Diffuse Some Noise: Diffusion Models for Measurement Noise Removal in Side-channel Analysis

Sengim Karayalçin, Guilherme Perin, **Stjepan Picek**

Selected Areas in Cryptography, 13.08.2025

Introduction

- Side-channel attacks (SCAs) represent a powerful attack category on crypto devices.
- We commonly divide SCAs into direct and profiling attacks.
- Deep Learning-based SCA (DLSCA) represents the most powerful category of profiling SCAs.
- Their main advantage is that they require little pre-processing/feature engineering effort and can break protected targets with a small number of attack traces.
- But, the situation is not so simple.

Introduction

- While DLSCA can break protected targets (mostly considered Boolean masking and desynchronization), it does not mean the attack is not more difficult.
- As such, any improvement in reducing the difficulty of attacking the target is important.
- One countermeasure (or environmental effect) that did not receive much attention is the Gaussian noise.

Motivation

- Can we reduce the effect of Gaussian noise (improve SNR)?
- A traditional approach would be to add more measurements.
- A deep learning approach may be to use a denoising autoencoder.
- But there are limitations - these approaches generally require considerable expertise to be effectively employed or necessitate the ability of the attacker to capture a 'clean' set of traces without the noise.

Goal

- We propose a novel approach to denoise traces based on Denoising Diffusion Probabilistic Models (DDPMs).
- Using these models, we can effectively remove environmental (Gaussian) noise from side-channel traces without requiring a reference set of 'clean' traces or profiling labels.
- We experimentally validate our approach against several datasets and show improved attack performance for non-profiled collision attacks, non-profiled attacks using deep learning, higher order correlation power analysis (HO-CPA), and horizontal attacks.

Generative vs. Discriminative

- Machine learning algorithms can be divided into two categories: generative and discriminative.
- The goal for discriminative algorithms is to simulate the conditional probability distribution of the output labels given the input features and understand the decision boundary.
- Generative algorithms are designed to simulate the joint probability distribution of the input features (possibly conditioned on labels).
- To create new samples, their goal is to learn the underlying data distribution.
- Template attack is generative!

Algorithmic Noise vs. Measurement Noise

- We consider algorithmic noise to be the parts of the computation that are happening in parallel with the intermediate values we target.
- Measurement noise is the part of the trace that is due to taking the physical measurements.
- We generally assume this noise follows or is similar to, a Gaussian distribution.
- The main difference between these types of noise for the purposes of unsupervised pre-processing of side-channel traces is that the algorithmic noise is part of the signal and is, therefore, not removed.

Denoising Diffusion Probabilistic Models (DDPMs)

- DDPM training is based on a relatively straightforward paradigm: during training, we iteratively add some noise to an image (or some other type of data) for T steps; this is referred to as the forward process.
- Then, for an image x_t where noise has been added t times, we train the model to predict x_{t-1} and thereby remove noise.
- This is called the backward process.
- The central idea here is that when we start from fully random noise and iteratively remove noise, we can generate realistic-looking images as the diffusion models try to 'amplify' patterns in the noise.

Denoising Diffusion Probabilistic Models (DDPMs)

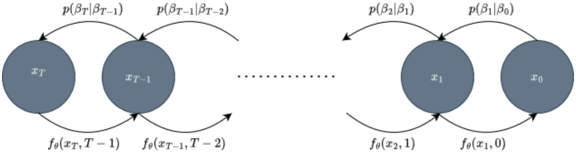


Figure: Diagram illustrating the forward and backward process for training DDPMs.

Approach

- The key idea here is to take a diffusion model parameterized with θ, $f_\theta : X^m \times T \mapsto X^m$, where X^m is a side-channel trace with m samples and $T = \mathbf{Z}_n$ that we train using standard diffusion model training on our measured traces.
- After training, we then input actual traces (or x_0) and try to remove noise (or predict x_{-1}) from these traces.
- This then results in every original trace being transformed into a denoised version.

Approach

- To keep the focus of this work on the viability of DDPMs for denoising traces in an unsupervised context, we only use synchronized traces.
- This allows us to restrict our architecture to shallow MLPs as these have been shown to be effective for processing synchronized side-channel traces.

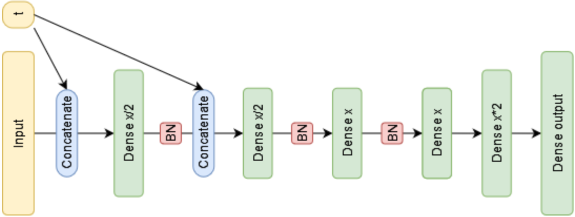
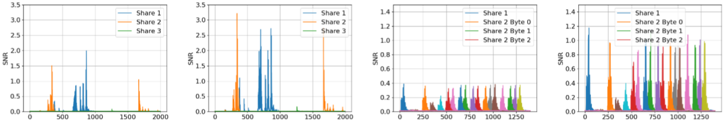


Figure: General model architecture for the input of size X.

Proof-of-concept

- We consider the ASCADv2 target where the leakage of the masked output is noisy (SNR around 0.08) and the ESHARD target provides measurements of a software implementation where both the mask and masked Sbox output leak with relatively low SNRs.



(a) Original (b) Diffused (c) Original (d) Diffused

Figure: SNR values for secret shares for ASCADv2(left) and ESHARD(right).

Proof-of-concept

- The results suggest that diffusion models learn to differentiate the side-channel signal from noise by looking for correlated features in the trace.
- By finding and combining information from those related points, the model can decrease the error in its output.
- This is relevant for real-world side-channel traces when we take several measurements during an operation that leaks some sensitive value, e.g., the oscilloscope has a high sampling rate or some sensitive value is manipulated in several trace points.

Multi Output Regression Enhanced (MORE)

- The basic idea of this attack is to train one model labeled for every possible key and conduct the regression task.
- As the labels generated using the correct key are the only ones that are related to the trace, the model should then most accurately predict labels of the correct key.

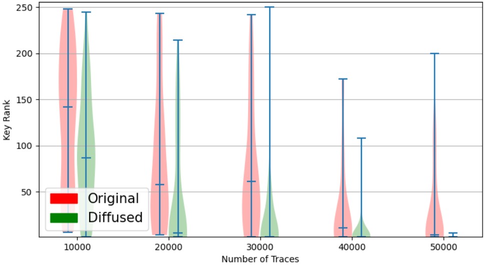


Figure: MORE results for ESHARD.

Collision Attack against ASCADv2

- Collision attacks aim to recover the bitwise difference between sub-keys (key-deltas).
- These key-deltas can then be used to brute-force one key byte, leading to full key recovery (given correct key-deltas).

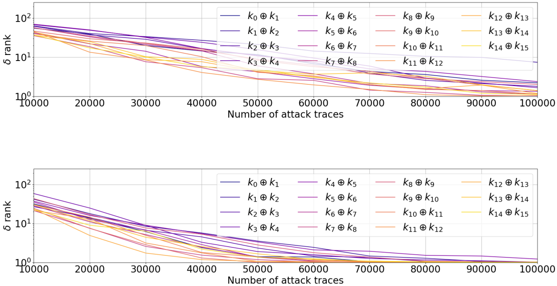
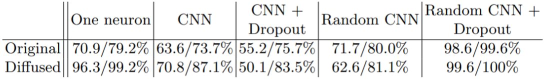


Figure: ASCADv2 collision attacks for Original (top) and Diffused (bottom) traces.

Horizontal Attacks against Public Key Implementation

- The horizontal attack that targets individual bits of the ECC key by classifying trace segments.
- In this attack, initial labeling that is only slightly better than random guessing (around 52%) is iteratively improved upon using CNNs.



	One neuron	CNN	CNN + Dropout	Random CNN	Random CNN + Dropout
Original	70.9/79.2%	63.6/73.7%	55.2/75.7%	71.7/80.0%	98.6/99.6%
Diffused	96.3/99.2%	70.8/87.1%	50.1/83.5%	62.6/81.1%	99.6/100%

Figure: Comparison of Average/Max single trace accuracy for key bits using the one neuron perceptron and CNN setups.

Correlation Power Analysis

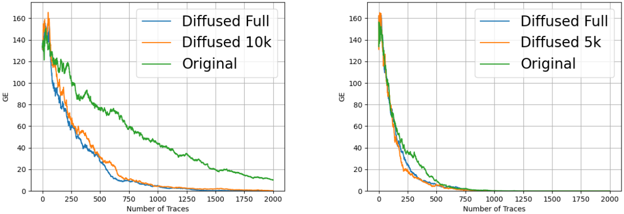


Figure: CPA results for ESHARD(left) and ASCAD(right). Diffused 5k/10k refers to denoising with DDPMs trained with 5 000 or 10 000 traces, respectively.

Conclusions

- Our results showcase that DDPM models can learn useful representations of side-channel traces in unsupervised contexts.
- To remove noise from a leaky sample point, the network needs more information about the leaking value.
- To accomplish this, it can find features that leak the same value and combine the information from these features to arrive at a less noisy version of the feature.
- In effect, we compress the information from several leaky samples into a singular sample.

Limitations

- While our results show significant gains for the showcased attacks against some targets, it is clear that these benefits are not universal.
- Our method does not improve the SNR for datasets that contain mostly algorithmic noise.

Questions?

Thank you for your attention!

I am happy to answer your questions.

stjepan.picek@ru.nl