Power Side-Channel Vulnerabilities of a RISC-V Cryptography Accelerator Integrated into CVA6 via Core-V eXtension Interface (CV-X-IF)

Power Side-Channel Vulnerabilities of a RISC-V Cryptography Accelerator Integrated into CVA6 via Core-V eXtension Interface (CV-X-IF)

Behnam Farnaghinejad*, Davide Bellizia†, Alessandra Dolmeta‡, Guido Masera‡,
Antonio Porsia*, Annachiara Ruospo*, Stefano Di Carlo*, Alessandro Savino*, and Ernesto Sanchez*
* Department of Control and Computer Engineering, Politecnico di Torino, Turin, Italy
‡ Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy
† Telsy S.p.A, Rome, Italy

***Abstract*—Modern RISC-V designs increasingly integrate cryptographic accelerators to provide better security features while enhancing performance; however, their vulnerability to power side-channel attacks remains insufficiently investigated. This paper presents a comprehensive evaluation of such vulnerabilities in a RISCV-based AES accelerator connected via the Core-V eXtension Interface (CV-X-IF). The analysis begins at the RTL using simulated power traces, employing KL (Kullback–Leibler) divergence alongside established statistical attacks such as Correlation Power Analysis (CPA) and Differential Power Analysis (DPA). Although the former serves as an early indicator of potential leakage, the simulation results demonstrate its limitations compared to CPA and DPA. To validate these findings, leakage trends are further examined through power measurements in an FPGA implementation.**

The proposed methodology is designed to be broadly applicable to a range of cryptographic workloads and accelerator architectures. It is demonstrated on an AES accelerator implementing the scalar cryptographic extension (Zk) with pre-expanded keys. Our findings reveal that side-channel vulnerabilities can persist even in tightly integrated instruction pipelines, underscoring the importance of early-stage leakage assessment. Notably, the close alignment between RTL-level simulations and FPGA-based measurements highlights the effectiveness of the approach and its practical value for guiding secure hardware design in RISC-V ecosystems. In particular, AES serves only as a case of study; the proposed RTL + FPGA validation flow is generic and can be applied to any cryptographic accelerator.

***Index Terms*—RISC-V, CVA6, Core-V eXtension Interface (CV-X-IF), Cryptographic Accelerator, Early-Stage Side-Channel Evaluation, RTL-Level Analysis, Hardware Security**

I. Introduction

With the growing adoption of RISC-V in performance-critical and security-sensitive applications, hardware cryptographic accelerators are increasingly integrated into processor pipelines to boost efficiency and reduce computational overhead. These accelerators enable fast and energy-efficient execution of encryption algorithms such as the Advanced Encryption Standard (AES). However, integrating cryptographic logic directly into the processor's datapath introduces new attack surfaces, particularly to side-channel attacks (SCA)—a class of attacks that exploit data-dependent variations in power consumption to extract secret keys.

CVA6 [1] (formerly Ariane) is a high-performance, 64-bit open-source RISC-V core that supports standard ISA extensions (IMAC), virtual memory, privilege-level separation, and a modular design suitable for both research and deployment. Its rich feature set and open development model have made it a popular platform for exploring custom extensions and accelerators. To support such extensions without modifying the main processor pipeline, the *Core-V eXtension Interface (CV-X-IF)* [2] provides a standardized mechanism for integrating custom instruction logic. This interface allows new instructions, such as cryptographic operations, to be executed either in a single cycle or over multiple cycles while remaining tightly coupled to the processor pipeline.

While tight integration offers performance benefits, it also introduces new challenges for side-channel security. Despite growing interest in RISC-V cryptographic extensions [4], most studies focus on software AES or loosely integrated hardware IPs. In contrast, the leakage behavior of *tightly coupled* accelerators, especially those embedded via standardized interfaces like CV-X-IF, remains underexplored. These designs may exhibit unique leakage patterns due to synchronized execution, shared datapaths, and fine-grained control signals.

This paper addresses this gap by evaluating the power side-channel leakage of a tightly integrated AES accelerator connected to the CVA6 core through CV-X-IF. The proposed analysis is performed at the simulation level, resorting to the RTL description of the device, as well as through real power measurements directly in an FPGA, allowing for early-stage leakage modeling and post-silicon validation.

The main contributions of this work are as follows:

- We provide a reusable methodology for early-stage side-channel evaluation of cryptographic accelerators integrated into RISC-V cores. This methodology enables pre-silicon identification of security risks through RTL-level analysis and supports validation through FPGA-based measurements.

This work was funded by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

- We propose a simulation-to-silicon validation methodology that combines RTL-based trace analysis with FPGA-based measurements to confirm leakage trends and key recovery feasibility.
- We perform an RTL-level side-channel leakage analysis of a CV-X-IF-integrated AES accelerator using Kullback–Leibler (KL) divergence and standard statistical attacks such as Correlation Power Analysis (CPA) and Differential Power Analysis (DPA).

The rest of the paper is organized as follows. Section II reviews related work on RISC-V cryptographic accelerators and provides background on power side-channel attacks and leakage modeling. Section III describes our threat model, RTL simulation setup, FPGA measurement platform, and attack implementation. Section IV presents experimental results from both simulation and hardware. Section V discusses limitations of RTL-level leakage analysis. Finally, Section VI summarizes the findings and outlines future directions.

II. Background and Related Work

The growing use of RISC-V has driven the development of cryptographic accelerators integrated directly into processor cores. While these designs offer performance and efficiency benefits, their resilience to side-channel attacks (SCA) is poorly investigated. This section reviews RISC-V crypto extensions, introduces key side-channel concepts, and outlines the leakage models and metrics used in this work.

A. RISC-V Cryptographic Accelerators

The open-source nature and modularity of the RISC-V Instruction Set Architecture (ISA) have encouraged the development of domain-specific extensions, including cryptographic accelerators. These accelerators significantly improve performance and energy efficiency over software implementations and are increasingly adopted in academic and industrial designs. Depending on the integration model, cryptographic hardware may be loosely coupled (e.g., a memory-mapped peripheral) or tightly coupled into the main pipeline.

One of the most influential efforts toward standardization was led by Marshall *et al.* [3], whose study explored various Instruction Set Extensions (ISEs) for AES, ultimately contributing to the ratified RISC-V scalar cryptography extension [4]. In a follow-up work [5], the authors presented the first full implementation of this extension on the SCARV core, analyzing the trade-offs in performance, area, and instruction efficiency. However, none of these works considered power side-channel vulnerabilities. More recent work by Szymkowiak *et al.* [7] introduced vector-based cryptographic extensions (Zvk) into the CVA6 processor via the Ara vector unit, demonstrating scalability and performance. However, side-channel security was not investigated, nor was the CV-X-IF-based integration analyzed from a leakage perspective.

Earlier, Fritzmann *et al.* [8] developed Post-Quantum Cryptography (PQC) accelerators that were tightly coupled to lightweight RISC-V cores such as PULPino [21] and VexRiscv [22]. Their work reused existing datapath resources and introduced custom crypto instructions, achieving speedups for lattice-based schemes. However, similar to the scalar and vector efforts, their work did not consider power side-channel leakage or include analysis at the RTL level.

These prior efforts demonstrate the growing interest in efficient cryptographic execution in RISC-V cores. Nonetheless, a critical gap remains: none of the cited works assess the side-channel leakage of cryptographic accelerators *tightly integrated* within a general-purpose RISC-V core such as CVA6. Furthermore, no prior work has evaluated the leakage implications of using CV-X-IF as an integration point. This paper addresses this gap by evaluating the leakage behavior of a CV-X-IF-connected AES accelerator at both the RTL and FPGA levels and demonstrates that the proposed methodology can also be implemented in different scenarios, including those employing processors other than RISC-V.

B. Side-Channel Attack Taxonomy

Side-channel attacks exploit unintentional information leakage from physical phenomena such as power consumption, electromagnetic radiation, or timing behavior. Among these, power SCAs are particularly effective and well-studied in cryptographic contexts. The simplest variant, *Simple Power Analysis* (SPA), visually inspects power traces to infer control flow or sensitive operations, but it is rarely sufficient for modern algorithms like AES. More advanced techniques include:

- **Differential Power Analysis (DPA)**: partitions traces into groups based on key-dependent intermediate values, using statistical tests to detect differences caused by correct versus incorrect key guesses [9].
- **Correlation Power Analysis (CPA)**: computes the Pearson correlation coefficient between measured traces and hypothetical leakage values (e.g., Hamming weight or Hamming distance) based on key hypotheses [10].
- **Advanced attacks**: such as template attacks [11], machine learning-based methods [12] or Soft Analytical SCAs(SASCA) [13], which often require profiling and more attacker knowledge but can extract keys with fewer traces.

Both CPA and DPA are considered *non-profiled attacks*, requiring only known plaintexts and power measurements, and are widely used in practical security evaluations.

This work focuses on CPA and DPA as representative techniques for assessing side-channel leakage in both RTL simulation and FPGA-based experiments.

C. CMOS Power Model and Leakage

At the hardware level, digital circuits consume dynamic power primarily due to switching activity [14].

For CMOS-based designs, the dynamic power consumption is modeled as follows:

$$P_{\text{dynamic}} = \alpha C V^2 f \tag{1}$$

where α is the switching activity factor, C is the load capacitance, V is the supply voltage, and f is the clock frequency.

Since α depends on input data and logic transitions, cryptographic operations exhibit data-dependent power consumption, even when using secure algorithms such as AES. To model this leakage, attackers often rely on simplified abstractions such as the *Hamming Weight* (HW) or *Hamming Distance* (HD) of internal values. The HW model assumes power to be proportional to the number of bits set to 1. In contrast, the HD model approximates the number of bit flips between two states, reflecting the switching activity across clock cycles [15]. Although these models capture the primary data-dependent behavior, they may overlook important physical effects in tightly integrated designs. Additional leakage sources, such as propagation of errors, interconnect capacitance, and unbalanced logic paths, can become significant, especially when encryption operations are embedded deep within the processor pipeline. These effects are best observed through the analysis at the RTL level, highlighting the importance of assessing early-stage leakage in modern hardware designs.

D. KL Divergence and Adversary Failure Probability

Kullback–Leibler (KL) divergence is widely used as a metric to detect leakage in side-channel analysis [16]–[18]. When the power leakage probability distributions for two different keys are distinguishable, the KL divergence is expected to be high, implying that an adversary can correlate power consumption with the key hypothesis. This allows the leakage assessment to proceed without full key recovery attacks. The KL divergence is computed empirically, using the discrete definition:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \tag{2}$$

where $P(x)$ and $Q(x)$ are the estimated probability distributions of switching activity or trace values under two key hypotheses.

Importantly, KL divergence can be quantitatively linked to the attacker's *failure probability* (Pr_F), defined as the likelihood that a key guess is incorrect. As shown in [16]–[18], KL thresholds correspond to various failure probabilities. For instance, a $\mathrm{Pr}_F > 0.90$ requires a KL divergence of less than 0.03 for the attacker to fail 90% of the time. Table I summarizes these thresholds.

TABLE I: KL Divergence Thresholds for Different Failure Probabilities (Pr_F)

Pr_F	KL	Pr_F	KL
> 0.96	< 0.01	> 0.53	< 0.78
> 0.90	< 0.03	> 0.45	< 1.12
> 0.80	< 0.12	> 0.38	< 1.53
> 0.71	< 0.28	> 0.32	< 2.00
> 0.61	< 0.50	> 0.26	< 2.53

Furthermore, the number of traces N also plays a critical role. As noted in [17], to assert with confidence level $(1-\alpha)$ that distributions $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$ and $Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$ are distinguishable, the following must hold:

$$N \geq \frac{(\sigma_X + \sigma_Y)^2 \cdot z_{1-\alpha/2}^2}{\epsilon^2 (\mu_X - \mu_Y)^2} \tag{3}$$

This expression bounds the minimum number of traces required to distinguish between keys with KL-based confidence.

Finally, KL divergence is directly connected to the attacker's *success rate (SR)* under a Gaussian assumption:

$$\mathrm{SR} \approx \Pr\left[\hat{k} = k^*\right] = \Pr\left[L(k^*;t) - L(\bar{k};t) > 0\right] \tag{4}$$

where the expectation of $L(k^*) - L(\bar{k})$ is equal to the KL divergence between the correct and incorrect key distributions [16], [17]. This mathematical relationship justifies the interpretation of KL as a predictor of attack success, and we use it in this work to assess leakage at the simulation level.

III. Methodology

This section outlines the methodology used to evaluate side-channel leakage in the CVA6-integrated AES accelerator. It defines the threat model and assumptions, describes the RTL design and its integration via CV-X-IF, presents the RTL simulation and FPGA measurement setups, details the implementation of side-channel attacks, and discusses the limitations of RTL-level leakage analysis.

A. Threat Model and Assumptions

We assume a powerful adversary with full knowledge of the cryptographic algorithm and its hardware implementation. The attacker has physical access to the device and can measure its instantaneous power consumption through the $\mathrm{V_{DD}}$ rail or a dedicated probe circuit. The goal is to recover the secret key by analyzing the variations in power consumption in multiple encryption executions. In particular, mainly targeting a RISC-V-based device, the cryptographic extension used in this study is the scalar AES extension (Zk), as specified in the RISC-V scalar cryptography specification [4]. The key schedule is not included in the monitored execution: key expansion is performed prior to the monitored AES instructions and stored in memory before encryption begins.

To maximize the effectiveness of side-channel attacks, the adversary can insert software or hardware triggers to isolate the encryption window and achieve trace alignment. We further assume that the attacker can control or observe plaintext inputs and collect many power traces. While CPA and DPA are concrete key-recovery techniques, this work employs KL divergence as a pre-silicon security evaluation metric. This enables early detection of data-dependent leakage at the RTL level without requiring full key recovery attacks. It measures the distinguishability of power distributions under different key hypotheses, helping to identify vulnerable cycles or modules before fabrication.

B. RTL Design and Integration via CV-X-IF

The AES accelerator under evaluation is integrated into the CVA6 core through the CV-X-IF. This interface enables the definition of standard or custom instructions without modifying the main pipeline stages, allowing the AES module to behave as a tightly coupled functional unit inside the execution stage. The implemented design supports the scalar RISC-V cryptographic extension and executes AES instructions in either single-cycle or multi-cycle configurations, depending on the operation and microarchitecture. This work focuses on a single-cycle execution design, which raises mainly two distinct side-channel concerns.

- **Single-Cycle Side-Channel Considerations:** When AES operations are executed in a single cycle, multiple logic blocks switch simultaneously, including SubBytes (S-box), ShiftRows, and MixColumns. This consolidated activity may increase the switching noise but also amplify key-dependent transitions. From a security standpoint, this can create short but intense leakage windows that an attacker can exploit if the trace alignment is precise.
- **Pipeline-Level Observability:** Since the AES execution is embedded within the pipeline, the power trace reflects the AES logic and the interaction with core components such as the register file, forwarding paths, and issue logic. These interactions may introduce indirect leakage vectors or increase measurement noise, which must be considered during evaluation.

This tightly integrated configuration exemplifies the challenges of performing early-stage leakage analysis in modern processor designs.

C. Experimental Setup

We evaluate side-channel leakage using both RTL simulation and physical FPGA measurements.

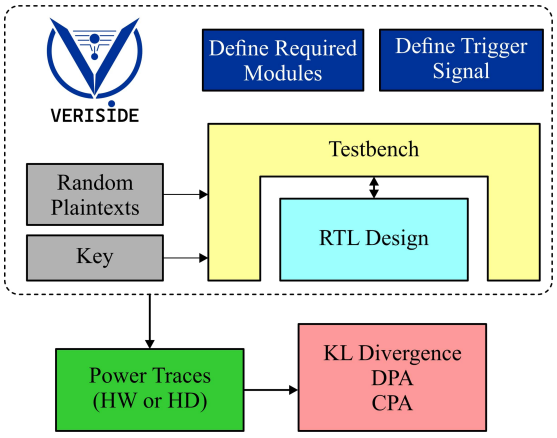


Fig. 1: RTL simulation setup and trace extraction.

For RTL-level simulations, we use the VeriSide framework [19] to extract power traces (modelled as HW or HD) from simulations of the CVA6 core integrated with the AES accelerator. The design is instrumented to produce SIDE files, which log the accumulated switching activity of internal signals cycle by cycle during encryption. Unlike traditional waveform-based approaches, VeriSide enables leakage analysis without generating large Value Change Dump (VCD) or Switching Activity Interchange Format (SAIF) files, greatly reducing storage overhead and reducing memory usage during post-processing. This flow is shown in Figure 1. Test vectors of known plaintexts are injected via a bare-metal test program running on CVA6, and each encryption generates one RTL-level trace. A wide set of plaintexts is simulated under a fixed key to build a trace matrix for analysis via KL divergence and correlation-based attacks. To validate the simulation results, we replicate the same CVA6 + AES RTL design on a CW305 Artix FPGA development board [6]. The board is equipped with a shunt resistor on the V_{DD} rail to measure the absorbed current, along with a 20dB low-noise preamplifier, and uses a digital trigger to isolate the encryption window [20]. A custom test harness sends plaintexts via UART and collects corresponding power traces using a Picoscope 6404D oscilloscope. This setup, shown in Figure 2, enables a direct comparison between simulation-based leakage indicators and real measured traces, bridging the gap between pre-silicon analysis and physical side-channel behavior.

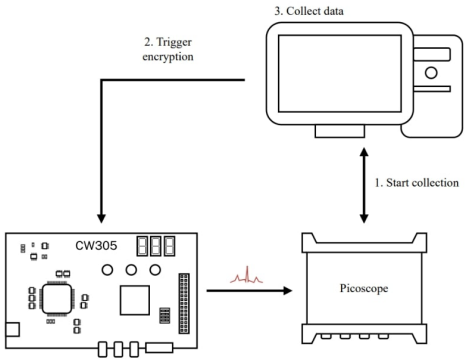


Fig. 2: Physical setup for FPGA power measurements.

D. Attack Implementation

We conduct CPA and DPA attacks on both RTL simulation traces and measured FPGA power data. Algorithm 1 summarizes the CPA procedure, which targets the SBox input of the final AES round. This point was selected because of its strong data dependency and consistency with the leakage patterns observed in RTL simulations.

Algorithm 1 CPA on AES **last-round** SBox input using Hamming-weight model

Input: Ciphertexts $\{c_i\}_{i=1}^{D}$, power traces $\{t_i\}_{i=1}^{D}$, key hypotheses $\{k_j\}_{j=1}^{K}$
Output: Correlation matrix $R \in \mathbb{R}^{K \times T}$

for $j \leftarrow 1$ **to** K **do**
  for $i \leftarrow 1$ **to** D **do**
    $v_{i,j} \leftarrow \texttt{SBoxInput}(c_i, k_j)$
    $h_{i,j} \leftarrow \texttt{HW}(v_{i,j})$
  end
  for $t \leftarrow 1$ **to** T **do**
    $R[j,t] \leftarrow \texttt{PearsonCorr}(\{h_{i,j}\}, \{t_{i,t}\})$
  end
end
return R

In the CPA attack, for each key hypothesis, we compute hypothetical intermediate values (Line 3) and use the HW model to estimate their power consumption (Line 3). These predicted leakage vectors are then correlated with actual power traces using Pearson correlation (Line 6), yielding a correlation matrix indexed by key and time sample. The key hypothesis with the highest correlation peak is selected as the most likely. In the DPA attack, the intermediate values are used to partition the traces into two groups based on a selected bit (e.g., MSB or LSB) of the hypothetical SBox input. The difference of mean traces between the two groups is computed for each time point. Peaks in the resulting difference-of-means signal may indicate key-dependent leakage. The DPA flow is summarized in Algorithm 2.

As shown in Lines 12–15, traces are partitioned into two sets depending on the output of the `BitSelect` function applied to intermediate values (Line 12). The mean trace of each group is computed, and their difference is used to construct a DPA trace (Line 19). Peaks in these difference traces may reveal key-dependent leakage. Unlike CPA, this approach does not require a leakage model, making it more general but typically less sensitive.

Algorithm 2 DPA on AES last round SBox input using bit-based partitioning

Input: Ciphertexts $\{c_i\}_{i=1}^{D}$, power traces $\{t_i\}_{i=1}^{D}$, key hypotheses $\{k_j\}_{j=1}^{K}$
Output: DPA traces $D_j \in \mathbb{R}^T$ for each key guess

10 **for** $j \leftarrow 1$ **to** K **do**
11 Initialize $G_0 \leftarrow \emptyset, G_1 \leftarrow \emptyset$ **for** $i \leftarrow 1$ **to** D **do**
12 $v_{i,j} \leftarrow \texttt{SBoxInput}(c_i, k_j)$
  if $\mathit{BitSelect}(v_{i,j}) == 0$ **then**
13 Append t_i to G_0
14 **end**
15 **else**
16 Append t_i to G_1
17 **end**
18 **end**
19 $D_j \leftarrow \texttt{Mean}(G_1) - \texttt{Mean}(G_0)$
20 **end**
21 **return** $\{D_j\}$

Although CPA is known to outperform DPA on AES; DPA is included *for completeness and cross-validation* with the RTL Hamming-distance traces. In this way, it is possible to guarantee that the methodology remains suitable for different attacker models. Both algorithms were applied independently to traces from RTL simulations and FPGA measurements. Our experimental results demonstrate that CPA was consistently successful in recovering the correct key bytes from the last round, while DPA often failed to yield distinguishable leakage due to overlapping operations and pipeline noise in the tightly integrated design.

IV. Experimental Results

This section reports results from RTL simulations and FPGA measurements, covering leakage detection, key recovery, trace count analysis, RTL vs. physical correlation, and the performance-security trade-off.

A. KL Divergence and Early Leakage Detection

To evaluate power leakage at the RTL level, we applied the Kullback–Leibler (KL) divergence metric across clock cycles using switching activity extracted from 10,000 AES encryption simulations. The aim is to assess whether data-dependent variations between key hypotheses result in statistically distinguishable power traces.

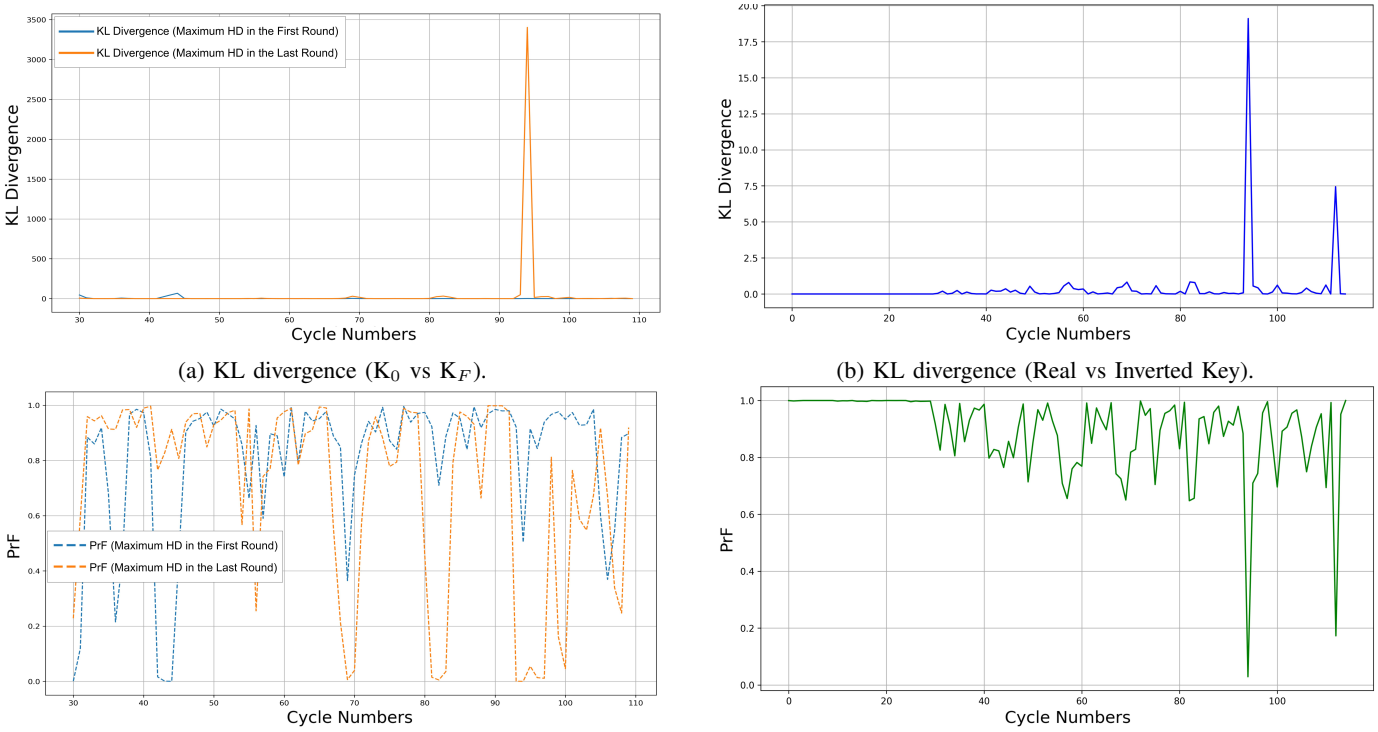
We evaluated KL divergence under two key models:

- **Key Model A:** Two synthetic key pairs designed to induce maximum Hamming distance in intermediate values: (i) an all-zero 128-bit key vs. an all-one key, targeting values in the first round; (ii) a custom key pair selected to produce maximum Hamming distance in the last round: `0x15f151742eb20b8a1dd1b66ce46cd389` versus `0xbd1561abc35fd594e9259b67f67ef268`.
- **Key Model B:** A realistic key used in the CPA/DPA experiments (`0x00ff00ff11ee22dd33cc44bb55aa6699`) compared against its bitwise-inverted version. This alternating structure promotes balanced switching while providing a meaningful attack target.

We highlight key insights from the KL divergence and PrF analysis under two key models:

- **First-Round Leakage (Model A)** Figure 3a shows clear KL divergence peaks around cycles 34–36 and 44–46 when comparing all-zero and all-one keys for the first round. While these peaks suggest key-dependent activity, neither CPA nor DPA attacks targeting these regions resulted in successful key recovery.
- **Last-Round Leakage (Model A):** The same key pair comparison applied to the last round produces a distinct KL divergence peak (KL $\approx$ 19.1 at cycle 94), clearly revealing key-dependent leakage. This aligns with the cycle where CPA successfully recovers the correct key bytes. However, the corresponding PrF curve indicates that multiple other cycles also exhibit low failure probabilities, despite not leading to successful key recovery. This suggests that while KL can identify genuine leakage in this case, PrF may produce false positives and should not be solely relied upon for assessing attack feasibility.
- **Real vs Inverted Key (Model B):** Using the actual key from CPA/DPA experiments and its inverted version reveals a substantial KL peak ($\approx$19.1) at cycle 94, as shown in Figure 3b. This aligns with the last round and the successful CPA results, indicating that key choice significantly influences leakage observability in simulation.
- **Failure Probability (PrF):** Figures 3c and 3d show the estimated adversary failure probability based on KL values under Gaussian assumptions. In the first-round case, several cycles exhibit low PrF (< 0.5), suggesting possible leakage, yet key recovery attacks failed—indicating false positives. In contrast, the last-round case yields both a clear KL peak and a low PrF (high leakage confidence) near cycle 94, which correlates with successful CPA results. These findings reinforce that while PrF can support early-stage analysis, it may overestimate leakage and should be validated through practical attacks.

From this analysis, several important insights emerge regarding the use of KL divergence and PrF as early leakage



(a) KL divergence (K_0 vs K_F).

(b) KL divergence (Real vs Inverted Key).

(c) PrF – Max HD (K_0 vs K_F).

(d) PrF – Real vs Inverted Key.

Fig. 3: KL divergence and failure probability (PrF) across clock cycles under two key models.

indicators:

- **Sensitivity to Key Hypotheses:** KL divergence is highly sensitive to the choice of key hypotheses and intermediate values. Synthetic key pairs that maximize Hamming distance may exaggerate leakage that is not exploitable in practice.
- **Limitations of PrF:** PrF derived from KL can misrepresent security: low PrF cycles may not yield successful attacks, and high PrF cycles may still leak exploitable information.
- **Importance of Realistic Keys:** Realistic key models provide more meaningful leakage evaluation and better align with CPA success.
- **Need for Validation:** RTL-based metrics like KL and PrF should be interpreted cautiously and always confirmed through trace-based statistical attacks.

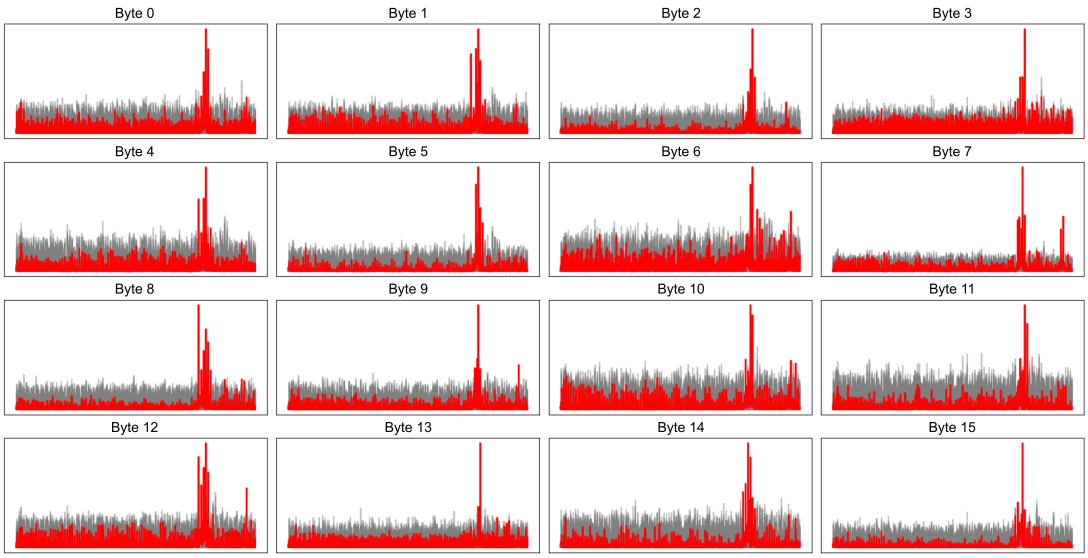
B. CPA/DPA results on RTL and FPGA

We evaluated the effectiveness of both CPA and DPA on the AES accelerator using power traces from RTL simulation and FPGA execution. The attacks focus on the input to the SubBytes operation in the final AES round, where each round is executed in a single instruction cycle via the scalar cryptography extension.

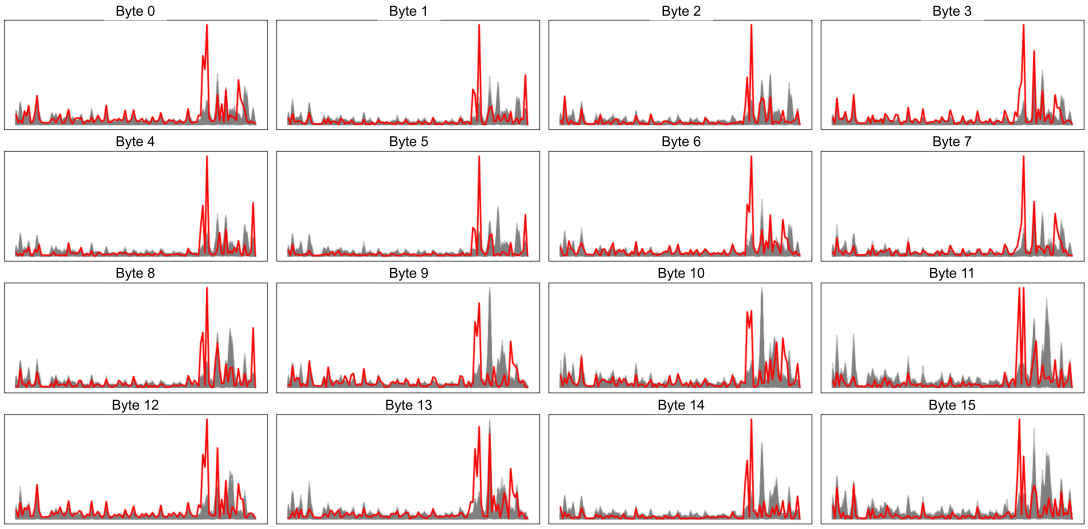
CPA (FPGA and Simulation). Figure 4 displays the CPA results for all 16 key bytes using traces from the FPGA and RTL simulation, respectively. Given the 64-bit CVA6 architecture and the synchronous triggering of the AES instruction, subtle differences in byte-level leakage may arise due to layout, datapath interactions, and bit-flip locality. Including the full set of bytes ensures completeness and strengthens the case for end-to-end validation. In both analyses, the highest correlation value for each byte aligns with the correct key hypothesis. This indicates that even in the presence of pipeline compression and instruction-level integration, CPA remains highly effective at recovering the AES subkey in the last round.

DPA (FPGA and Simulation). Figure 5 shows the corresponding DPA results. Here, the analysis highlights that peaks do not consistently align with the last round and often fail to isolate the correct key values. In the FPGA case, we observed successful recovery for 7 bytes consistently. For RTL simulation, the results improve but are still less definitive than CPA. The CPA aligns with the last round, while DPA leakage does not. Moreover, since DPA relies on mean differences of grouped traces based on a single key bit, its leakage signal is more susceptible to noise and interleaved switching. Increasing the number of traces or applying enhanced preprocessing techniques (e.g., template matching or alignment heuristics) may be required to improve DPA's success rate.

The results highlight the resilience of CPA across platforms and abstraction levels, particularly in the last round of AES. In contrast, DPA was only partially successful, primarily outside the last round, and never achieved full key extraction. These findings confirm that compressed, instruction-bound AES operations are more robust against DPA in practice but not immune to CPA unless specific countermeasures are



(a) CPA results – FPGA traces.



(b) CPA results – RTL simulation traces.

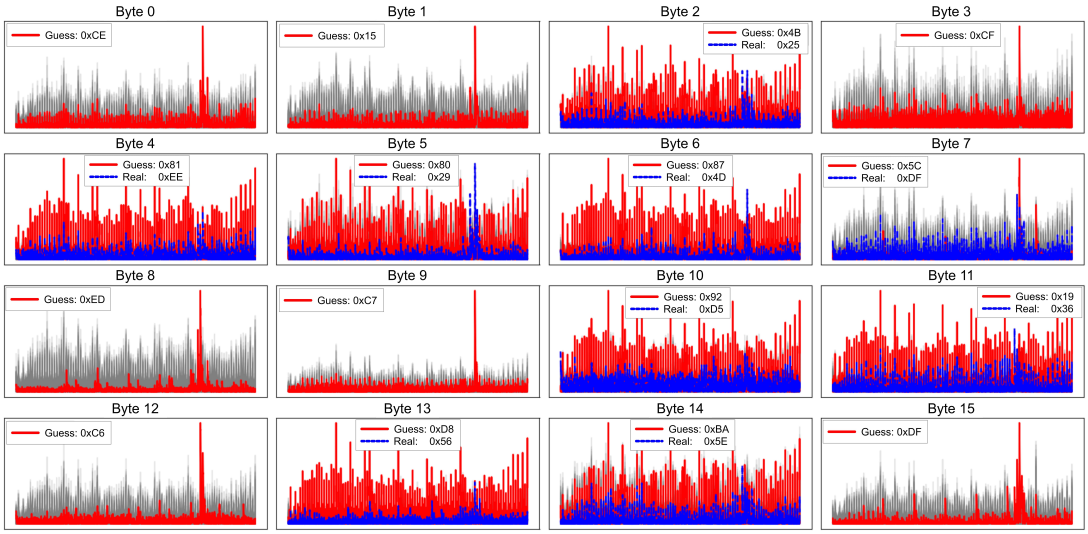
Fig. 4: CPA correlation for all AES key bytes using FPGA and RTL simulation traces. In both cases, the correct key hypothesis yields the highest correlation, confirming successful key recovery in the last round.

introduced.

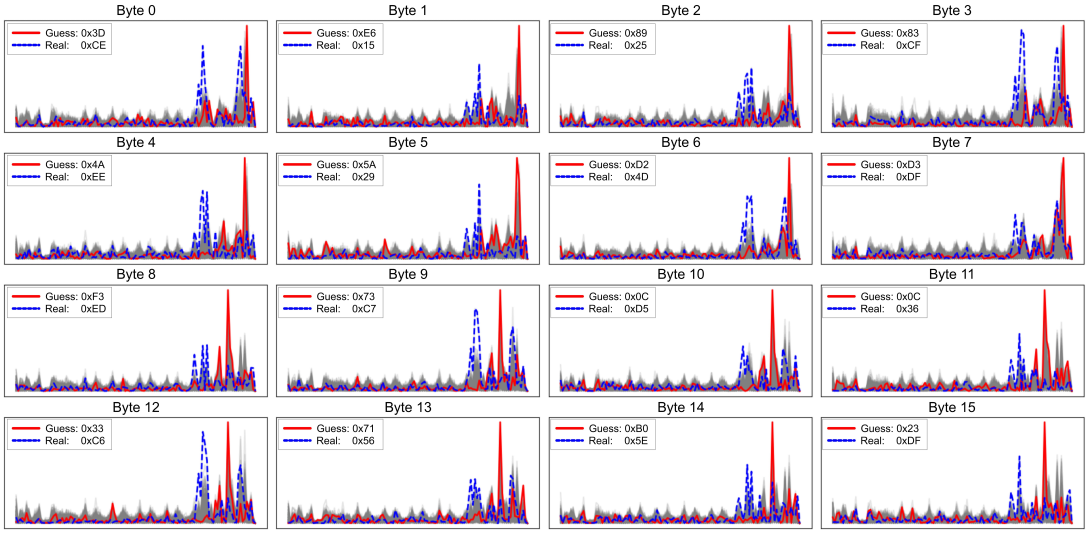
C. Number of Traces for Key Recovery

To evaluate attack complexity and the statistical effort required for key recovery, we analyze how CPA correlation evolves with increasing trace counts. Figure 6 shows the correlation trends for each AES key byte, computed separately using physical traces from the FPGA (left) and simulated traces from RTL activity (right). In both experiments, the attacks target the input of the last round SubBytes transformation. For each byte, two curves are plotted: the correlation for the correct key guess and the highest correlation observed among all incorrect guesses. In the FPGA dataset, clear separation emerges within approximately 8,000–10,000 traces for most key bytes. Simulated data demonstrates similar correlation behavior, with several key bytes becoming distinguishable after 1,000–2,000 traces. While the required trace count varies per byte due to data-dependent activity, the consistency of correct key dominance highlights CPA's effectiveness.

These results further confirm a key insight from this work: even in the absence of notable KL divergence at the RTL level (as shown in previous sections), CPA remains capable of extracting keys at both pre- and post-silicon stages. Thus,



(a) DPA results – FPGA traces.



(b) DPA results – RTL simulation traces.

Fig. 5: DPA results for all AES key bytes. While some bytes show distinguishable peaks, most fail to identify the correct key in the last round. DPA success is weaker and may require more traces or alignment refinement.

statistical success in CPA does not necessarily correlate with RTL leakage metrics such as KL divergence or PrF, underscoring the importance of end-to-end trace-based evaluations for comprehensive leakage assessment.

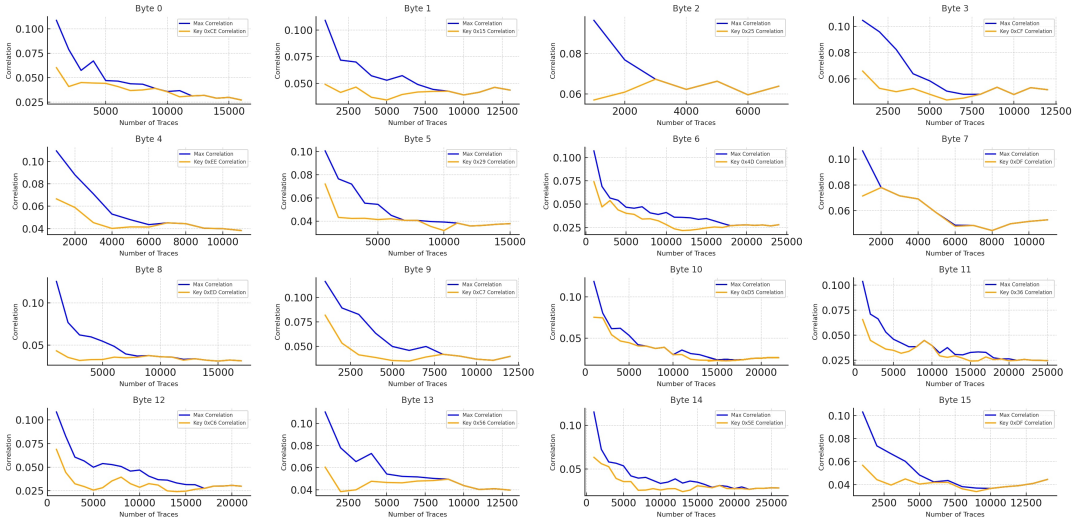
D. RTL vs. Physical Correlation Trends

To assess consistency between simulation-based and physical measurements, we compute a cross-correlation matrix between power traces collected from RTL simulations and those acquired from the FPGA board. Figure 7 visualizes the correlation coefficients for 40K randomly selected traces from each domain. The diagonal structure indicates that corresponding traces from simulation and FPGA share similar temporal and data-dependent leakage behavior. Localized red regions along the diagonal, despite measurement noise, confirm that the simulation captures realistic switching patterns observable during post-silicon experiments. The average diagonal correlation coefficient is 0.012, demonstrating a non-negligible alignment between the domains and supporting the use of RTL-based power modeling for early leakage diagnosis.

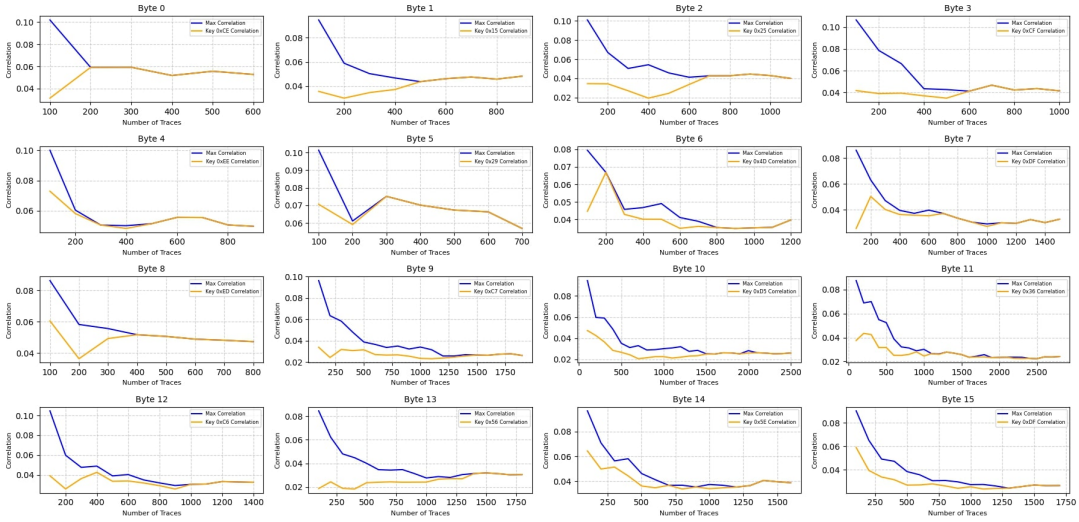
E. Performance vs. Security

The AES accelerator implemented via scalar cryptography extensions offers significant performance advantages. However, our experimental results demonstrate that such acceleration does not inherently prevent side-channel leakage. Leakage remains observable even under dense execution patterns and low-level switching activity.

While KL divergence and PrF provide early indications of potential leakage, our findings show that they may produce false



(a) FPGA-based correlation vs. trace count for last round key bytes.



(b) RTL simulation correlation vs. trace count for last round key bytes.

Fig. 6: CPA convergence per byte. Orange curve – correlation of the *correct* key guess; blue curve – highest correlation observed among all *incorrect* guesses. A byte is deemed recovered once the orange curve overlaps the blue curve.

positives or overlook subtle but exploitable vulnerabilities. For instance, KL divergence revealed key-dependent variations in the last round when appropriate key hypotheses were chosen, aligning with successful CPA results. Conversely, multiple cycles with low PrF did not lead to successful key recovery, highlighting the limitations of relying solely on statistical metrics.

V. Discussion: Limitations of RTL-Level Leakage Analysis

Although RTL-level simulations provide early insight into switching activity and functional behavior, they lack access to several critical physical effects that influence side-channel leakage in real implementations. As a result, RTL-based leakage indicators can offer useful trends but must be interpreted with caution. Key limitations of RTL-level analysis include:

- **Glitch propagation:** RTL abstractions assume ideal, hazard-free transitions. In post-synthesis or gate-level designs, glitches caused by logic hazards can significantly increase switching activity and leakage, particularly in combinational blocks.
- **Signal arrival time and race conditions:** RTL models evaluate all signals at discrete clock boundaries without modeling within-cycle transition timing. In practice, varying arrival times can impact correlation or mask leakage.
- **Physical capacitance and fan-out:** The power consumed by real hardware also depends on wire length, load capacitance, and fan-out—none of which are captured in pre-synthesis RTL. Functionally equivalent operations may exhibit different power signatures due to layout.
- **Switching concurrency:** Operations that occur simulta-

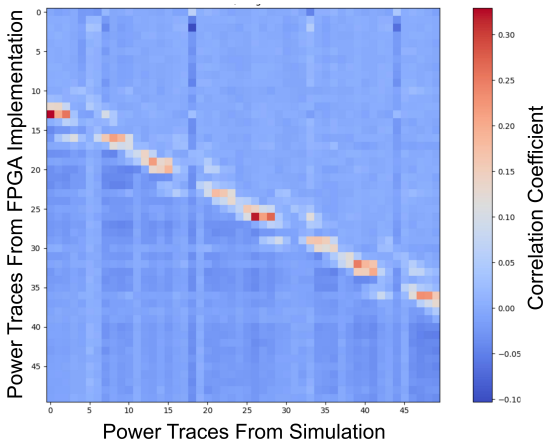


Fig. 7: Trace-to-trace correlation matrix between simulation and FPGA traces

neously at the RTL level may cause overlapping transitions in hardware, potentially amplifying or masking leakage depending on toggle patterns.

VI. Conclusion and Future Work

This work investigated power side-channel vulnerabilities in a RISC-V AES accelerator tightly integrated into the CVA6 core via CV-X-IF, while concurrently evaluating the effectiveness of RTL-level analysis as a practical tool for early-stage leakage detection. Our results show that RTL-level simulation reveals data-dependent switching patterns that align with post-silicon leakage measured on an FPGA, validating its utility as a first-pass screening method.

This analysis confirms that meaningful leakage can be detected before synthesis, enabling the insertion of countermeasures—such as masking, balancing, or microarchitectural changes—during design, thereby reducing costly post-silicon fixes. While RTL screening is insufficient, it plays a critical role in a staged evaluation flow that must include gate- and layout-level analysis.

Although focused on a scalar AES accelerator, the methodology generalizes to other cryptographic workloads and designs. Future work will extend the flow to pipelined and multicycle CV-X-IF accelerators and incorporate masking and shielding countermeasures into the RTL analysis framework to quantify their effectiveness.

References

[1] F. Zaruba and L. Benini, The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-Ready 1.7-GHz 64-Bit RISC-V Core in 22-nm FDSOI Technology. (Jul. 2019). doi: 10.1109/TVLSI.2019.2926114.

[2] "OpenHW Group Specification: Core-V eXtension interface (CV-X-IF) - Development — Core-V eXtension interface (CV-X-IF) v1.0.0-rc.3-dev.3 documentation." Accessed: Feb. 21, 2025. [Online]. Available: https://github.com/openhwgroup/core-v-xif/tree/v1.0.0

[3] B. Marshall, G. R. Newell, D. Page, M.-J. O. Saarinen, and C. Wolf, "The design of scalar AES Instruction Set Extensions for RISC-V," IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 109–136, 2021, doi: 10.46586/tches.v2021.i1.109-136.

[4] "RISC-V Cryptography Extensions Volume I: Scalar & Entropy Source Instructions (v1.0.0-rc3)"

[5] B. Marshall, D. Page, and T. Pham, "Implementing the Draft RISC-V Scalar Cryptography Extensions," in Hardware and Architectural Support for Security and Privacy, Virtual Greece: ACM, Oct. 2020, pp. 1–8. doi: 10.1145/3458903.3458904.

[6] C. O'Flynn, "CW305 Artix Target Board," *ChipWhisperer Documentation*. Accessed: Apr. 3, 2025. [Online]. Available: https://rtfm.newae.com/Targets/CW305

[7] T. Szymkowiak, E. Isufi, and M.-J. Saarinen, "Marian: An Open Source RISC-V Processor with Zvk Vector Cryptography Extensions," 2024, 2024/1449. Accessed: Feb. 24, 2025. [Online]. Available: https://eprint.iacr.org/2024/1449

[8] T. Fritzmann, G. Sigl, and J. Sepúlveda, "RISQ-V: Tightly Coupled RISC-V Accelerators for Post-Quantum Cryptography," IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 239–280, Aug. 2020, doi: 10.13154/tches.v2020.i4.239-280.

[9] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in Advances in Cryptology — CRYPTO' 99, vol. 1666, M. Wiener, Ed., in Lecture Notes in Computer Science, vol. 1666. , Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397. doi: 10.1007/3-540-48405-1.

[10] E. Brier, C. Clavier, and F. Olivier, "Correlation Power Analysis with a Leakage Model," in Cryptographic Hardware and Embedded Systems - CHES 2004, vol. 3156, M. Joye and J.-J. Quisquater, Eds., in Lecture Notes in Computer Science, vol. 3156. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 16–29. doi: 10.1007/978-3-540-28632-5.

[11] S. Chari, J. R. Rao, and P. Rohatgi, "Template Attacks," in Cryptographic Hardware and Embedded Systems - CHES 2002, vol. 2523, B. S. Kaliski, Ç. K. Koç, and C. Paar, Eds., in Lecture Notes in Computer Science, vol. 2523. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 13–28. doi: 10.1007/3-540-36400-5_3.

[12] L. Weissbart, S. Picek, and L. Batina, "One trace is all it takes: Machine Learning-based Side-channel Attack on EdDSA," 2019, 2019/358. Accessed: Apr. 04, 2025. [Online]. Available: https://eprint.iacr.org/2019/358

[13] N. Veyrat-Charvillon, B. Gérard, and F.-X. Standaert, "Soft Analytical Side-Channel Attacks," 2014, 2014/410. Accessed: Apr. 04, 2025. [Online]. Available: https://eprint.iacr.org/2014/410

[14] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed., Upper Saddle River, NJ: Prentice Hall, 2003.

[15] "Power Analysis Attacks: Revealing the Secrets of Smart Cards — SpringerLink." Accessed: Jan. 06, 2025. [Online]. Available: https://link.springer.com/book/10.1007/978-0-387-38162-6.

[16] J. Park and A. Tyagi, "Security Metrics for Power Based SCA Resistant Hardware Implementation," in 2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), Jan. 2016, pp. 541–546. doi: 10.1109/VLSID.2016.43.

[17] N. Pundir, J. Park, F. Farahmandi, and M. Tehranipoor, "Power Side-Channel Leakage Assessment Framework at Register-Transfer Level," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 30, no. 9, pp. 1207–1218, Sep. 2022, doi: 10.1109/TVLSI.2022.3175067.

[18] M. He, J. Park, A. Nahiyan, A. Vassilev, Y. Jin, and M. Tehranipoor, "RTL-PSC: Automated Power Side-Channel Leakage Assessment at Register-Transfer Level," in 2019 IEEE 37th VLSI Test Symposium (VTS), Apr. 2019, pp. 1–6. doi: 10.1109/VTS.2019.8758600.

[19] B. Farnaghinejad, A. Porsia, A. Ruospo, A. Savino, S. Di Carlo, and E. Sanchez, "Late Contribution: VeriSide: A Modified Verilator for Leakage Assessment at the RTL Level," in 2025 IEEE 26th Latin American Test Symposium (LATS), Mar. 2025, pp. 1–2. doi: 10.1109/LATS65346.2025.10963943.

[20] D. Bellizia, B. Udvarhelyi, and F.-X. Standaert, "Towards a Better Understanding of Side-Channel Analysis Measurements Setups," in Smart Card Research and Advanced Applications: 20th International Conference, CARDIS 2021, Lübeck, Germany, November 11–12, 2021, Revised Selected Papers, Berlin, Heidelberg: Springer-Verlag, Nov. 2021, pp. 64–79. doi: 10.1007/978-3-030-97348-3_4.

[21] pulp-platform/pulpino. (Apr. 01, 2025). C. pulp-platform. Accessed: Apr. 04, 2025. [Online]. Available: https://github.com/pulp-platform/pulpino

[22] SpinalHDL/VexRiscv. (Apr. 03, 2025). SpinalHDL. Accessed: Apr. 04, 2025. [Online]. Available: https://github.com/SpinalHDL/VexRiscv