electronics

Article

Noise2Clean: Cross-Device Side-Channel Traces Denoising with Unsupervised Deep Learning

Honggang Yu [1], Mei Wang [2,*], Xiyu Song [1], Haoqi Shan [3], Hongbing Qiu [1], Junyi Wang [1] and Kaichen Yang [4]

[1] Guilin University of Electronic Technology , Guilin 541004, China
[2] Guilin University of Technology, Guilin 541004, China
[3] University of Florida, Gainesville, FL 32611, USA
[4] Michigan Technological University, Houghton, MI 49931, USA
* Correspondence: mwang@guet.edu.cn

Abstract: Deep learning (DL)-based side-channel analysis (SCA) has posed a severe challenge to the security and privacy of embedded devices. During its execution, an attacker exploits physical SCA leakages collected from profiling devices to create a DL model for recovering secret information from victim devices. Despite this success, recent works have demonstrated that certain countermeasures, such as random delay interrupts or clock jitters, would make these attacks more complex and less practical in real-world scenarios. To address this challenge, we present a novel denoising scheme that exploits the U-Net model to pre-process SCA traces for "noises" (i.e., countermeasures) removal. Specifically, we first pre-train the U-Net model on the paired noisy-clean profiling traces to obtain suitable parameters. This model is then fine-tuned on the noisy-only traces collected from the attacking device. The well-trained model will be finally deployed on the attacking device to remove the noises (i.e., countermeasures) from the measured power traces. In particular, a new inductive transfer learning method is also utilized in our scheme to transfer knowledge learned from the source domain (i.e., profiling device) to the target domain (i.e., attacking device) to improve the model's generalization ability. During our experimental evaluations, we conduct a detailed analysis of various countermeasures separately or combined and show that the proposed denoising model outperforms current state-of-the-art work by a large margin, e.g., a reduction of at least 30% in computation costs and 5× in guessing entropy.

Keywords: deep learning; side-channel analysis; denoising model

Citation: Yu, H.; Wang, M.; Song, X.; Shan, H.; Qiu, H.; Wang, J.; Yang, K. Noise2Clean: Cross-Device Side-Channel Traces Denoising with Unsupervised Deep Learning. *Electronics* **2023**, *12*, 1054. https://doi.org/10.3390/electronics12041054

Academic Editor: Andrei Kelarev

Received: 31 December 2022
Revised: 31 January 2023
Accepted: 15 February 2023
Published: 20 February 2023

1. Introduction

Deep learning-based profiled side-channel analysis (SCA) has posed a significant security and privacy threat to cryptographic circuits. During its execution, an attacker leverages physical SCA leakages (e.g., power [1], electromagnetic emissions [2], etc.) collected from profiling devices to create DL models for recovering secret information from target devices. For example, Maghrebi et al. [3] demonstrated that deep convolutional neural networks (DCNNs) can be applied for SCA and even exceed traditional statistical methods due to their high-level feature representation capacity and translation-invariance property. Since then, many following works have been proposed to improve the performance of SCA attacks by using specialized DL models or data pre-processing algorithms. For instance, Cagli et al. [4] proposed to embed DCNNs with data augmentation techniques such as shifting deformation to reduce the number of traces required to recover secret information from target devices. Kim et al. [5] showed that the addition of Gaussian noise to traces can address the overfitting problem of DL models and is therefore beneficial for all DL-based SCA attacks. However, these attacks would become less effective or even useless if the distribution of the training set deviates from that of the testing set.

To tackle this issue, Das et al. [6] proposed a cross-device attack method that builds the DL model with traces from multiple devices and eventually achieves lower minimum

traces to disclosure (MTD) than previous attacks, even in the preference of high inter-device variations. Zhang et al. [7] considered a more practical scenario and introduced a new attack method, known as frequency and learning-based power analysis, to address the challenge caused by homogeneous and even heterogeneous devices. Yu et al. [8] took a further step towards utilizing the meta-transfer learning to optimize the parameters of the DL model and make the attack more powerful. Cao et al. [9] proposed to leverage adversarial networks to learn device-invariant features for SCA attacks. However, these DL-based SCA attacks routinely depend on the signal's correction characteristics to build the DL model for cross-device SCA attacks. The countermeasures designed by vendors can easily destroy such signal patterns and eventually degrade the attack's performance.

Recent works apply the DL technique for making implicit feature selection of SCA measurements when facing the challenges from the countermeasures [10]. However, they mainly address a particular type of countermeasures. The performance of these methods will rapidly degrade if they are used for the pre-processing of multiple source countermeasures. Wu et al. [11] further treat single/multi-source countermeasures as noise and utilize autoencoder models to denoise SCA traces for profiled SCA attacks. Nevertheless, they mainly considered a strong white-box assumption that an attacker has complete control over target devices so that they can turn off countermeasures to obtain clean traces from these devices, making it impractical in real-world scenarios. Moreover, they build their DL models and perform the noise removal only on similar devices (i.e., the same circuit architecture). Their method's effectiveness on cross-devices still needs to be thoroughly evaluated.

To overcome the above drawbacks, in this paper, we propose a novel DL-based denoiser to pre-process side-channel traces for noise removal. Different from current state-of-the-art works, the proposed method particularly focuses on a realistic scenario, i.e., black-box setting. Specifically, we assume that attackers have full access to profiling devices so that they can turn on/off countermeasures to obtain paired noisy/clean traces for building a DL model. We further assume that attackers have no control over attacking devices but can observe and collect noisy traces without knowing their clean counterparts (i.e., unsupervised learning). The key idea of our method is that we utilize a novel U-Net-based denoiser to efficiently remove the noises (i.e., countermeasures) from the measured power traces. To the best of our knowledge, such a technique has never been used in the SCA community. Figure 1 shows the overview of the proposed denoising framework. Our denoising framework mainly consists of three key stages: DCANNs pre-training, DCNNs Fine-tuning, and DCNNs inference and denoising. Specifically, we first pre-train the U-Net model on the paired noisy-clean traces collected from the profiled device. In particular, since the U-shaped structure of the model allows for the use of global location and feature representation, the resulting denoiser is able to achieve higher accuracy and lower computation cost simultaneously, which is much better than the results achieved by current state-of-the-art work. Then, we fine-tune the DL model on the noisy-only traces captured from the target device. During the DCNNs inference and denoising stage, the well-trained model will be finally deployed on the target device to remove various types of noises and countermeasures. We also use a simple but efficient learning scheme, known as inductive transfer learning to optimize the parameters of the DL model for cross-device side-channel attacks (i.e., secret key recovery). Since this transfer learning scheme effectively regularizes the feature mapping, the fine-tuned model can converge faster while reducing the probability of overfitting problems.

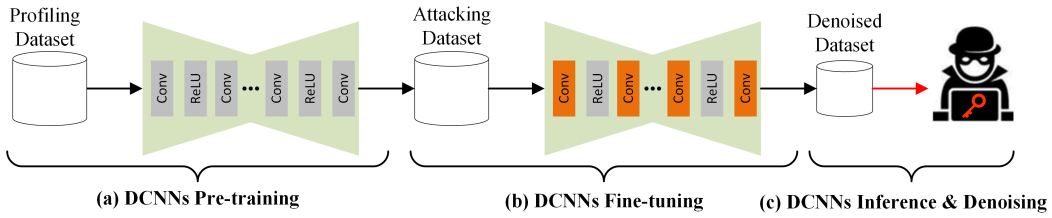


Figure 1. Overview of the proposed denoising framework for profiled SCA attacks: (**a**) Pre-training the DCNNs model (i.e., U-Net) on the paired noisy/clean profiling dataset; (**b**) Fine-tuning the trained DL model on noisy only target dataset; (**c**) Deploying the well-trained DCNNs model on newly collected SCA traces for the noise removal.

In a nutshell, the main contributions of our work are as follows:

- We propose the first method that utilizes a novel U-Net-based denoising scheme to remove certain types of countermeasures for DL-based SCA attacks even in the black-box setting.
- We use a simple but efficient transfer learning technique to transfer knowledge learned from the source domain (i.e., profiling device) to the target domain (i.e., attacking device) to improve the model's generalization ability.
- Extensive evaluation on both local datasets and publicly available datasets shows that the proposed scheme outperforms state-of-the-art DL-based denoising methods by a large margin.

The paper is organized as follows. In Section 2, we discuss some background and related works. Section 3 describes our scheme of the proposed denoising model. Section 4 presents the experimental results and analysis. Section 5 concludes the paper.

2. Background and Related Works

2.1. Profiled Side-Channel Attack

Profiled side-channel analysis has posed a significant threat to embedded devices [12–14]. It aims to exploit various physical side-channel leakages such as power consumption or electromagnetic (EM) emanations to reveal confidential information (e.g., secret keys) from victim cryptographic algorithms. The DL technique has become increasingly common in the SCA domain and even outperforms traditional statistical methods due to its high-level feature representation ability and translation-invariance property. A typical DL-based SCA attack can be divided into two key stages: the profiling and the attacking stages. For the profiling stage, an attacker captures side-channel measurements from a profiling device and uses these measurements along with their labels as a dataset to train DL models. An attacker usually selects the architecture of DL models from the model zoo based on their experiences. During the attacking stage, such a trained DL model would be re-used by an attacker to determine the target device's secret keys from the collected SCA leakages. The guessing entropy (GE) is often used to assess the attack's performance [5]. An attacker routinely hopes to break target devices using as fewer SCA traces as possible (i.e., lower guessing entropy value). However, background noises and complicated countermeasures often compromise this goal and lead to high computing costs while an attacker works towards building a DL model for breaking victim implementations.

2.2. Transfer Learning

Transfer learning (TL) has been widely used in various real-world scenarios, such as computer vision and natural language processing. In general, it is used to fine-tune DL models so that the knowledge learned from the source domain can be effectively transferred to related but different target domains. For example, Ge et al. [15] introduced a transfer learning method that uses a selective joint fine-tuning technique to optimize the parameters of DL models even in the context of insufficient training data. Yang et al. [16] proposed to utilize the learned latent relational graphs to capture dependencies between data points,

thus improving the DL model's performance on downstream tasks. Guo et al. [17] took a further step to use an adaptive fine-tuning approach to find the optimal strategy for target data. Moreover, many researchers have recently explored the possibility of using dedicated regularization approaches and showed that the transfer learning technique with explicit inductive bias promotes the similarity between fine-tuned parameters and original parameters [17]. As a result, such approaches outperform standard fine-tuning on most baseline datasets while performing transfer operations from the source domain to the target domain. In this paper, we utilize such an inductive transfer learning scheme to transfer features learned from the source domain (i.e., profiling devices) to the target domain (i.e., attacking devices). Consequently, our transfer learning scheme accelerates the model denoising process and provides performance gains by using well-trained DL models.

2.3. Dcnns Based Denoisers

Deep convolutional neural networks (DCNNs) have achieved great success in various security-crucial tasks [18,19]. For example, Zhang et al. [20] introduced an effective and efficient clustering framework that uses deviation-sparse fuzzy c-means w/o neighbor information constraint to build the deep learning model. Similarly, Tang et al. [21] further proposed a new viewpoint-based weighted kernel fuzzy clustering method that is superior to previous clustering algorithms especially when processing high-dimensional data. Moreover, Zhang et al. [22] took one step towards leveraging residual learning and batch normalization to accelerate the training process and improve the performance of the DCNNs model. Lehtinen et al. [23] presented novel DL models that learned to restore images by only applying noisy data with the performance exceeding training using clean data. Recently, researchers have investigated that network regularization and transfer learning enables DCNNs to converge faster so that the network can quickly learn target data and recover the clean counterparts from real-noise images with a high signal-to-noise ratio (SNR). Motivated by these prior works, this paper proposes an effective and efficient denoising scheme that utilizes a few novel DL techniques, including U-Net-based architecture, residual learning, and transfer learning to remove single and/or multiple source countermeasures from the measured SCA traces. Since the proposed denoising scheme does not require any clean traces from target devices, it is even more potent than previous white-box DL-based noise removal models.

3. Methodology

In recent years, DL-based SCA attack methods have posed serious threats to embedded devices. Although these existing works achieved progress, their effectiveness would rapidly degrade if target devices are protected with various types of countermeasures, such as random delay interrupts or clock jitters. To address this challenge, we treat these countermeasures as particular "noise" due to their randomness and then apply a novel DL-based denoising method to remove that noise. The overall scheme of our proposed denoising method is shown in Figure 1. Specifically, we first pre-train the U-Net model with paired noisy/clean trances captured from profiling devices. The model is then fine-tuned on the noisy-only traces captured from target devices. In particular, we also utilize a simple but efficient algorithm, known as inductive transfer learning, to optimize the parameters of the DL model for cross-device side-channel attacks. Since the proposed method leverages the advantages of the very deep architecture (i.e., U-Net) and inductive transfer learning, our attack can significantly outperform the current state-of-the-art works by a large margin.

3.1. DCNNs Denoiser Pre-Training

The DCNNs pre-training phase is similar to the classic model training stage. Let $\mathcal{X}$ and $\mathcal{Y}$ denote noisy traces and their clean counterparts, respectively. For the denoiser, we employ a widely-used U-Net [24] architecture which takes $\mathcal{X}$ as inputs to give a prediction, the $\hat{\mathbf{X}}$ of the noise-free clean traces (the DL model architecture used in this paper will be discussed in the following sections). During the pre-training stage, we use the stochastic

gradient descent (SGD) algorithm to minimize the cross-entropy loss J over the training set $\mathcal{D}_t$ to obtain the model's parameters θ (e.g., weights):

$$\mathcal{L}_{\mathcal{D}_i}(f_\theta) = \sum_{(\mathcal{X},\mathcal{Y})\in\mathcal{D}_t} J(f_\theta(x), y) \tag{1}$$

where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are paired noisy/clean SCA traces. In the proposed denoising scheme, we collect paired noisy/clean SCA traces from profiling devices to pre-train the U-Net model. After the pre-training stage, we can obtain initial parameters which would serve as good starting points for the DL model while transferring generic features from the source domain (i.e., profiling devices) to the target domain (i.e., target devices).

3.2. DCNNs Denoiser Fine-Tuning and SCA Attacks

To improve the efficiency of our proposed method, we further fine-tune the pre-trained model on the target domain using a simple but powerful DL technique, i.e., inductive transfer learning. We start by introducing the classic transfer learning scheme for DL models. However, as mentioned in [17], it is hard for us to use this scheme to preserve the knowledge learned from the source domain during the fine-tuning stage on the target domain, leading to worse signal-to-noise ratio (SNR) and thus lower guessing entropy. Motivated by prior works [17,25], we address this challenge by using a novel network regularization term $L^2 - SP$ to optimize the parameters of DL models. Note that the reason why we choose the $L^2 - SP$ as the regularization term is that, compared to other regularization terms such as L^1 or L^2, $L^2 - SP$ can lead to faster convergence and better performance of our DL models for SCA attacks.

Formally, we define the regularization term $\Omega(\theta)$ as follows:

$$\Omega(\theta) = \frac{\alpha}{2}\left\|\theta_{\mathcal{S}} - \theta^0_{\mathcal{S}}\right\|_2^2 + \frac{\beta}{2}\|\theta_{\mathcal{S}'}\|_2^2 \tag{2}$$

where θ^0 is the parameter of the model pre-trained on the source domain. This will be acting as the starting point at the fine-tuning stage. Using this initial vector as the reference in the $L^2 - SP$ penalty, we obtain our final loss function $\tilde{J}$:

$$\tilde{J}(\theta) = J(\theta) + \gamma \cdot \Omega(\theta) \tag{3}$$

where γ is the parameter that is used to trade off the data-fitting term and the compound regularization term during the training stage of DL models. Furthermore, this particular parameter could also minimize the two terms simultaneously while using SGD to update/fine-tune the network parameters. With the help of Equation (3), the DL model could address the challenge exposed by the classic transfer learning technique, thus keeping the original control of overfitting.

The working process of our denoiser and the resulting SCA attacks are presented in Algorithm 1. Similar to the classic DL-based denoising method in the SCA domain, we first turn on/off the countermeasures deployed on the profiling device (i.e., source domain) to collect the noisy/clean pairs for pre-training the U-Net model. In comparison to the random variables used by the existing DL-based method, such pre-trained network parameters (e.g., weights) would serve as a good starting point while we transfer the generic features from the profiling device to the target device. Then, we assume that an attacker has no active control over the target device but passively observes and collects the noisy traces from the target device. During the fine-tuning stage, we capture these noisy traces and generate the corresponding synthetic dataset $D_t(x)$ for fine-tuning the DL model. The resulting model will be finally applied for pre-processing the new SCA traces collected from the target device for the "noise" removal. In particular, the inductive transfer learning technique is used to optimize the parameters for efficient SCA attacks. Since the proposed method combine the advantages of the very deep architecture (i.e., U-Net) and inductive

transfer learning, our method can outperform the state-of-the-art work by a large margin (i.e., a significantly reduction in the computation cost and the guessing entropy).

Algorithm 1 DCNNs Denoiser and SCA Attacks: For the datasets $D_s(x)$ and $D_t(x)$ from source (i.e., profiling device) and target domains (i.e., target device), the DL models $F(x)$ with parameters $\boldsymbol{w}$ (e.g., weights, bias)

Input: $D_s(x)$, $D_t(x)$, $F(x)$
Output: Model parameters $\boldsymbol{w}$

```
Divide datasets D_s(x) into different subsets.
Initialize DL model's parameters with random variables
Pre-train an original DL model F_s(x) with dataset D_s(x)
Obtain the initialization parameters of DL models w_0
while not done do
    Generate fine-tuning set D_t(x) from the target domain.
    Update the parameters w on D_t(x) using Equation (3)
end while
Pre-process the newly collected traces from the target device
Optimize the DL model with inductive transfer learning for efficient SCA attacks.
```

4. Experimental Results

4.1. Experiment Setup

Instead of only evaluating our approach with existing side-channel database, i.e., ASCAD dataset (https://github.com/ANSSI-FR/ASCAD (accessed on 9 June 2021)), we also collect power and EM traces with our customized side-channel acquisition platform and perform side-channel analysis with them. As shown in Figure 2, our acquisition platform enables us to perform side-channel analysis on popular microprocessors and FPGAs while running cryptography algorithms or hardware designs. We use Chipwhisperer UFO board as the motherboard to provide power and data supply for our target CPU and FPGA. The Chipwhisperer UFO board also has a shunt resister design which allows us to collect power traces. We use our customized 3D printer controller software to automatically move the EM probe and collect EM and power traces using a high sample rate Keysight oscilloscope, mso-x 4154a, which is produced by Keysight in Colorado, USA. The collected trace is further pre-analyzed to identify the most leaked position from the CPU/FPGA package and is then used as an instruction to guide the 3D printer in completing EM trace collection. For microprocessor targets, we customize the popular AES algorithm, tiny-AES-c, which is written in C language, and run such an algorithm while collecting power traces.

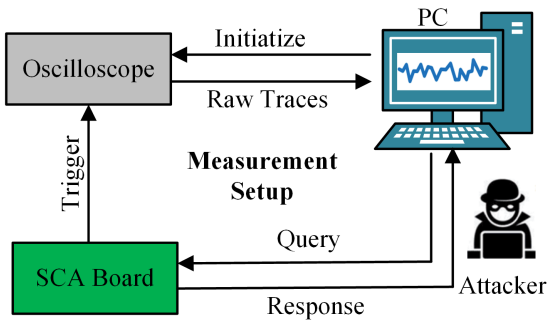


Figure 2. Overview of the SCA trace acquisition platform.

To further evaluate the generality of our approach, we collect power traces on different microprocessors with respect to different microarchitecture designs. More specifically, we run the same AES algorithm on both ARM and AVR-based microprocessors to collect the side channel traces. For ARM based microprocessors, we further run trace acquisition on

different variations. As shown in Table 1, we use ARM Cortex-M0, Cortex-M3, Cortex-M4 chips, which are distributed by Future Electronics in Mississippi, USA as the evaluation targets for ARM platforms. These different ARM architecture designs pose challenges to regular side-channel attack methods as normally requires the attacker to specifically train and run attack on the same platforms. For AVR-based microprocessors, we use ATXMEGA microprocessor as it uses AVRxm architecture design, different than regular AVR which is used by the ASCAD dataset. For the algorithm source code compilation process, we use the same optimization option for all platforms. We also perform reverse engineering for the object files generated for different microprocessors. Such a process provides us information to check if all platforms run similar assembly instructions with similar control flow so that the only variation of the acquisition comes from architecture designs.

Table 1. Summary of all considered SCA datasets.

Datasets	Platform	Nr Features	Nr Traces
ATXMEGA	AVRxm	700	30,000
STM32F0	ARM Cortex-M0	700	30,000
STM32F1	ARM Cortex-M3	700	30,000
STM32F3	ARM Cortex-M4	700	30,000
STM32F4	ARM Cortex-M4	700	30,000
ASCAD	AVR	700	30,000

4.2. Network Architecture

In this paper, we conduct all the experiments on a server of Intel Xeon(R) E5-2623 v4 2.60 GHz CPU, NVIDIA Tesla V100 GPU, and 128GB memory. It is worth noting that the server is built by the authors. We set the epoch and learning rate to fixed values 30 and 1×10^{-3}, respectively. Motivated by existing works [24,26], we use the 16-layer U-Net-based DCNN model where symmetric skip connections, strided convolutions, and transpose convolutions are utilized for extracting multi-scale and generic feature mappings from SCA traces (see Figure 3). We also use rectified linear activation function (ReLU) to down-sample the features generated by convolution operations. The pre-trained weights are first used to initialize our DL models for noise removal. We then fine-tune the last few layers (also called classification layers) of the DL model with our inductive transfer learning scheme. During the pre-training and fine-tuning stages, SGD algorithms are utilized in our method to minimize the loss function in Equation (3). Additionally, the DL model used for SCA attacks consists of two convolutional layers and four fully-connected layers. During the training stage, we keep the model architecture fixed and set the epoch and learning rate to 50, 1×10^{-3}, respectively. The trained model would be used to recover the secret information from the victim device at the testing stage.

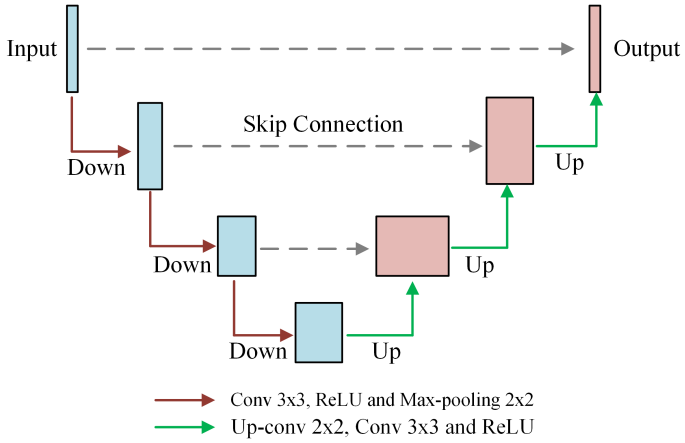
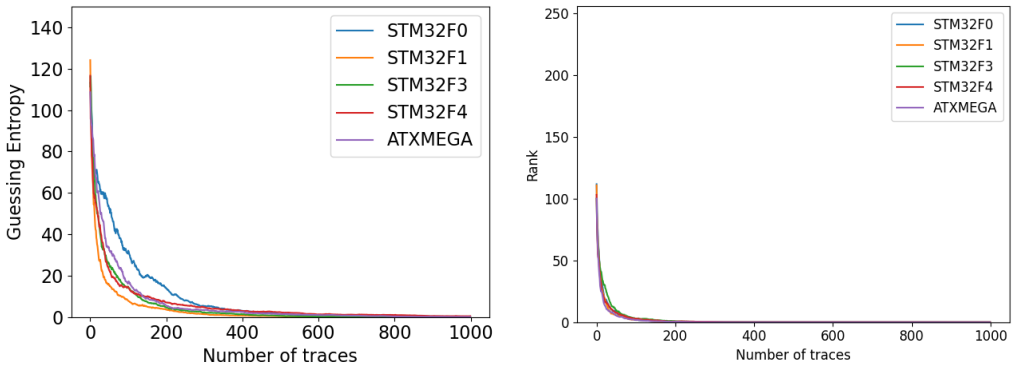


Figure 3. The U-Net architecture of our denoising model.

4.3. Cross-Device Denoising for Local and Public Datasets

To evaluate the effectiveness of our proposed scheme, we conduct extensive experiments on a group of embedded devices with different micro-architectures. In our experiments, we simulated four common types of countermeasures existing in side-channel traces: Gaussian noise, random delay interrupts, clock jitters, and shuffling. For these countermeasures, we use the same setting as mentioned in [11] so that we can make a fair comparison in this paper. Furthermore, we consider a more complicated scenario in which the target device is protected with a combination of all four types of countermeasures.

During the training stage, we first pre-train the U-Net model with 20,000 paired clean/noisy traces from profiling devices. We then randomly collect 20,000 noisy traces from target devices, which are different from profiling devices in terms of models and instruction set architectures (ISA). The pre-trained U-Net model is further fine-tuned with these noisy traces. During the testing stage, we utilize the trained U-Net model to denoise the noisy traces captured from the target device. The resulting traces (i.e., denoised traces) are used for building the DL model to recover the secret keys from the victim device. We also apply the inductive transfer learning technique for optimizing the parameters of the DL model for performing such attacks. The experimental results of the Gaussian noise are demonstrated in Figure 4. As we can see from Figure 4a, with the Gaussian noise, all DL models trained on local datasets can not even converge towards guessing entropy of 0 within 700 traces while attacking the ASCAD dataset. We then use the proposed U-Net-based denoiser to pre-process the noise traces from the target dataset (i.e., ASCAD dataset) protected with the Gaussian noise. The results are shown in Figure 4b. We can observe that all DL models can converge within 230 traces, which is much better than the results obtained by the averaging denoising as shown in Figure 4a. Moreover, we compare the proposed denoising method with the state-of-the-art work in [11] and the results are shown in Table 2. As shown in the table, we can see that our results are much better than the results achieved by the state-of-the-art work [11] in terms of computation time (by 30% on average) and N_{tGE} (by 5× on average).



(**a**) Denoising with averaging (**b**) Denoising with U-Net

Figure 4. Comparison with different strategies for Gaussian noise removal of ASCAD dataset.

Table 2. A comparison to current state-of-the-art DL-based denoising method. ITL—Inductive Transfer Learning. The summary of the training and inference time is reported (on average). N_{tGE} represents the number of traces required to recover the secret key from the target device (on average). min—minutes.

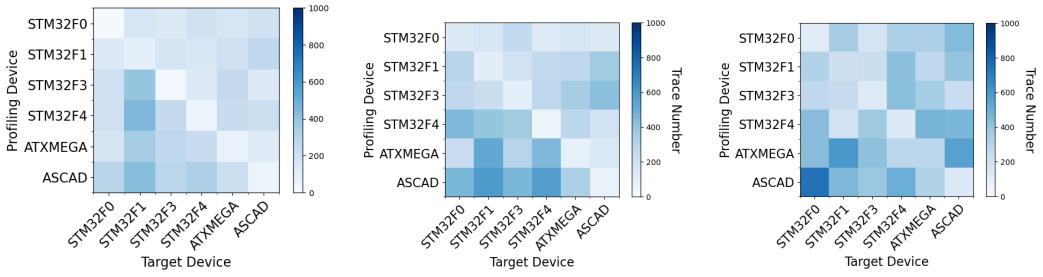
Method	Scenario	Pre-Train	Fine-Tune	Model	Time	N_{tGE}
State-of-the-art [11]	Identical	✓	✗	Autoencoder	22 min	136
This Work	Non-Identical	✓	✓ (ITL)	U-Net	15 min	25

In our experiments, we also test the effectiveness of state-of-the-art single- and/or cross-device attacks on the target device, which are protected with the Gaussian noise, including DL-SCA [3], FL-SCA [7], TL-SCA [27], CD-PA [28] and MTL-SCA [8]. The experimental results are shown in Table 3. We can see that, with the proposed denoising method, we improve the performance of these attacks by a large margin, i.e., significantly reducing the number of traces required to recover the secret information from the victim device. For example, in our evaluation, we build the DL model with the traces denoised by our method and find that such a DL model can break the cryptographic implementation with only 8 traces, which is much better than the results achieved by the current state-of-the-art MTL-SCA attack (i.e., 35 traces). These experimental results further demonstrate that by leveraging the advantages of both very deep architecture (i.e., U-Net) and inductive transfer learning, our denoising model can lead to a better signal-to-noise ratio (SNR), making cross-device attacks more efficient than state-of-the-art works.

Table 3. A comparison to related works with/without our U-Net-based denoising model. N_{tGE} represents the number of traces required to recover the secret key from the target device (on average). FFT—Fast Fourier Transform.

Method	Cross-Device	Pre-Processing	N_{tGE}
DL-SCA [3]	✗	✗/U-Net	300/120
FL-SCA [7]	✓	FFT/ U-Net	210/98
TL-SCA [27]	✓	✗/U-Net	180/72
CD-PA [28]	✓	✗/U-Net	57/32
MTL-SCA [8]	✓	✗/U-Net	35/8

In this paper, we further evaluate the performance of our proposed U-Net-based denoiser on the target devices protected with other complex types of countermeasures, including random delay interrupts, clock jitter, and shuffling. The experimental results are shown in Figure 5. Upon quantitative analysis, we observe that for the individual countermeasure (see Figure 5a–c), our denoising model can effectively remove noises from the traces and therefore is able to help an attacker build a DL model with the high accuracy. Take the clock jitter as an example (see Figure 5b), we apply the traces denoised by our method to train the DL model for cross-device SCA attacks. We find that such a trained model can break the victim device with less than 50 traces, which is much better than the results achieved by the DL model trained with noisy only traces. Further, we consider a more complicated situation in which a vendor deploys the combined countermeasures (i.e., Gaussian noise, random delay interrupts, clock jitter, and shuffling) on the target device to keep its secret and privacy. During the evaluation, we find that, even in this difficult scenario, the denoising model proposed in this paper is still efficient. That is, the DL model trained with the denoised traces can converge within only 100 traces, which is much better than existing works.



(**a**) Random Delay (**b**) Clock jitters (**c**) Shuffling

Figure 5. Guessing entropy: denoising different types of noises with our U-Net-based model.

5. Conclusions

In this paper, we propose a novel denoising method that applies the U-Net model to remove noises from the measured SCA traces. To the best of our knowledge, this is the first time such DL techniques are used in the SCA community. During the evaluation, we validate our proposed method on various widely-used countermeasures, including Gaussian, random delay, clock jitters and Shuffling. Our experimental results show that, in comparison to existing works, the proposed method can effectively denoise the SCA traces even in the black-box setting. Consequently, the DL models trained with such denoised traces can recover the secret information from the victim device with fewer SCA traces and lower computing costs, while existing works fail in at least one or two of these aspects. In the future, we will mainly focus on developing more effective and efficient methods to remove the noses (i.e., countermeasures) from the SCA traces. As a result, an attacker, who wants to target the victim device, can create the DL model with fewer computation costs (e.g., training, inference) while maximizing the denoising performance simultaneously.

Author Contributions: The contributions of authors are as follows: Methodology, H.Y.; Software, H.Q.; Validation, X.S.; Investigation, H.S.; Writing—original draft, M.W.; Writing—review & editing, J.W.; Visualization, K.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (Nos. 62071135, 61261017), the Guangxi Natural Science Foundation (No. 2020GXNSFAA159105), and the Study Abroad Program for Graduate Students of Guilin University of Electronic Technology.

Data Availability Statement: The data included in this study is available upon request by contact with honggang.yu.uf@gmail.com.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kocher, P.; Jaffe, J.; Jun, B. Differential Power Analysis. In *Advances in Cryptology—CRYPTO' 99*; Wiener, M., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1999; pp. 388–397. [CrossRef]
2. Gandolfi, K.; Mourtel, C.; Olivier, F. Electromagnetic Analysis: Concrete Results. In *Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2001, Third International Workshop, Paris, France, 14–16 May 2001*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2162, pp. 251–261. [CrossRef]
3. Maghrebi, H.; Portigliatti, T.; Prouff, E. Breaking Cryptographic Implementations Using Deep Learning Techniques. *IACR Cryptol. EPrint Arch.* **2016**, *2016*, 921.
4. Cagli, E.; Dumas, C.; Prouff, E. Convolutional neural networks with data augmentation against jitter-based countermeasures. In *Proceedings of the 19th International Conference on Cryptographic Hardware and Embedded Systems, Taipei, Taiwan, 25–28 September 2017*; Springer: Cham, Switzerland, 2017; pp. 45–68. [CrossRef]
5. Kim, J.; Picek, S.; Heuser, A.; Bhasin, S.; Hanjalic, A. Make Some Noise. Unleashing the Power of Convolutional Neural Networks for Profiled Side-channel Analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2019**, *2019*, 148–179. [CrossRef]
6. Das, D.; Golder, A.; Danial, J.; Ghosh, S.; Raychowdhury, A.; Sen, S. X-DeepSCA: Cross-Device Deep Learning Side Channel Attack. In Proceedings of the 56th Annual Design Automation Conference 2019, Las Vegas, NV, USA, 2–6 June 2019; pp. 1–6.
7. Zhang, F.; Shao, B.; Xu, G.; Yang, B.; Yang, Z.; Qin, Z.; Ren, K. From Homogeneous to Heterogeneous: Leveraging Deep Learning based Power Analysis across Devices. In Proceedings of the 2020 57th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 20–24 July 2020; pp. 1–6. [CrossRef]
8. Yu, H.; Shan, H.; Panoff, M.; Jin, Y. Cross-Device Profiled Side-Channel Attacks Using Meta-Transfer Learning. In Proceedings of the 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 5–9 December 2021; pp. 703–708. [CrossRef]
9. Cao, P.; Zhang, H.; Gu, D.; Lu, Y.; Yuan, Y. AL-PA: Cross-Device Profiled Side-Channel Attack Using Adversarial Learning. In Proceedings of the 59th ACM/IEEE Design Automation Conference, San Francisco, CA, USA, 10–14 July 2022; pp. 691–696. [CrossRef]
10. Zaid, G.; Bossuet, L.; Habrard, A.; Venelli, A. Methodology for Efficient CNN Architectures in Profiling Attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2019**, *2020*, 1–36. [CrossRef]
11. Wu, L.; Picek, S. Remove Some Noise: On Pre-processing of Side-channel Measurements with Autoencoders. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2020**, *2020*, 389–415. [CrossRef]
12. Thapar, D.; Alam, M.; Mukhopadhyay, D. TranSCA: Cross-Family Profiled Side-Channel Attacks using Transfer Learning on Deep Neural Networks. Cryptology ePrint Archive, Report 2020/1258. 2020. Available online: https://eprint.iacr.org/2020/1258 (accessed on 14 October 2020).

13. Panoff, M.; Yu, H.; Shan, H.; Jin, Y. A Review and Comparison of AI-Enhanced Side Channel Analysis. *J. Emerg. Technol. Comput. Syst.* **2022**, *18*, 62. [CrossRef]
14. Shan, H.; Zhang, B.; Zhan, Z.; Sullivan, D.; Wang, S.; Jin, Y. Invisible Finger: Practical Electromagnetic Interference Attack on Touchscreen-based Electronic Devices. In Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 22–26 May 2022; pp. 1246–1262. [CrossRef]
15. Ge, W.; Yu, Y. Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint Fine-Tuning. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
16. Yang, Z.; Zhao, J.; Dhingra, B.; He, K.; Cohen, W.W.; Salakhutdinov, R.; LeCun, Y. Glomo: Unsupervisedly learned relational graphs as transferable representations. *arXiv* **2018**, arXiv:1806.05662.
17. Guo, Y.; Shi, H.; Kumar, A.; Grauman, K.; Rosing, T.; Feris, R. Spottune: Transfer learning through adaptive fine-tuning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 4805–4814.
18. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
19. Xiao, L.; Heide, F.; Heidrich, W.; Schölkopf, B.; Hirsch, M. Discriminative transfer learning for general image restoration. *IEEE Trans. Image Process.* **2018**, *27*, 4091–4104. [CrossRef] [PubMed]
20. Zhang, Y.; Bai, X.; Fan, R.; Wang, Z. Deviation-Sparse Fuzzy C-Means With Neighbor Information Constraint. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 185–199. [CrossRef]
21. Tang, Y.; Pan, Z.; Pedrycz, W.; Ren, F.; Song, X. Viewpoint-Based Kernel Fuzzy Clustering with Weight Information Granules. *IEEE Trans. Emerg. Top. Comput. Intell.* **2022**, 1–15. . [CrossRef]
22. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [CrossRef] [PubMed]
23. Lehtinen, J.; Munkberg, J.; Hasselgren, J.; Laine, S.; Karras, T.; Aittala, M.; Aila, T. Noise2Noise: Learning Image Restoration without Clean Data. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Dy, J., Krause, A., Eds.; PMLR, Cambridge MA, USA, 2018; Volume 80, pp. 2965–2974.
24. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2015; pp. 234–241.
25. Xuhong, L.; Grandvalet, Y.; Davoine, F. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*; PMLR: Cambridge MA, USA, 2018; pp. 2825–2834.
26. Guo, S.; Yan, Z.; Zhang, K.; Zuo, W.; Zhang, L. Toward convolutional blind denoising of real photographs. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 1712–1722.
27. Genevey-Metat, C.; Gérard, B.; Heuser, A. On What to Learn: Train or Adapt a Deeply Learned Profile? Cryptology ePrint Archive, Report 2020/952. 2020. Available online: https://eprint.iacr.org/2020/952 (accessed on 11 August 2020).
28. Cao, P.; Zhang, C.; Lu, X.; Gu, D. Cross-Device Profiled Side-Channel Attack with Unsupervised Domain Adaptation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2021**, *2021*, 27–56. [CrossRef]