made for unification [20].

The literature on FIA-specific security metrics remains comparatively fragmented. Metrics such as fault sensitivity index, effective fault coverage, attack-cost vectors, and detection latency have been proposed, but no broadly adopted taxonomy yet unifies them. This issue is not unique to cryptographic hardware. Although AI/ML fault-injection research is extensive and draws heavily from the test and reliability literature, the security-oriented interpretation of such experiments can differ from the reliability-oriented one: the latter often emphasizes correctness and resilience, whereas the former focuses on adversarial exploitability, targeted manipulation, and attacker-constrained success. In summary, three gaps motivate this survey.

1) Most widely used metrics are developed for post-silicon measurement and do not transfer cleanly to pre-silicon design stages, where simulation fidelity, observability, and localization needs differ.
2) Leakage-detection metrics, information-theoretic metrics, and attack-based metrics answer different questions, yet they are often reported interchangeably or interpreted inconsistently.
3) For FIA in particular, the literature lacks standardized reporting conventions and shared reference points, which limit reproducibility and cross-study comparison.

To address these gaps, we survey the SCA and FIA metric landscape, identify persistent limitations, and outline open research questions for the *IEEE Design & Test* community. We make three contributions: (i) We propose a unified taxonomy of SCA and FIA metrics, organized by evaluation stage, security objective, and attacker model. (ii) We examine common

errors in metric interpretation, including cases where leakage-detection and exploitability metrics may lead to different conclusions. (iii) We provide guidance for metric selection and reporting, and we define the minimum metadata needed for reproducible evaluations and cross-study comparisons.

Rather than introducing a new metric, this study aims to establish a principled basis for metrics that are reliable, comparable, and reproducible. The remainder of this paper is organized as follows. Section II provides an overview of SCA and FIA. Section III dissects challenges in SCA metrics, including pre-silicon applicability, AI/ML integration, and TVLA calibration. Section IV performs the same analysis on FIA metrics, highlighting the need for common definitions, metadata, and reproducibility. Section VI concludes with prospects for a unified benchmark suite.

II. Background

This section provides an overview of two dominant physical threats: side-channel analysis and fault-injection attacks.

A. Side-Channel Analysis

SCA exploits physical leakages from cryptographic devices, such as electromagnetic emissions, power consumption, or timing variations, to recover secret information without directly breaking the underlying algorithm. These leakages arise due to the physical behavior of transistors switching between logic states ($0 \rightarrow 1$ or $1 \rightarrow 0$), where the instantaneous power consumption depends on circuit capacitance, voltage, and switching frequency. The total power consumption of a device consists of a relatively constant static power and a data-dependent dynamic power. This dynamic power is consumed during transistor switching and can be modeled as $P = 1/2 \cdot \alpha \cdot C \cdot V^2 \cdot f$, where α represents the switching activity factor, C the load capacitance, V the supply voltage, and f the clock frequency. Because α depends on the data being processed, the total power draw fluctuates in a way that is correlated with secret-dependent operations. Furthermore, the time-varying currents responsible for dynamic power consumption also generate electromagnetic (EM) emissions according to Maxwell's equations. This makes EM leakage fundamentally linked to power consumption, providing another rich source of information for an attacker.

In practice, attackers collect multiple traces of such leakage signals and apply analytical techniques, which can be categorized as non-profiled and profiled attacks. Non-profiled attacks, such as differential power analysis, apply statistical methods directly to the captured traces to distinguish the correct key guess from incorrect ones. In contrast, profiled attacks operate in two stages. First, in a profiling stage, an attacker characterizes the leakage of a fully controlled, identical device to build a precise model. This model is then used in the attack stage to recover the secret key from a few traces captured from the target device. Figure 1 illustrates the difference between the non-profiled and profiled side-channel attack process, from physical leakage capture to secret key recovery.

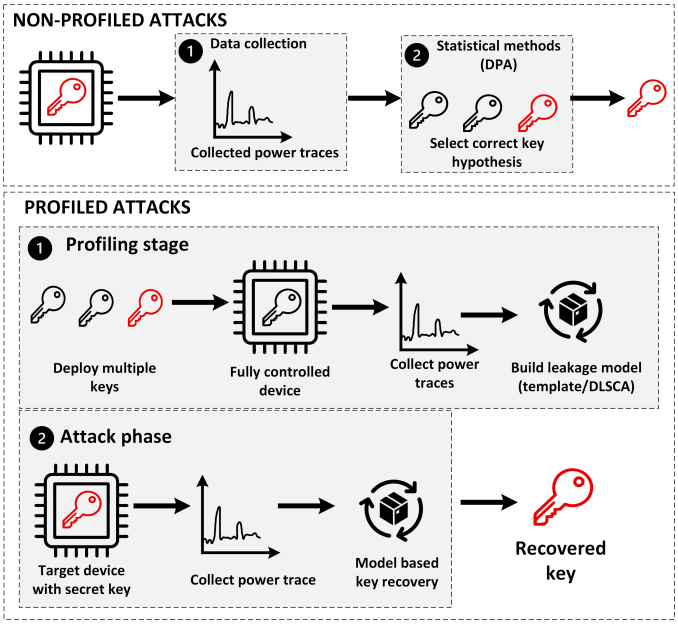

Fig. 1. Illustration of a side-channel attack process: in non-profiled attacks, power traces are collected from a target device during computation and statistical analysis, such as differential power analysis (DPA), is used to recover the secret key. Profiled attacks occur in two stages. The first stage is profiling, where all possible key values are deployed on a controlled device to collect power traces and build a leakage model using techniques such as template matching or by training a neural network. During the attack phase, the trace collected during computation with the secret key is matched against the developed model to obtain the key.

B. Fault Injection Attacks

Fault injection attacks (FIAs) constitute a class of physical security threats where an adversary induces deliberate perturbations—voltage or clock glitches, electromagnetic interference, or laser-induced faults—into hardware systems to disrupt their intended execution, as shown in Figure 2. These artificially introduced faults can circumvent security mechanisms, extract sensitive assets (*e.g.,* cryptographic keys), or transition systems into erroneous operational states. Unlike algorithmic vulnerabilities, these attacks exploit the physical implementation of the system, rendering them particularly critical in embedded platforms, secure tokens, and AI accelerators, where the compromise of a single computation may lead to complete system failure.

The physical manifestation of these attacks varies by modality. In *voltage glitching*, the supply voltage V_{DD} is transiently reduced or raised to violate timing constraints, effectively shrinking setup (t_{setup}) time window, corrupting flip-flop behavior. Similarly, *clock glitching* perturbs the system clock to introduce race conditions, *e.g.,* reducing the clock period (T_{clk}) below the minimum propagation delay (t_{pd}) can induce premature latching of intermediate values. *Electromagnetic fault injection (EMFI)* relies on rapidly changing magnetic fields to induce Eddy currents in on-chip interconnects, governed by $V_{\text{induced}} \propto \frac{d\Phi}{dt}$, where Φ is the magnetic flux linkage. *Laser-based attacks* employ focused photon bursts to locally ionize the silicon substrate, introducing charge into transistor junctions and flipping logic states.

Each type of fault changes the system's behavior, breaking the expected reliability of digital logic. For instance, fault-

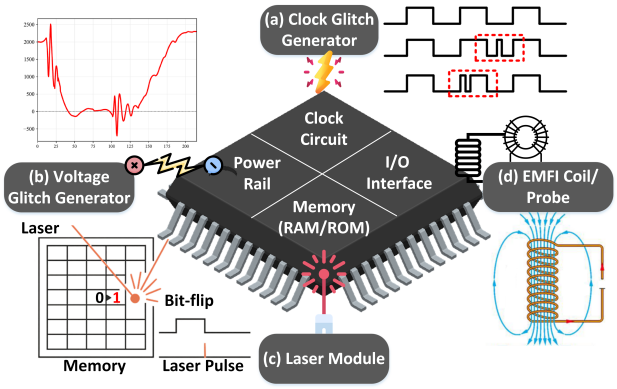

Fig. 2. Example fault injection attack (FIA) setup targeting an integrated circuit (IC). The figure illustrates four representative physical fault injection techniques: (a) clock glitching, where external timing perturbations are injected into the clock circuitry, (b) voltage glitching, which introduces transient disturbances on the power rails to destabilize internal logic, (c) laser fault injection, targeting memory components (*e.g.,* RAM or ROM) to induce localized disruptions, and (d) electromagnetic fault injection (EMFI), where a coil probe emits high-frequency pulses near the I/O interface to couple faults into internal signal paths. Each technique aims to disrupt the system's expected behavior to compromise computation integrity or extract secret information.

induced operand corruption during modular multiplication or discrete Gaussian sampling may cause leakage through erroneous rejection behavior, or alter cryptographic signatures in ways that enable secret key reconstruction. In effect, fault injection compromises the foundational assumption of correct hardware execution, with consequences ranging from denial-of-service (DoS) to cryptanalytic attacks. Such risks highlight the necessity of carefully engineered hardware-level protections in the design of security-critical systems.

III. Challenges in Side-Channel Analysis Metrics

In this section, we first present an overview of side-channel analysis metrics, followed by a brief description of the Test Vector Leakage Assessment methodology. We then discuss the challenges associated with SCA metrics across four contexts: general challenges applicable to all evaluations, those specific to pre-silicon and post-silicon analyses, and challenges arising in emerging application domains. Table I summarizes the key metrics, core concepts, and our insights from prior work on evaluating implementation security against SCA.

A. Overview of Side-Channel Analysis Metrics

To evaluate and compare device resistance against side-channel attacks, researchers have proposed a range of quantitative metrics [20]. Among the most widely adopted are the signal-to-noise ratio (SNR), which quantifies the strength of leakage relative to background noise (see Section III-D); Mutual Information analysis, which measures the statistical dependence between observed leakage and sensitive variables (see Section III-D); and the Test Vector Leakage Assessment, which detects the presence of leakage without requiring a key hypothesis (see Section III-A1).

Additional metrics identified in the literature include success rate (SR) and guessing entropy, which quantify the number of traces or effort required for key recovery; specifically, SR quantifies the probability that an adversary correctly identifies the secret key as the most likely candidate:

$$SR = \Pr\left[L(k^*;t) > L(k_i;t),\ \forall i \neq k^*\right], \tag{1}$$

where $L(k;t)$ denotes the log-likelihood of observing trace t under key hypothesis k. This metric measures the leakage in terms of information-theoretic divergence and practical key distinguishability.

Distribution-based metrics include Kullback–Leibler (KL) divergence; learning parity with noise complexity estimations; and template attack efficiency measures. These frameworks often incorporate information-theoretic measures such as the KL divergence between switching activity distributions:

$$D_{KL}(k_i \| k_j) = \int f_{T|k_i}(t) \log \frac{f_{T|k_i}(t)}{f_{T|k_j}(t)}\, dt, \tag{2}$$

where $f_{T|k_i}(t)$ and $f_{T|k_j}(t)$ denote the conditional probability density functions of the side-channel traces given keys k_i and k_j, respectively.

Additionally, model-based statistical approaches are also adopted to assess side-channel leakage. One of the most common is the Pearson correlation coefficient, which quantifies the linear relationship between observed power proxies and hypothetical leakage models. It is defined as

$$\rho = \frac{\sum_{i=1}^{n}(L_i - \overline{L})(P_i - \overline{P})}{\sqrt{\sum_{i=1}^{n}(L_i - \overline{L})^2} \cdot \sqrt{\sum_{i=1}^{n}(P_i - \overline{P})^2}}, \tag{3}$$

where L_i denotes the predicted leakage values derived from a specific leakage model (e.g., Hamming weight or Hamming distance), P_i represents the observed power samples, and $\overline{L}$ and $\overline{P}$ are their respective means. Higher absolute values of ρ indicate stronger correlation and, consequently, higher susceptibility to side-channel attacks. In contrast to the other metrics, the Pearson correlation coefficient serves as a model-based metric that evaluates how strongly the measured power aligns with the hypothesized leakage behavior.

Beyond the metrics discussed thus far, side-channel evaluations also depend on several fundamental performance indicators that quantify the practical difficulty of key extraction. Among these, the number of traces required for a successful attack remains the most intuitive and widely reported measure, representing how many side-channel observations an adversary needs to recover the secret key. In addition, for a thorough security assessment, evaluators must employ metrics that distinguish between attacker capabilities and implementation resistance. As an example, while the SR metric measures the probability that the correct key appears as the top-ranked candidate, the guessing entropy (GE) metric quantifies the average remaining search effort as $\log_2(\text{rank of correct key})$. Both metrics require known-key analysis, enabling evaluators to assess security even when attacks fail to achieve full key recovery. Moreover, several metrics can serve purposes beyond their original intent. For example, the Pearson correlation coefficient functions not only as an attack distinguisher in correlation power analysis (CPA) but also supports security projections through analytical models that relate correlation

strength to the number of required traces and expected success rates.

1) Test Vector Leakage Assessment (TVLA)

The methodology of TVLA for hardware implementations is designed to detect whether confidential information is inadvertently revealed through power side-channel signals from the hardware implementation [7]. In recent years, the TVLA methodology has seen widespread adoption, accompanied by ongoing efforts by ISO to formalize its standardization [23]. As shown in Figure 3, the process begins with test generation, typically based on hamming weight (HW) or hamming distance (HD) metrics, aimed at provoking observable power variations. The hardware is then executed (in post-silicon analysis) or simulated (in pre-silicon analysis) using generated patterns, and switching activity is extracted from the power traces. From this data, statistical metrics are used to measure discrepancies between the resulting power profiles.

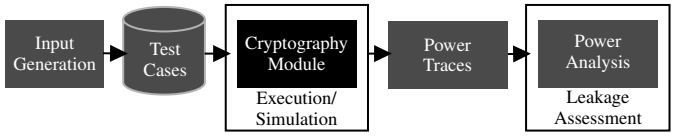

Fig. 3. Major steps of Test vector leakage assessment (TVLA) of hardware cryptographic implementations.

The most widely adopted statistical analysis technique with TVLA is the *Welch's t-test* to compare two sets of side-channel traces, typically, collected under fixed and random input conditions, as defined below:

$$t = \frac{\mu_0 - \mu_1}{\sqrt{\frac{\sigma_0^2}{n_0} + \frac{\sigma_1^2}{n_1}}}, \tag{4}$$

where μ_0 and μ_1 denote the sample means of the two trace populations, σ_0^2 and σ_1^2 are their sample variances, and n_0 and n_1 represent the respective numbers of traces. The computed t-value quantifies the statistical separation between the two distributions. A commonly used decision rule considers $|t| > 4.5$ as evidence of significant leakage, corresponding approximately to a p-value below 10^{-5}. Under this criterion, exceeding the threshold indicates that the device's power distributions under fixed and random inputs are distinguishable, implying potential side-channel leakage. Additional variations within the TVLA framework include the standard Welch's t-test with its non-specific and specific variants, while other statistical tests such as the χ^2 test have been explored as complementary approaches outside the canonical TVLA methodology.

B. Common Challenges with Side-Channel Analysis Metrics

The choice of metric can impact security assessments, sometimes leading to contradictory conclusions about the same implementation. While many of the limitations of commonly used metrics have been previously discussed, these insights are often scattered across the literature and lack a unified presentation. In this section, we consolidate and synthesize these discussions, providing a structured overview of the common challenges associated with using these metrics for side-channel analysis, along with their practical implications.

Limitations of TVLA and Statistical Testing: Welch's t-test-based TVLA remains one of the most widely used statistical methods for side-channel leakage detection. However, it has important limitations. In its standard fixed-versus-random setting, TVLA is mainly sensitive to particular leakage forms and can miss higher-order or masked leakages unless additional preprocessing or dedicated higher-order analysis is used [4]. Furthermore, a TVLA pass does not prove the absence of leakage; it only shows that no statistically significant difference was detected under the chosen test conditions, and therefore should not be interpreted as a sufficient guarantee of security [26]. As shown by He et al. [8], leakage can remain undetected if test vectors or measured power traces fail to sufficiently activate internal transitions, leaving vulnerabilities unobserved. Practical challenges also persist: too few traces may obscure subtle leakage, while excessive sampling can inflate false positives, particularly in simulation-based or low-noise environments [30].

Domain-Specific Challenges: Additional challenges arise when applying TVLA to asymmetric cryptographic implementations, such as post-quantum cryptography (PQC). Specifically, these computations often involve structured randomness, ephemeral secrets, and transforms such as the Number-Theoretic Transform (NTT). These characteristics violate the assumptions underpinning the fixed-vs-random model, frequently resulting in inconclusive or misleading outcomes [23]. Furthermore, standard TVLA configurations may overlook corner cases where leakage manifests only under rare or specific input conditions. For example, in FALCON, a zero-check subroutine in floating-point arithmetic was not exercised under the random-vs-fixed setup, requiring a fixed-vs-fixed strategy to properly capture leakage and validate masking security [12]. Collectively, these findings emphasize that although TVLA is becoming the de facto standard, it remains far from a universally applicable or comprehensive solution across all cryptographic domains.

Another practical challenge lies in defining trace-count thresholds for acceptable security evaluation. Existing certification-oriented leakage-assessment frameworks often rely on finite trace budgets, with ISO/IEC 17825 commonly discussed in terms of 10,000-trace and 100,000-trace assurance settings [10]. Yet the security meaning of these thresholds is incomplete, because success depends strongly on the attack strategy, leakage model, and analysis assumptions. A device that appears secure under one testing configuration may still be vulnerable under another configuration.

Complementarity and Contradictions Among Metrics: While TVLA and other statistical tests can reveal measurable differences between power distributions, they do not necessarily indicate practical exploitability. For instance, statistical tests like TVLA's t-test may signal the presence of leakage (by rejecting the null hypothesis of security), yet the same device might appear secure under success rate (SR) or guessing entropy (GE) metrics if the detected leakage is insufficient for practical key recovery [20]. This discrepancy arises because t-tests detect any statistical deviation between distributions,

TABLE I

COMPARISON OF REPRESENTATIVE SIDE-CHANNEL-ANALYSIS (SCA) METRIC CLASSES. THE TABLE DISTINGUISHES WHETHER A METRIC IS INTENDED FOR LEAKAGE DETECTION, INFORMATION-THEORETIC CHARACTERIZATION, ATTACK-LEVEL EVALUATION, OR LOCALIZATION/INTERPRETATION.

Metric	Primary purpose	Typical stage	Output	Main limitation
Leakage-detection metrics				
TVLA (Welch's t-test) [7, 11, 23, 26]	Detects statistically significant differences between trace distributions.	Post-silicon	t-statistic/pass-fail	Indicates detectable leakage under the chosen test setup, but does not directly quantify exploitability or guarantee security.
Signal-to-noise ratio (SNR) [14]	Quantifies the separability of data-dependent signal variation from noise.	Pre-/post-silicon	Numerical ratio	Useful for screening, but it depends on trace partitioning and does not directly predict attack success.
Information-theoretic metrics				
Mutual information (MI) [27]	Quantifies statistical dependence between sensitive variables and observations.	Pre-/post-silicon	Information value	More attack-agnostic than correlation-based metrics, but sensitive to density estimation and sample quality.
Perceived information (PI) [20]	Measures information leakage under an assumed leakage model.	Pre-/post-silicon	Information value	Provides a model-aware leakage estimate, but depends strongly on the assumed model and estimator quality.
Hypothetical information (HI) [20]	Evaluates the effectiveness of theorized leakage models without physical profiling, using hypothetical distributions.	Pre-silicon	Information value	Useful for early-stage evaluation, but ignores actual physical noise and real-world device discrepancies.
Entropy-based metrics [20]	Characterize uncertainty or concentration in observed or inferred distributions.	Pre-/post-silicon	Entropy value	Reduced entropy does not necessarily imply exploitable leakage, especially when interpretation is not aligned with the threat model.
Attack-based metrics				
Success rate (SR) [20]	Measures the probability that an attack recovers the correct secret.	Post-silicon	Probability	Operationally meaningful, but highly dependent on attack assumptions, preprocessing, and profiling resources.
Guessing entropy (GE) [20]	Measures the average rank of the correct key hypothesis during attack evaluation.	Post-silicon	Average rank	Reflects practical attack difficulty, but remains attack- and dataset-dependent.
Traces-to-disclosure (TTD) [20]	Measures the number of traces needed to achieve successful key recovery.	Post-silicon	Trace count	Easy to interpret, but strongly dependent on attack method, leakage model, and stopping criterion.
Localization/interpretation metrics				
Localization-oriented analysis [20]	Identifies where leakage originates within the design or trace.	Pre-/post-silicon	Hotspot ranking	Improves interpretability, but is generally less standardized than detection or attack-based metrics.
Normalized inter-class variance (NICV) [20]	Identifies points of interest and localizes leakage without requiring the secret key or a specific leakage model.	Pre-/post-silicon	Variance ratio (0 to 1)	Efficient for localization, but performance degrades in extremely noisy environments and relies on public inputs (like plaintext) for class partitioning.

whereas SR and GE measure the actual success or effort required for key extraction. Conversely, a device might pass TVLA testing (no detectable leakage within the available trace budget) but fail under MI Analysis or correlation-based projections, which can predict that key recovery would become feasible given additional traces. These observations emphasize the complementary nature of attack-independent metrics (*e.g.,* t-test, MI, SNR) and attack-based metrics (*e.g.,* SR, GE, correlation): the former detect potential leakage, while the latter confirm its exploitability. A balanced evaluation thus requires both perspectives to avoid misleading conclusions about security strength.

State-Space Challenge: Side-channel attacks commonly employ divide-and-conquer strategies that target individual key bytes or intermediate values. However, translating these local results into a reliable full-key security assessment remains a major challenge. Metrics such as key enumeration and rank estimation attempt to bridge this gap. In unknown-key scenarios, key enumeration systematically lists full-key candidates in decreasing likelihood order until the correct key is found, while in known-key evaluations, rank estimation efficiently computes the correct key's position in this ordering; often through histogram-based convolution or similar probabilistic techniques. These methods provide a way to extend byte-level leakage metrics to full-key security projections. Nevertheless, the exponential growth of the key space makes exhaustive enumeration computationally infeasible for realistic key sizes, and rank estimation itself becomes sensitive to modeling assumptions, leakage correlations, and numerical precision. As a result, accurately quantifying full-key security remains an open and computationally demanding problem in post-silicon leakage evaluation.

Recent work has emphasized more holistic and multidimensional assessment frameworks, proposing metrics that integrate multiple statistical perspectives, such as multivariate mutual information, cross-correlation metrics, and dimensionality-reduction-based leakage measures to better capture complex leakages [1]. Others have studied the interplay between metrics, showing that certain metrics correlate strongly while others provide complementary insights, suggesting the need for

multi-metric evaluations [3]. Despite the diversity of available metrics, there remains no consensus on a universal standard or an agreement on which use-cases will require which specific metric; this often depends on experimental setup, threat model, and analyst preference.

C. Challenges Specific to Pre-silicon Side-Channel Analysis

Pre-silicon verification of side-channel security has emerged as a vital component of modern hardware design methodologies. Among the available evaluation techniques, TVLA (see Section III-A1) remains one of the most widely adopted, while other metrics discussed in Section III-A are also frequently employed. However, pre-silicon side-channel analysis faces unique challenges. Conventional leakage metrics primarily reveal whether a measurable dependence exists in side-channel traces, but they generally provide limited spatial localization within the underlying design. Moreover, because these metrics are usually trace-based, they are not readily compatible with hardware analysis frameworks that operate on register-transfer level (RTL) descriptions, gate-level netlists, FSMs, or other symbolic and formally analyzable design models. These limitations hinder both formal verification and the effective deployment of countermeasures such as targeted masking or logic-level redesign.

Shannon entropy quantifies the uncertainty of a random variable X based on its probability distribution. Formally, the entropy is defined as

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i), \tag{5}$$

where $p(x_i)$ is the probability of outcome x_i. A higher entropy indicates greater uncertainty in the distribution of the observed variable X, whereas a lower entropy reflects a more concentrated distribution. In cryptographic hardware, entropy-based analysis can be useful as one indicator of how structured or predictable internal activity may be under a given observation model. However, entropy should not be interpreted as a direct measure of exploitability. For (hardware or software) implementations that do not employ explicit randomization mechanisms, such as masking, shuffling, or blinding, reduced entropy may simply reflect deterministic functional behavior of the implementation rather than practically exploitable information leakage. Entropy-based results should therefore be interpreted together with the underlying threat model and, where possible, complemented by attack-based or leakage-detection metrics.

The side-channel vulnerability (SCV) metric enables efficient pre-silicon leakage evaluation using only a limited number of simulated traces. Specifically, SCV can be applied in formal verification through Information Flow Tracking to estimate side-channel leakage early in the design stage [30]. It is defined as:

$$\text{SCV} = \frac{P_{signal}}{P_{noise}} = \frac{P_{T.h_i} - P_{T.h_j}}{P_{noise}} \tag{6}$$

where $P_{T.h_i}$ and $P_{T.h_j}$ represent the average power for output Hamming weights h_i and h_j, respectively. The difference between them indicates the signal power contributing to side-channel vulnerability. In this metric, the signal power is computed based on the power model defined by the designer. Consequently, the accuracy of the evaluation depends heavily on the fidelity of the selected power model. If the model fails to reflect the actual power behavior of the design, the resulting analysis may lead to incorrect or misleading conclusions.

Newer frameworks emphasize localization and scalability of leakage detection. Architectural correlation analysis (ACA) refines correlation analysis by searching for leaky time intervals and ranking gates with the leakage impact factor (LIF), combining correlation coefficients with each gate's power contribution [2]. LIF incorporates the total power of the design in its computation; hence, the accuracy of the power estimation model directly influences the reliability of the results. LIF is mainly used to identify gates that are more susceptible to side-channel leakage by analyzing distinct LIF peaks, which indicate components critical from a security perspective. However, LIF does not quantify the exact amount of side-channel leakage; rather, it highlights gates that are relatively more vulnerable compared to others within the design.

In summary, these metrics reveal both the potential and challenges of pre-silicon side-channel evaluation. While new metrics like SCV attempt to go beyond TVLA's black-box nature, pre-silicon evaluations still lack widely accepted, standardized metrics. This gap highlights the need for methodologies tailored to early design stages that can provide both detection and diagnostic guidance. Although pre-silicon simulation cannot fully match the fidelity of post-silicon measurements, and an inherent gap between simulation and physical measurement is unavoidable, pre-silicon analysis remains underutilized relative to its potential value. Even with reduced accuracy, it can provide actionable early-stage insights, enabling the identification of relative leakage trends, structural vulnerabilities, and high-risk design regions prior to fabrication. Such early detection can significantly reduce design iteration costs and improve overall system security, while complementing, rather than replacing, post-silicon validation.

D. Challenges Specific to Post-Silicon Side-Channel Analysis

Unlike in the pre-silicon side-channel analysis, post-silicon analysis is carried out on real devices where the adversary has to deal with noise from the environment. The SNR offers an attack-independent alternative by quantifying the ratio between useful signal variance and noise variance (SNR = Var$(L_d)/Var(L_n)$), enabling quick identification of leaky time samples without performing actual attacks. However, SNR relies on simplified univariate leakage models and cannot directly predict attack success. Score and rank metrics provide more granular insight by showing how key candidates are ordered during an attack; the score reflects the attack distinguisher value while the rank indicates each candidate's position in the sorted list. These metrics visualize attack convergence but require multiple experimental repetitions to ensure statistical stability, and they focus on individual key bytes rather than full keys.

In post-silicon evaluations, more sophisticated information-theoretic metrics are employed to establish attack-independent

security bounds directly from measured leakage. The mutual information (MI) metric quantifies the amount of information leaked about the secret key K in bits, expressed as $MI(L;K) = H(K) - H(K|L)$, where $H(K)$ denotes the entropy of the secret key, and $H(K|L)$ denotes the conditional entropy of the key given the observed leakage L, representing the remaining uncertainty after observing side-channel information. This formulation inherently supports multivariate analysis, making it suitable for capturing complex leakage interactions across multiple points of interest. Hypothetical Information (HI) extends this concept by estimating leakage using profiled or simulated models, while Perceived Information (PI) further refines the evaluation by validating these models against actual silicon measurements. Negative PI values can reveal inaccurate leakage models or insufficient profiling data, highlighting potential mismatches between theoretical assumptions and real device behavior. However, applying these metrics post-silicon introduces significant challenges, including the high computational cost of estimating probability distributions from noisy analog traces, sensitivity to measurement alignment and environmental variability, and the difficulty of obtaining statistically meaningful results within limited trace budgets.

E. Metric Transfer to Emerging Workloads

Side-channel attacks on AI/ML implementations have emerged as a distinct threat, targeting information beyond cryptographic keys. By exploiting power, electromagnetic (EM) emission, timing, and memory-access leakages, adversaries can infer sensitive details *e.g.,* model architecture, hyperparameters, weights/biases, activation functions, and even input data. Both profile-based techniques and non-profile methods (*e.g.,* correlation analysis) have been used, particularly during inference, to reverse engineer inputs or proprietary model behavior; moreover, exact parameter recovery is often unnecessary, since attackers can build surrogate models that closely replicate the victim's outputs.

To evaluate the effectiveness of these attacks, researchers have adopted a range of metrics tailored to the type of information being extracted. Specifically, the following metrics have been used to evaluate SCA attacks targeting model architecture extraction. *Levenshtein Distance Accuracy (LDA)* is used to evaluate how closely the recovered model structure matches the ground truth, with higher LDA values indicating fewer edits are needed to transform the predicted architecture into the correct one [6]. *Segment Accuracy (SA)* instead measures the percentage of sampling points in the side-channel trace that are correctly assigned to their corresponding ML layer types, reflecting the precision of trace segmentation [6]. Together, these two metrics capture both the global accuracy of model architecture reconstruction and the local accuracy of layer-wise segmentation. The effectiveness of the recovered model architectures can also be assessed using the *average test error (ATE)* metric. ATE measures the mean discrepancy between the predictions of the victim model F_v and the recovered model F_s over the test set D_t, and is defined as

$$ATE = \frac{1}{|D_t|} \sum_{(x,y)\in D_t} d(F_v(x), F_s(x)), \tag{7}$$

where $d(\cdot)$ is a p-norm distance function. A lower ATE value implies that the recovered model provides outputs closer to those of the victim model, indicating a successful recovery.

To evaluate the quality of the recovered inputs, most often input images, researchers typically rely on the following metrics. The *recognition accuracy* [9] is computed by reclassifying the reconstructed images with the victim's original neural network, which reflects how well the semantic features of the inputs are preserved. The *mean structural similarity index (MSSIM)* [9] is employed to compare luminance, contrast, and structural information between the recovered and original images, where values range from -1 (completely dissimilar) to $+1$ (identical). The *pixel-level distance* [9] metric is defined as

$$\alpha_{\text{pixel}} = \frac{\sum_{x\in I} \|p_v(x) - p_{vg}(x)\|^2}{|I|}, \tag{8}$$

where $p_v(x)$ and $p_{vg}(x)$ denote the pixel values of the recovered and original images, respectively, and $|I|$ is the number of pixels in the image. A lower α_{pixel} indicates a higher similarity between the reconstructed and ground-truth inputs. The *cross-correlation (CCR)* and its normalized form (CCR_{norm}) have also been used to evaluate the similarity between original and reconstructed images [17]. The CCR is defined as

$$CCR = \sum_{(i,j)\in N_{n\times m}} \big((A[i,j] - \bar{A}) \times (B[i,j] - \bar{B})\big), \tag{9}$$

where A and B are the pixel matrices of the original and recovered images, size of $n \times m$ pixels, and $\bar{A}, \bar{B}$ are their respective mean values. The normalized cross-correlation is then given by

$$CCR_{\text{norm}} = \frac{CCR}{\sqrt{\sum (A[i,j] - \bar{A})^2} \cdot \sqrt{\sum (B[i,j] - \bar{B})^2}}, \tag{10}$$

which scales the correlation to the range $[-1, 1]$, with values closer to 1 indicating higher similarity.

Moreover, when profiled side-channel attacks employ ML models, their evaluation typically follows standard ML performance metrics. For classification-based profilers, studies commonly report metrics, such as *precision, recall, F-measure, and accuracy*. *Precision* captures the fraction of predicted positives that are correct, whereas *recall* quantifies the fraction of true positives that are successfully identified. Accordingly, low *recall* implies many missed positives (false negatives), while low precision implies many incorrect positives (false positives). The *F-measure* summarizes this trade-off by combining precision and recall into a single balanced score.

IV. Challenges in Fault Injection Attack Metrics

FIAs are active, physical methods that deliberately introduce faults into hardware systems (*e.g.* via clock/voltage glitches, electromagnetic (EM) pulses, lasers) to evaluate or compromise security. Several studies have proposed methodologies and frameworks that quantify vulnerability using purpose-built metrics. In light of this, we conduct a large-scale survey of earlier works, define a taxonomy for the FIA campaigns, and introduce the metric catalog in Table II to help analysts navigate the complex FIA process. Table II captures key

TABLE II

OVERVIEW OF KEY METRICS USED IN FAULT INJECTION ATTACK ASSESSMENTS. THIS TABLE COMPARES VARIOUS HARDWARE- AND SOFTWARE-ORIENTED METRICS THAT QUANTIFY THE SECURITY VULNERABILITIES OF CRYPTOGRAPHIC SYSTEMS AGAINST FAULT INJECTION ATTACKS. EACH METRIC IS DEFINED WITH ITS CORRESPONDING UNIT AND CATEGORY, OFFERING A CLEAR VIEW OF HOW DIFFERENT VULNERABILITIES, INFORMATION LEAKAGE, AND FAULT COVERAGE CAN BE EVALUATED ACROSS DIFFERENT DESIGNS.

Metric	Definition	Unit	Category
Vulnerability (V) [5]	Expected probability that an adversary can guess the secret in one attempt based on observed outputs. $V[S \mid Y, Y'] = \sum_{y,y'} \max_s \left(\Pr[s] \Pr[y, y' \mid s]\right)$	Probability	Hardware Oriented Metrics
Information Leakage (L) [5]	Measure of information (bits) an adversary can learn about the secret after an observation. $\mathbb{L}[S \mid Y] = H_\infty(S) - H_\infty(S \mid Y)$	Numerical value	
Timing Violation [29] Vulnerability Factor (TVVF)	Evaluates vulnerability to timing-violation fault attacks from probabilities of fault injection and propagation. $\text{TVVF}(C, A) = \sum_{i=1}^{N} P_{G_i} \cdot P_{\text{obs}}(O_i)$	Probability	
Vulnerability to [25] Timing FA	Vulnerability of a hardware design to DFA via timing fault attacks, using post-layout timing information. $V_{\text{TFI}} = \frac{1}{N_{\text{DBsp}}} \sum_{p=1}^{N_{\text{DBsp}}} P_{\text{sp}}$	Probability	
Architectural [18] Vulnerability Factor (AVF)	Probability that a random transient fault in a hardware structure results in a user-visible error.	Probability	
Susceptibility [19] Factor (SF)	Ratio of the timing difference between the paths of the state FF to be violated and the path to be met via FIAs to cause a vulnerable FSM transition and the average path delays. $SF = \frac{Path_{Difference}}{avg(Path_{FS})}$	Numerical value	
Vulnerability factor [19] for fault-injection (VF_{FI})	Composed of two parameters namely Percentage Vulnerable Transitions (PVT(%)) and Average Susceptibility Factor (ASF).	Numerical value	
Feasible Faults [28]	Number of faults that can realistically occur under a certain injection technique. $\text{TF}_{\text{global}} = \text{Tf} \times \sum_{i=1}^{\text{CFth}} \binom{\text{NS}}{i}$, $\text{TF}_{\text{local}} = \text{Tf} \times \sum_{i=1}^{\text{CFth}} \binom{\text{NT}}{i}$	Numerical value	
Fault Coverage (FC) [22]	Number of faults detected relative to the total number identified exploitable faults.	Percentage	Countermeasures Evaluation Oriented Metrics
Fault Coverage Corrected (FCC) [22]	Weighted calculation of fault coverage based on number of faults against the severity of faults $\text{FCC} = \text{TFC} \times 0.125 \times \text{FC}$	Percentage	
System Security Factor (SSF) [15]	Probability that an attack would create an 'illegal transaction' in order to bypass existing security mechanisms. $\hat{\text{SSF}} = \frac{1}{N} \sum_{t_i, p_i \sim f_{T,P}} e(t_i, p_i)$	Probability	
Automatic Leakage [24] Assessment Factor (ALAFA)	A score to determine potential information leakage resulting from fault attacks by analyzing the differential between correct and faulty ciphertexts. $\text{L}_{FA} = \Delta C = F(f, P, K)$, $\text{I}(\Delta C; K \mid P) = 0$	Numerical value	
Information-Theoretic [16] Security Factor (γ)	Proportional reduction of actual information leakage compared to theoretical information leakage. $\gamma = \frac{m - m'}{m}$	Numerical value	
Fault Parameters and Timing [13]	Metadata should include the exact glitch or fault parameters used: clock frequency, glitch width (pulse duration), glitch timing (offset relative to trigger), voltage level, number of attempts, etc. so that a third party should be able to validate the outcome.	Probability	Reproducibility Oriented Metrics
Environmental Conditions [13]	Reproducible experiments must control or record temperature, ambient EM noise, and ensure the target is in a consistent state before each injection.	N/A	
Outcome Classification [13]	Recording what happened on each injection (*e.g.* no effect, corrupted output, crash/reset) to analyze success rates.	Percentage	

metrics, critical concepts, including survey insights, and shows how current work evaluates implementation security against FIAs. Different use cases motivate distinct metric classes for hardware and software evaluation. Thus, the available metrics for FIAs can be grouped and summarized as follows:

- **Hardware-oriented metrics:** Define circuit properties mathematically or through formal verification rather than measuring them during device operation. They are primarily employed for design-time evaluations, facilitating precise localization of vulnerabilities within the circuit. This enables the development of targeted countermeasures at the early stages of the design process and allows comparative evaluations across different implementations of the same logic design.
- **Countermeasure evaluation-oriented metrics:** Quantify the strength of defensive mechanisms by combining fault detection, system resilience, information leakage, and reproducibility. They assess not only whether faults are detected but also how fault severity is reflected, how injected faults may bypass protections, and how much

sensitive information remains exposed. Precise reporting of injection parameters—such as timing, voltage, and attempt counts—is emphasized to ensure verifiability and comparability across studies, thereby providing a structured basis for evaluating countermeasure reliability.

- **Reproducibility-oriented metrics:** Measure the consistency of fault outcomes, either as a binary property or as a degree of stability across repeated attempts. Reproducibility is essential for both evaluation and advanced attack scenarios, such as differential fault analysis, which often assumes that identical faults can be injected multiple times. Accordingly, academic literature frequently considers reproducibility a fundamental requirement for fault-injection platforms.

With this taxonomy in place, the literature offers concrete instantiations that target specific fault models and design abstractions. One of the earliest examples, the *architectural vulnerability factor (AVF)*, refines pessimistic transient-fault assumptions by identifying the fraction of architecturally critical execution (ACE) bits, thereby estimating a microprocessor's soft-error rate during early design [18]. Extending this idea to timing faults, *timing violation vulnerability factor (TVVF)*, and the analyzing vulnerabilities in finite state machine (AVFSM) frameworks evaluate data– and control–path risks, respectively. TVVF provides a probabilistic estimate of a design's susceptibility to setup-time violation attacks from a netlist-level view, modeling both fault injectability and propagation to observable outputs [29].

AVFSM addresses synthesis-induced weaknesses in finite-state machines (FSMs) through two metrics: the *susceptibility factor (SF)*, which measures the likelihood of false state transitions under injection, and the *fault-injection vulnerability factor* (VF_{FI}), which aggregates SF and the percentage of false transitions to quantify overall FSM vulnerability [19]. More recently, LDTFI introduced a layout-aware framework for clock-glitch timing fault injection and defined vulnerability to timing fault attacks, which assesses susceptibility to differential fault analysis (DFA) using standard delay format (SDF) simulations on post-layout netlists [25].

Formal circuit models support additional hardware-centric metrics. The SoFI framework encodes formal security properties (SP) over gate-level netlists and derives a *feasible faults* metric by filtering injected faults to those that violate an SP and remain practically realizable [28]. Designers then harden critical locations instead of protecting the entire chip. Feldtkeller et al. introduced a quantitative information flow (QIF) analysis to move beyond binary secure/insecure classifications and enable fine-grained comparisons among hardware implementations under fault injection [5]. Their metric formally analyzes how injected faults reduce the uncertainty regarding secrets, yielding an information-theoretic view of leakage. Automatic leakage assessment for fault attack countermeasures (ALAFA) is a statistical framework used to evaluate the security of cryptographic systems against fault attacks [24]. ALAFA detects potential leakage by analyzing the differences between correct and faulty ciphertexts, which may reveal secret keys or fault values. It simulates faults using a fault model and adapts the Welch's 't-test' to determine if ciphertext distributions show significant leakage.

Beyond design and implementation assessments, several works focus on quantifying the protection offered by countermeasures. Potestad *et al.* compared AES countermeasures against DFA using *Fault Coverage (FC)* and its refinement, *Fault Coverage Corrected (FCC)*, which accounts for eight DFA fault classes [22]. Li *et al.* model injection and propagation as probabilistic processes and estimate the *system security factor (SSF)*—the probability of illegal FSM transitions—via Monte-Carlo evaluation [15]. In another work, Liu *et al.* introduced *information theoretic security factor*, a metric that directly measures leakage, avoiding reliance on secondary indicators such as error rates or the success of a specific cryptanalytic attack [16].

The literature reveals a fragmented and inconsistent landscape of FIA metrics. As shown in Table II, some studies prioritize probabilistic measures of attack success, while others quantify information leakage in terms of bits [5]. Unlike performance, power, and area (PPA) metrics—which benefit from standardized benchmarks, such as floating-point operations per second (FLOPs)—FIA metrics lack a cohesive taxonomy and shared reference points. This heterogeneity undermines cross-study comparisons. For instance, researchers cannot readily determine whether a system that consistently leaks a single bit is more secure than one that rarely leaks an entire key. As attacks grow more sophisticated and system-level assumptions vary widely, the field can no longer rely on isolated, ad hoc evaluations. A unified framework for FIA metrics is urgently needed to support reproducible, quantitative, and broadly comparable security assessments.

In conclusion, while individual studies have advanced FIA measurement across various dimensions, the field lacks a unified taxonomy. Establishing a standardized metrics framework for implementation security is now imperative. Such a foundation would facilitate clearer communication of security guarantees, enable principled design trade-offs, and foster the development of more resilient hardware systems. Taken together, the above discussion suggests that the central difficulty is not merely the lack of metrics, but the lack of clear guidance on how existing metrics should be selected and interpreted under different assumptions. Since leakage detection, exploitability estimation, localization, and countermeasure assessment are distinct objectives, no single metric can serve all of them equally well. A practical survey should therefore not only catalog available metrics, but also clarify how they may be matched to concrete evaluation scenarios.

V. Open Research Questions

1) How should leakage-detection and exploitability metrics be combined? Metrics such as TVLA and SNR indicate detectable dependence, whereas attack-based metrics such as success rate or guessing entropy reflect adversarial usefulness. A principled methodology for combining these views without conflating them remains lacking.

2) What reporting standard is sufficient for FIA reproducibility? Unlike SCA, FIA studies often vary widely

in injection modality, timing precision, spatial resolution, and outcome classification. While we have metrics like TVLA for SCA, the community still lacks a minimally accepted reporting standard that supports meaningful cross-study comparison.

3) How should security metrics generalize beyond classical cryptographic targets? Emerging workloads may preserve similar physical leakage mechanisms while changing the adversarial objective. More work is needed to determine which existing metrics transfer cleanly and which require reinterpretation or redesign.

4) What properties should a useful security metric satisfy? Future metrics should be evaluated not only by convenience, but also by properties such as reproducibility, threat-model clarity, sensitivity, robustness to preprocessing choices, and interpretability for designers and evaluators.

VI. Conclusion

While individual studies have advanced the measurement of fault injection and side-channel attacks, the field remains a fragmented landscape of ad-hoc metrics. Consequently, security assessments remain difficult to compare across studies and may not always be reproducible in a way that supports rigorous engineering practice. Moving forward, the goal should not be to force implementation security into a single universal score, analogous to power, performance, or area, but rather to establish a common evaluation framework that defines standard reporting criteria, clarifies which metrics are appropriate for which threat models, and enables results to be interpreted consistently across designs and platforms. Such a framework would allow designers to assess security risk more systematically, would support evaluators and certification laboratories in making reproducible and defensible judgments, and would improve the quality of design trade-off analysis. Establishing this common metrics language is the essential next step to enable clearer design trade-offs and build provably secure systems.

References

[1] Alric Althoff, Jeremy Blackstone, and Ryan Kastner. Holistic power side-channel leakage assessment: Towards a robust multidimensional metric. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8, 2019. doi: 10.1109/ICCAD45719.2019.8942098.

[2] Katayoon Basharkhah, Zahra Hojati, and Zainalabedin Navabi. An event-based gate-level framework for power side-channel leakage assessment. In *2025 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, volume 1, pages 1–6. IEEE, 2025.

[3] Julien Béguinot, Wei Cheng, Sylvain Guilley, and Olivier Rioul. Be my guesses: The interplay between side-channel leakage metrics. *Microprocessors and Microsystems*, 107:105045, 2024. ISSN 0141-9331. doi: https://doi.org/10.1016/j.micpro.2024.105045. URL https://www.sciencedirect.com/science/article/pii/S0141933124000401.

[4] François Durvaux and François-Xavier Standaert. From improved leakage detection to the detection of points of interests in leakage traces. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 240–262. Springer, 2016.

[5] Feldtkeller, Jakob and Güneysu, Tim and Schaumont, Patrick. Quantitative fault injection analysis. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 302–336. Springer, 2023.

[6] Yansong Gao, Huming Qiu, Zhi Zhang, Binghui Wang, Hua Ma, Alsharif Abuadbba, Minhui Xue, Anmin Fu, and Surya Nepal. Deeptheft: Stealing dnn model architectures through power side channel. *arXiv preprint arXiv:2309.11894*, 2023.

[7] Benjamin Jun Gilbert Goodwill, Josh Jaffe, Pankaj Rohatgi, et al. A testing methodology for side-channel resistance validation. In *NIST non-invasive attack testing workshop*, volume 7, pages 115–136, 2011.

[8] Miao He, Jungmin Park, Adib Nahiyan, Apostol Vassilev, Yier Jin, and Mark Tehranipoor. RTL-PSC: Automated power side-channel leakage assessment at register-transfer level. In *IEEE VLSI Test Symposium (VTS)*, pages 1–6, 2019.

[9] Lukas Huegle, Martin Gotthard, Vincent Meyers, Jonas Krautter, Dennis RE Gnad, and Mehdi B Tahoori. Power2picture: Using generative cnns for input recovery of neural network accelerators through power side-channels on fpgas. In *2023 IEEE 31st Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pages 155–161. IEEE, 2023.

[10] ISO/IEC. Information technology — Security techniques — Testing methods for the mitigation of non-invasive attack classes against cryptographic modules. Standard ISO/IEC 17825:2024, International Organization for Standardization, Geneva, CH, 2024. URL https://www.iso.org/standard/82422.html.

[11] Aruna Jayasena, Emma Andrews, and Prabhat Mishra. TVLA*: Test Vector Leakage Assessment on Hardware Implementations of Asymmetric Cryptography Algorithms. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2023.

[12] Emre Karabulut and Aydin Aysu. Masking FALCON's Floating-Point Multiplication in Hardware. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2024(4):483–508, 2024.

[13] Kazemi, Zahra and Hely, David and Fazeli, Mahdi and Beroulle, Vincent. A review on evaluation and configuration of fault injection attack instruments to design attack resistant MCU-based IoT applications. *Electronics*, 9(7):1153, 2020.

[14] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999. doi: 10.1007/3-540-48405-1_25.

[15] Meng Li, Liangzhen Lai, Vikas Chandra, and David Z Pan. Cross-level monte carlo framework for system vulnerability evaluation against fault attack. In *Proceedings of the 54th Annual Design Automation Conference 2017*, pages 1–6, 2017.

[16] Qiang Liu, Bo Ning, and Pengjie Deng. Information theory-based quantitative evaluation method for countermeasures against fault injection attacks. *IEEE Access*, 7:141920–141928, 2019.

[17] Shayan Moini, Shanquan Tian, Daniel Holcomb, Jakub Szefer, and Russell Tessier. Power side-channel attacks on bnn accelerators in remote fpgas. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 11:357–370, 6 2021. ISSN 21563365. doi: 10.1109/JETCAS.2021.3074608.

[18] Mukherjee, Shubhendu S and Weaver, Christopher and Emer, Joel and Reinhardt, Steven K and Austin, Todd. A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor. In *Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36.*, pages 29–40. IEEE, 2003.

[19] Nahiyan, Adib and Xiao, Kan and Yang, Kun and Jin, Yeir and Forte, Domenic and Tehranipoor, Mark. AVFSM: A framework for identifying and mitigating vulnerabilities in FSMs. In *Proceedings of the 53rd Annual Design Automation Conference*, pages 1–6, 2016.

[20] Kostas Papagiannopoulos, Ognjen Glamočanin, Melissa Azouaoui, Dorian Ros, Francesco Regazzoni, and Mirjana Stojilović. The side-channel metrics cheat sheet. *ACM Computing Surveys*, 55(10):1–38, 2023.

[21] Shari Lawrence Pfleeger and Robert K. Cunningham. Why Measuring Security Is Hard. *IEEE Security & Privacy*, 8(4):46–54, 2010. doi: 10.1109/MSP.2010.60.

[22] Francisco Eugenio Potestad-Ordóñez, Erica Tena-Sánchez, Antonio José Acosta-Jiménez, Carlos Jesús Jiménez-Fernández, and Ricardo Chaves. Hardware countermeasures benchmarking against fault attacks. *Applied Sciences*, 12(5):2443, 2022.

[23] Markku-Juhani O Saarinen. Wip: Applicability of iso standard side-channel leakage tests to nist post-quantum cryptography. In *2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 69–72. IEEE, 2022.

[24] Sayandeep Saha, S Nishok Kumar, Sikhar Patranabis, Debdeep Mukhopadhyay, and Pallab Dasgupta. ALAFA: automatic leakage assessment for fault attack countermeasures. In *Proceedings of the 56th Annual Design Automation Conference 2019*, pages 1–6, 2019.

[25] Shuvo, Amit Mazumder and Pundir, Nitin and Park, Jungmin and Farahmandi, Farimah and Tehranipoor, Mark. LDTFI: Layout-aware timing fault-injection attack assessment against differential fault analysis. In *2022 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 134–139. IEEE, 2022.

[26] François-Xavier Standaert. How (not) to use welch's t-test in side-channel security evaluations. In *International conference on smart card research and advanced applications*, pages 65–79. Springer, 2018.

[27] François-Xavier Standaert. How (not) to use mutual information analysis for evaluating sca security. *CHES*, 2015.

[28] Wang, Huanyu and Li, Henian and Rahman, Fahim and Tehranipoor, Mark M and Farahmandi, Farimah. Sofi: Security property-driven vulnerability assessments of ics against fault-injection attacks. *IEEE*

Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(3):452–465, 2021.

[29] Yuce, Bilgiday and Ghalaty, Nahid Farhady and Schaumont, Patrick. TVVF: Estimating the vulnerability of hardware cryptosystems against timing violation attacks. In *2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 72–77. IEEE, 2015.

[30] Tao Zhang, Jungmin Park, Mark Tehranipoor, and Farimah Farahmandi. PSC-TG: RTL power side-channel leakage assessment with test pattern generation. In *ACM/IEEE Design Automation Conference (DAC)*, pages 709–714, 2021.

Arsalan Ali Malik received the B.S. degree in Electrical Engineering in 2015 and the M.S. degree in Computer Engineering in 2020 from the Sir Syed CASE Institute of Technology, Pakistan. He is currently a Ph.D. candidate in Computer Engineering at North Carolina State University, with research interests in hardware security, fault-injection attacks, and secure AI systems.

Ashley Kurian is a Ph.D. student at North Carolina State University, where her research focuses on machine learning security. Her work explores attacks such as side-channel and cryptanalytic model extraction, and the development of defensive techniques to enhance the resilience of neural networks against these threats.

Harshvadan Mihir is a Ph.D. student in the Department of Electrical and Computer Engineering at North Carolina State University. His research interests include hardware security, fault-injection attacks and pre-silicon vulnerability assessment in heterogeneous AI systems.

Sahan Sanjaya is a Ph.D. student in the Department of Computer and Information Science and Engineering at the University of Florida. His research focuses on hardware security and quantum computing.

Aruna Jayasena is an Assistant Professor in the Department of Computer Science and Engineering at the University of Tennessee, Chattanooga. He received his Ph.D. from the University of Florida in 2025. His research focuses on systems security and secure computing architectures.

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 is an IEEE Fellow, an AAAS Fellow, and an ACM Distinguished Scientist.

Aydin Aysu is an Associate Professor and University Faculty Scholar in the Electrical and Computer Engineering Department at North Carolina State University, where he leads the Hardware Cybersecurity Research Lab. His research has earned NSF CAREER, Google RSP, and Goodnight Innovation Fellow awards, along with three best paper awards (DATE, GLS-VLSI, HASP), two hardware security top picks (IEEE CEDA), an IEEE Micro top picks, and a publicity paper award (DAC). He is an IEEE senior member.