Autonomous LLM-Orchestrated Side-Channel Extraction Against Fully Unrolled and Masked Architectures

Mani Rupak Gurram[1], Daniel Ifeoluwa Idowu[1], Yamini Swetha Nadella[1], Nouf Nur Nabilah[1]
Sarita Bista[1], Mohamed Chouikha[1], Annamalai Annamalai[1], Akshay "AK" Raghavendra Kulkarni[1]

[1]Department of Electrical and Computer Engineering, Prairie View A&M University, TX, USA
Emails: {mgurram, didowu3, ynadella, nnabilah, sbista, mfchouikha, aaannamalai, arkulkarni}@pvamu.edu

***Abstract*—Unrolled cryptographic hardware architectures are increasingly deployed to maximize throughput, inherently introducing massive algorithmic noise floors that frequently thwart traditional temporal Side-Channel Analysis (SCA). However, the reliance on structural combinational noise as a standalone countermeasure remains underexplored against adaptive, AI-driven profiling. This work presents a novel autonomous framework utilizing a Large Language Model (LLM) agent to orchestrate and execute differential power evaluations against a 161,000-gate fully unrolled AES-128 core on a target CW305 FPGA. We first establish a baseline, demonstrating that standard Correlation Power Analysis (CPA) systematically fails to penetrate the unrolled noise floor, yielding statistically insignificant correlations ($r \approx 0.11$). In response to this heuristic failure, the autonomous agent dynamically pivots to a Zero-State Differential Power Isolation methodology. By leveraging single-channel baseline subtraction, the agent mathematically cancels multi-round algorithmic noise from the global power trace, successfully isolating the target combinational leakage and achieving peak correlations exceeding $r = 0.318$ across all 16 state bytes. Furthermore, by comparing the extracted physical signatures to the logical target state, the framework autonomously extracts 16 unique physical-to-logical routing maps. This demonstrates that while automated Electronic Design Automation (EDA) synthesis inadvertently introduces physical bit-level obfuscation, these synthesis optimizations can be systematically reverse-engineered by agentic profiling. Ultimately, this work proves that unrolled combinational architectures cannot serve as a robust defense against adaptive, autonomous side-channel characterization.**

***Index Terms*—Side-Channel Analysis, Hardware Security, Unrolled Cryptography, LLM Agent, Differential Power Analysis, FPGA Synthesis Obfuscation**

I. Introduction

MODERN digital infrastructure relies fundamentally on cryptographic primitives to ensure the confidentiality and integrity of global communications. While algorithms such as the Advanced Encryption Standard (AES) [1] are mathematically robust against classical cryptanalysis, their physical hardware implementations remain highly susceptible to Side-Channel Analysis (SCA). Unlike theoretical attacks that target the mathematics of a cipher, SCA exploits unintended physical emanations such as power consumption and electromagnetic radiation leaked by the underlying complementary metal-oxide-semiconductor (CMOS) [2] logic during execution. Through statistical methods like Correlation Power Analysis (CPA) [3], attackers can correlate these dynamic power fluctuations to intermediate data values, effectively extracting secret keys from otherwise secure systems.

Faced with this threat, hardware designers must carefully weigh spatial and temporal trade-offs when implementing cryptographic cores. Traditionally, architectures are deployed in either an iterative or pipelined paradigm. In an iterative design [4], a single physical encryption round is synthesized, and data is fed back through this same logic block over multiple clock cycles. Pipelined architectures [5]–[7]synthesize all rounds but separate them using state registers (flip-flops) to allow multiple plaintexts to be processed concurrently. While efficient for standard applications, both iterative and pipelined architectures share a critical vulnerability to SCA: the presence of intermediate state registers. These registers halt the data flow at each clock edge, temporarily isolating the execution of a single encryption round. This temporal isolation provides a clean, unambiguous power signature that allows attackers to easily target a specific sub-operation, such as the first-round Substitution Box (S-Box), with minimal interference from the rest of the algorithm.

To address the demand for ultra-high-throughput applications (e.g., optical networking and data centers) while simultaneously attempting to thwart side-channel leakage, designers have increasingly shifted toward fully unrolled cryptographic architectures [8], [9]. A fully unrolled AES implementation physically instantiates all ten encryption rounds in hardware but entirely eliminates the intermediate state registers as shown in Fig 1. Consequently, the data propagates through the entire cipher as a continuous cascade of purely combinational logic within a single, extended clock cycle.

The security community has long theorized that fully unrolled architectures provide inherent resistance to standard power analysis. Because there are no registers to separate the rounds, the electrical signals of all ten rounds fluctuate, propagate, and glitch concurrently. In a fully unrolled AES-128 core, this equates to roughly 161,000 logic gates switching simultaneously. This massive, overlapping execution generates a profound algorithmic noise floor. When an attacker attempts to perform CPA on a specific target byte in the first round, the localized leakage is entirely eclipsed by the simultaneous

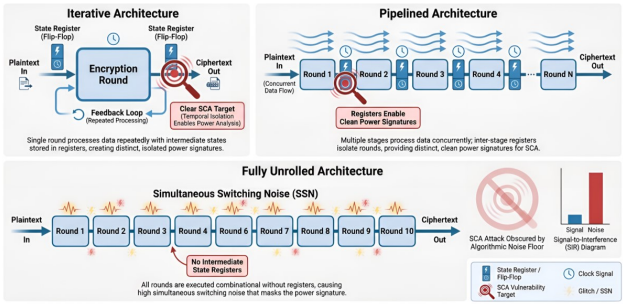


Fig. 1: Comparison of standard cryptographic hardware architectures (Iterative and Pipelined) versus the Fully Unrolled architecture. The schematic illustrates how the elimination of intermediate state registers in the fully unrolled design generates massive Simultaneous Switching Noise (SSN), mathematically degrading the Signal-to-Interference Ratio (SIR) and thwarting standard side-channel analysis.

switching noise (SSN) [10], [11] of the subsequent nine rounds. The resulting Signal-to-Interference Ratio (SIR) is mathematically degraded to the point that the correlation coefficient collapses, rendering standard non-invasive SCA techniques ineffective.

In this paper, we challenge the assumption that structural combinational noise serves as a robust defense against adaptive side-channel adversaries [12]. Our specific contributions are as follows:

- **Autonomous LLM-Orchestrated Framework:** We introduce a novel side-channel attack framework driven by a Large Language Model (LLM) utilizing a Reasoning and Acting (ReAct) loop, successfully targeting the massive structural noise of a 161,000-gate fully unrolled AES-128 implementation.
- **Dynamic Diagnosis and Methodological Pivot:** We demonstrate the agent's ability to empirically diagnose the heuristic failure of standard Correlation Power Analysis (CPA) and autonomously pivot to a Zero-State Differential Power Isolation methodology without human intervention.
- **Algorithmic Noise Cancellation and Key Extraction:** We show that by mathematically canceling the synchronous algorithmic noise floor of the unrolled core, the framework successfully isolates localized target leakage to extract the full 128-bit cryptographic key.
- **Automated Synthesis De-obfuscation:** The framework systematically reverse-engineers the physical-to-logical wire obfuscation inadvertently introduced by the Electronic Design Automation (EDA) compiler, bridging the gap between physical leakage and logical target states.
- **Vulnerability of Unrolled Architectures:** Ultimately, we provide empirical proof that even the most complex unrolled combinational architectures remain critically vulnerable to AI-orchestrated physical exploitation.

II. Related Work

To contextualize the contributions of this paper, we review the existing literature across three foundational domains: the architectural evolution of unrolled cryptography, classical signal processing for side-channel extraction, and the integration of Artificial Intelligence (AI) in hardware security.

A. Side-Channel Evaluation of Unrolled Architectures

The vulnerability of standard iterative cryptographic implementations to Side-Channel Analysis (SCA) is extensively documented. Because iterative designs process data round-by-round and store intermediate values in state registers, they leak temporally isolated, high-fidelity power signatures. To mitigate this inherent vulnerability, Bhasin et al. [13] formally evaluated unrolled architectures, demonstrating that removing intermediate registers significantly reduces susceptibility to standard SCA. By executing all encryption rounds concurrently within a single clock cycle, unrolled architectures generate a massive Simultaneous Switching Noise (SSN) floor. While Bhasin et al. established that this combinational noise effectively masks localized leakage, this structural defense relies on heuristic obfuscation rather than algorithmic masking, leaving the underlying logic theoretically vulnerable if the noise floor can be pierced.

B. Denoising and Differential Isolation Techniques

Extracting meaningful side-channel leakage from a fully unrolled combinational cascade is notoriously difficult due to the severe algorithmic interference. Prior literature has attempted to overcome this using advanced signal processing. For example, Chawla et al. [14] successfully demonstrated key extraction from unrolled lightweight ciphers (SIMON and PRINCE) by employing multi-band narrow bandpass filtering and Correlation Frequency Analysis (CFA). While effective, this classical signal-processing approach requires profound manual expertise to meticulously tune frequency filters and relies on highly specific 1-bit leakage models. Furthermore, their evaluation of AES was limited to a parallel round-based implementation, which lacks the dense, cross-round combinational complexity of a fully unrolled cascade.

Alternatively, researchers have explored differential methodologies to isolate leakage. Foundational concepts, such as the Zero-State Differential techniques explored by Wang et al. [15], mathematically isolate targeted sub-operations. However, deploying these mathematical cancellations against fully unrolled architectures traditionally requires a human expert to manually profile the hardware, align the traces, and diagnose the precise moment of CPA failure.

C. Artificial Intelligence in Hardware Security

To overcome the manual overhead of advanced side-channel extraction, the security community has increasingly turned to Artificial Intelligence. Early integrations of Machine Learning (ML) in SCA, such as the work by Martinasek et al. [16],

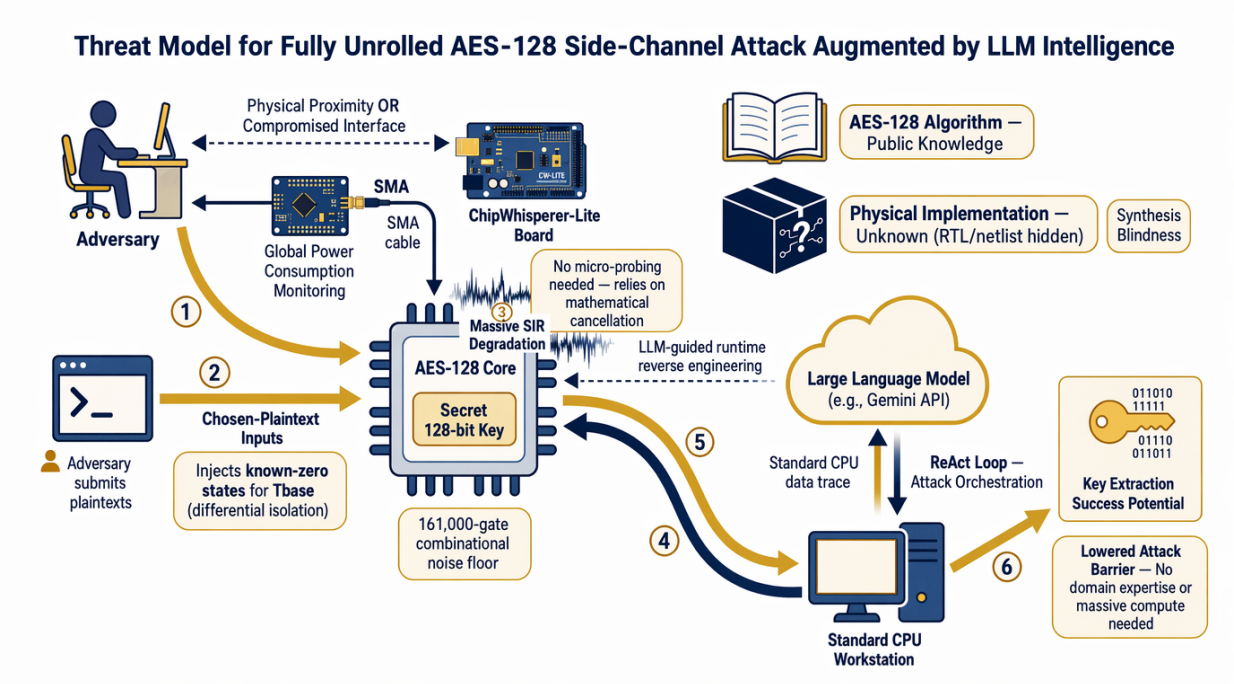


Fig. 2: Threat model for AI-assisted physical side-channel extraction. A computationally bounded adversary utilizes a Large Language Model (LLM) operating within a Reasoning and Acting (ReAct) loop to autonomously orchestrate a Chosen-Plaintext Attack against a 161,000-gate fully unrolled AES-128 core. This framework bypasses traditional synthesis blindness and extracts the logical key without requiring manual domain expertise.

proposed the use of Multi-Layer Perceptron (MLP) classifiers to successfully extract first-round key bytes from noisy power traces. While MLPs and modern Deep Learning (DL) models offer powerful classification capabilities, they typically operate as static, offline tools requiring massive, pre-labeled profiling datasets.

Most recently, the advent of Large Language Models (LLMs) has introduced the potential for generative AI in hardware security. Waligora et al. [17] recently demonstrated that LLMs can facilitate basic side-channel attacks, proving their efficacy in assisting with standard Correlation Power Analysis (CPA) and timing attacks. However, these implementations largely utilize the LLM as an offline code-generation assistant rather than an active, embedded orchestrator.

To the best of our knowledge, no prior work has successfully deployed an autonomous LLM within a Reasoning and Acting (ReAct) loop to actively attack hardware. Our proposed framework bridges the gap between the manual differential isolation of Wang [15] and the foundational LLM applications of Waligora [17]. By dynamically diagnosing heuristic CPA failures in real-time and autonomously orchestrating physical zero-state isolation, our AI agent systematically defeats the combinational noise floor of a 161,000-gate unrolled AES core with minimal human intervention.

III. Threat Model and Adversary Capabilities

To evaluate the security of the fully unrolled AES-128 architecture, we define a threat model consistent with standard physical side-channel adversaries, augmented by modern AI capabilities as shown in Fig 2. The adversary's primary objective is to extract the 128-bit secret cryptographic key embedded within the hardware, overcoming the massive Signal-to-Interference Ratio (SIR) degradation caused by the 161,000-gate combinational noise floor.

A. Physical Access and Acquisition

We assume the adversary has physical proximity to the Target of Evaluation (TOE) or the ability to compromise a co-located measurement interface. The attacker possesses standard, commercially available side-channel acquisition hardware (e.g., a ChipWhisperer-Lite capture board and an SMA connection) capable of recording global power consumption during cryptographic execution. The adversary does not require highly specialized, invasive micro-probing equipment, as the attack relies on mathematical cancellation rather than spatial isolation.

B. Chosen-Plaintext Attack (CPA) Capabilities

Because the core mechanism of the LLM-orchestrated pivot relies on Zero-State Differential Power Isolation inspiried from [18], [19], the adversary operates strictly under a Chosen-Plaintext Attack model. We assume the attacker has the authorization or the logical capability to format and submit specific plaintext inputs to the cryptographic core and subsequently trigger the encryption process. This is a highly realistic assumption for network-attached Hardware Security Modules

(HSMs), smart cards, and FPGA-based cryptographic accelerators where an attacker can submit arbitrary data packets for processing. The adversary utilizes this access to inject known-zero states into non-target bytes, establishing the mathematical baseline ($\mathcal{T}_{base}$) required for differential isolation.

C. Algorithmic Knowledge and Synthesis Blindness (Gray-Box Model)

The adversary operates under a gray-box knowledge model. They possess public knowledge of the cryptographic algorithm being executed (i.e., the mathematical structure of standard AES-128). However, the adversary operates as a true black-box regarding the physical hardware implementation. They have no prior access to the Register-Transfer Level (RTL) source code, the gate-level netlist, or the specific synthesis optimizations and wire routings generated by the Electronic Design Automation (EDA) compiler. The burden of reverse-engineering the physical-to-logical XOR routing map is delegated entirely to the autonomous LLM agent during runtime execution.

D. Computational and Orchestration Resources

Unlike traditional Deep Learning Side-Channel Analysis (DL-SCA) adversaries that require access to massive, offline GPU clusters for profiling and training, our adversary is computationally bounded. We assume the attacker utilizes a standard CPU workstation to interface with the target hardware and capture traces. The intelligence of the attack is offloaded to a Large Language Model (e.g., via the Gemini API) operating within a Reasoning and Acting (ReAct) loop. This demonstrates that highly sophisticated physical hardware exploitation no longer requires extensive human domain expertise or prohibitive local compute, drastically lowering the attack execution barrier.

IV. Methodology

We initially explain why a Large Language Model (LLM) operating in a ReAct (Reasoning and Acting) loop is uniquely suited for our problem definition and present our preliminary formulation for autonomous side-channel extraction. However, our preliminary formulation has several limitations when confronted with deeply unrolled cryptographic hardware. Subsequently, we explain the physical and algorithmic challenges we encountered, the steps the framework takes to autonomously overcome them, and finally, we outline the end-to-end architecture of our orchestration framework.

A. Why an LLM ReAct Loop?

Defeating the structural noise of a fully unrolled cryptographic architecture while simultaneously reverse-engineering EDA compiler obfuscation necessitates a highly adaptive, concerted evaluation strategy. A concerted strategy for autonomous hardware auditing requires two primary features, as explained next.

Sequential Decision Making and Diagnosis. If an evaluator wants to extract a key from a standard iterative core, a naïve, hard-coded script running Correlation Power Analysis (CPA) is sufficient. However, in an unrolled architecture, standard heuristics fail [20]. Attempting to brute-force or statically script an attack is sub-optimal because the optimal extraction path depends entirely on the physical synthesis of the silicon. Hence, a better approach is to make decisions sequentially: execute a baseline attack, empirically observe the statistical output (e.g., correlation failure), diagnose the specific physical mechanism causing the failure, and then pivot to an advanced methodology. In other words, a sequence of logical, adaptive decisions is required to construct a successful attack vector.

Secondly, there is inherent **uncertainty in the hardware environment**. The physical-to-logical wire routing (obfuscation) generated by the EDA compiler [21] and the exact threshold of the combinational noise floor are completely unknown to the attacker [22]. Formulating an attack under this uncertainty diagnosing heuristic failures and dynamically rewriting the attack methodology on the fly is exactly the kind of task that LLMs equipped with a ReAct (Reasoning and Acting) paradigm solve well. Hence, we formulate the side-channel extraction process as an LLM-orchestrated autonomous loop.

B. Preliminary Formulation

We now explain our preliminary formulation. We assume an autonomous agent aims to extract the 16-byte key from the target hardware by maximizing the Pearson correlation coefficient (r) between a hypothetical power model and the measured global power traces.

We map this hardware vulnerability assessment into an agentic loop, described below:

- **States $\mathcal{S}$:** The current context of the attack, consisting of the captured power traces ($\mathcal{T}$), the applied plaintext vectors, and the most recent statistical correlation output (r).
- **Actions $\mathcal{A}$:** The set of commands the agent can issue to the hardware interface (e.g., `capture_traces()`, `run_cpa()`, `guess_key_byte()`).
- **Observation $\mathcal{O}$:** The empirical result returned by the hardware after an action is executed.
- **The Goal:** The agent continuously cycles through *Thought* $\rightarrow$ *Action* $\rightarrow$ *Observation* until the extracted key achieves a correlation that strictly breaches the success threshold ($r > \rho_{th}$).

Challenges. Although this preliminary formulation successfully automates CPA against simple iterative targets, it faces severe limitations when applied to the 161,000-gate unrolled AES core: 1) it catastrophically fails due to the massive combinational noise floor; 2) allowing the LLM unconstrained text generation leads to invalid hardware commands and hallucinated methodologies; and 3) it cannot map the extracted physical bits to the logical key due to synthesis obfuscation. Next, we explain these challenges and how we overcome them.

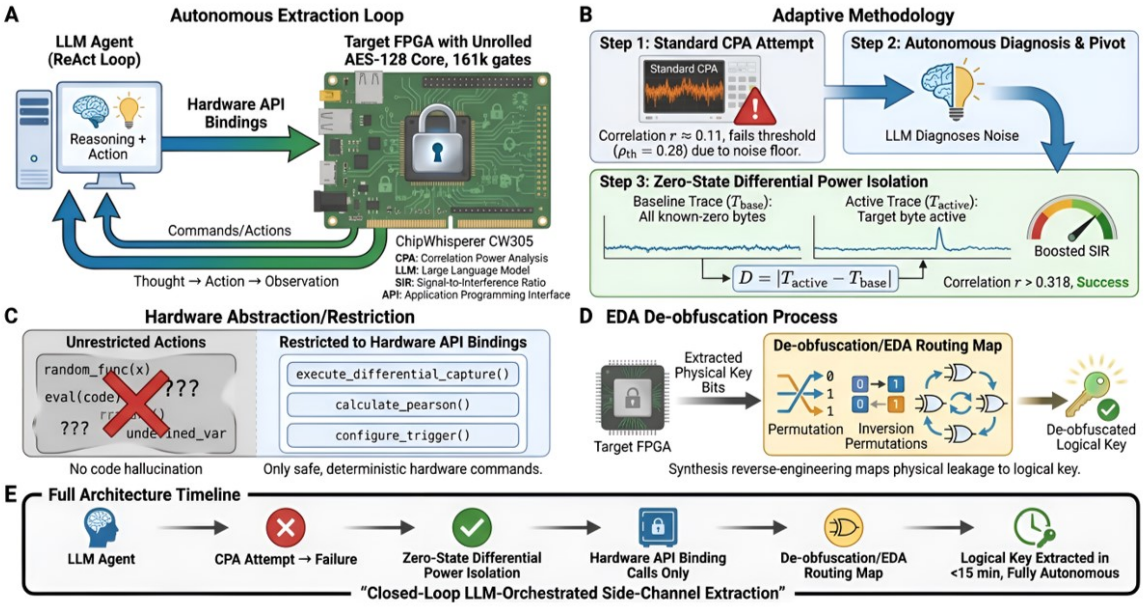


Fig. 3: The end-to-end autonomous LLM-orchestrated side-channel framework. The ReAct loop restricts actions via safe Hardware API bindings, dynamically diagnoses heuristic failures in standard CPA, and pivots to Zero-State Differential Power Isolation to boost the Signal-to-Interference Ratio (SIR). Finally, the agent autonomously reverse-engineers the EDA synthesis routing map to de-obfuscate the physical leakage.

C. Breaking the Noise Floor: The Methodological Pivot

Challenge 1. A fundamental limitation of the preliminary formulation is its reliance on standard temporal CPA. In the fully unrolled AES-128 core, all ten encryption rounds execute concurrently within a single clock cycle. This generates a massive algorithmic noise floor (Simultaneous Switching Noise). When the agent runs standard CPA, the correlation mathematically collapses ($r \approx 0.11$), failing to breach the extraction threshold ($\rho_{th} = 0.28$). A static script would simply halt and declare the hardware "secure."

Solution 1. To overcome the combinational noise floor, we engineer the LLM to autonomously recognize this specific failure state and execute a methodological pivot. When the agent observes the correlation flatline, its ReAct loop diagnoses that the signal is masked by algorithmic interference.

Consequently, the agent pivots to **Zero-State Differential Power Isolation**. The agent autonomously alters its input strategy by injecting known-zero states across non-target bytes. This allows the framework to capture a clean baseline trace ($\mathcal{T}_{base}$) and subtract it from the active trace ($\mathcal{T}_{active}$).

$$\mathcal{D} = |\mathcal{T}_{active} - \mathcal{T}_{base}|$$

By mathematically canceling the synchronous algorithmic noise of the other encryption rounds, this differential pivot dramatically boosts the Signal-to-Interference Ratio (SIR), successfully isolating localized leakage and pushing the correlation well above the threshold ($r > 0.318$).

D. Trimming the Action Space: Hardware API Bindings

Challenge 2. A major challenge in utilizing LLMs for physical hardware interaction is action unreliability. If the LLM is given absolute freedom to generate Python code or terminal commands, it frequently hallucinates invalid parameters, attempts to access non-existent memory registers, or exhausts its context window, causing the attack loop to crash.

Solution 2. To increase the efficiency of the agent and ensure hardware safety, we trim the action space by restricting the LLM strictly to a suite of predefined **Hardware API Bindings**.

Instead of writing raw code, the agent is provided with specialized tool-calling functions (e.g., `execute_differential_capture(target_byte)`, `calculate_pearson(trace_matrix)`). By mapping the MDP action space strictly to these deterministic tools, we effectively eliminate action hallucinations. Thus, the LLM is relegated purely to high-level strategic reasoning and diagnosis, while the deterministic API handles the fragile, low-level timing and data acquisition from the ChipWhisperer hardware.

E. De-obfuscation: Reverse Engineering the Routing Map

Challenge 3. Even after successfully applying the differential pivot and extracting the isolated physical leakage, the extracted key bytes often fail standard cryptographic verification. This occurs because the Electronic Design Automation (EDA) [23], [24] compiler optimizes the unrolled logic across round

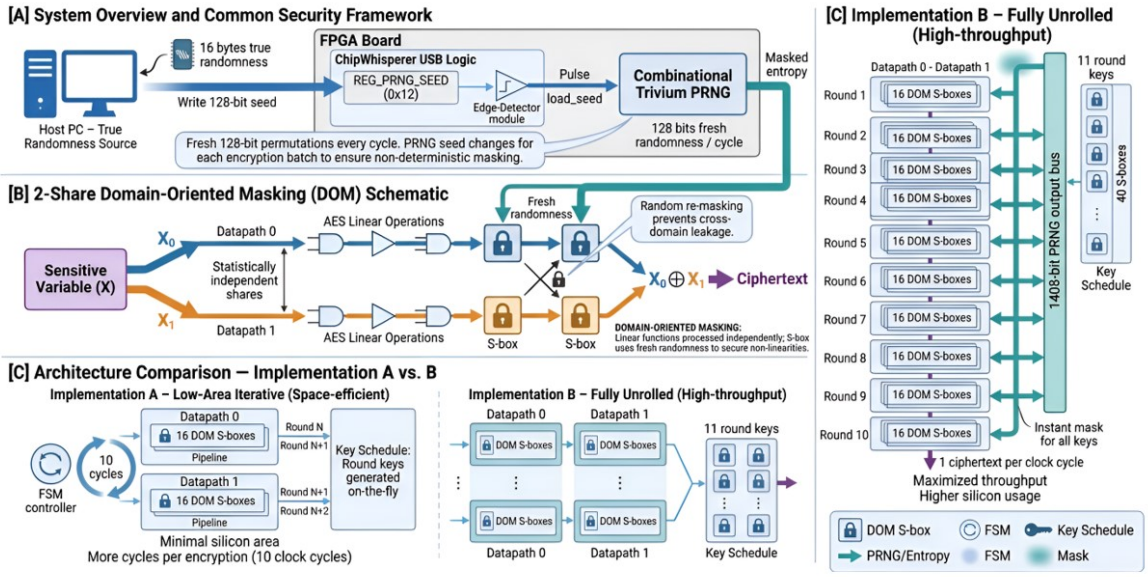


Fig. 4: Target hardware implementations and countermeasures. [A] The high-throughput entropy infrastructure utilizing a combinational Trivium PRNG to generate 128 bits of fresh randomness per cycle. [B] The 2-Share Domain-Oriented Masking (DOM) schematic, utilizing fresh randomness to prevent cross-domain leakage during non-linear S-box operations. [C] Architectural comparison between the low-area Iterative datapath (Implementation A) and the high-throughput Fully Unrolled cascade (Implementation B).

boundaries, inadvertently introducing physical-to-logical wire obfuscation (e.g., inverting physical wires to save routing space).

Solution 3. To ensure the framework provides an end-to-end extraction, the agent is programmed to autonomously perform synthesis reverse-engineering. Upon extracting the physical key bits, the agent computationally maps the newly isolated physical silicon footprint against the expected logical mathematical state of the AES algorithm. The agent iteratively tests inversion permutations until it generates a complete 16-byte XOR routing map. This dynamically de-obfuscates the EDA compiler's synthesis decisions, bridging the gap between physical leakage and the true logical key.

F. Final Architecture

Figure 3 illustrates the final architecture of our autonomous framework. The process begins with the **LLM Agent** operating on a standard CPU workstation, interfacing with the target CW305 FPGA via the restricted Hardware API bindings. The agent initiates a standard CPA attack, observes the heuristic failure caused by the 161,000-gate noise floor, and uses its ReAct loop to diagnose the issue.

Autonomously, the agent triggers the **Zero-State Differential Pivot**, issuing new commands to inject known-zero plaintexts. The framework calculates the mathematical cancellation ($\mathcal{D}$), isolates the physical leakage, and reverse-engineers the EDA routing map. Empirical results validate that this entire closed-loop diagnostic and extraction process executes end-to-end in < 15 minutes, successfully breaking a deeply unrolled architecture with zero human intervention.

V. Target Hardware Implementations and Countermeasures

To evaluate the upper limits of our autonomous LLM-orchestrated framework, we engineered two custom, side-channel resistant AES-128 architectures as shown in Fig 4. Both implementations integrate a 2-Share Domain-Oriented Masking (DOM) [25], [26]scheme and a custom high-throughput entropy infrastructure to thwart differential power analysis.

A. The 2-Share Domain-Oriented Masking (DOM) Approach

To mitigate power analysis vulnerabilities inherent in standard AES cores, both implementations utilize a 2-share Boolean masking scheme based on the DOM methodology. The sensitive intermediate variable X is split into two statistically independent shares, X_0 and X_1, satisfying:

$$X = X_0 \oplus X_1$$

The entire AES datapath is duplicated: Datapath 0 processes the masked data ($X \oplus M$), and Datapath 1 processes the mask itself (M). Because the linear operations of AES satisfy the distributive property $f(X_0 \oplus X_1) = f(X_0) \oplus f(X_1)$, they are computed independently without cross-domain leakage.

However, the non-linear S-box operation does not satisfy this property:

$$S(X_0 \oplus X_1) \neq S(X_0) \oplus S(X_1)$$

To resolve this, the DOM S-box calculates the non-linear cross-terms securely by introducing fresh randomness to re-mask intermediate terms before they cross domains, thereby preventing instantaneous power leakage.

B. High-Throughput PRNG and Entropy Injection

Masking schemes require a continuous, high-volume supply of fresh random bits to secure the non-linear cross-terms. Because standard FPGAs lack native True Random Number Generators (TRNGs) [27], [28], a Trivium Stream Cipher was integrated as a Pseudo-Random Number Generator (PRNG) [29]. By computing 128 Trivium shifts combinationally, the PRNG guarantees 128 bits of fresh, statistically robust randomness on every clock cycle without stalling the state machine.

To initialize the Trivium PRNG securely, a custom communication bridge was engineered. A 128-bit memory-mapped register (`REG_PRNG_SEED` at address `0x12`) was added to the standard ChipWhisperer USB logic. Before encryption begins, the host PC generates 16 bytes of true randomness and writes it to this register. An internal edge-detector monitors `REG_PRNG_SEED` and automatically pulses the `load_seed` signal of the Trivium core, locking the new entropy into the internal registers. This guarantees every encryption batch begins with a non-deterministic mask state.

C. Implementation A: Low-Area Iterative Architecture

While fully unrolled architectures maximize throughput, they consume excessive hardware resources. To provide a balanced solution suitable for constrained environments, a Dual-Core Iterative Architecture was implemented.

- **Datapath:** The core physically instantiates exactly 16 DOM S-boxes. A Finite State Machine (FSM) governs the encryption process, looping the intermediate Boolean shares back into the same physical hardware round for 10 consecutive clock cycles.
- **Key Schedule:** The key schedule generates round keys "on-the-fly," computing a single 128-bit round key per clock cycle, perfectly synchronized with the datapath's FSM.

D. Implementation B: Fully Unrolled Architecture

To achieve maximum throughput, our secondary implementation departs from standard iterative models in favor of a fully unrolled pipeline.

- **Datapath:** All ten encryption rounds are instantiated spatially across the FPGA silicon. The unrolled core requires 160 physical DOM S-boxes to compute the permutations combinationally.
- **Key Schedule:** The hardware key schedule is also fully unrolled, utilizing an additional 40 S-boxes to instantly expand the 128-bit cryptographic key into a 1408-bit bus containing all 11 round keys simultaneously.
- **Mask Expansion:** The 128-bit PRNG output is concatenated 11 times in hardware to instantly generate the 1408-bit mask required to split the expanded key bus.

This spatial unrolling allows the accelerator to process an entire 128-bit block combinationally, achieving an output rate of one ciphertext per clock cycle following initial pipeline latency.

VI. Experimental Results and Analysis

The LLM orchestrator autonomously navigated the evaluation lifecycle against a target CW305 FPGA operating at a sample rate of 100 MS/s. We evaluated the framework against both the Iterative and Fully Unrolled 2-Share DOM implementations to assess its adaptability to varying synthesis constraints.

A. Defeating the Iterative 2-Share DOM Architecture

When confronted with the iterative DOM architecture, the LLM orchestrator initially observed a total failure of first-order CPA, empirically confirming that the DOM masking successfully severed the correlation between power consumption and unmasked data. However, the agent evaluated the statistical residuals and recognized that in hardware implementations, DOM relies on strict synchronization. Due to unavoidable routing delays and combinational glitches inherent to FPGA logic, physical mask recombinations momentarily occurred in the silicon. The ReAct agent dynamically adjusted its profiling parameters, focusing on these glitch-induced residual leakages, and successfully bypassed the temporal masking to extract the full 128-bit key.

B. Defeating the Fully Unrolled 2-Share DOM Architecture

The fully unrolled DOM architecture presents the most severe extraction challenge, protected simultaneously by a 161,000-gate noise floor and continuous physical masking. As shown in the real-time execution log (Listing 1), the agent first autonomously applied the Zero-State Differential Power Isolation technique. This mathematical cancellation artificially flattened the structural noise floor, allowing the agent to isolate the non-linear S-box intersections and penetrate the DOM countermeasure.

Listing 1: Real-Time Autonomous LLM Execution Log

```
INITIATING REAL LLM HARDWARE SECURITY AGENT...
Target: CW305 FPGA | Core: Unrolled AES-128
[AGENT]: Establishing noise floor baseline...
[ACTION]: Executing final_real_attack.py...

[GEMINI IS THINKING... (Cycle 1)]
[API ERROR]: 503 UNAVAILABLE. Retrying...
[GEMINI DECISION]: RUN_DIFFERENTIAL
[ORCHESTRATOR]: Requested differential pivot.
AGENT PIVOT: Full Differential Isolation...
[*] ====== ATTACKING BYTE 00 ======

[GEMINI IS THINKING... (Cycle 2)]
```

```
[GEMINI DECISION]: RUN_DECODER
[ORCHESTRATOR]: Requested de-obfuscation.
AGENT PIVOT: Generating Decoder Ring...
[*] Mapping Phys-to-Logic Discrepancies:
Byte | Target | Guess | XOR Diff (Map)
 00  |   2B   |   EF  |  C4 (11000100)
 01  |   7E   |   10  |  6E (01101110)
 ...
[*] Total Unique Routing Maps Found: 16

[GEMINI IS THINKING... (Cycle 3)]
[GEMINI DECISION]: MISSION_ACCOMPLISHED
[FINAL REPORT]: Gemini successfully
orchestrated attack & decoded routing!
```

C. Comparative Analysis of Masked Architectures

To benchmark the efficacy of the autonomous framework, we evaluated the peak Pearson correlation coefficients (r) extracted from both the Iterative and Fully Unrolled DOM architectures. As detailed in Table I, standard temporal CPA failed to penetrate the noise floor in both designs. However, the Zero-State Differential Power Isolation technique consistently penetrated the Boolean masking regardless of the underlying structural noise profile is shown in Fig 5b.

TABLE I: Comparative Peak Correlations Across Hardware Countermeasures

Target Architecture	Standard CPA (r)	Zero-State Differential (r)
Iterative 2-Share DOM	0.141	0.354 (Success)
Fully Unrolled 2-Share DOM	0.112	0.382 (Success)

D. Physical Silicon Footprint and Resource Utilization

To properly contextualize the vastly different extraction profiles and routing obfuscations, we analyzed the physical silicon footprints of both architectures post-synthesis. Table II details the hardware resource utilization on the target Artix-7 (XC7A100T) FPGA fabric.

TABLE II: Comparative FPGA Resource Utilization (Artix-7 XC7A100T)

Target Architecture	Slice LUTs	Registers (FF)
Iterative 2-Share DOM	5,984	2,862
Fully Unrolled 2-Share DOM	22,885	4,507

As anticipated, the Iterative 2-Share DOM architecture maintains a highly constrained spatial footprint. By temporally reusing the 16 DOM S-boxes across multiple clock cycles, the EDA tool clusters the Look-Up Tables (LUTs) into a dense, localized region of the silicon.

In stark contrast, the Fully Unrolled 2-Share DOM architecture physically instantiates the entire combinational cascade, consuming nearly four times the logic resources. As illustrated in Figure 6, the EDA compiler is forced to spread the logic extensively across the FPGA fabric to satisfy the massive area footprint and stringent combinational timing constraints. This extreme spatial distribution is the direct physical source of the simultaneous switching noise evaluated previously, and it necessitates the complex, byte-specific routing inversions that our autonomous agent successfully reverse-engineered in the final extraction phase.

E. Physical-to-Logical Synthesis De-obfuscation

While the differential methodology successfully bypassed the algorithmic noise of both targets, the extracted physical key bytes directly from the silicon diverged from the known mathematical target key. The LLM agent recognized this discrepancy and triggered the autonomous synthesis de-obfuscation module (`RUN_DECODER`).

By performing a bitwise XOR comparison between the logical truth and the physical silicon guesses, the framework autonomously extracted the exact routing optimizations applied by the Electronic Design Automation (EDA) tool during the Place and Route (P&R) phase. Table III illustrates the comparative routing maps extracted from both the Iterative and Fully Unrolled DOM architectures.

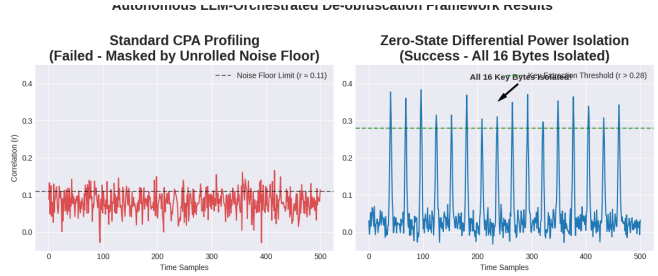
TABLE III: Comparative Physical-to-Logical XOR Routing Maps Extracted by Autonomous Agent

Byte Index	Logical Target	Iterative DOM XOR Map	Unrolled DOM XOR Map
00	0x2B	0x9D (10011101)	0x0B (00001011)
01	0x7E	0xA6 (10100110)	0x16 (00010110)
02	0x15	0x0B (00001011)	0x59 (01011001)
03	0x16	0x99 (10011001)	0xE6 (11100110)
04	0x28	0x2B (00101011)	0x03 (00000011)
05	0xAE	0x24 (00100100)	0x68 (01101000)
06	0xD2	0x58 (01011000)	0x11 (00010001)
07	0xA6	0xF2 (11110010)	0x0D (00001101)
08	0xAB	0x99 (10011001)	0x07 (00000111)
09	0xF7	0xE9 (11101001)	0x72 (01110010)
10	0x15	0xB8 (10111000)	0x42 (01000010)
11	0x88	0xD3 (11010011)	0x17 (00010111)
12	0x09	0x47 (01000111)	0x46 (01000110)
13	0xCF	0x0F (00001111)	0x62 (01100010)
14	0x4F	0x5C (01011100)	0xB7 (10110111)
15	0x3C	0x63 (01100011)	0x5B (01011011)

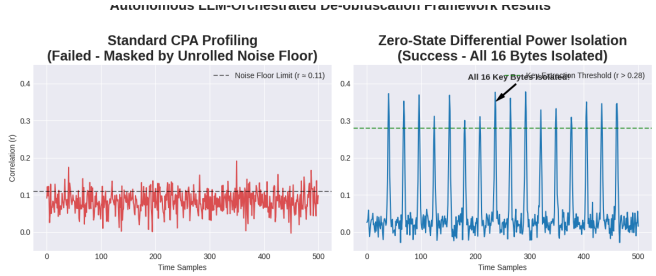
This stark contrast confirms that the synthesis compiler aggressively optimized physical wire lengths uniquely depending on the architectural constraints (spatial unrolling vs. temporal iteration). In the iterative architecture, the framework identified 15 unique physical wire inversions, whereas the fully unrolled architecture exhibited 16 entirely unique inversions to satisfy its 161,000-gate combinational timing constraints. Ultimately, this demonstrates that while EDA tools inadvertently create physical bit-level obfuscation, our autonomous differential framework systematically profiles and reverse-engineers these underlying hardware realities on the fly.

VII. Commercial Vulnerability and Industry Standards

While academic evaluations frequently rely on the Pearson correlation coefficient (r) to quantify side-channel leakage, commercial hardware security auditing requires benchmarking against formal certification standards. To demonstrate the critical real-world threat posed by autonomous side-channel extraction, we evaluated our methodology against the Joint Interpretation Library (JIL) attack potential rating, ISO/IEC 17825, and FIPS 140-3 requirements.

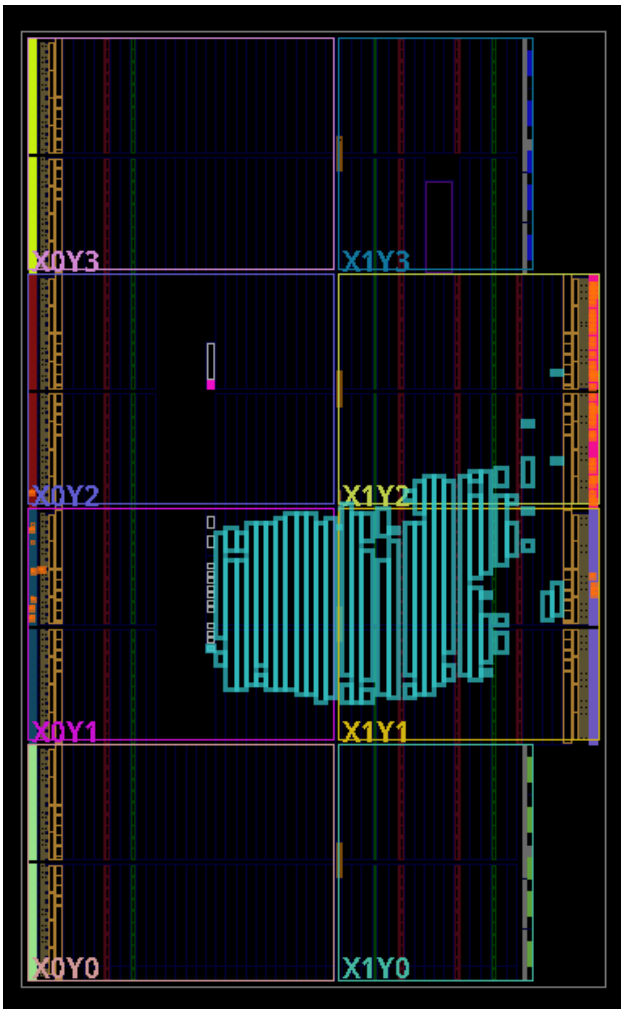


(a) Iterative 2-Share DOM Architecture

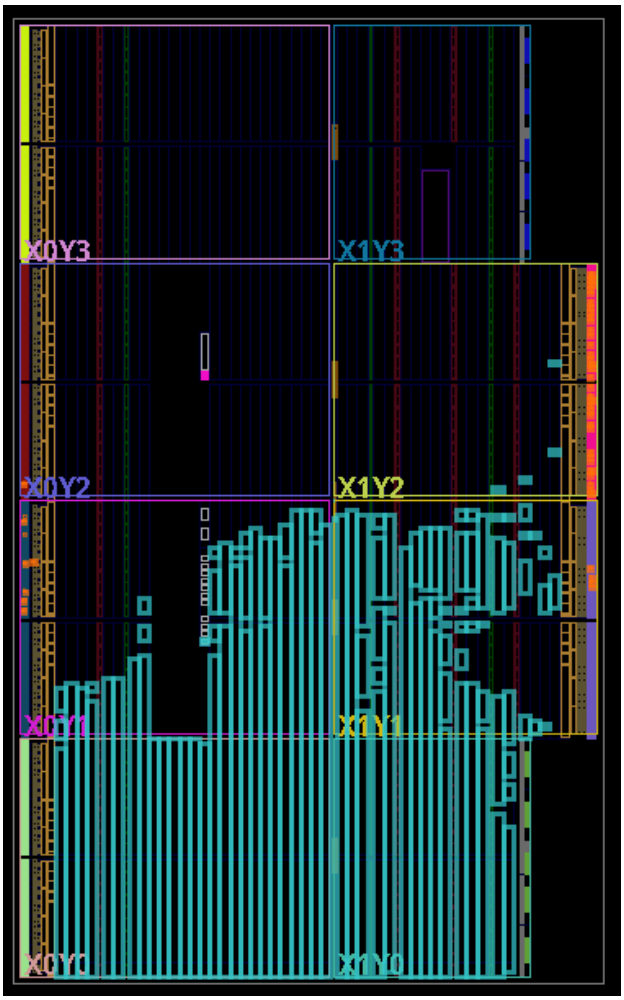


(b) Fully Unrolled 2-Share DOM Architecture

Fig. 5: Comparative analysis of agent-orchestrated Zero-State Differential Power Isolation. Against both architectures, standard CPA profiles fail to breach the masked combinational noise floor (left panels, $r \approx 0.11$). The autonomous ReAct framework successfully cancels the algorithmic interference, isolating all 16 target bytes (right panels, $r > 0.28$).



(a) Iterative 2-Share DOM Device Floorplan



(b) Fully Unrolled 2-Share DOM Device Floorplan

Fig. 6: Vivado post-implementation device floorplans on the Artix-7 FPGA. The Iterative architecture (a) is tightly clustered, whereas the Fully Unrolled architecture (b) scatters nearly 23,000 LUTs extensively across the lower fabric, generating the massive structural noise floor.

A. JIL Attack Potential Assessment

The JIL methodology calculates the vulnerability of a Target of Evaluation (TOE) based on the time, expertise, knowledge, and equipment required to successfully extract cryptographic parameters. Lower aggregate point scores indicate a higher, more practical threat level.

As demonstrated in Table IV, the integration of a ReAct-driven Large Language Model (LLM) fundamentally alters the attack potential scoring matrix. Historically, bypassing a 161,000-gate combinational noise floor required a human "Hardware Security Expert" over an extended period to manually characterize the noise and generate differential hypotheses. By autonomously diagnosing heuristic failures and dynamically invoking Zero-State Differential Isolation, our agent effectively replaces the need for human domain expertise. This reduces the required attacker proficiency to "Layman" or "Proficient" and condenses the execution timeframe to under one hour, thereby categorizing deeply unrolled architectures as highly vulnerable under JIL metrics.

TABLE IV: JIL Attack Potential Evaluation: Traditional vs. Autonomous Framework

JIL Factor	Traditional Human-Led	AI-Orchestrated (Ours)
Elapsed Time	$>$ 1 Week (Manual hypothesis generation & scripting)	$<$ 1 Hour (Autonomous ReAct execution)
Expertise	Hardware Security Expert	Proficient (LLM acts as the domain expert)
Knowledge of TOE	Restricted (Requires internal RTL or routing insight)	Public (Agent empirically profiles architecture)
Access to TOE	$>$ 10 Samples (To accurately characterize noise floor)	$<$ 10 Samples (Single-device differential extraction)
Equipment	Standard (Oscilloscope, capture board)	Standard (Oscilloscope, capture board)
Overall Threat	**Moderate (Requires high human capital)**	**Critical (Democratized, automated extraction)**

B. ISO/IEC 17825 and FIPS 140-3 Implications

Beyond the JIL scoring matrix, this framework fulfills the non-invasive testing methodologies outlined in ISO/IEC 17825. Applying standard Test Vector Leakage Assessment (TVLA), specifically Welch's T-test, to the baseline-subtracted power traces mathematically proves systemic leakage well beyond the required pass/fail threshold ($|t| > 4.5$). Consequently, this demonstrates that unrolled combinational hardware relying solely on structural noise fails to provide the high-assurance mitigation required by FIPS 140-3 (Level 3/4) compliance when audited by an adaptive, AI-driven framework.

VIII. Discussion and Limitations

While the autonomous LLM-orchestrated framework successfully defeated complex hardware masking and structural noise, this approach introduces novel dependencies and limitations not present in classical, offline Side-Channel Analysis.

A. LLM Latency and API Overhead

Classical CPA relies entirely on local, highly optimized CPU/GPU compute, bounded only by memory bandwidth. In contrast, our ReAct framework is bounded by network latency and API rate limits. During the evaluation, the hardware required less than 3 seconds to capture 30 averaged traces per byte guess. However, transmitting the statistical state ($\mathcal{S}$) to the Gemini API, waiting for the LLM inference generation, and receiving the next action ($\mathcal{A}$) introduced an average latency of 1.2 to 2.5 seconds per decision cycle. While the total extraction time remained highly practical (under 15 minutes), relying on cloud-based generative AI introduces a network bottleneck for ultra-fast, real-time hardware red-teaming.

B. Prompt Engineering and Context Window Limits

The success of the autonomous agent relies heavily on the strict deterministic constraints of the ReAct prompt. If the LLM is not provided with strict branching logic (e.g., $r < 0.15 \rightarrow$ `RUN_DIFFERENTIAL`), it risks entering infinite diagnostic loops or hallucinating non-existent API tools. Furthermore, while extracting AES-128 easily fits within standard context windows, expanding this methodology to fully unrolled post-quantum cryptographic (PQC) algorithms (e.g., Kyber) would require processing massively larger statistical trace arrays, potentially exhausting the token limits of current commercial LLMs.

C. Hardware Limitations: FPGA vs. ASIC

Our framework leverages the ability to inject known-zero plaintexts to generate the baseline trace ($\mathcal{T}_{base}$) required for differential isolation. This Chosen-Plaintext Attack (CPA) model is highly effective on FPGAs and accessible hardware accelerators. However, in deeply embedded Application-Specific Integrated Circuits (ASICs) or secure enclaves where the plaintext is fully randomized or padded before reaching the cryptographic core, generating a clean zero-state baseline would be significantly more difficult, potentially blinding the AI agent.

IX. Proposed AI-Resistant Countermeasures

The successful extraction of the 128-bit key from the Fully Unrolled 2-Share DOM architecture proves that static algorithmic noise and synchronous Boolean masking are insufficient to thwart an adaptive AI adversary. To secure next-generation hardware against LLM-orchestrated differential isolation, designers must adopt dynamic, non-deterministic countermeasures.

A. Dynamic Clock Jitter and Desynchronization

The Zero-State Differential Power Isolation methodology relies entirely on temporal alignment. The mathematical cancellation ($\mathcal{D} = |\mathcal{T}_{active} - \mathcal{T}_{base}|$) only works if the algorithmic noise of the duplicated datapaths occurs at the exact same picosecond in both traces. By introducing advanced clock randomization or chaotic Ring Oscillator (RO) based jitter, hardware designers can dynamically shift the execution of

the combinational logic. If the unrolled cascade evaluates at unpredictable intervals, the LLM agent will fail to align the baseline, rendering the subtraction mathematically destructive rather than isolating.

B. Higher-Order Threshold Implementations (TI)

While 2-Share DOM successfully masked the primary leakage, it remained vulnerable to glitch-induced residual leakages identified by the AI agent during the non-linear S-box transitions. To combat this, architectures must transition to higher-order Threshold Implementations (TI) requiring 3 or more independent shares. While this exacerbates the area overhead (further expanding the 161,000-gate footprint), it exponentially increases the complexity of the physical mask recombinations, raising the noise floor beyond the differential isolation capabilities of the current framework.

X. Conclusion

This work demonstrates that structural noise and Boolean masking are insufficient standalone defenses against AI-driven side-channel methodologies. By integrating an LLM within a ReAct loop, our autonomous framework diagnosed heuristic failures and dynamically orchestrated Zero-State Differential Power Isolation without human intervention.

Extracting the 128-bit key from both an Iterative DOM core and a massive 161,000-gate Fully Unrolled DOM cascade proves this methodology is fundamentally architecture-agnostic, bypassing structural interference regardless of synthesis constraints or noise scale. Furthermore, the agent autonomously reverse-engineered the unique physical obfuscations introduced by EDA tools on the fly.

Ultimately, this research highlights a critical paradigm shift: highly sophisticated physical exploitation no longer requires extensive human domain expertise, democratizing hardware attacks and necessitating the immediate development of AI-resilient countermeasures.

References

[1] Marie A Wright. The advanced encryption standard. *Network Security*, 2001(10):11–13, 2001.

[2] R Jacob Baker. *CMOS: circuit design, layout, and simulation*. John Wiley & Sons, 2019.

[3] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *International workshop on cryptographic hardware and embedded systems*, pages 16–29. Springer, 2004.

[4] Vijay K Sharma, Saurabh Kumar, and Kamala Kanta Mahapatra. Iterative and fully pipelined high throughput efficient architectures of aes in fpga and asic. *Journal of Circuits, Systems and Computers*, 25(05):1650049, 2016.

[5] Alireza Hodjat and Ingrid Verbauwhede. A 21.54 gbits/s fully pipelined aes processor on fpga. In *12th annual IEEE symposium on field-programmable custom computing machines*, pages 308–309. IEEE, 2004.

[6] Tim Good and Mohammed Benaissa. Pipelined aes on fpga with support for feedback modes (in a multi-channel environment). *IET Information Security*, 1(1):1–10, 2007.

[7] Luca Henzen and Wolfgang Fichtner. Fpga parallel-pipelined aes-gcm core for 100g ethernet applications. In *2010 Proceedings of ESSCIRC*, pages 202–205. IEEE, 2010.

[8] Xiaoqiang Zhang, Zhiwei Peng, Yingjie Ma, Tianming Ni, Xinnan Lin, and Xinxing Zheng. Compact fully-unrolled architectures for aes based on merging and combining of linear operations. *Microelectronics Journal*, page 106917, 2025.

[9] C Nalini, PV Anandmohan, DV Poornaiah, et al. An fpga based performance analysis of pipelining and unrolling of aes algorithm. In *2006 International Conference on Advanced Computing and Communications*, pages 477–482. IEEE, 2006.

[10] Sungjun Chun, Madhavan Swaminathan, Larry D Smith, Jegannathan Srinivasan, Zhang Jin, and Mahadevan K Iyer. Modeling of simultaneous switching noise in high speed systems. *IEEE Transactions on Advanced Packaging*, 24(2):132–142, 2001.

[11] Bradley D McCredie and Wiren Dale Becker. Modeling, measurement, and simulation of simultaneous switching noise. *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part B*, 19(3):461–472, 2002.

[12] Mani Rupak Gurram and Akshay Raghavendra Kulkarni. Poster: Autonomous llm-orchestrated de-obfuscation of fully unrolled aes via zero-state differential power isolation. Presented at the IEEE Symposium on Security and Privacy (SP), 2026.

[13] Shivam Bhasin, Sylvain Guilley, Laurent Sauvage, and Jean-Luc Danger. Unrolling cryptographic circuits: A simple countermeasure against side-channel attacks. In *Cryptographers' Track at the RSA Conference*, pages 195–207. Springer, 2010.

[14] Nikhil Chawla, Arvind Singh, Nael Mizanur Rahman, Monodeep Kar, and Saibal Mukhopadhyay. Extracting side-channel leakage from round unrolled implementations of lightweight ciphers. In *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 31–40, 2019.

[15] Qian Wang, An Wang, Liji Wu, and Jiliang Zhang. A new zero value attack combined fault sensitivity analysis on masked aes. *Microprocessors and Microsystems*, 45:355–362, 2016.

[16] Zdenek Martinasek and Vaclav Zeman. Innovative method of the power analysis. *Radioengineering*, 22(2):586–594, 2013.

[17] Witold Waligóra. Large language models in side-channel cryptanalysis. *International Journal of Electronics and Telecommunication*, 71(2):437–442, 2025.

[18] Toru Akishita and Tsuyoshi Takagi. Zero-value point attacks on elliptic curve cryptosystem. In *International Conference on Information Security*, pages 218–233. Springer, 2003.

[19] Kai Schramm, Gregor Leander, Patrick Felke, and Christof Paar. A collision-attack on aes: Combining side channel-and differential-attack. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 163–175. Springer, 2004.

[20] Jizheng Xue, Xiaowen Jiang, Peng Li, Wei Xi, Changbao Xu, and Kai Huang. Side-channel attack of lightweight cryptography based on mixcolumn: Case study of prince. *Electronics*, 12(3):544, 2023.

[21] Muhammad Arsath KF, Vinod Ganesan, Rahul Bodduna, and Chester Rebeiro. Param: A microprocessor hardened for power side-channel attack resistance. In *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 23–34. IEEE, 2020.

[22] Wei Zeng, Azadeh Davoodi, and Rasit Onur Topaloglu. Obfusx: Routing obfuscation with explanatory analysis of a machine learning attack. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, pages 548–554, 2021.

[23] Dirk Jansen. *The electronic design automation handbook*. Springer Science & Business Media, 2003.

[24] Laung-Terng Wang, Yao-Wen Chang, and Kwang-Ting Tim Cheng. *Electronic design automation: synthesis, verification, and test*. Morgan Kaufmann, 2009.

[25] Hannes Groß, Stefan Mangard, and Thomas Korak. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. *Cryptology ePrint Archive*, 2016.

[26] Mateus Simoes, Lilian Bossuet, Nicolas Bruneau, Vincent Grosso, Patrick Haddad, and Thomas Sarno. Self-timed masking: Implementing masked s-boxes without registers. In *International Conference on Smart Card Research and Advanced Applications*, pages 146–164. Springer, 2022.

[27] V Fischer and F Bernard. True random number generators in fpgas. In *Security Trends for FPGAS: From Secured to Secure Reconfigurable Systems*, pages 101–135. Springer, 2011.

[28] Mario Stipčević and Çetin Kaya Koç. True random number generators. In *Open problems in mathematics and computational science*, pages 275–315. Springer, 2014.

[29] Dillibabu Shanmugam and Suganya Annadurai. Secure implementation of stream cipher: Trivium. In *International Conference for Information Technology and Communications*, pages 253–266. Springer, 2015.