Less is More: Latent Diffusion for Efficient IoT Side-Channel Analysis

Zekai Zhang, Donglong Chen, *Member, IEEE*, Wangchen Dai, Jinfa Hong, Yu Hin Chan, Çetin Kaya Koç, *Life Fellow, IEEE*, Patrick S.Y. Hung and Ray C.C. Cheung, *Senior Member, IEEE*

***Abstract*—The proliferation of cryptographic primitives in resource-constrained Internet of Things (IoT) devices has made them prime targets for Side-Channel Analysis (SCA). However, designing effective defenses against these attacks has become increasingly complex, as traditional deep learning approaches rely heavily on extensive profiling datasets that are difficult to obtain in the context of widely distributed and physically restricted IoT environments. This challenge is further exacerbated by countermeasures such as clock jitter and random delays. To overcome this limitation, this paper introduces a novel and data-efficient three-stage framework for generating high-fidelity synthetic side-channel traces. First, we employ a Supervised Variational Autoencoder (S-VAE) to map noisy, high-dimensional raw traces into a compact and denoised latent space, effectively creating an information-rich manifold. Second, a conditional Denoising Diffusion Implicit Model (DDIM), powered by an advanced attention-augmented U-Net, is trained exclusively on this computationally tractable latent space to learn the complex conditional data distribution. Finally, we empirically validate our framework on the public ASCAD benchmark and ChipWhisperer CW308T UFO platform. The results are compelling: an attack model trained solely on our synthetic data successfully recovers the secret key in the most challenging ASCAD_desync100 scenario using only 4107 traces and using only 2560 traces, 97.7% accuracy can be achieved on the Chipwhisphere platform . This work provides a practical and efficient pathway for the robust security evaluation of cryptographic implementations in data-scarce IoT environments, significantly lowering the barrier for thorough side-channel vulnerability analysis.**

***Index Terms*—Side-channel attacks, Diffusion model, Deep learning (DL), Conditional generation**

I. Introduction

EMBEDDED cryptographic devices are foundational to modern digital security, safeguarding sensitive information in a vast array of applications, from smart cards to Internet of Things (IoT) devices. In typical IoT scenarios, such as smart metering infrastructures, industrial control systems, and wearable health monitors, these devices often operate in distributed and physically unrestricted environments. Such exposure makes them easily accessible to potential adversaries, while strict resource constraints limit the complexity of their hardware defenses. However, these devices are vulnerable to Side-Channel Analysis (SCA), a class of potent physical attacks that do not rely on cryptographic weaknesses but rather on exploiting physical leakages like power consumption [1], electromagnetic emissions [2], or timing variations that occur during computation. Profiled attacks [3], a formidable category of SCA, involve characterizing a target device's physical leakages in a controlled "profiling" phase to build a predictive model, which is later used in an "attack" phase to extract secret keys from a new set of measurements.

In recent years, deep learning has revolutionized profiled SCA, with models such as Multi-Layer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs) demonstrating remarkable success in key recovery, even against implementations protected by countermeasures. These deep learning-based attacks, however, are notoriously data-hungry, requiring large and diverse datasets of side-channel traces to train effectively. In practice, acquiring a sufficient number of high-quality traces can be a significant bottleneck due to measurement costs, time constraints, or limited access to the target device. Furthermore, the effectiveness of these models can be hampered by countermeasures like clock jitter [4] or random delays (desynchronization), which are specifically designed to misalign traces and frustrate pattern recognition.

To overcome the challenge of data scarcity and improve attack resilience, data augmentation has emerged as a promising strategy. Generative models, in particular, offer a powerful paradigm for creating synthetic yet highly realistic side-channel traces to enrich training datasets. While Generative Adversarial Networks (GANs) [5] have been explored for this purpose, they can sometimes suffer from training instability

This work was supported in part by the CityUHK Internal Grant under Grant 9239083; in part by the National Natural Science Foundation of China under Grant 62372417; in part by the Jiangsu Province 100 Foreign Experts Introduction Plan under Grant BX2022012; in part by the Scientific Research Innovation Capability Support Project for Young Faculty under Grant SRICSPYF-BS2025137; in part by the Guangdong Provincial Key Laboratory of IRADS under Grant 2022B1212010006; in part by the Guangdong and Hong Kong Universities "1+1+1" Joint Research Collaboration Scheme under Grant 2025A0505000001; in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515011274; in part by the Guangdong Province General Universities Key Field Project (New Generation Information Technology) under Grant 2023ZDZX1033; in part by the UIC Research Grant under Grant UICR04202401-21; and in part by the CCF-Huawei Populus Grove Fund under Grant CCF-HuaweiTC202310.

Z. Zhang, J. Hong, Y. Chan, P.S.Y. Hung and R.C.C. Cheung are with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China. E-mail: zekazhang2-c@my.cityu.edu.hk, jinfahong2-c@my.cityu.edu.hk, yhchan96@cityu.edu.hk, psyhung@cityu.edu.hk, r.cheung@cityu.edu.hk.

D. Chen is with Guangdong Provincial/Zhuhai Key Laboratory of IRADS, Beijing Normal-Hong Kong Baptist University, Zhuhai 519088, China. E-mail: donglongchen@bnbu.edu.cn.

W. Dai is with Sun Yat-sen University, Shenzhen, China. (e-mail: w.dai@my.cityu.edu.hk).

Çetin Kaya Koç is with Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China, and also with University of California, Santa Barbara, Santa Barbara, CA 93106 USA. (e-mail: cetinkoc@ucsb.edu).

Corresponding author: Donglong Chen.

or mode collapse. More recently, diffusion models [6], [7] have emerged as a new class of state-of-the-art generative models capable of producing exceptionally high-fidelity samples across various domains, including time-series data. These models learn to reverse a gradual noising process, enabling them to generate diverse and realistic data from a simple noise distribution. The main contributions of this work are shown as follows:

1) We introduce a Supervised Variational Autoencoder (S-VAE) not merely as a pre-processor, but as a powerful feature purifier. We demonstrate that an attack conducted directly within our compact 32-dimensional latent space can, in a synchronized scenario, significantly outperform a state-of-the-art attack on the original 700-dimensional raw data. This validates our latent space as a more robust and information-dense representation for side-channel analysis, effectively filtering noise to isolate the core leakage manifold.
2) We introduce, for the first time, a framework that synergistically combines an S-VAE with a conditional Denoising Diffusion Implicit Model (DDIM) for side-channel trace synthesis. By operating on a reduced-dimension latent space, our approach stands in stark contrast to contemporary generative models like GANs and raw-trace Denoising Diffusion Probabilistic Models (DDPMs) that contend with high-dimensional, noisy data. This design dramatically reduces the computational burden, significantly shortening both training and generation times without sacrificing the quality of the synthetic traces and offering a practical and scalable solution for data augmentation.
3) To the best of our knowledge, our work demonstrates superior performance against reported baselines, presenting the first successful attack using synthetically generated data from a diffusion model on the highly challenging ASCAD_desync100 dataset. At the same time, the goal of recovering key using only a small number of traces was achieved on the Chipwhisphere platform. Experimental results demonstrate that an attack model trained exclusively on our generated data achieves highly competitive performance compared to those trained on real data. This demonstrates the exceptional robustness of our framework and its capability to overcome strong, real-world countermeasures, providing a potential blueprint for tackling other challenging cryptographic algorithms and implementations.

II. Related Works

Our research integrates three distinct but complementary domains within side-channel analysis: deep learning for leakage modeling, generative methods for data synthesis, and latent space representations for data efficiency.

The effectiveness of recent profiling attacks hinges on the capacity of deep neural networks to autonomously discern subtle leakage patterns from noisy, high-dimensional side-channel traces. Convolutional Neural Networks (CNNs) have proven particularly adept at this task, demonstrating resilience to desynchronization countermeasures by learning spatially invariant leakage features [8], [9]. The deep CNN used in our final validation phase is representative of this established approach. While powerful, the performance of these models is fundamentally constrained by the quality and quantity of the available profiling data, motivating the critical need for robust data synthesis.

To address this data dependency, the field has increasingly explored generative techniques. Beyond the specific domain of SCA, generative AI has recently demonstrated transformative potential across the broader landscape of wireless network security and IoT. Comprehensive surveys have highlighted the fundamental architecture and state-of-the-art of generative diffusion models for wireless networks [10], while other works have established their utility in emerging paradigms such as secure semantic communications [11] and physical-layer authentication [12]. These advancements underscore the growing capability of generative models to capture complex physical-layer distributions.

The first wave of research in SCA was dominated by Generative Adversarial Networks (GANs), which were successfully used to augment training sets and generate traces for specific cryptographic intermediates [13], [14]. However, the adversarial dynamics of GANs often lead to training instability and mode collapse, potentially limiting the diversity and fidelity of the generated data. More recently, Denoising Diffusion Models have emerged as a more stable and powerful alternative, offering state-of-the-art generation quality [6]. Their initial applications in SCA have confirmed their ability to produce high-fidelity traces that retain critical leakage information [15], [16]. Crucially, these pioneering works apply the diffusion process directly to the raw trace data. This direct approach, while effective, is inherently inefficient; it forces the model to learn the distribution of the entire signal, including noise and redundant features, rendering it computationally expensive and critically dependent on vast training datasets.

An orthogonal line of research has focused on creating more compact and noise-resilient representations of side-channel traces. Autoencoders (AEs) and Variational Autoencoders (VAEs) [17], [18] have been employed for tasks such as denoising and feature extraction. This body of work establishes a critical precedent: a compact latent space can effectively encapsulate the salient leakage characteristics of the original signal while discarding irrelevant noise.

Our work is positioned at the confluence of these research streams, proposing a novel synthesis of latent representation and generative modeling. Rather than applying the computationally burdensome diffusion process directly to raw traces, we pioneer a three-stage strategy. We first leverage a Supervised Variational Autoencoder (S-VAE) to learn an efficient and information-rich latent representation, a technique validated for its feature purification capabilities. Subsequently, we train an advanced, attention-based Conditional Denoising Diffusion Implicit Model (DDIM) exclusively within this compact and well-structured latent space. This strategic fusion of techniques yields a framework that is not only computationally tractable but also remarkably data-efficient, enabling the generation of high-fidelity synthetic traces from training

TABLE I
LIST OF NOTATIONS USED IN THIS PAPER.

Symbol	Description	Symbol	Description
General and Data Symbols			
$\mathbf{x}$	Raw power trace vector in $\mathbb{R}^L$	$\mathbf{z}$	Latent vector in $\mathbb{R}^{D_z}$
c	Integer cryptographic label (S-box output)	L	Length of raw power traces (700)
D_z	Dimensionality of the latent space (32)	$X_{\text{sync/desync}}$	Datasets of synchronized/desynchronized traces
Supervised Variational Autoencoder (S-VAE) Symbols			
$f_{\text{enc}}(\cdot;\phi)$	S-VAE encoder network with parameters ϕ	$\mathcal{X}_{\text{profile}}, \mathcal{Y}_{\text{profile}}$	Datasets of traces and corresponding labels
$f_{\text{dec}}(\cdot;\theta)$	S-VAE decoder network with parameters θ	$\mathbf{logits}$	Output logits from the classification head
$f_{\text{class}}(\cdot;\omega)$	S-VAE classification head with parameters ω	$\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2$	Mean and log variance vectors from the encoder
Loss Function Components			
$\mathcal{L}_{\text{total}}$	Total loss function for the S-VAE	β	Weight of the KL divergence in the S-VAE loss
$\mathcal{L}_{\text{MSE}}$	Mean Squared Error reconstruction loss	γ	Weight of the CE classification loss in the S-VAE loss
$\mathcal{L}_{\text{KL}}$	Kullback-Leibler divergence loss	D_{KL}	Kullback-Leibler divergence function
$\mathcal{L}_{\text{CE}}$	Cross-Entropy classification loss		
Denoising Diffusion Implicit Model (DDIM) Symbols			
$\mathbf{z}_0$	Original clean latent vector	$\mathbf{z}_t$	Noisy latent vector at timestep t
$\mathbf{z}_T$	Final noisy vector, equivalent to pure noise	$\boldsymbol{\epsilon}$	Gaussian noise vector sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$
t	Diffusion timestep index, $t \in \{1, \ldots, T\}$	T	Total number of diffusion timesteps
β_t	Noise variance schedule at timestep t	N	Number of inference steps for sampling
α_t	Defined as $1 - \beta_t$	$\bar{\alpha}_t$	Cumulative product $\prod_{i=1}^{t} \alpha_i$
$\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, c, t)$	Predicted noise by the U-Net with parameters θ	$\hat{\mathbf{z}}_0$	Predicted clean latent vector during sampling
$\mathcal{L}_{\text{DDIM}}$	MSE loss function for training the U-Net	θ	Trainable parameters of the U-Net model

datasets. The final stage then validates the efficacy of our approach by using these synthetically generated traces to train a powerful attack model and successfully compromise the real-world target.

III. PRELIMINARY

The methodology outlined in this paper is based on a formal mathematical framework that utilizes a variety of symbols and variables to define its essential components. To enhance clarity and consistency, we have compiled all key notations used throughout this study into a single, comprehensive reference list for the reader's convenience. This list, presented in Table I, details the symbols pertaining to the datasets, the architecture of the Supervised Variational Autoencoder (S-VAE), the mechanics of the Denoising Diffusion Implicit Model (DDIM), and other relevant variables.

A. Supervised Variational Autoencoder for Leakage-Aware Latent Space Formulation

The foundational challenge in generating side-channel traces lies in the inherent nature of the raw data. Raw traces are afflicted by the curse of dimensionality, residing in a high-dimensional space that is sparsely populated and dominated by noise and redundant features. A generative model applied directly to this raw space would waste significant capacity learning to replicate irrelevant signal components, failing to capture the subtle cryptographic leakage.

To overcome this, our methodology commences with the formulation of a structured, low-dimensional latent space. However, we posit that a standard Variational Autoencoder (VAE), trained solely on a reconstruction objective, is fundamentally misaligned with the goals of side-channel analysis. A standard VAE may learn to compress and denoise a trace, but it has no explicit incentive to preserve the very high-frequency, low-amplitude signals that constitute the information leakage. These critical features, while vital for classification, often contribute minimally to the overall reconstruction error and are thus discarded as noise.

To address this fundamental limitation, our primary contribution is the design and implementation of a Supervised Variational Autoencoder (S-VAE) [19], an architecture tailored to create a "leakage-aware" latent space. The S-VAE enhances the traditional encoder-decoder structure with a third component: a lightweight classification head. This head operates directly on the latent mean vector $\boldsymbol{\mu}$ produced by the encoder.

This architectural modification enables a paradigm shift in the training objective. Instead of the conventional Evidence Lower Bound (ELBO), we introduce a novel multi-component loss function that formally decomposes the side-channel signal into deterministic leakage and stochastic noise. Our objective function is a carefully weighted sum of three distinct losses, each serving a specific physical role in the context of SCA:

$$\mathcal{L}_{\text{S-VAE}} = \mathcal{L}_{\text{MSE}} + \beta \cdot D_{KL} + \gamma \cdot \mathcal{L}_{\text{CE}} \tag{1}$$

where:

- $\mathcal{L}_{\text{MSE}}$ is the Mean Squared Error reconstruction loss. Physically, this term enforces signal fidelity, ensuring that the latent vector $\mathbf{z}$ captures the dominant energy variations and temporal structure of the raw power trace $\mathbf{x}$.
- D_{KL} is the Kullback-Leibler divergence, acting as a structural regularizer. By forcing the latent distribution to approximate a Gaussian prior, it imposes a "smoothness" constraint on the learned manifold. In the context of IoT devices, this effectively filters out high-frequency, non-Gaussian stochastic noise (e.g., electronic thermal noise) which typically does not conform to the compact latent structure.

- $\mathcal{L}_{\text{CE}}$ is the Cross-Entropy classification loss. This term acts as a semantic anchor for the secret key information. Side-channel leakage manifests as minute variations in dynamic power consumption correlated with the sensitive intermediate value y (the S-box output). This loss component explicitly forces the encoder to isolate these subtle, key-dependent signal features from high-amplitude background activity, ensuring that the latent space $\mathbf{z}$ preserves the critical cryptographic information required for key recovery.

The inclusion of the classification loss term, weighted by the hyperparameter γ, introduces a critical trade-off between signal fidelity and leakage extraction. While the reconstruction loss ($\mathcal{L}_{\text{MSE}}$) aims to preserve the exact waveform, the classification loss ($\mathcal{L}_{\text{CE}}$) explicitly guides the encoder to prioritize features discriminative for the cryptographic operation. By tuning γ, we balance the accuracy of trace reconstruction against the robustness of the latent representation. This compels the S-VAE to learn a manifold where vectors corresponding to different intermediate values are maximally separable, effectively filtering out noise that does not contribute to the security analysis.

Through empirical analysis, we identified a latent dimensionality of $D = 32$ as a critical balance point between computational complexity and model expressiveness. This dimension is compact enough to strictly limit the modeling of irrelevant high-frequency noise, yet sufficiently high-dimensional to capture the subtle, non-linear dependencies of the leakage. Theoretically, this approach relies on optimizing the Evidence Lower Bound (ELBO) as an approximation to the intractable true data likelihood. This variational relaxation allows us to transform the computationally prohibitive problem of modeling raw, high-dimensional traces into a well-posed, computationally efficient task within a compact latent space—a formulation specifically optimized for the resource constraints typical of IoT security evaluations.

B. Denoising Diffusion Implicit Models (DDIMs)

Our proposed framework for generating high-fidelity side-channel traces is a three-stage process. First, a Supervised Variational Autoencoder (S-VAE) is trained on the real trace dataset to learn a compressed and information-rich latent space. Second, a conditional Denoising Diffusion Implicit Model (DDIM) [20] is trained exclusively within this low-dimensional latent space to generate new traces. Finally, these synthetic traces are used to train a neural network to perform a profiled attack. This section details the workings of the DDIM component.

The Denoising Diffusion Implicit Model is a generative model that learns to reverse a predefined process that gradually adds noise to data. Its operation can be understood in three key parts: the fixed Forward Process, the learnable Reverse Process, and the efficient DDIM Generation Process.

It is critical to emphasize that the update rules and training objectives employed in this framework are not heuristic designs but are strictly derived from a well-established optimization framework. The training objective is mathematically equivalent to maximizing the Evidence Lower Bound (ELBO) on the data log-likelihood, a fundamental principle in variational inference. Furthermore, the deterministic sampling process (update rules) used in DDIM is rigorously derived from the discretization of the Probability Flow Ordinary Differential Equation (ODE). This theoretical grounding ensures that our model approximates the score function of the data distribution, providing convergence guarantees consistent with the theory of stochastic differential equations.

1) Forward Process (Noising): The forward process, also known as the diffusion process, is a fixed Markov chain that systematically corrupts an input data point by gradually adding Gaussian noise over a series of T discrete timesteps. Let $\mathbf{z}_0$ be a clean latent vector from the S-VAE encoder. The forward process generates a sequence of increasingly noisy latent vectors $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T$, where $\mathbf{z}_T$ is indistinguishable from pure Gaussian noise.

This process is governed by a predefined noise schedule, $\{\beta_t\}_{t=1}^{T}$, which determines the variance of the noise added at each step t. In our implementation, we use a cosine schedule, which has been shown to improve generation quality. A key property of this process is that we can directly sample a noisy vector $\mathbf{z}_t$ at any arbitrary timestep t from the original data $\mathbf{z}_0$ in a closed form:

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \tag{2}$$

where:

- $\boldsymbol{\epsilon}$ is a random noise vector sampled from a standard normal distribution, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- $\alpha_t = 1-\beta_t$, and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ is the cumulative product of the α values.
- The terms $\sqrt{\bar{\alpha}_t}$ and $\sqrt{1-\bar{\alpha}_t}$ are scaling factors that control the signal-to-noise ratio at timestep t. As t approaches T, $\bar{\alpha}_t$ approaches 0, causing the signal component ($\mathbf{z}_0$) to vanish and leaving only noise.

This forward process is not learned; it is a fixed procedure used to prepare the training data for the neural network.

2) Reverse Process: The core of the diffusion model's intelligence lies in the reverse process. The goal is to train a neural network, denoted as $\boldsymbol{\epsilon}_\theta$, that can reverse the noising process. Specifically, given a noisy latent vector $\mathbf{z}_t$ at timestep t, the network is trained to predict the noise component $\boldsymbol{\epsilon}$ that was added to create it.

Our network $\boldsymbol{\epsilon}_\theta$ is a U-Net architecture conditioned on both the timestep t and the corresponding cryptographic label c (i.e., the S-box output). The model takes the noisy latent vector $\mathbf{z}_t$, the timestep embedding for t, and the label embedding for c as input, and outputs a prediction of the noise, $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, c)$.

The training objective is to minimize the difference between the predicted noise and the actual noise $\boldsymbol{\epsilon}$ used in the forward process. This is achieved using a Mean Squared Error (MSE) loss function:

$$\mathcal{L} = \mathbb{E}_{\mathbf{z}_0, c, \boldsymbol{\epsilon}, t}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, c)\right\|^2\right] \tag{3}$$

By learning to predict the noise for any given noisy input and timestep, the model implicitly learns the distribution of the clean data.

3) Generation Process: Once the noise prediction network $\boldsymbol{\epsilon}_\theta$ is trained, we can use it to generate new synthetic latent vectors. While standard Denoising Diffusion Probabilistic Models (DDPMs) use a slow, stochastic sampling process that requires all T steps, the DDIM formulation allows for a much faster, deterministic generation path. This enables us to generate high-quality samples in a fraction of the steps.

The DDIM generation process starts with a random noise vector, $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and iteratively refines it over a sequence of $N \ll T$ inference timesteps (from $t = T$ down to $t = 1$). At each step, the model performs the following calculations to estimate the latent vector at the previous timestep, $\mathbf{z}_{t-1}$, from the current one, $\mathbf{z}_t$:

1) **Predict Noise:** The trained U-Net predicts the noise component from the current vector: $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, c)$.
2) **Predict Original Data:** An estimate of the original clean data, denoted $\hat{\mathbf{z}}_0$, is derived using the predicted noise:

$$\hat{\mathbf{z}}_0 = \frac{\mathbf{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, c)}{\sqrt{\bar{\alpha}_t}} \tag{4}$$

3) **Compute Direction Pointers:** The model now has two directions: one pointing towards the predicted clean data ($\hat{\mathbf{z}}_0$) and one pointing towards the predicted noise ($\boldsymbol{\epsilon}_\theta$).
4) **Update Step:** The next latent vector, $\mathbf{z}_{t-1}$, is calculated as a linear combination of these two directions, controlled by the noise schedule coefficients:

$$\mathbf{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \cdot (\hat{\mathbf{z}}_0) + \sqrt{1-\bar{\alpha}_{t-1}} \cdot \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, c) \tag{5}$$

This deterministic process (for $\eta = 0$) is repeated for the desired number of inference steps. The final output, $\mathbf{z}_0$, is a synthetically generated latent vector. This vector is then passed through the S-VAE's decoder to reconstruct a full-length, high-fidelity side-channel trace corresponding to the conditioning label c.

IV. Methodology

Current state-of-the-art generative methods in SCA typically apply diffusion models directly to raw power traces [15], [16]. While theoretically sound, this direct adaptation faces a fundamental hurdle in IoT environments: the curse of dimensionality combined with low Signal-to-Noise Ratio (SNR). Raw traces are dominated by non-informative components, forcing the generative model to expend the vast majority of its capacity learning irrelevant distributions. This leads to slow convergence and poor performance on limited datasets.

To address the challenge of data scarcity in side-channel analysis, we introduce a novel three-stage pipeline that synthesizes and validates high-fidelity labeled power traces.

To facilitate reproducibility and clarify the data flow, we explicitly summarize the algorithmic workflow and dimensional transformations at each stage as follows:

1) **Stage 1: Latent Space Formulation (Algorithm 1).** The first stage employs a Supervised Variational Autoencoder (S-VAE) to compress noisy, high-dimensional raw traces ($\mathbf{x} \in \mathbb{R}^{700}$) into a robust, low-dimensional latent representation ($\mathbf{z} \in \mathbb{R}^{32}$). This process effectively filters noise and extracts salient leakage features. Notations are detailed in Table I.
2) **Stage 2: Conditional Diffusion Training (Algorithm 2).** The second stage utilizes a conditional Denoising Diffusion Implicit Model (DDIM). The attention-augmented U-Net (architecture defined in Table II) is trained exclusively on these compact vectors ($\mathbf{z} \in \mathbb{R}^{32}$) to model the conditional distribution $p(\mathbf{z}|c)$, bypassing the complexity of raw signal processing.
3) **Stage 3: Synthetic Generation and Validation (Algorithm 3).** Finally, the trained framework generates synthetic latent vectors which are decoded back to the time domain ($\hat{\mathbf{x}} \in \mathbb{R}^{700}$). These traces are used to train a CNN-based attack model, serving as the ultimate measure of the fidelity and practical utility of our generated data.

The following sections detail each component of this pipeline, from data representation and generation to the final attack-based validation. The entire architecture of our proposed framework is illustrated in Figure 1.

A. Dataset and Latent Space Pre-processing

The foundation of a successful generative model is a representative and challenging dataset. To ensure our framework is robust against common side-channel countermeasures, we utilize the publicly available ASCAD database. A critical aspect of our methodology is evaluating our framework across multiple distinct datasets from this collection to demonstrate its resilience. Specifically, we independently leverage the profiling traces from three distinct versions: ASCAD_sync, ASCAD_desync50, and ASCAD_desync100. A detailed description of the dataset will be provided in the experimental section. By training and evaluating our model on these diverse conditions independently, we demonstrate that our S-VAE can successfully learn the fundamental leakage features of the cryptographic operation even under severe temporal artifacts, without relying on synchronized data to attack desynchronized targets.

Each raw trace in the datasets consists of $L = 700$ time samples, a window established by the ASCAD creators to comprehensively cover the relevant leakage window for the target cryptographic operation. The labels for our conditional model are derived from the standard target for ASCAD: the output of the third S-box operation. This intermediate value is dependent on the third byte of the plaintext and the secret key ($c = \text{AES\_Sbox}[p_2 \oplus k_2]$), making it an ideal target for profiled attacks.

The initial pre-processing pipeline is as follows. First, the raw traces from all three profiling sets are combined and normalized using a 'StandardScaler' to have a zero mean and unit variance. This step is crucial for stabilizing the training of the neural network. The resulting normalized traces, while cleaner, still reside in a high-dimensional space, burdened with significant noise and redundancy.

To overcome this, we employ our tailored S-VAE architecture as a powerful non-linear pre-processor. The normalized traces from all three datasets are used to train a single S-VAE,

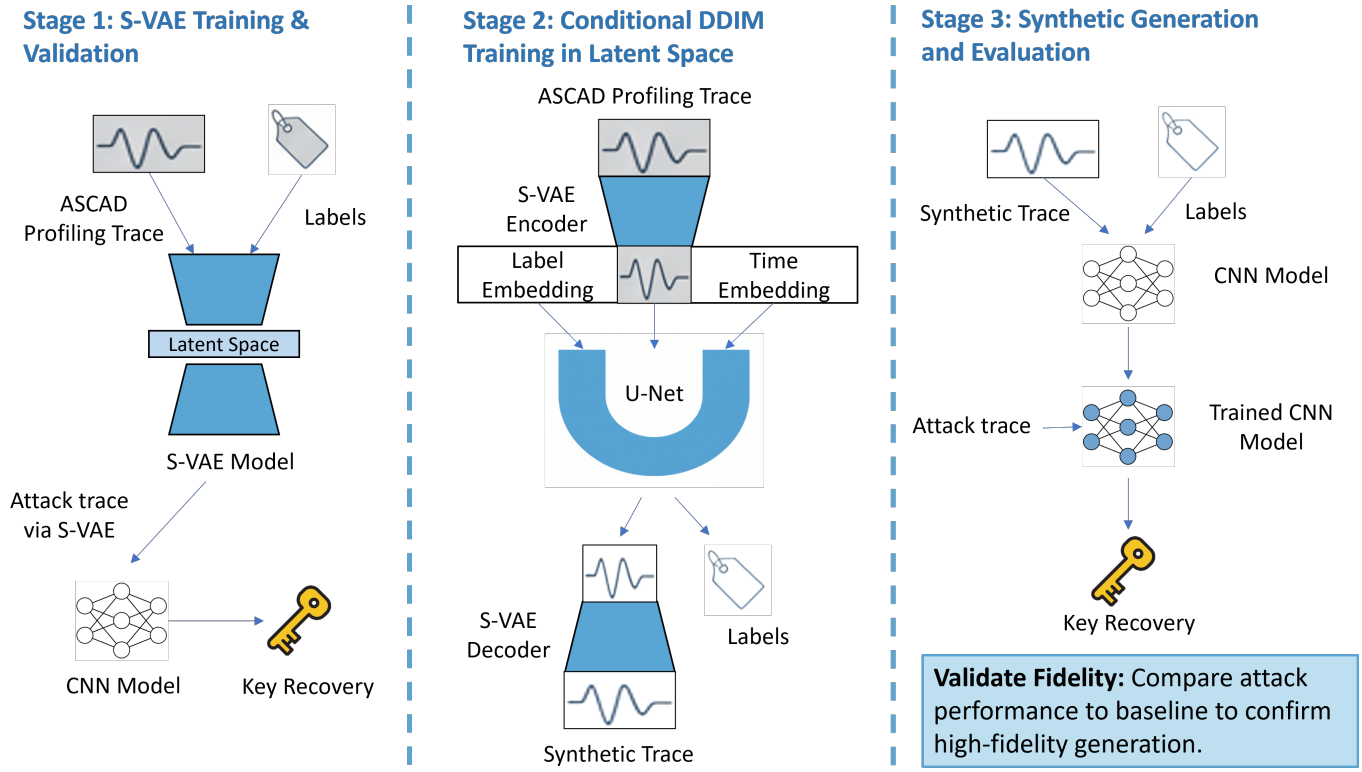


Fig. 1. Overview of the proposed three-stage framework. **Stage 1 (Left):** The S-VAE is trained to compress noisy raw traces into a compact, leakage-aware latent space. **Stage 2 (Center):** A Conditional DDIM, powered by an attention-augmented U-Net, is trained to model the distribution of these latent vectors conditioned on cryptographic labels. **Stage 3 (Right):** Synthetic latent vectors are sampled and decoded back to the time domain to train the CNN attack model. The diagram illustrates the data flow from raw profiling traces to the final key recovery validation.

whose primary role is to perform a probabilistic encoding into a compact latent space. This S-VAE is designed to distill the essential leakage information while aggressively filtering stochastic noise. We empirically determined an optimal latent dimensionality of $D = 32$. The encoder thus performs the mapping:

$$f_{\text{enc}} : \mathbb{R}^{700} \to \mathbb{R}^{32} \tag{6}$$

The output of this stage is a new, unified dataset consisting of low-dimensional latent vectors ($\mathbf{z} \in \mathbb{R}^{32}$) and their corresponding cryptographic labels. This transformed dataset is not only computationally efficient but also represents a denoised and structurally enhanced version of the original data, priming it for the subsequent high-fidelity generation by our conditional DDIM. The complete procedure for training the VAE and formulating this latent space is formally summarized in Algorithm 1.

B. Conditional DDIM for Latent Vector Generation

The core of our generative framework is the conditional diffusion model trained on the S-VAE's latent space. Its function is to approximate the complex conditional distribution $p(\mathbf{z}|c)$ by learning to reverse the diffusion process. The learnable engine that powers this reversal is a sophisticated, attention-augmented U-Net, which we denote as $\boldsymbol{\epsilon}_\theta$. This section details the architecture of this network, our strategy for injecting conditional information, the training objective, and the final conditional generation process.

1) Advanced U-Net Architecture: A standard U-Net is effective due to its symmetric encoder-decoder structure and skip connections, which enable multi-scale feature processing. However, to model the subtle and potentially non-local dependencies inherent in side-channel leakage, we enhance this foundational design with residual connections and self-attention mechanisms. The architecture is specifically designed to operate on 1D sequential data (the latent vectors $\mathbf{z}_t \in \mathbb{R}^{32}$). A comprehensive layer-by-layer breakdown of the full architecture, including all parameters and output shapes, is provided in Table II.

The encoder path is responsible for progressively extracting more abstract, high-level features from the noisy latent input $\mathbf{z}_t$. It consists of a series of downsampling stages. Each stage contains two ResidualBlock modules followed by a strided convolution for downsampling. Each ResidualBlock itself contains two 1D convolutional layers with batch normalization and ReLU activations, and a skip connection that adds the block's input to its output. This residual design is crucial for enabling the training of a deep network by mitigating the vanishing gradient problem.

A key innovation in our U-Net is the integration of a MultiHeadAttentionBlock within the deepest, lowest-resolution layers of the encoder and the bottleneck. While convolutional layers excel at identifying local patterns, their ability to model long-range dependencies is limited by the size of their receptive field. Side-channel leakage, even in a compressed

TABLE II
DETAILED ARCHITECTURE OF THE U-NET MODEL USED FOR NOISE PREDICTION. THE MODEL OPERATES ON A LATENT VECTOR OF LENGTH $L = 32$. WE DENOTE THE BATCH SIZE AS N AND THE BASE CHANNEL COUNT AS $bc = 256$.

Path	Layer / Block	Parameters / Details	Output Shape	Connected From
Input and Conditioning				
-	Input Latent Vector (z_t)	-	$(N, 1, L)$	-
-	Input Projection	1×1 Conv, $1 \to bc$ channels	(N, bc, L)	Input z_t
-	Additive Conditioning	Time & Label Embeddings	(N, bc, L)	Previous Layer
Encoder (Downsampling Path)				
Encoder 1	ResidualBlock $\times 2$	$k = 5$, same padding	(N, bc, L)	Previous Layer
	Downsample	$k = 4, s = 2, p = 1$ Conv1D	$(N, 2bc, L/2)$	Output of ResBlocks (creates `skip1`)
Encoder 2	ResidualBlock $\times 2$	$k = 5$, same padding	$(N, 2bc, L/2)$	Previous Layer
	Downsample	$k = 4, s = 2, p = 1$ Conv1D	$(N, 4bc, L/4)$	Output of ResBlocks (creates `skip2`)
Encoder 3	ResBlock + AttnBlock $\times 2$	$k = 5$, $nh = 4$ heads	$(N, 4bc, L/4)$	Previous Layer
	Downsample	$k = 4, s = 2, p = 1$ Conv1D	$(N, 8bc, L/8)$	Output of AttnBlocks (creates `skip3`)
Bottleneck				
Bottleneck	ResidualBlock	$k = 5$, same padding	$(N, 8bc, L/8)$	Previous Layer
	MultiHeadAttentionBlock	$nh = 4$ heads	$(N, 8bc, L/8)$	Previous Layer
	ResidualBlock	$k = 5$, same padding	$(N, 8bc, L/8)$	Previous Layer
Decoder (Upsampling Path)				
Decoder 1	Upsample	$k = 4, s = 2, p = 1$ ConvTranspose1D	$(N, 4bc, L/4)$	Previous Layer
	Concatenate	Concatenate along channel dim	$(N, 8bc, L/4)$	Previous Layer + `skip3`
	ResBlock + AttnBlock $\times 2$	$k = 5$, $nh = 4$ heads	$(N, 4bc, L/4)$	Previous Layer
Decoder 2	Upsample	$k = 4, s = 2, p = 1$ ConvTranspose1D	$(N, 2bc, L/2)$	Previous Layer
	Concatenate	Concatenate along channel dim	$(N, 4bc, L/2)$	Previous Layer + `skip2`
	ResidualBlock $\times 2$	$k = 5$, same padding	$(N, 2bc, L/2)$	Previous Layer
Decoder 3	Upsample	$k = 4, s = 2, p = 1$ ConvTranspose1D	(N, bc, L)	Previous Layer
	Concatenate	Concatenate along channel dim	$(N, 2bc, L)$	Previous Layer + `skip1`
	ResidualBlock $\times 2$	$k = 5$, same padding	(N, bc, L)	Previous Layer
Output				
-	Final Projection	1×1 Conv, $bc \to 1$ channels	$(N, 1, L)$	Previous Layer
-	Predicted Noise (ϵ_θ)	-	$(N, 1, L)$	Previous Layer

latent space, may contain causally related features that are not spatially adjacent. The self-attention mechanism overcomes this limitation by computing the response at a position in the sequence as a weighted sum of the features at all other positions. This allows the model to capture a global, holistic understanding of the entire latent vector, modeling complex interactions that would be missed by convolutions alone.

The bottleneck of the network sits at the lowest spatial resolution and serves to process the most abstract feature representations, containing further residual and attention blocks. The decoder path then symmetrically mirrors the encoder. It uses transposed convolutions to progressively upsample the feature maps back to the original resolution. At each upsampling stage, the feature maps are concatenated with the corresponding feature maps from the encoder path via long-range skip connections. This is the defining feature of the U-Net, as it allows the network to fuse high-level, semantic context from the decoder with the fine-grained, low-level spatial details preserved by the encoder, leading to a highly accurate noise prediction.

2) Conditional Information Encoding: To guide the generation process, the U-Net must be conditioned on both the diffusion timestep t and the cryptographic label c. We employ a sophisticated embedding strategy to inject this information effectively.

- **Timestep Embedding:** The scalar timestep t is transformed into a high-dimensional vector using a sinusoidal positional embedding, inspired by the Transformer architecture. This allows the network to easily distinguish between different stages of the denoising process.
- **Label Embedding:** The integer label $c \in [0, 255]$ is mapped to a dense embedding vector via a learnable layer.

These two embedding vectors are then passed through separate dense layers and are added to the initial feature representation of the latent vector $\mathbf{z}_t$. By injecting the conditional information at the very beginning of the network, we allow it to influence the feature extraction process at every subsequent layer, ensuring the final noise prediction is strongly conditioned on both t and c.

3) Training Objective and Procedure: The U-Net $\boldsymbol{\epsilon}_\theta$ is trained using a simple yet powerful objective: to minimize the Mean Squared Error (MSE) between the true noise $\boldsymbol{\epsilon}$ added during the forward process and the noise predicted by the network. The training procedure, formalized in Algorithm 2, can be summarized as follows: for each training step, we sample a clean latent vector $\mathbf{z}_0$ and its label c from our S-VAE processed dataset. We then sample a random timestep $t \sim \text{Uniform}(1, \ldots, T)$ and a random noise vector $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We use these to create a single noisy sample $\mathbf{z}_t$ via forward diffusion. This sample, along with t and c, is fed to the U-Net to produce a predicted noise $\hat{\boldsymbol{\epsilon}}$. The parameters θ are then updated by performing a gradient descent step on the MSE loss, $\mathcal{L} = \|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}\|^2$.

Algorithm 1 Supervised VAE (S-VAE) Training for Latent Space Formulation

Require: Profiling dataset $(\mathcal{X}_{\text{profile}}, \mathcal{Y}_{\text{profile}})$, Hyperparameters (learning rate η, epochs E, batch size B, KL weight β, CE classification weight γ).
Ensure: Optimized S-VAE parameters ϕ^* (encoder), θ^* (decoder), and ω^* (classifier head).
Initialize S-VAE parameters ϕ, θ, ω.
$(\mathcal{X}_{\text{train}}, \mathcal{Y}_{\text{train}}) \leftarrow \texttt{Preprocess}(\mathcal{X}_{\text{profile}}, \mathcal{Y}_{\text{profile}})$. ▷ Split data and normalize traces.
Initialize AdamW Optimizer with learning rate η.
for epoch $\leftarrow 1$ to E **do**
  Shuffle $(\mathcal{X}_{\text{train}}, \mathcal{Y}_{\text{train}})$ jointly. ▷ Preserve trace-label correspondence.
  for each batch $(\mathbf{x}_{\text{batch}}, \mathbf{y}_{\text{batch}})$ from $(\mathcal{X}_{\text{train}}, \mathcal{Y}_{\text{train}})$ **do**
    $\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2 \leftarrow f_{\text{enc}}(\mathbf{x}_{\text{batch}}; \phi)$. ▷ Encoder maps traces to a distribution.
    $\mathbf{z} \leftarrow \texttt{Reparameterize}(\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2)$. ▷ Sample latent vector $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$.
    $\mathbf{x}_{\text{recon}} \leftarrow f_{\text{dec}}(\mathbf{z}; \theta)$. ▷ Decoder reconstructs traces from latent vector.
    $\mathbf{logits} \leftarrow f_{\text{class}}(\boldsymbol{\mu}; \omega)$. ▷ Classify using the latent mean vector.
    $\mathcal{L}_{\text{MSE}} \leftarrow \texttt{MSE}(\mathbf{x}_{\text{batch}}, \mathbf{x}_{\text{recon}})$. ▷ Calculate reconstruction error.
    $\mathcal{L}_{\text{KL}} \leftarrow D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_{\text{batch}}) \| p(\mathbf{z}))$. ▷ Calculate KL divergence.
    $\mathcal{L}_{\text{CE}} \leftarrow \texttt{CrossEntropy}(\mathbf{logits}, \mathbf{y}_{\text{batch}})$. ▷ Calculate classification error.
    $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{MSE}} + \beta \cdot \mathcal{L}_{\text{KL}} + \gamma \cdot \mathcal{L}_{\text{CE}}$. ▷ Compute total S-VAE loss.
    $(\phi, \theta, \omega) \leftarrow \texttt{Optimize}(\phi, \theta, \omega, \mathcal{L}_{\text{total}})$. ▷ Update all parameters via backpropagation.
  end for
end for
return $\phi^*, \theta^*, \omega^*$

4) Conditional Generation with DDIM Sampling: Once trained, the U-Net serves as the engine for a DDIM-based conditional sampler. This process allows us to generate a novel latent vector corresponding to any desired label c. The generation begins with a vector of pure Gaussian noise, $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. It then iteratively denoises this vector over a sequence of $N \ll T$ inference steps.

The critical aspect of this process is the role of the conditioning label c. At every single step of the reverse process (from t down to $t-1$), the current noisy vector $\mathbf{z}_t$ and the constant target label c are passed to the U-Net $\boldsymbol{\epsilon}_\theta$. The network's prediction is thus always guided by the target label, ensuring that the denoising trajectory is steered towards a final sample $\mathbf{z}_0$ that belongs to the manifold of class c. The DDIM formulation provides an accelerated sampling scheme to compute $\mathbf{z}_{t-1}$ from $\mathbf{z}_t$ and the guided noise prediction. After N steps, the process yields a synthetic latent vector $\mathbf{z}_0$ which is then passed to the S-VAE decoder to reconstruct a full-length power trace. This guided generation process is detailed in Algorithm 3.

Algorithm 2 Conditional DDIM Training on Latent Space

Require: S-VAE processed dataset $D_z = \{(\mathbf{z}_0, c)\}$, U-Net $\boldsymbol{\epsilon}_\theta$, Training settings (epochs E, steps T, schedule $\bar{\alpha}_t$).
Ensure: Optimized U-Net parameters θ^*.
Initialize U-Net parameters θ.
Initialize AdamW Optimizer and Learning Rate Scheduler.
for epoch $\leftarrow 1$ to E **do**
  Shuffle dataset D_z. ▷ Ensure stochasticity for each epoch.
  for each batch $(\mathbf{z}_{0,\text{batch}}, c_{\text{batch}})$ from D_z **do**
    $t \leftarrow \texttt{UniformSample}(1, \ldots, T)$. ▷ Sample a random timestep for each item in the batch.
    $\boldsymbol{\epsilon} \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$. ▷ Sample Gaussian noise from a standard normal distribution.
    $\mathbf{z}_{t,\text{batch}} \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{z}_{0,\text{batch}} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$. ▷ Create noisy latent vectors.
    $\boldsymbol{\epsilon}_{\text{pred}} \leftarrow \boldsymbol{\epsilon}_\theta(\mathbf{z}_{t,\text{batch}}, c_{\text{batch}}, t)$. ▷ Predict noise using the conditional U-Net.
    $\mathcal{L} \leftarrow \texttt{MSE}(\boldsymbol{\epsilon}, \boldsymbol{\epsilon}_{\text{pred}})$. ▷ Calculate loss based on true and predicted noise.
    $\theta \leftarrow \texttt{OptimizeStep}(\theta, \mathcal{L})$. ▷ Update U-Net parameters via backpropagation.
  end for
end for
return θ^*

Algorithm 3 Conditional Latent Vector Generation

Require: Trained U-Net $\boldsymbol{\epsilon}_{\theta^*}$, target label c, number of inference steps N, noise schedule $\bar{\alpha}_t$.
Ensure: A synthetic latent vector $\mathbf{z}_0$ corresponding to label c.
$\mathbf{z}_T \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$. ▷ Start with a random noise vector.
Let timesteps be $(\tau_i)_{i=1}^N$ be a sequence from T down to 1.
for $i \leftarrow N$ down to 1 **do**
  $t \leftarrow t_i$.
  $t_{\text{prev}} \leftarrow t_{i-1}$ (or 0 if $i = 1$).
  $\hat{\boldsymbol{\epsilon}} \leftarrow \boldsymbol{\epsilon}_{\theta^*}(\mathbf{z}_t, c, t)$. ▷ Predict noise for the current step, conditioned on target label c.
  $\hat{\mathbf{z}}_0 \leftarrow (\mathbf{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}})/\sqrt{\bar{\alpha}_t}$. ▷ Predict the original clean data.
  $\mathbf{z}_{t_{\text{prev}}} \leftarrow \sqrt{\bar{\alpha}_{t_{\text{prev}}}}\hat{\mathbf{z}}_0 + \sqrt{1-\bar{\alpha}_{t_{\text{prev}}}}\hat{\boldsymbol{\epsilon}}$. ▷ Update the vector to the previous timestep.
end for
return $\mathbf{z}_0$

V. Experiments and Results

This section details the empirical validation of our proposed latent diffusion framework. We designed a series of experiments to rigorously assess the quality of the generated synthetic traces and their effectiveness in mounting a successful profiled side-channel attack. The primary objective of our experiments is to demonstrate that our generative model, trained on a limited set of real traces, can produce synthetic

data of sufficient quality to successfully attack a challenging, real-world target protected by desynchronization countermeasures. All experiments were conducted on a high-performance workstation equipped with an Intel Core i9-14900K CPU and a single NVIDIA GeForce RTX 5090 GPU.

To explicitly address the high computational overhead typically associated with diffusion models and to demonstrate the practical relevance of our framework for IoT security evaluations, we briefly quantify our training and sampling costs. By strategically compressing the high-dimensional raw traces into a 32-dimensional latent space via the S-VAE, the overall computational burden is drastically reduced. Specifically, training the S-VAE takes approximately 5 minutes. Operating exclusively within this compact latent space, training the conditional DDIM requires only about 2 hours to fully converge. Furthermore, generating a complete synthetic dataset of $50,000$ traces takes merely 2 hours by exploiting the accelerated sampling capability of DDIM. Compared with a standard DDPM formulation with $T = 16{,}000$ diffusion timesteps, our DDIM uses only $N = 1{,}000$ inference steps, corresponding to an approximate $16\times$ reduction in sampling cost. This substantial reduction in computational burden and hardware requirements highlights the "Less is More" advantage of our latent approach, making high-fidelity synthetic trace generation both computationally tractable and practically accessible.

A. Dataset

To rigorously evaluate the performance of our generative model and ensure a fair comparison with state-of-the-art methods, we conduct our experiments on two distinct hardware platforms.

First, we utilize the ASCAD database [21], which is widely recognized as the standard benchmark for deep learning-based SCA. Unlike proprietary datasets, ASCAD enables reproducible research by providing a common baseline for profiling attacks on an 8-bit AVR microcontroller (ATmega8515) running a masked AES-128 implementation. Its adoption in numerous recent works (e.g., [22], [23]) validates its suitability for benchmarking generative defenses. Second, to further demonstrate the robustness and generalization capability of our work across different hardware architectures, we introduce a dataset collected from the CW308T-XMEGA target board. This board is equipped with an ATXMEGA128D4 processor, an 8-bit microcontroller featuring 8 KB of SRAM and 128 KB of FLASH. The target board is mounted on the ChipWhisperer CW308T UFO platform, a widely used capture hardware that allows for consistent power trace acquisition across various microcontrollers. Our experimental design focuses on four specific scenarios. These scenarios address different levels of desynchronization and hardware variations, directly evaluating the robustness of our model against common countermeasures and architectural differences:

- **Experiment 1: Synchronized Baseline (ASCAD.h5):** This dataset contains the original, perfectly aligned traces. It serves as our baseline condition, representing an ideal, countermeasure-free environment to establish the maximum potential quality of our synthetic data.
- **Experiment 2: Moderate Desynchronization (ASCAD desync50.h5):** This dataset contains traces that have been artificially desynchronized within a random window of up to 50 samples. This represents a common and practical real-world scenario where moderate clock jitter is present.
- **Experiment 3: High Desynchronization (ASCAD desync100.h5):** This dataset contains traces with a more aggressive desynchronization of up to 100 samples. It provides a highly challenging scenario designed to test the upper limits of our model's ability to learn and generate leakage patterns in the presence of strong countermeasures.
- **Experiment 4: Hardware Generalization (CW308T-XMEGA):** This dataset consists of power traces acquired from the ATXMEGA128D4 microcontroller. By extending our evaluation to this distinct 8-bit architecture, we verify that our generative method is not overfitted to the specific leakage characteristics of the ATmega8515 (ASCAD) and remains effective on alternative platforms designated by the ChipWhisperer ecosystem.

For all scenarios, we follow a strict and independent separation of data. The entire generative pipeline is trained exclusively on the profiling set (50,000 traces for ASCAD experiments; equivalent splitting for XMEGA). The quality of the resulting synthetic data is then evaluated by training an attack model and using it to attack the corresponding traces from the separate test set. Each attack set uses a different, unknown key, ensuring a realistic black-box attack evaluation.

The target for both generation and attack remains consistent across all experiments: the third byte of the AES key, following the standard leakage model where the label corresponds to the S-box output: $c = \text{AES\_Sbox}[p_2 \oplus k_2]$. This comparative structure allows us to directly measure the impact of desynchronization and hardware architecture on our model's generative capabilities and the subsequent attack success.

B. Results

This section presents a comprehensive empirical evaluation of our proposed methodology. We designed a structured, three-part experimental study to rigorously assess our contributions. The overarching goal is to demonstrate that our proposed latent diffusion framework is not only viable but also highly effective and robust for generating synthetic side-channel traces, particularly in the presence of desynchronization countermeasures. Our evaluation is structured as a comparative analysis across three distinct experimental setups, each performed on the three ASCAD scenarios (synchronized, desync50, and desync100).

First, we establish a performance baseline, or "upper bound," by training the attack model directly on the real profiling traces. This represents the best-case performance achievable with the given attack architecture. Second, we conduct a critical ablation study to validate the S-VAE stage. We attack using the real traces after they have been encoded into the 32-dimensional latent space. This experiment is designed to answer a key question: does our S-VAE based dimensionality reduction preserve the critical leakage information necessary for a successful attack? Finally, we evaluate our full, end-to-end framework. We train the attack model exclusively on the

synthetic traces generated by our S-VAE conditional DDIM pipeline and attack the real traces. This directly measures the quality and functional utility of our generated data.

1) E1: Baseline Performance on Real Data: To establish an unequivocal benchmark for comparison, we first conducted a direct attack using the original, unmodified ASCAD datasets. This experiment quantifies the maximum achievable performance with our chosen attack architecture, representing the "gold standard" against which any generative model should be measured.

The comprehensive results, quantified by the Number of Traces to Guessing Entropy zero (NTGE), are presented in Table III. This table not only shows the performance of our baseline attack (i.e. Ours-Origin‘) but also contextualizes it by comparing it against several key publications that have previously benchmarked on the ASCAD datasets.

The comprehensive results, quantified by the Number of Traces to Guessing Entropy zero (NTGE), are presented in Table III. This table not only shows the performance of our baseline attack (i.e., Ours-Origin) but also contextualizes it by comparing it against several state of the art methods that have previously been benchmarked on the ASCAD datasets.

As shown in the results, our baseline attack is highly effective across all scenarios. To explicitly address the inherent stochasticity of deep learning training and to ensure a rigorous evaluation, we conducted three experimental runs for each configuration. We report the best-performing results achieved across these runs to highlight the optimal capability of our trained models. To validate the robustness of our experimental setup, we analyze our performance in relation to prior works. In the ideal, countermeasure-free scenario, our attack model recovers the key with only 69 traces (best of 69, 85, 95). This performance is highly competitive and aligns with the state-of-the-art. It is on par with the results from Zaid et al. [23] (191 traces) and Libang et al. [24] (202 traces), confirming that our chosen CNN architecture and training regimen are correctly implemented and highly effective. Furthermore, it represents a significant improvement over the initial benchmark reported by Prouff et al. [21], which required over 1000 traces, highlighting the advancements in deep learning-based attacks.

The true test of an attack model's robustness is its performance against countermeasures. The robustness of our model becomes particularly evident on the ASCAD_desync50 dataset, where our attack requires only 101 traces for key recovery (best of 101, 118, 199) . This result demonstrates a substantial improvement in efficiency, significantly outperforming the 193 traces reported by Libang et al. [24] and proving to be more than twice as efficient as the 244 traces required by Zaid et al. [23]. This further validates that our baseline is performing at the expected state-of-the-art level.

Most notably, on the highly challenging ASCAD_desync100 dataset, our attack model outperforms previously published results, requiring only 229 traces for key recovery (best of 229, 306, 386). This is a marked improvement over the 270 traces required by Zaid et al. [23] and the 231 traces required by Libang et al. [24]. This superior performance on the most difficult dataset suggests that our implementation is particularly well-suited for handling high levels of desynchronization. In contrast, the original analysis by Prouff et al. [21] failed to recover the key within 5000 traces on either desynchronized dataset, underscoring the formidable challenge posed by this countermeasure.

2) E2: Ablation Study - Viability of the VAE Latent Space: A cornerstone of our methodology is the strategic compression of high-dimensional traces into a compact latent space. This ablation study critically evaluates a central hypothesis: can a 32-dimensional latent space, learned by our S-VAE, effectively preserve the salient leakage features required for a successful attack? This experiment is designed to quantify the information fidelity of our S-VAE stage and justify its role as a foundational component for the subsequent diffusion model.

To test this, we conducted an attack directly within the latent space. For each of the three scenarios, the real profiling and attack traces were first passed through the pre-trained VAE encoder to obtain their 32-dimensional latent vector representations. A CNN attack model was then trained on the latent vectors from the profiling set and used to attack the latent vectors from the corresponding attack set. The quantitative results of this experiment are presented in the Ours-S-VAE column of Table III, and we analyze them in detail below. Consistent with our baseline evaluations, each experiment in this ablation study was repeated three times to account for stochastic variance. We report the best-performing result across these runs to highlight the optimal capability of the latent space representation.

In the synchronized scenario, we observe that the attack on the S-VAE's latent space requires only 74 traces to recover the key (best of 74, 128, 209). This performance is remarkably close to the baseline attack on the original data 69 traces), representing a negligible performance trade-off. By learning to reconstruct the principal components of the signal, the S-VAE effectively filters out stochastic, high-frequency noise present in the raw traces, creating a "cleaner" and more information-dense feature space. As expected, for the more challenging desynchronized scenarios, a performance trade-off is observed. On ASCAD_desync50, the number of traces required increases from 101 to 248 (best of 248, 251, 367). On ASCAD_desync100, the number increases from 200 to 335 (best of 335, 338, 520). This degradation is an inevitable consequence of the aggressive dimensionality reduction. The desynchronization introduces significant temporal variance, and it is plausible that some subtle timing-related leakage information is lost during the S-VAE's encoding-decoding process. However, it is crucial to note that even with this information loss, the key is still recovered with a very reasonable number of traces.

This experiment provides compelling evidence for our core hypothesis. The attack on the S-VAE-based latent space remains highly effective across all scenarios even when factoring in experimental variance, confirming that the vast majority of critical leakage information is preserved with high fidelity. The slight performance degradation in desynchronized cases is a well-justified and acceptable trade-off for the immense practical benefits a 32-dimensional representation offers. By operating in this compact space, the subsequent DDIM training becomes computationally tractable, requiring significantly less

TABLE III
COMPARISON OF NTGE WITH STATE-OF-THE-ART METHODS (ALL ON ASCAD PLATFORM)

Attack Data Source	[21]	[24]	[23]	[25]-CGAN	[15]-DDPM	Ours-Origin	Ours-S-VAE	Ours-DDIM
ASCAD_sync	1146	202	191	191	1531	69	74	88
ASCAD_desync50	5000+	193	244	-	5730	101	248	694
ASCAD_desync100	5000+	231	270	-	-	229	355	4107

memory and time.

3) E3: Attack on Synthetically Generated Data: This experiment evaluates the performance of our method. The objective is to demonstrate that an attack model trained exclusively on data produced by our S-VAE conditional DDIM framework can successfully compromise a real-world target and to benchmark this performance against both the real-data baseline and other state-of-the-art generative models. For each of the three ASCAD scenarios, we first trained our conditional latent diffusion model on the corresponding S-VAE processed profiling set. We then used this model to generate a new, fully synthetic dataset of 50,000 labeled traces. The performance of our proposed method is presented in the Ours-DDIM column of Table III. To rigorously validate the stability of our generative framework and account for the stochastic nature of diffusion models, we executed three runs for each scenario and report the optimal attack performance alongside the distribution of results.

Our framework successfully recovers the secret key in all three scenarios, requiring 88 (best of 88, 238, 488) traces for ASCAD_sync, 694 (best of 694, 756, 953) for ASCAD_desync50, and 4107 (best of 4107, 4119, 4211) for ASCAD_desync100. When compared to our own real-data baseline in Ours-Origin column, an expected performance gap exists, as any generative process entails some degree of information loss. However, this gap is remarkably narrow in the synchronized scenario (i.e. 88 vs. 69 traces). While the gap widens for the desynchronized cases, the key is still recovered efficiently. This confirms that our framework generates synthetic data that is not merely statistically plausible but functionally potent, retaining the critical leakage characteristics necessary for a successful attack.

Compared to contemporary GAN-based methods, our model demonstrates superior efficiency; on the synchronized dataset, our method requires only 88 traces, making it more than twice as effective as the CGAN-based approach of Wang et al. [25] (191 traces). However, a more stark and compelling contrast is seen when comparing our results to the DDPM-based method of Yap et al. [15], which operates directly on raw traces. On ASCAD_sync, our model is approximately 17 times more efficient , and this performance gap widens dramatically on the ASCAD_desync50 dataset, where our model is over 7.5 times more efficient. This vast improvement provides a direct and powerful validation of our core hypothesis: by training the diffusion model on a compact, denoised 32-D latent space instead of the noisy, high-dimensional 700-D raw trace space, we achieve a monumental gain in both generative quality and attack efficacy.

Crucially, our work represents, to the best of our knowledge, the first successful demonstration of a generative model producing synthetic data capable of compromising the highly challenging ASCAD_desync100 target. Our method successfully recovers the key with 4107 traces. This breakthrough is a direct testament to the exceptional robustness of our S-VAE Conditional DDIM pipeline. The ability to generate functional attack traces from a source afflicted by such severe misalignment firmly establishes our framework as a state-of-the-art solution for security evaluation against strong, real-world countermeasures.

4) E4: Comparative Analysis and Data Efficiency: To validate the superiority of our framework, we extend our evaluation beyond internal consistency checks to a direct comparison with recent state-of-the-art (SOTA) methods. As in our previous experiments, we account for the stochastic nature of model training by conducting three runs and reporting the highest achieved accuracy. To rigorously evaluate the data efficiency of our framework, we compare the classification accuracy and the required profiling set size against these methods in Table IV.

As detailed in the table, traditional deep learning-based attacks and heavy generative models often demand extensive datasets to converge. For instance, while the DDPM-based approach by Karayal et al. [16] achieves near-perfect accuracy (99.60%), it requires a massive profiling set of 63,750 traces. Similarly, standard CNNs and metric-learning approaches like TripletPower [26] show significant performance degradation when the training data is reduced (e.g., CNN accuracy drops to 57.56% at 4,000 traces).

In contrast, our method achieves a peak accuracy of 97.7% (best of 97.7%, 96.5%, 94.4%) using only 2,560 synthetic traces generated via DDIM. This result highlights the "Less is More" advantage: by effectively learning the leakage manifold within a compact latent space, our model generates highly informative training samples. These synthetic traces capture the essential leakage features more efficiently than larger datasets used in traditional raw-trace or GAN-based approaches [27], allowing for rapid model convergence with minimal profiling effort.

C. Hyperparameter settings and analysis

To ensure the reproducibility of our results and to provide full transparency into our experimental design, this section details the complete set of hyperparameters used for each stage of our framework. The selected values are based on a combination of established best practices from the relevant literature and empirical tuning on a small validation set. A consolidated list of all settings is presented in Table V.

TABLE IV
COMPARISON OF ACCURACY WITH STATE-OF-THE-ART METHODS (ALL ON CHIPWHISPERER PLATFORM)

Work	Method / Model	Training Traces	Accuracy
[27]	CGAN	4,500	91.25%
[26]	CNN	16,000	98.26%
	TripletPower	16,000	98.32%
	CNN	4,000	57.56%
	TripletPower	4,000	97.02%
[28]	MLP/CNN	5,000	97.00%
	CNN+LSTM	5,000	96.00%
This work	DDIM	2560	97.7%

While many of the chosen hyperparameters, such as the learning rates and batch sizes, are standard for their respective tasks, several decisions are critical to the success of our framework.

For the S-VAE, the most impactful hyperparameter is the KL weight, $\beta = 4.0$. A standard VAE ($\beta = 1.0$) prioritizes perfect reconstruction, which can lead to overfitting on noise. By increasing β, we place a stronger regularization penalty on the latent space, forcing the encoder to learn a more generalized and disentangled representation. This is crucial for its function as a denoiser and feature purifier, as validated in our ablation study.

For the DDIM, the architectural choices are paramount. The choice of 1000 epochs reflects the significant complexity of learning the conditional distribution of the latent space. The U-Net's capacity, governed by the base channel count of 256 and 4 attention heads, was determined to be a robust configuration capable of modeling the intricate leakage patterns without excessive computational overhead. The large number of total diffusion timesteps ($T = 16,000$) provides a very smooth noising process, which aids the model in learning a precise denoising function. This high-fidelity learning is then leveraged by the DDIM sampler, which can produce high-quality samples in a much smaller number of inference steps ($N = 1000$), providing a practical balance between training fidelity and generation speed.

TABLE V
CONSOLIDATED LIST OF HYPERPARAMETERS USED FOR ALL STAGES OF THE EXPERIMENTAL PIPELINE.

Stage	Parameter	Value
Stage 1: Supervised Variational Autoencoder (S-VAE)		
	Optimizer	AdamW
	Learning Rate	1×10^{-4}
	Epochs	75
	Batch Size	256
	Latent Dimension (D_z)	32
	KL Weight (β)	4.0
	CE Weight (γ)	150.0
Stage 2: Conditional Denoising Diffusion Implicit Model (DDIM)		
	Optimizer	AdamW
	Learning Rate	1×10^{-4}
	Epochs	1,000
	Batch Size	256
	Total Timesteps (T)	16,000
	Inference Steps (N)	1,000
	U-Net Base Channels (bc)	256
	U-Net Attention Heads (nh)	4
Stage 3: CNN Attack Model		
	Optimizer	AdamW
	Learning Rate	1×10^{-4}
	Epochs	75
	Batch Size	256

D. Discussion

The empirical results presented in the previous section robustly validate our three-stage generative framework. However, the raw performance metrics only tell part of the story. This section provides a deeper interpretation of our findings, reflects on the underlying mechanics of our approach, discusses its broader implications for IoT security, and outlines the limitations and promising avenues for future research.

1) The Synergy of S-VAE and Conditional DDIM: A Division of Labor: Our central hypothesis was that a three-stage approach would outperform a monolithic model operating on raw data. The results provide a resounding confirmation. The success of our framework can be attributed to an effective "division of labor" between the S-VAE and the Conditional DDIM, a synergy that is key to its performance.

The S-VAE's role is far more significant than mere dimensionality reduction. As demonstrated in our ablation study (E2), it acts as a powerful desynchronization filter and feature purifier. By learning a low-dimensional representation, the S-VAE is forced to discard high-frequency, stochastic noise and focus on the principal, invariant components of the signal—the leakage itself. The surprising performance improvement on the synchronized dataset (74 vs 167 traces) is a testament to this denoising capability. For desynchronized traces, it effectively creates a more stable, temporally consistent manifold for the subsequent model to learn from.

With the S-VAE handling the difficult task of cleaning and structuring the data, the conditional DDIM is freed to do what it does best: high-fidelity distributional learning. Instead of wasting its vast capacity on modeling noise and jitter in a 700-dimensional space, it can focus its full expressive power on the much simpler, well-behaved 32-dimensional latent space. This explains the monumental performance gap observed between our method and the raw DDPM approach of Yap et al. [15].

Our framework does not just generate traces; it first learns what is important in a trace, and then generates from that purified knowledge.

2) Implications for IoT Security and Side-Channel Analysis: The practical implications of this data-efficient generative approach are significant, particularly for the IoT ecosystem.

- **Democratizing Security Evaluation:** The primary barrier to advanced SCA in IoT is the acquisition of large-scale datasets. Our method dramatically lowers this barrier. Security analysts can now leverage a small, easily obtainable set of traces to generate a vast, high-quality synthetic dataset, enabling thorough, deep learning-based vulnerability assessments that were previously impractical.
- **Enabling Proactive Design:** Chip designers can use this framework to evaluate the effectiveness of countermeasures before fabrication. By capturing a limited number of traces from a prototype, they can generate a massive dataset to simulate and quantify the resilience of their designs against state-of-the-art attacks, accelerating the design-test-iterate cycle.
- **A New Paradigm for Trace Sharing:** Instead of sharing large, cumbersome datasets, research institutions or certification labs could share highly compact, pre-trained generative models. This would allow for reproducible research and standardized testing without the logistical overhead of transferring terabytes of data.

3) Convergence and Stability Analysis: A key concern in generative modeling is training stability. Unlike GANs, which rely on finding a Nash equilibrium in a min-max game and often suffer from mode collapse or oscillation, our Latent Diffusion approach benefits from a stable optimization landscape. The training objective is a weighted regression problem, which is inherently more stable. Empirically, we observed consistent convergence of the validation loss across all experimental scenarios (synchronized and desynchronized). While the individual optimization objectives (such as the ELBO) are theoretically sound, our results provide strong empirical evidence that the joint pipeline exhibits reliable behavior without requiring the delicate hyperparameter tuning typical of adversarial methods.

Finally, we acknowledge the inherent stochasticity involved in training deep learning models, particularly complex generative architectures such as diffusion models. Variations in random weight initialization and the stochastic nature of the diffusion sampling process can introduce slight variances in the final generative quality and the resulting Side-Channel Analysis (SCA) metrics (e.g., NTGE and accuracy). Due to the substantial computational resources required to fully train these models across multiple large-scale datasets, conducting numerous independent runs to report robust statistical means and standard deviations was computationally prohibitive. Consequently, the results reported in Table III and Table IV represent the performance of a best-effort converging model.

4) Discussion on Generalizability: While our empirical evaluation focuses on the widely adopted ASCAD datasets and the ChipWhisperer platform (targeting AES implementations), the fundamental architecture of our latent diffusion framework exhibits strong potential for broader generalizability. First, our framework is inherently data-driven and algorithm-agnostic. The S-VAE and DDIM components learn the underlying statistical distribution of time-series side-channel leakages rather than algorithm-specific cryptographic structures. Consequently, by adapting the conditional labels (e.g., switching from Identity leakage to Hamming Weight or Hamming Distance), our methodology can be naturally extended to other symmetric algorithms, as well as asymmetric ciphers (e.g., RSA, ECC) and emerging Post-Quantum Cryptography (PQC) schemes. Second, the robust 32-D latent space projection is specifically designed to filter out stochastic noise and temporal misalignments. This architectural advantage suggests that our approach can maintain its efficacy across traces collected from diverse hardware architectures, such as FPGAs, secure smart cards, and various IoT microcontrollers. Exploring the application of this latent generative framework to a wider array of cryptographic settings and device types remains an exciting direction for future work.

VI. Conclusion and Future Work

In conclusion, this work presents more than just a new method for data augmentation; it offers a strategic shift from data-hungry analysis to data-efficient, model-driven security evaluation. By intelligently combining representation learning via a S-VAE with high-fidelity generative modeling using a DDIM, we have empirically demonstrated a framework capable of producing synthetic side-channel traces of exceptional quality, leading to successful key recovery even on highly desynchronized targets. While the optimization objectives of the individual components are theoretically grounded, our results provide robust empirical validation of the joint pipeline's stability and convergence. While our framework demonstrates significant promise, its limitations highlight several exciting avenues for future investigation. A crucial next step is to assess the framework's generalizability across diverse cryptographic algorithms (e.g., ECC, post-quantum cryptosystems) and different hardware platforms, which may exhibit vastly different leakage characteristics. Furthermore, while generation is accelerated, the initial training of diffusion models remains computationally intensive; future research could explore more efficient training paradigms, such as consistency models, to further enhance practicality. Additionally, a systematic study on the sensitivity and automated optimization of key hyperparameters, such as the latent space dimensionality and the β value, would be a valuable contribution. Finally, the structured latent space learned by the S-VAE presents a research opportunity in its own right, with potential for novel attack vectors or for visualizing the impact of physical countermeasures. By addressing these future challenges, we believe this approach of modeling compact, purified feature spaces provides a robust and practical solution to one of the most pressing challenges in the physical security assessment of modern cryptographic systems.

References

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology — CRYPTO' 99*, M. Wiener, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397.

[2] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Cryptographic Hardware and Embedded Systems — CHES 2001*, Ç. K. Koç, D. Naccache, and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 251–261.

[3] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2002*, B. S. Kaliski, ç. K. Koç, and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 13–28.

[4] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure dpa resistant asic or fpga implementation," in *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, vol. 1, 2004, pp. 246–251 Vol.1.

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[7] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *9th International Conference on Learning Representations, ICLR*, 2021.

[8] E. Cagli, C. Dumas, and E. Prouff, "Convolutional neural networks with data augmentation against jitter-based countermeasures," in *Cryptographic Hardware and Embedded Systems – CHES 2017*, 08 2017, pp. 45–68.

[9] G. Zaid, L. Bossuet, A. Habrard, and A. Venelli, "Methodology for efficient cnn architectures in profiling attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 1, p. 1–36, Nov. 2019. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/8391

[10] D. Fan, R. Meng, X. Xu, Y. Liu, G. Nan, C. Feng, S. Han, S. Gao, B. Xu, D. Niyato, T. Q. S. Quek, and P. Zhang, "Generative diffusion models for wireless networks: Fundamental, architecture, and state-of-the-art," 2025. [Online]. Available: https://arxiv.org/abs/2507.16733

[11] R. Meng, S. Gao, D. Fan, H. Gao, Y. Wang, X. Xu, B. Wang, S. Lv, Z. Zhang, M. Sun, S. Han, C. Dong, X. Tao, and P. Zhang, "A survey of secure semantic communications," 2025. [Online]. Available: https://arxiv.org/abs/2501.00842

[12] R. Meng, X. Cheng, S. Gao, X. Xu, C. Dong, G. Nan, X. Tao, P. Zhang, and T. Q. S. Quek, "Generative ai for physical-layer authentication," 2025. [Online]. Available: https://arxiv.org/abs/2504.18175

[13] S. Karayalçın, M. Krček, L. Wu, S. Picek, and G. Perin, "It's a kind of magic: A novel conditional gan framework for efficient profiling side-channel analysis," in *Advances in Cryptology – ASIACRYPT 2024*, K.-M. Chung and Y. Sasaki, Eds. Singapore: Springer Nature Singapore, 2025, pp. 99–131.

[14] N. Mukhtar, L. Batina, S. Picek, and Y. Kong, "Fake it till you make it: Data augmentation using generative adversarial networks for all the crypto you need on small devices," in *Topics in Cryptology – CT-RSA 2022*, S. D. Galbraith, Ed. Cham: Springer International Publishing, 2022, pp. 297–321.

[15] T. Yap and D. Jap, "Creating from noise: Trace generations using diffusion model for side-channel attack," in *Applied Cryptography and Network Security Workshops*, M. Andreoni, Ed. Cham: Springer Nature Switzerland, 2024, pp. 102–120.

[16] S. Karayalcin, G. Perin, and S. Picek, "Diffuse some noise: Diffusion models for measurement noise removal in side-channel analysis," *Selected Areas in Cryptography*, 2025.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[18] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014. [Online]. Available: http://arxiv.org/abs/1312.6114

[19] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 3581–3589.

[20] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: https://openreview.net/forum?id=St1giarCHLP

[21] E. Prouff, R. Strullu, R. Benadjila, E. Cagli, and C. Canovas, "Study of deep learning techniques for side-channel analysis and introduction to ascad database," *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 53, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:41991837

[22] A. Golder, D. Das, J. Danial, S. Ghosh, S. Sen, and A. Raychowdhury, "Practical approaches toward deep-learning-based cross-device power side-channel attack," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 12, pp. 2720–2733, 2019.

[23] G. Zaid, L. Bossuet, A. Habrard, and A. Venelli, "Methodology for efficient cnn architectures in profiling attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 1, p. 1–36, Nov. 2019. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/8391

[24] L. Zhang, X. Xing, J. Fan, Z. Wang, and S. Wang, "Multi-label deep learning based side channel attack," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. PP, pp. 1–1, 10 2020.

[25] H. Wang, T. Feng, and C. Liu, "Wavelet coefficients based generative adversarial networks for side-channel attack preprocessing," *Signal, Image and Video Processing*, vol. 19, 06 2025.

[26] C. Wang, J. Dani, S. Reilly, A. Brownfield, B. Wang, and J. M. Emmert, "Tripletpower: Deep-learning side-channel attacks over few traces," in *2023 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2023, pp. 167–178.

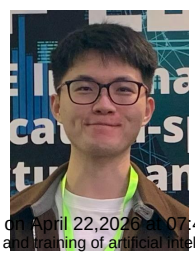
[27] W. Wan, W. Jun-Nian, H. Fan-Liang, and N. Feng, "Sca-cgan: A new side-channel attack method for imbalanced small samples," *Radioengineering*, vol. 32, no. 1, p. 125, 2023.

[28] W. MingDeng, Y. YingJian, and G. PengFei, "Power analysis based on deep learning for multiple implementations of cryptographic algorithms," in *2023 IEEE 17th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, 2023, pp. 82–86.

Zekai Zhang received the First-Class Honors B.Eng. degree in Electrical and Electronic Engineering from University College London (UCL), in 2022. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, City University of Hong Kong. His research interests include side-channel analysis, hardware security, and deep learning.

Donglong Chen received the PhD degree from the Department of Electronic Engineering, City University of Hong Kong, in 2015. He was a visiting research scholar of COSIC, KU Leuven, Belgium, in 2013. After completing his PhD degree study, he spent four years with the industry including Huawei Technology Co., Ltd. and Tencent Technology Co., Ltd. He is currently an associate professor with the Faculty of Science and Technology, Beijing Normal-Hong Kong Baptist University, China. His research interests include cryptographic engineering, software/hardware co-design for algorithms, and privacy computing.

Wangchen Dai received the Ph.D. degree in electronic engineering from the City University of Hong Kong in 2018. After that, he had appointments at Hardware Security Lab, Huawei Technologies Company Ltd., and the Department of CSSE, Shenzhen University. He is currently working as a Senior Researcher with Zhejiang Lab, Hangzhou, China. His research interests include cryptographic hardware and embedded systems, fully homomorphic encryption, and reconfigurable computing.



Jinfa Hong received the B.S. degree in Electronic Engineering from Fuzhou University, Fuzhou, China, in 2022, and the M.S. degree in Electrical Engineering from the City University of Hong Kong, Hong Kong, in 2023. His research focuses on edge AI hardware design, with an emphasis on heterogeneous computing and operator optimization by algorithm-hardware co-design methodology.

Yu Hin Chan received his B.Eng. degree in Electronic and Communication Engineering from City University of Hong Kong in 2022, and his M.Sc. degree in Computing from Cardiff University, UK, in 2024. He is currently a Research Assistant at the Department of Electrical Engineering, City University of Hong Kong. His research interests include hardware-software co-design, secure communication protocols for RISC-V systems, and cryptography.

Çetin Kaya Koç (Fellow, IEEE) earned his B.S. in Electrical Engineering (summa cum laude) from İstanbul Technical University in 1980 and went on to complete his Ph.D. in Electrical and Computer Engineering at the University of California, Santa Barbara in 1988. His research spans a variety of areas, including cryptographic engineering, random number generation, finite field and ring arithmetic, homomorphic encryption, and machine learning. Dr. Koç holds positions in multiple organizations and leads a research group known as Koç Lab, where his work is supported by various funding institutions. The lab comprises postdoctoral researchers, Ph.D. and M.S. candidates, along with advanced undergraduate students, all collaborating on cutting-edge research in cryptography.

Patrick S. Y. Hung received his B.Sc. in Electrical Engineering from the University of Hong Kong and his M.Sc. and Ph.D. in Electrical Engineering from Stanford University. His research interests span computer architecture and computer arithmetic. He was awarded the CBI Overseas Scholarship in the United Kingdom and the Taishan Scholar Award in China. Dr. Hung previously served as a Consulting Assistant Professor at Stanford University and is currently an Adjunct Professor at the City University of Hong Kong.

Ray C. C. Cheung received the Ph.D. degrees in computing from Imperial College London, London, U.K., in 2007. He conducted his postdoctoral research work with UCLA and his visiting fellowship with Princeton University. He is a Professor with the Department of Electrical Engineering, and an Associate Provost (Digital Learning), CityUHK. His current research interests include cryptographic processor designs and embedded system designs. He is now an Associate Editor of the Journal of Cryptographic Engineering, Springer, and Frontiers in High Performance Computing. He served as the Technical Chair of FPT'02, the General Chair/Co-Chair of ARC'12, FPT'22, and ASAP'24. He is currently the Chair of the IEEE Hong Kong Section.