

Attacking and Securing Hybrid Homomorphic Encryption Against Power Analysis

Aikata Aikata, Maciej Czuprynko, Nedžma Mušović, Emira Salkić and Sujoy Sinha Roy

Graz University of Technology, Austria

{aikata,maciej.czuprynko,sujoy.sinharoy}@tugraz.at, {nedzma.musovic,emira.salkic}@student.tugraz.at

Abstract—We present the first power side-channel analysis of a Hybrid Homomorphic Encryption (HHE) tailored symmetric encryption scheme. HHE combines lightweight client-side Symmetric Encryption (SE) with server-side homomorphic evaluation, enabling efficient privacy-preserving computation for the client and minimizing the communication overhead. Recent integer-based HHE designs such as PASTA, MASTA, HERA, and Rubato rely on prime-field arithmetic, but their side-channel security has not been studied. This gap is critical, as modular arithmetic and large key spaces in integer-based schemes introduce new leakage vectors distinct from those in conventional Boolean symmetric ciphers. In this work, we close this gap by presenting the first power side-channel analysis of an HHE-tailored scheme - HERA. Our results demonstrate a successful key recovery from as few as 40 power traces using Correlation Power Analysis.

In addition to showing that such attacks are feasible, we develop the first masking framework for integer-based SE schemes to mitigate them. Our design integrates PINI-secure gadgets with assembly-level countermeasures to address transition leakage, and we validate its effectiveness using the Test Vector Leakage Assessment. Our experiments confirm both the practicality of the attack and the strength of the proposed countermeasures. We also demonstrate that the framework extends to other integer-based HHE schemes, by applying our technique to PASTA. Thus, we provide leakage models, identify relevant attack targets, and define evaluation benchmarks for integer-based HHE-tailored SE schemes, thereby filling a longstanding gap and laying the foundation for side-channel-resilient design in this area.

Index Terms—HHE, FHE, Power Analysis, DPA, CPA, Masking

I. INTRODUCTION

Fully Homomorphic Encryption (FHE) enables performing arbitrary computations on encrypted data, preserving confidentiality in untrusted environments. This powerful paradigm supports a wide range of applications, ranging from secure machine learning to private data outsourcing. However, it remains challenging to deploy it efficiently as it incurs heavy communication and computation overheads, typically $10,000\times$ to $100,000\times$ compared to plaintext operations.

To reduce the communication and client's computation overhead, the Hybrid Homomorphic Encryption (HHE) paradigm was introduced [1], which combines lightweight symmetric encryption (SE) on the client side with FHE on the server side. This significantly lowers the client's workload while retaining the security and flexibility of homomorphic evaluation. Recent schemes like PASTA [2], MASTA [3], and HERA [4] extend

This work was supported in part by CONFIDENTIAL-6G (Grant No: 101096435) and the Austrian FWF grant PAT640202.

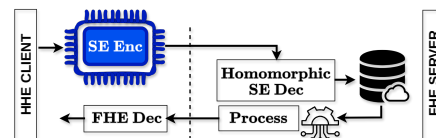


Fig. 1. The HHE process where client data is first protected with symmetric encryption and sent to the server, where a specialized homomorphic decryption step transforms it into a ciphertext compatible with homomorphic computation.

this idea to prime-field arithmetic, enabling more efficient operations for practical workloads. This is due to the fact that these schemes are compatible with efficient FHE schemes over integers [5]–[7], and deliver high performance.

As these schemes move from theory to implementation [8], [9], especially edge-level [8], they face new challenges. Edge devices are resource-constrained, operate in potentially adversarial environments, and are physically accessible, making them especially vulnerable to practical side-channel attacks. Even though the cryptography itself is mathematically sound, the physical realizations of HHE can inadvertently leak information through side-channels, such as power consumption traces.

A. Research Gap

The practical implementations of HHE remain unexplored in the context of power analysis. To the best of our knowledge, no prior work has investigated power side-channel attacks on HHE-tailored schemes, although several studies have examined their vulnerability to fault attacks [10]–[12]. This absence of exploration leaves a critical gap: while the underlying cryptographic constructions offer strong theoretical security, their physical realizations may still leak exploitable information.

Moreover, in the absence of any prior power analysis attacks, there has likewise been no study of countermeasures designed to protect HHE schemes against such threats. As a result, both the offensive and defensive dimensions of side-channel security for HHE remain open problems. Investigating this gap introduces several unique challenges:

■ **High-Dimensional Leakage and Large Key-space.** Unlike lightweight block ciphers (e.g., AES [13], ASCON [14]), where leakage often comes from compact operations like S-boxes, integer-based HHE-tailored schemes (e.g., PASTA, MASTA, HERA) rely on modular arithmetic. This spreads side-channel leakage across many intermediate values and low-level processor operations, making it harder to locate exploitable points.

Keys also consist of many integer coefficients, creating much larger secret and state spaces. As a result, standard side-channel methods that target small subkeys (like 8-bit AES bytes) do not apply naturally. Understanding this leakage is crucial as integer-based designs support integer FHE schemes (e.g., CKKS [5], BGV/FV [6], [7]) and thus efficient HHE [8].

■ **Randomness and Transition Leakage.** Securing integer-based HHE-tailored schemes against side-channel analysis requires repeated masking of multiplications with fresh randomness and frequent mask refreshes, which is problematic on constrained devices where randomness generation and management are costly. Moreover, while formal proofs cover the probing model [15], software implementations face transition leakage when multiple shares reside in the same stack/register. Balancing robustness against transition leakage with practical efficiency thus remains a key challenge for masking.

B. Our Contributions

① **First CPA on HHE.** This work addresses the lack of side-channel analysis and masking techniques for integer-based HHE schemes. We present the first comprehensive CPA (Correlation Power Analysis) evaluation of HERA. We also show that our technique is applicable to all the other HHE-tailored SE schemes defined over integers. Prior fault-based attacks on such schemes [12] rely heavily on nonce reuse, while the only exception [10] introduced a model that bypasses this restriction but still requires knowledge of the function evaluated. In contrast, our CPA approach does not require any of these assumptions and still achieves full key recovery with 40 traces, making it a far more practical and general threat. We demonstrate the attack using the Chipwhisperer CW308 UFO board with STM32F303 Target, and the capture is done using the CW1173 ChipWhisperer-Lite.

② **First Masking Framework for HHE.** On the defensive side, we propose the first masked HERA realization, combining glitch-robust Probe Isolating Non-Interference (PINI) secure gadgets with assembly-level countermeasures against transition leakage. Along with providing proofs for security, we also validate them by performing Test Vector Leakage Assessment (TVLA). We further discuss how these masking techniques can be extended to other HHE-tailored schemes by extending it to PASTA. Our work, therefore, establishes leakage models, attack targets, and benchmarks as a foundation for systematic analysis. We further study the trade-offs between performance and leakage resistance, showing that even partial leakage can be neutralized with careful design.

③ **Open-Source Artifacts.** To support reproducibility and foster future research, we release the complete artifact package, including implementations and the power-trace dataset of our CPA attack on HERA, masked implementations of HERA and PASTA, as well as their corresponding leakage assessments, at https://anonymous.4open.science/r/hera_cpa_masking/.

II. BACKGROUND

Notations: We use lowercase letters (e.g., x) to denote scalars, uppercase letters (e.g., X) for vectors, and bold uppercase

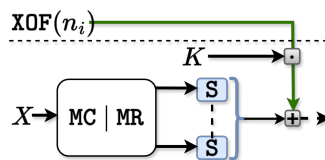


Fig. 2. The HERA permutation takes as input the key (K), nonce (n), and counter (i), and generates the truncated result- key stream (KS).

letters (e.g., M) for matrices. Subscripts (e.g., X_i) are used to index elements in a vector or matrix. Superscripts (e.g., X^i) are used to distinguish variable names.

A. Hybrid Homomorphic Encryption

In the HHE framework [1]–[4], [16] (as shown in Fig. 1), the client begins by encrypting the SE private key K with the public key under homomorphic encryption and transmits it to the server. Once this setup phase is complete, the client uses K to symmetrically encrypt the message blocks m_i and sends the resulting ciphertexts to the server whenever data storage or computation in the cloud is required. The server, in turn, homomorphically evaluates the decryption circuit, yielding a ciphertext that remains encrypted under the homomorphic scheme but can still be further processed. Finally, the client retrieves the homomorphically generated ciphertext from the server and uses the secret key to decrypt and obtain the result. Thus, as shown in Fig. 2, values such as the nonce (n) and counter (i) are considered public, since both the client and server are aware of them. Conversely, the encryption key K is strictly private to the client.

B. HERA Scheme Design Overview

We have chosen HERA [4] for our case study as it is very efficient, and other HHE scheme designs can be viewed as adaptations or variations of HERA. HERA is compatible with FHE schemes over \mathbb{F}_p (BGV, BFV, and CKKS). The state size is 16 for HERA. These 16 coefficients are stored in a 16-element vector X (4×4 matrix state X) and then processed via permutation, as shown in Fig. 2. HERA [4] permutation comprises five rounds, all utilizing cube S-boxes, thus requiring a modulus such that $\gcd(p-1, 3) = 1$. Here, p can be any prime between 16 and 60 bits depending on specific requirements of the underlying FHE scheme. The resulting KeyStream is added to the plaintext block (size t) for encryption. Each of the five HERA round consists of several layers, described as follows:

- **A (Fixed Affine Layer):** The primary distinguishing feature of HERA is its approach to use a ‘fixed’ affine layer. In this layer, matrix multiplication (MC|MR) utilizes a constant low-hamming weight *invertible* matrix M . Then, the MixColumn (MC) and MixRow (MR) are computed, where the same matrix M is multiplied column/row-wise.
- **S (S-Box Layer):** The next layer involves the S-Box operation and uses the cube S-Box ($S(X) = X^3 \pmod{p}$), which is *invertible*.
- **ARK[K, n, i] (Add Round Key):** The output of the XOF (extendable-output function using SHAKE128) is multi-

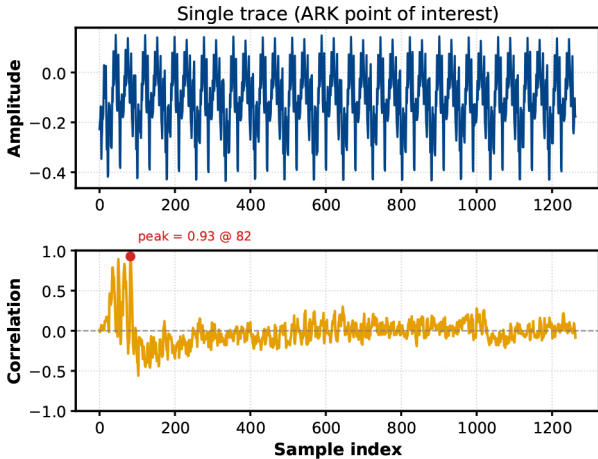


Fig. 3. Representative power trace of HERA (top) and correlation for the correct key guess (bottom), highlighting the leakage peak at the point of interest.

plied by the encryption key ($K \in \mathbb{Z}_t^{16}$), creating a key-schedule-like effect ($K \odot RC^i$) and added to the state X .

C. Correlation Power Analysis and Masking

Correlation Power Analysis (CPA) is a side-channel attack that exploits the relation between secret values and power consumption during computations. First introduced as an extension of Differential Power Analysis (DPA) [17], CPA was studied in [18]–[20]. A CPA attack targets an intermediate computation of the form $r = f(v, s)$, where v is a known input, s is the secret, and r is the result [21]. Power consumption is commonly modeled using Hamming weight leakage model, which assumes that the power consumption is proportional to the number of set bits in an intermediate register value r . Over many executions, this leakage reveals information about s . The attacker collects power traces from multiple runs, guesses intermediate values for all possible secrets, and maps them to predicted power using the leakage model. By correlating the predictions with the measured traces (via Pearson correlation), the secret yielding the highest correlation is identified. In practice, many traces are required for accurate recovery.

Masking is a key countermeasure against power analysis attacks, including Simple Power Analysis, Differential Power Analysis, and Correlation Power Analysis. The fundamental concept involves splitting each sensitive variable x into $d + 1$ uniformly distributed random masked variables, referred to as shares, where d represents the masking order. We then conduct all operations on these shares. Chari et al. [22] were the first to propose this technique, demonstrating that the measurement complexity of a single-bit DPA increases exponentially with the value of d [22].

Masking implementations usually rely on *gadgets* - small circuits that perform operations on masked shares. By carefully composing these gadgets, we can realize a complete masked algorithm, whose security can then be formally verified, for instance, using PINI proofs. Notably, according to the widely used t -probing model [23], a masked implementation is t -probing secure if any t intermediate values (probes) reveal

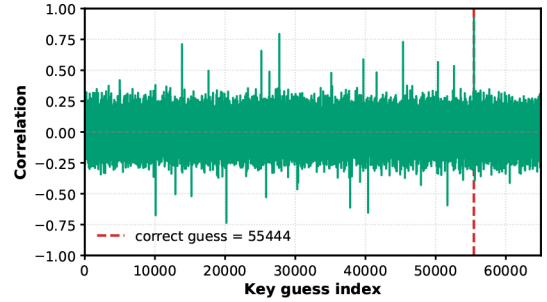


Fig. 4. Correlation values for all key guesses at the point of interest, with the correct key clearly distinguished by the peak at index 55,444.

no information about the secret, assuming the input shares are freshly randomized, where t equals the masking order.

III. THE CORRELATION POWER ANALYSIS OF HERA

In this work, we focus on HERA for evaluating side-channel resistance of prime-field lightweight ciphers. This is because HERA is specifically optimized for efficient arithmetic and employs low Hamming-weight matrices, making it well suited for hardware implementations and resource-constrained platforms. The structure of HERA can also be regarded as an adaptation of earlier designs, combining linear diffusion layers with nonlinear S-Box operations in a manner similar to other schemes. As a result, insights gained from the cryptanalysis of HERA naturally extend to other ciphers, such as PASTA, MASTA, and RUBATO.

Among the different components of HERA, the ARK step is particularly vulnerable because it directly combines the secret key K with known intermediate values, making it a prime target for a CPA attack. The ARK operation is computed as

$$\text{ARK}[K, n, i](X) = X + K \odot RC^i \pmod{p} \quad (1)$$

where RC^i is the i -th round constant, and X is the state. Since the round constants can be deterministically regenerated from the public nonce, an attacker is able to predict the key-dependent intermediate values and correlate them with power traces. This makes it possible to mount a CPA on the modular multiplication (resp. addition).

A. Practical CPA on HERA: Experimental Setup and Results

To demonstrate the feasibility of the attack, we implemented HERA with parameters $\text{HERA}(5, 16, p = 2^{16} + 1)$ on the ChipWhisperer CW308 UFO board with an STM32F303 target and captured traces using the CW1173 ChipWhisperer-Lite. The target of the attack is the first ARK operation in the first round. To perform the attack, we first collect multiple traces and the outputs of the targeted ARK operation in Eq. 1 (one such trace is represented in Fig. 3) with a fixed key and varying data. Then, we compute the predicted leakage as the hamming weight of the targeted operation for each $2^{16} + 1$ possible values of the first element of the secret key. Finally, we compute the Pearson correlation coefficient between the predicted leakage and the captured one, choosing the guess with the highest coefficient as the correct guess. We illustrate this procedure in the second image of Fig. 3 by showing the correlation of the correct guess

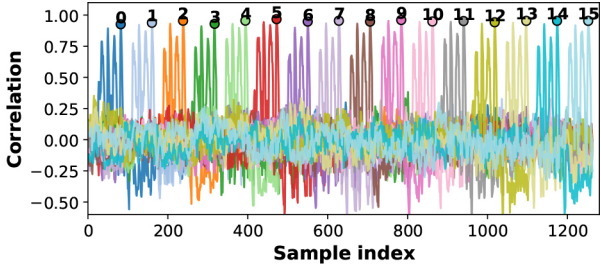


Fig. 5. CPA results for all 16 key elements. Curves show correlation across hypotheses, with dots marking the peak values that reveal the correct elements.

at each sample point. Fig. 4 illustrates the correlation values for all key guesses at the peak, and the correct key guess stands out with a dominant peak.

To recover the entire key, the same procedure is applied iteratively, with the only difference being the positions of the correlation peaks, as illustrated in Fig. 5. We perform the attack on 100 random keys to assess the success rate. Fig. 6 shows the success rate of the key recovery with respect to the amount of observed traces. The results demonstrate that the correct key can be reliably recovered with fewer than 40 traces, highlighting the vulnerability of the unmasked design.

While this demonstration shows that CPA is highly effective on the chosen parameter set, it is important to emphasize that the straightforward approach requires searching the entire space of an element of the secret key. For a modulus of size $2^{16} + 1$, this corresponds to roughly 2^{16} candidates, which remains practical. However, as the bit size of the modulus increases, the complexity of brute-force searching grows exponentially. Advanced techniques, such as those proposed in [24] and applied to a multiplication in [25], address this limitation by incrementally recovering the secret, guessing only a few bits at a time and treating the rest as noise. The trade-off of such approaches is an increase in the number of traces required due to the added statistical noise.

It is important to note that the operation under consideration involves a multiplication with the key, a feature that is common across all recently proposed integer-based HHE schemes. For example, while RUBATO employs the same key-schedule-like ARK step as HERA, both PASTA and MASTA initialize the state with the key and subsequently apply a matrix multiplication with the state, where the matrix itself is derived from the nonce and hence public. Our attack extends naturally to this operation, as well. Thus, all of these schemes are susceptible to the established CPA technique.

IV. MASKING TECHNIQUE

To prevent DPA, one of the most established protection techniques is *masking*, which randomizes intermediate computations so that single observations reveal no exploitable information about the underlying secrets. In particular, we focus on *first-order arithmetic masking*, where each sensitive variable $x \in \mathbb{Z}_p$ is decomposed into two uniformly distributed random shares, x_0 and x_1 , such that: $x = x_0 + x_1 \pmod{p}$.

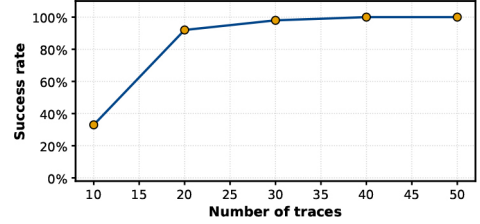


Fig. 6. Success rate of the CPA attack as a function of the number of traces for 100 random keys, converging to full recovery after ≈ 40 traces.

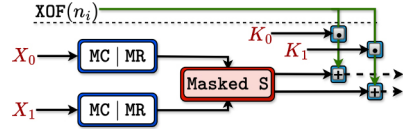


Fig. 7. Dataflow of first-order arithmetic masking in HERA. Each sensitive intermediate x is decomposed into two shares (x_0, x_1) , which are processed independently throughout the cipher. Linear operations (e.g., addition, modular reduction) are applied share-wise, while nonlinear operations (S-box) are adapted to operate on masked inputs.

The key $K \in \mathbb{Z}_p^n$ is decomposed into two uniformly random shares, $K_0, K_1 \in \mathbb{Z}_p^n$, with $n = 16$ corresponding to the HERA's state size (see Fig. 7). The masked key shares are then processed through the Affine Layer, with ARK, MC, and MR applied independently to each share (Alg. 1), and MskSB operating jointly on both shares (Alg. 3). This sequence repeats over five rounds. Throughout the computation, the state is maintained as two shares X_0 and X_1 , satisfying $X_0 + X_1 \equiv X \pmod{p}$. The output of each gadget directly serves as the input to the next, preserving the masked representation.

Proposition 1. *Independently applying linear gadgets ARK, MC, MR is first-order probing secure. Their output shares show uniform distribution and are independent of the input shares.*

Proof. The ARK, MC, and MR operations of HERA are inherently linear and preserve the masking order by construction. Therefore, we can apply these operations independently to each share (see Alg. 1). Each operation acts linearly on a single share, and each input share is uniformly random; therefore, any internal variable or output share is uniformly distributed and independent of the other shares. As a result, any adversarial probe reveals at most one share, making the operations naturally 1-PINI secure [15]. \square

HERA's S-box introduces the nonlinear confusion by cubing the state. The core challenge in masking this operation lies in realizing a secure masked multiplication. We therefore propose using the gadget MskMul [26] (as presented in Alg. 2) and apply it twice, to square the state and then to cube it, as shown in Alg. 3. Hence, the security of masking this layer primarily relies on the security of the multiplication gadget.

Proposition 2. *The gadget MskMul is first-order probing secure. Its output shares show uniform distribution and are independent of the input shares satisfying $z_0 + z_1 = x \cdot y \pmod{p}$.*

Algorithm 1 Masked ARK+MC+MR

Require: State shares $S_0, S_1 \in \mathbb{Z}_p^n$, round constants $RC^i \in \mathbb{Z}_p^n$
Require: Key shares $K_0, K_1 \in \mathbb{Z}_p^n$, Matrix $M \in \mathbb{Z}_p^{n \times n}$
Ensure: Output shares $V_0, V_1 \in \mathbb{Z}_p^n$
1: $T_0 \leftarrow S_0 + K_0 \cdot RC^i \pmod{p}$ \triangleright ARK on share 0
2: $T_1 \leftarrow S_1 + K_1 \cdot RC^i \pmod{p}$ \triangleright ARK on share 1
3: $U_0 \leftarrow M T_0 \pmod{p}$, $U_1 \leftarrow M T_1 \pmod{p}$ \triangleright MC on share 0,1
4: $V_0 \leftarrow U_0 M^T \pmod{p}$, $V_1 \leftarrow U_1 M^T \pmod{p}$ \triangleright MR on share 0,1
5: **Return** V_0, V_1

Algorithm 2 MskMul (Masked Multiplication)

Require: Shares $x_0, x_1, y_0, y_1 \in \mathbb{Z}_p$, with $x = x_0 + x_1$, $y = y_0 + y_1$, Fresh randomness $r \in \mathbb{Z}_p$
Ensure: Shares $z_0, z_1 \in \mathbb{Z}_p$, s.t. $z_0 + z_1 \equiv x \cdot y \pmod{p}$
1. $t_{00} \leftarrow x_0 y_0$, $t_{01} \leftarrow x_0 y_1$, $t_{10} \leftarrow x_1 y_0$, $t_{11} \leftarrow x_1 y_1$
2. $v_0 \leftarrow r + t_{10}$, $v_1 \leftarrow v_0 + t_{01}$
3. $z_0 \leftarrow p - r + t_{00}$, $z_1 \leftarrow v_1 + t_{11}$
Return z_0, z_1

Proof. Assume that x_0, x_1 and y_0, y_1 are uniform arithmetic shares of $x, y \in \mathbb{Z}_p$, such that $x = x_0 + x_1 \pmod{p}$ and $y = y_0 + y_1 \pmod{p}$. We show that any probe on an intermediate value is independent of the secrets x and y . The intermediate terms $t_{00} = x_0 \cdot y_0$, $t_{01} = x_0 \cdot y_1$, $t_{10} = x_1 \cdot y_0$, and $t_{11} = x_1 \cdot y_1$ each involve at most one share of each input. Since no partial product simultaneously involves both x_0 and x_1 (or, analogously, y_0 and y_1), and every share is uniform and independent of the secret, these products are independent of x and y . Through the variable v_1 , we sum the intermediate multiplications t_{00} and t_{10} to obtain the output share z_1 .

To carry out this summation without revealing the secret x (since $t_{00} + t_{10} = y_0 \cdot x$), we must mask this intermediate result. To achieve this, we sample a random mask r uniformly and independently. Hence, $v_0 = r + t_{10}$ is uniform in \mathbb{Z}_p , independent of all secrets; consequently $v_1 = v_0 + t_{01}$ is also uniform and independently of the secret state x . The output share $z_0 = (p - r) + t_{00}$ is uniform because it represents an offset of the uniform variable, $(p - r)$ (independent of the secrets). The output share $z_1 = v_1 + t_{11}$ is an offset of the uniform variable v_1 ; hence, it is also uniformly distributed and independent of the secret, thanks to the randomness in v_1 . Therefore, in all steps, no probe leaks any information about the inputs x and y , while ensuring the correct computation of the desired multiplication, since:

$$z_0 + z_1 = p - r + t_{00} + r + t_{10} + t_{01} + t_{11} = x \cdot y \pmod{p}$$

Proposition 3. *The gadget MskSB is first-order probing secure. Its output shares show uniform distribution and are independent of the input shares, satisfying $z = z_0 + z_1 = x^3 \pmod{p}$.* \square

Proof. Since we have proven the first-order PINI security of MskMul, any gadget that uses it while preserving the masking structure is also first-order PINI secure by composition. In particular, this applies to the S-box layer, where we apply the multiplication gadget twice. Notably, we refresh the masks of the shares before and after each multiplication. Mask refreshing is essential to maintain the security of masked computations, as multiplications in arithmetic masking are nonlinear when both operands are secret, which may induce correlations between

Algorithm 3 MskSB (Masked S-box cube)

Input: Shares $X_0, X_1 \in \mathbb{Z}_p^n$
Requires: $R_1, R_2 \in \mathbb{Z}_p^n \leftarrow$ fresh randomness
Output: Shares $Z_0, Z_1 \in \mathbb{Z}_p^n$, s.t. $Z_0 + Z_1 = X^3 \pmod{p}$
for $i = 0$ to $n - 1$ **do**
 $y_0 \leftarrow X_{0i} + R_{1i} \pmod{p}$
 $y_1 \leftarrow X_{1i} + p - R_{1i} \pmod{p}$
 $v_0, v_1 \leftarrow \text{MskMul}(X_{0i}, y_0, X_{1i}, y_1)$
 $t_0 \leftarrow v_0 + R_{2i} \pmod{p}$ \triangleright Share Refresh
 $t_1 \leftarrow v_1 + p - R_{2i} \pmod{p}$ \triangleright Share Refresh
 $Z_{0i}, Z_{1i} \leftarrow \text{MskMul}(t_0, y_0, t_1, y_1)$
end for
Return Z_0, Z_1

shares and potentially leak secret information. The mask refreshing is first-order PINI secure by design, as it is performed independently on distinct shares - y_0, y_1, t_0 and t_1 each depend on a single share and fresh randomness (R_1 and R_2). Moreover, the output shares of MskSB are identical to those of MskMul, preserving their uniform distribution. Hence, the S-box layer is first-order PINI secure. \square

So far, we have analyzed individual gadgets within one HERA round and established their first-order probing security. We now show that repeated application of these gadgets within and across rounds extends this property to the full encryption.

Proposition 4. *The full encryption, composed of first-order probing-secure gadgets, is itself first-order probing secure.*

Proof. Sequentially composing 1-PINI secure gadgets over multiple rounds does not introduce any single-probe leakage, since each gadget exposes only masked shares and uses independent randomness. Therefore the first-order probing security is preserved for the full encryption. \square

A. Practical Implementation Strategies

Importantly, the PINI proofs presented do not account for implementation-specific issues, such as transition leakage. While classical Hamming-distance leakage mostly comes up in the discussion in the context of Boolean sharing [27], arithmetic shares can also be affected by transition leakage. In particular, if multiple arithmetic shares reside temporarily in the same register, the bitwise transitions caused by updates of arithmetic operations may leak information about the secret through correlations between successive register states.

To mitigate this risk, we enforce strict code-level isolation of operations on individual shares within the affine layer in the C implementation. For the nonlinear layer, we implement all share manipulations in assembly, giving us complete control over register allocation and lifetime. Moreover, all functions take only the addresses of shares as input, so the shares never reside in registers in an uncontrolled manner. To further reduce risk, we introduce non-functional memory reads and writes to separate consecutive accesses to the shares. By following these heuristics, we ensure each share resides in a dedicated register, with minimal lifetime and without unintended overlaps.

Concretely, the multiplication gadget MskMul, including mask refreshing and the subsequent share updates, is implemented entirely in assembly for Cortex-M4 and assembled with

TABLE I
COMPARISON OF NOT MASKED AND MASKED HERA IMPLEMENTATIONS IN
TERMS OF AVERAGE CLOCK CYCLES AND STACK USAGE.

	Clock cycles		Stack usage	
	Not Masked	Masked	Not Masked	Masked
HERA	38,249	4,853,801	700 bytes	852 bytes
PASTA	4,249,061	15,549,629	1,756 bytes	2,072 bytes

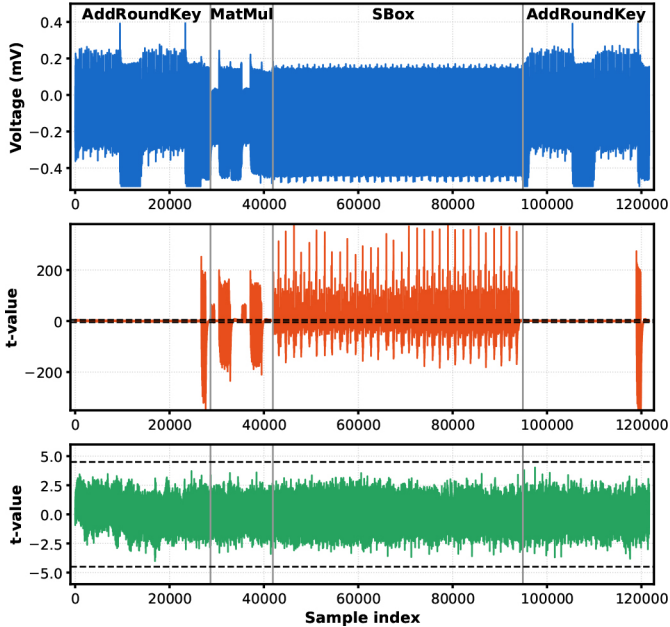


Fig. 8. HERA power traces and corresponding t-test evaluations. The top plot shows a single HERA encryption trace for one round, with vertical dashed lines marking major computation phases (AddRoundKey, MatMul, SBox, AddRoundKey). The middle plot presents TVLA results with faulty randomness, where the test statistic exceeds the ± 4.5 threshold lines in red, revealing clear leakage. The bottom plot shows the same TVLA evaluation with proper randomness for 10,000 traces, where the values remain within the thresholds, indicating the absence of significant leakage.

GCC 13.2.1. We compile and link with `-fno-lto` flag to avoid link-time transformations (e.g., cross-module inlining, or dead-code elimination) that could alter the call context, register/stack behavior, or remove security-relevant structure around the gadget. This approach preserves the independence of shares assumed in the formal PINI proofs and yields the results summarized in Tab. I.

B. Test Vector Leakage Assessment

We validated our implementation via the TVLA, which detects whether any first-order leakage remains in the implementation by evaluating the t-test scores across fixed versus random inputs [28], [29]. Passing first-order TVLA tests provides strong evidence that the implementation does not exhibit detectable first-order leakage beyond the scope of the theoretical model. Therefore, we run the t-tests for 10,000 traces and follow the common practice of using a detection threshold of ± 4.5 for the TVLA t-statistic. As depicted in Fig. 8, the t-values remain within this interval across all measurement points, indicating no statistically significant first-order leakage.

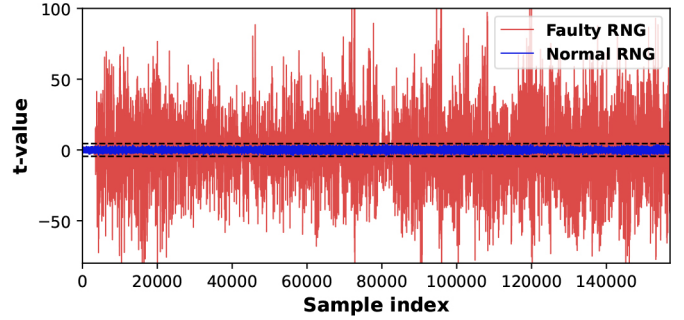


Fig. 9. Leakage assessment of the Feistel S-box in PASTA.

C. First-Order Masking Across Related Cipher Schemes

We can apply the masking techniques utilized for HERA across related HHE cipher schemes like MASTA [3] or PASTA [2], which also operate over an arithmetic ring. The affine layer of these ciphers allows trivial masking by applying each linear operation independently to each share. These ciphers introduce the nonlinearity through multiplication, in the same manner as in HERA. In particular, PASTA employs a Feistel-like S-box $(X^j + [j \neq 0](X^{j-1})^2) \pmod p$ in every round except the final one. We can implement the core of this S-box’s masking using our `MskMul` multiplication gadget. In its final round, PASTA instead uses the same cubic S-box layer as HERA, which further aligns the masking strategies of the two ciphers. Following this, we also masked PASTA (results in Tab. I) and validated it with TVLA, as shown in Fig. 9.

D. Discussion on Higher-order extension

An adversary may exploit higher-order power analysis by correlating information across multiple shares or intermediate values. It is therefore important to emphasize that first-order PINI proofs provide security only against single-probe attacks; achieving resistance against more advanced side-channel adversaries requires higher-order masking or additional countermeasures. The arithmetic masking scheme employed here can be extended to order d security by decomposing each secret into $d+1$ shares and preserving statistical independence throughout the computations. Linear layers apply directly to each share, while the nonlinear layer - particularly the multiplication gadget - must be adapted by combinatorially multiplying the shares of two secrets and redistributing the results into d output shares.

V. CONCLUSION

This work provides the first practical demonstration of power analysis attacks on integer-based HHE schemes, using HERA as a case study. We showed that full key recovery is achievable with very few traces, underscoring the importance of side-channel resistance in real deployments. To address this, we introduced a masking framework tailored to arithmetic ciphers and validated its effectiveness through leakage assessments. Beyond HERA, our approach generalizes to other HHE-tailored designs, as demonstrated using PASTA. By combining attack insights with practical countermeasures, this study bridges a critical gap and sets the stage for side-channel-resilient hybrid homomorphic encryption.

REFERENCES

- [1] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the AES circuit," in *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings* (R. Safavi-Naini and R. Canetti, eds.), vol. 7417 of *Lecture Notes in Computer Science*, pp. 850–867, Springer, 2012.
- [2] C. Dobraunig, L. Grassi, L. Helming, C. Rechberger, M. Schafneggler, and R. Walch, "Pasta: A case for hybrid homomorphic encryption," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 3, pp. 30–73, 2023.
- [3] J. Ha, S. Kim, W. Choi, J. Lee, D. Moon, H. Yoon, and J. Cho, "Masta: An he-friendly cipher using modular arithmetic," *IEEE Access*, vol. 8, pp. 194741–194751, 2020.
- [4] J. Cho, J. Ha, S. Kim, B. Lee, J. Lee, J. Lee, D. Moon, and H. Yoon, "Transciphering framework for approximate homomorphic encryption," in *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part III* (M. Tibouchi and H. Wang, eds.), vol. 13092 of *Lecture Notes in Computer Science*, pp. 640–669, Springer, 2021.
- [5] J. H. Cheon, A. Kim, M. Kim, and Y. S. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I* (T. Takagi and T. Peyrin, eds.), vol. 10624 of *Lecture Notes in Computer Science*, pp. 409–437, Springer, 2017.
- [6] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "Fully homomorphic encryption without bootstrapping," *Electron. Colloquium Comput. Complex.*, p. 111, 2011.
- [7] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *IACR Cryptol. ePrint Arch.*, p. 144, 2012.
- [8] A. Aikata, D. S. Sobrino, and S. S. Roy, "PASTA on edge: Cryptoprocessor for hybrid homomorphic encryption," *IEEE Design and Automation Conference*, p. 1919, 2025.
- [9] Y. Jeon, M. Erez, and M. Orshansky, "Presto: Hardware acceleration of ciphers for hybrid homomorphic encryption," *CoRR*, vol. abs/2507.00367, 2025.
- [10] Aikata, A. Dabholkar, D. Saha, and S. S. Roy, "SASTA: ambushing hybrid homomorphic encryption schemes with a single fault," *IACR Cryptol. ePrint Arch.*, p. 41, 2024.
- [11] R. Radheshwar, M. Kansal, P. Méaux, and D. Roy, "Differential Fault Attack on Rasta and FiLIP_{DSM}," *IEEE Trans. Computers*, vol. 72, no. 8, pp. 2418–2425, 2023.
- [12] D. Roy, B. N. Bathe, and S. Maitra, "Differential fault attack on kreyvium and FLIP," *IEEE Trans. Computers*, vol. 70, no. 12, pp. 2161–2167, 2021.
- [13] NIST, "Fips pub 197: Advanced encryption standard (aes)," p. 311, 2001.
- [14] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schl affer, "Ascon v1.2: Lightweight authenticated encryption and hashing," *J. Cryptol.*, vol. 34, no. 3, p. 33, 2021.
- [15] G. Cassiers and F. Standaert, "Trivially and efficiently composing masked gadgets with probe isolating non-interference," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2542–2555, 2020.
- [16] J. Ha, S. Kim, B. Lee, J. Lee, and M. Son, "Rubato: Noisy ciphers for approximate homomorphic encryption," in *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part I* (O. Dunkelman and S. Dziembowski, eds.), vol. 13275 of *Lecture Notes in Computer Science*, pp. 581–610, Springer, 2022.
- [17] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings* (M. J. Wiener, ed.), vol. 1666 of *Lecture Notes in Computer Science*, pp. 388–397, Springer, 1999.
- [18] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings* (M. Joye and J. Quisquater, eds.), vol. 3156 of *Lecture Notes in Computer Science*, pp. 16–29, Springer, 2004.
- [19] J. Coron, P. C. Kocher, and D. Naccache, "Statistics and secret leakage," in *Financial Cryptography, 4th International Conference, FC 2000 Anguilla, British West Indies, February 20-24, 2000, Proceedings* (Y. Frankel, ed.), vol. 1962 of *Lecture Notes in Computer Science*, pp. 157–173, Springer, 2000.
- [20] R. Mayer-Sommer, "Smartly analyzing the simplicity and the power of simple power analysis on smartcards," in *Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings* (C. K. Ko  and C. Paar, eds.), vol. 1965 of *Lecture Notes in Computer Science*, pp. 78–92, Springer, 2000.
- [21] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.
- [22] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers* (B. S. K. Jr., C. K. Ko , and C. Paar, eds.), vol. 2523 of *Lecture Notes in Computer Science*, pp. 13–28, Springer, 2002.
- [23] Y. Ishai, A. Sahai, and D. A. Wagner, "Private circuits: Securing hardware against probing attacks," in *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings* (D. Boneh, ed.), vol. 2729 of *Lecture Notes in Computer Science*, pp. 463–481, Springer, 2003.
- [24] M. Tunstall, N. Hanley, R. McEvoy, C. Whelan, C. Murphy, and W. Marnane, "Correlation power analysis of large word sizes," pp. 13–14.
- [25] Z. Chen, E. Karabulut, A. Aysu, Y. Ma, and J. Jing, "An efficient non-profiled side-channel attack on the crystals-dilithium post-quantum signature," in *2021 IEEE 39th International Conference on Computer Design (ICCD)*, pp. 583–590, 2021.
- [26] G. Cassiers, L. Masure, C. Momin, T. Moos, and F. Standaert, "Prime-field masking in hardware and its soundness against low-noise SCA attacks," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 2, pp. 482–518, 2023.
- [27] J. Balasch, B. Gierlichs, V. Grosso, O. Reparaz, and F. Standaert, "On the cost of lazy engineering for masked software implementations," in *Smart Card Research and Advanced Applications - 13th International Conference, CARDIS 2014, Paris, France, November 5-7, 2014. Revised Selected Papers* (M. Joye and A. Moradi, eds.), vol. 8968 of *Lecture Notes in Computer Science*, pp. 64–81, Springer, 2014.
- [28] B. J. Gilbert Goodwill, J. Jaffe, P. Rohatgi, et al., "A testing methodology for side-channel resistance validation," in *NIST non-invasive attack testing workshop*, vol. 7, pp. 115–136, 2011.
- [29] T. Schneider and A. Moradi, "Leakage assessment methodology - extended version," *J. Cryptogr. Eng.*, vol. 6, no. 2, pp. 85–99, 2016.