Dual-Leak: Deep Unsupervised Active Learning for Cross-Device Profiled Side-Channel Leakage Analysis

Honggang Yu*, Shuo Wang*, Haoqi Shan*†, Maximillian Panoff*,
Michael Lee*, Kaichen Yang‡ and Yier Jin*
*University of Florida, {honggang.yu, m.panoff, michael.lee}@ufl.edu, {shuo.wang, yier.jin}@ece.ufl.edu
†CertiK, haoqi.shan@certik.com
‡Michigan Technological University, kaicheny@mtu.edu

2023 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) | 979-8-3503-0062-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/HOST55118.2023.10133491

***Abstract*—Deep Learning (DL)-based side-channel analysis (SCA), as a new branch of SCA attacks, poses a significant privacy and security threat to implementations of cryptographic algorithms. Despite their impacts on hardware security, existing DL-based SCA attacks have not fully leveraged the potential of DL algorithms. Therefore, previously proposed DL-based SCA attacks may not show the real capability to extract sensitive information from target designs. In this paper, we propose a novel cross-device SCA method, named Dual-Leak, that applies Deep Unsupervised Active Learning to create a DL model for breaking cryptographic implementations, even with countermeasures deployed. The experimental results on both the local dataset and publicly available dataset show that our Dual-Leak attack significantly outperforms state-of-the-art works while no labeled traces are required from victim devices (i.e., unsupervised learning). Countermeasures are also discussed to assure hardware security against new attacks.**

***Index Terms*—Side-channel Analysis, Deep Learning, Active Learning**

I. Introduction

Side-channel analysis (SCA) is a class of cryptographic attack in which information from victim devices is extracted by exploiting various unintentional physical side-channel leakages such as power consumption [1], [2] and electromagnetic (EM) emissions [3], [4]. Recent works demonstrate that SCA against cryptographic devices achieve substantial progress even in the presence of various countermeasures such as masking and random delay. Among these new attacks, Deep Learning (DL)-based SCA is becoming increasingly common and even outperforms traditional statistical methodologies thanks to its capacity for high-level feature representation and translation invariance [5]. For instance, Hettwer *et al.* [6] introduced a side-channel attack scheme, which shows that DL models can break a cryptographic implementation running on a microcontroller. In the wake of this, many following works have been proposed to enhance their performance by designing specialized deep neural networks (DNNs) [5], [7]–[9]. In addition to the model itself, Kim *et al.* [10] demonstrated that adding artificial noise to power or EM side-channel traces helps with the regularization of neural networks. Wang *et al.* [11] introduced Conditional Generative Adversarial Networks (CGANs) to augment the side-channel traces for DL models. The authors in [12] further applied the U-Net model to remove the noise from the measured side-channel traces, hence making the attack more powerful.

Despite the research progress, DL-based SCA methods still suffer from various limitations. For example, the performance of DL models will rapidly degrade if the statistical distribution of SCA traces from profiling devices differs from those captured from victim devices, a phenomenon occurring even between two instances of the same architecture. To address this challenge, Das *et al.* [13] developed the first cross-device SCA methodology that used the combined power side-channel leakages from multiple profiling devices to create DL models. Zhang *et al.* [14] further proposed to apply the Fast Fourier transform (FFT) to improve the performance of cross-device SCA attacks further. More recently, Yu *et al.* [15] utilized the meta-transfer learning technique to optimize the parameters of DL models, thus reducing training costs and the required amount of traces for a successful attack. However, these DL-based SCA methods [12]–[15] still use supervised learning techniques, which rely on the availability of huge amounts of labeled data when creating these DL models. Labeling collected data is often impractical as an attacker generally is not allowed to access the label information from victim devices under real-world conditions.

One way to ease this problem is coming up with intelligent ways to partly reduce or eliminate the adversary's dependence on the label information from victim devices. For instance, Picek *et al.* [16] explored the possibility of reducing such labeling efforts by using a semi-supervised learning scheme. Cao *et al.* [17] further proposed a new attack method, known as AL-PA, which applies unsupervised learning as well as adversarial learning models to create the profiled model for cross-device attacks. Although the approach of [17] does not require labeled attacking traces, the inherent drawback of the adversarial learning model routinely makes the training stage unstable and slow. As a result, the model's parameters (e.g., weights) have difficulty finding optimized values for the training set. Additionally, existing works evaluate these DL-based attacks only on the different copies of the devices with the same circuit design. The effectiveness of the attack across non-identical devices needs to be justified.

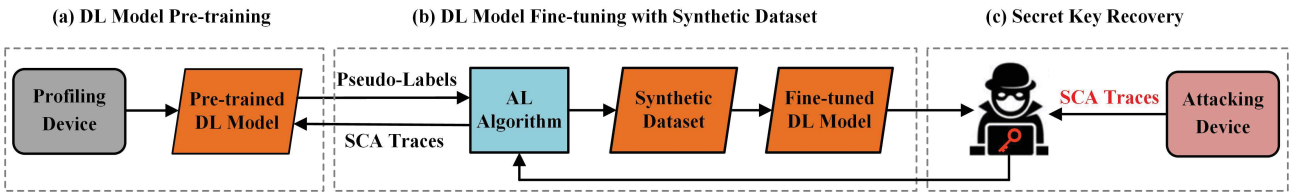


Fig. 1: Illustration of the proposed unsupervised active learning framework for SCA attacks: (a) Pre-train an initial DL model with the physical side-channel traces collected from profiling devices; (b) Use the AL algorithms to select the "informative" SCA traces from the unlabeled active pool and generate the synthetic dataset for fine-tuning the DL model; (c) Deploy the fine-tuned DL model on the attacking device to determine its secret key from the measured power or EM traces.

In this paper, we present a novel attack method that utilizes unsupervised active learning to build the DL model for efficient side-channel attacks without requiring any labeled attacking traces. We call the new attack Dual-Leak. In particular, we address the challenging problem with higher device discrepancy and surprisingly find much less expensive analytical solutions thanks to active learning algorithms. The key idea of our attack method is that we exploit a novel combination of active learning and unsupervised learning to create the deep learning model for efficient side-channel attacks. Figure 1 visualizes the structure of our proposed Dual-Leak. Specifically, the DL model is first pre-trained on the labeled profiling dataset to obtain suitable initial parameters. We then use active learning to choose the most informative traces from the unlabeled active pool and present them to be labeled by the pre-trained model (also known as the oracle). The resulting trace-prediction pairs can be viewed as the synthetic dataset to re-train the DL model for optimizing its parameters (e.g., weights). Since the selected traces lie approximately on the classification boundary, an adversary can greatly improve the efficiency and effectiveness when generating the synthetic dataset for fine-tuning the DL model. It is worth noting that this proposed approach is fundamentally different than other works such as [12] as we select a subset of complete traces to use in training through active learning, which is then used in transfer learning while [12] uses a parameter regularization to guide the transfer learning process. In addition to various separate AL algorithms, we also explore and analyze their combined strategies. Our results demonstrate that ensemble methods can be used for increasing the diversity of the informative traces lying on the classification boundary, thus making our DL based SCA attacks more powerful than current state-of-the-art works. As a result, an adversary can use our attack method to accurately recover the secret information from the victim devices even in an unsupervised learning scenario.

In summary, we make the following contributions:

- We propose the first method that integrates active learning and unsupervised learning algorithms for cross-device side-channel attacks. Most representative data from the unlabeled input SCA traces will be automatically selected to query an oracle DL model to obtain their labels and build a synthetic dataset to train the model for SCA attacking purposes.
- We assess a loss function designed to effectively identify valuable features for training from both the source and target domains concurrently, thus making the model converge faster while reducing the probability of overfitting. Moreover, this loss function is generic and can be easily integrated into existing DL based SCA attacks to improve their effectiveness.
- We conduct extensive experiments on both local datasets and publicly available datasets to evaluate the performance of our Dual-Leak SCA methodology. The experimental results show that the proposed attack outperforms state-of-the-art works while having no requirements on label information from the victim devices.

II. Background and Related Works

A. Deep Neural Network

Deep Neural Networks (DNNs) are gaining popularity in various security-critical tasks such as object recognition or autonomous vehicles [18]–[20]. Typically, a DNN model mainly contains three types of data layers: *an input layer*, *an output layer* and *hidden layers*. The input layer often acts as a special part of DNN models and usually varies with the dimensions of the input data. Given an input, the corresponding classification results, e.g., labels, confidence scores, etc. are routinely printed by output layers. The remaining hidden layers such as convolution, pooling and fully connected layers are often utilized to extract the linear/non-linear features of input data using a variety of dedicated computations. The convolutional layer is usually followed by a pooling layer, which performs average or max pooling operations to produce the sub-sampled feature maps during the sliding window. The fully-connected layers in a DNN model connect every neuron in one layer to all activations in previous layers, as seen in a traditional Multi-Layer Perceptron (MLP) network. The function of these layers is to predict the probability distribution of the input data for different labels by computing weighted summations, adding certain biases as well as using non-linear activation functions such as Tanh.

B. Profiled SCA

A profiled SCA poses a serious security and privacy threat to embedded devices. Recently, researchers show that DL models can improve the performance of traditional profiled methods against cryptographic devices in embedded systems and thus have become increasingly popular in SCA attacks [21]–[24]. Specifically, given physical side-channel traces X_i and the corresponding labels Y_i, an attacker can create the dataset

$\mathcal{D}_{\mathrm{t}} = \{(X_i, Y_i) \mid i = 1, 2, \ldots, N_m\}$ for training the DL model where N_m denotes the size of the dataset. In a typical DL-based profiled SCA, an attacker trains the model $f(x)$ on the dataset $\mathcal{D}_{\mathrm{t}}$ and utilizes it to obtain the secret keys of victim devices by feeding already known plaintexts P_i and the attacking traces into the DL model. In particular, the DL models used in the profiled SCA can filter and align the physical side-channel traces automatically, relaxing the trace alignment requirement in traditional SCA attacks.

C. Active Learning

Generally, Active Learning (AL) is applied to iteratively selected informative data points to present them to be labeled by an oracle, to maximize the performance of retrained deep neural networks [25]–[33]. Existing works on active learning mainly focus on how a user can quantify the importance of each point in the large-scale active pool such as "useful" or "unusable". For example, Gal *et al.* [34] introduced an uncertainty-based active algorithm that directly samples an informative subset of a very large collection and query labels with the ones with low confidence (i.e., those the model is least certain about). Beluch *et al.* [35] further improved their effectiveness by using an ensemble of neural networks to estimate the uncertainty of unlabeled examples so that the proposed method can achieve a good trade-off between a model's accuracy and computation costs. Ducoffe *et al.* [36] presented a novel active learning method that utilizes margin-based sampling to choose particular data points from the active pool. Since the resulting points lie approximately on the classification boundary, a user can greatly reduce labeling efforts while generating the synthetic dataset for training the DL model.

Inspired by the success of AL algorithms in computer vision and pattern recognition, this paper explores how an adversary can transplant this novel technique to build DL models adapted for efficient and practical SCA attacks. In particular, we consider active learning as a core set selection process in which a set of informative side-channel traces is selected from unlabeled traces and then presented to an oracle for labeling. Consequently, the model trained on the resulting input-output pairs (i.e., synthetic dataset) is competitive over the remaining traces. To the best of our knowledge, our work is the first research effort to introduce active learning algorithms in the SCA domain to improve the model's performance for cross-device attacks.

III. Threat Model

While recent DL-based SCA attacks follow similar but not the same threat models, for the proposed Dual-Leak attack, we follow a more relaxed (and more realistic) setting. We first assume that an adversary has no active control over victim devices, but can passively observe and collect their side-channel traces under encryption operations. Therefore, all traces captured from the victim devices are not linked to the precise intermediate variable, meaning that the adversary can only collect unlabeled side-channel traces. While an adversary can capture side-channel traces from the profiling devices with known secret keys (i.e., labeled traces), in this study we further assume a more realistic attack scenario where the adversary may not have the exact same profiling device as the victim device. The adversary only knows the secrecy related functionality of the victim device so they will purchase a profiling device with the same functionality (but not the same circuit structure). For example, an adversary knows the victim device runs the AES algorithm but does not know the underlying circuit structure or the software program. The adversary can then have an arbitrary AES design, either an AES accelerator or a software program running on a microprocessor as the profiling device. We believe that this threat model is more practical than existing works as adversaries rarely have active control over the victim device while passively observing side-channel information such as power consumption or electromagnetic emissions.

The main goal of the adversary is to apply those available traces, i.e., a set of labeled profiling traces and a set of unlabeled attacking traces to build a DL model for effectively recovering the confidential information (e.g., encryption keys) from the victim device. Different from current state-of-the-art side-channel attacks, the proposed method particularly focuses on the cross-architecture attack scenario where the profiling and attacking devices are different in terms of circuit designs and/or instruction set architectures (ISAs).

IV. Methodology

DNN models have gained increasing popularity in the side-channel community due to their high-level feature representation and translation invariance. Despite their success in hardware security, recent DL-based SCA attacks have not leveraged the potential of DL algorithms, thus, do not show the capability to extract secret information from victim cryptographic algorithms. In this paper, we, for the first time, propose to utilize active learning to build DL models for cross-device profiled SCA attacks. The overview of our Dual-Leak method is demonstrated in Figure 1 and the attack process is also shown in Algorithm 1. The proposed attack mainly consists of two steps: DL model pre-training and DL model fine-tuning with the synthetic dataset. Specifically, our attack starts with training a DL model on the labeled profiling dataset to obtain suitable initial parameters. Then, we use AL algorithms to select the most informative traces from the unlabeled attacking traces and present them to the pre-trained model (i.e., oracle) for obtaining the pseudo labels. The resulting input-output pairs will be viewed as the synthetic dataset to further fine-tune the DL model. Finally, we deploy such a well-trained deep learning model on the victim device to determine its confidential information (e.g., secret keys) from the measured power or EM traces. The key idea of our attack is that, by combining the advantages of active learning and unsupervised learning, we can compromise the victim cryptographic algorithms with lower computation costs and fewer side-channel traces than current state-of-the-art attacks.

A. Active Learning Sampling Strategy

In a typical cybersecurity oriented DL task, which poses restrictions on the capabilities of adversaries, obtaining large

amounts of labeled data (more precisely, individual *instances*) is not always practical [29], [32]. Active learning offers a promising solution to this issue. By choosing a set of informative samples from the unlabeled data and getting them labeled by an oracle, active learning aims to minimize labeling effort while simultaneously maximizing the performance of the DL model. The concept of active learning originated from the fact that only a few examples from the set of unlabeled data (also known as the active pool) are essential for determining the DL model's decision surface. Based on that, active learning algorithms have shown unprecedented success in many research areas in computer vision and pattern recognition due to its ability to mitigate the cost of creating a model by selecting the most representative data to use in training [30]–[32]. In this paper, we explore the application of these active learning methods in cross-device SCA attacks and develop a novel method to ensemble them, thus ensure that informative traces are labeled such that the model learned over the resulting trace-prediction pairs is competitive for the remaining traces. More precisely, we formally define the problem of finding an informative trace x' sampled by the multiclass active function $\mathcal{Q}_{\text{multiclass}}$ as follows:

$$\mathcal{Q}_{\text{multiclass}} : \underset{x' \in \mathcal{D}_u}{\arg\min}\ \kappa\left(x', y, \theta\right) \tag{1}$$

where $\mathcal{D}_u$ denotes the unlabeled dataset, κ denotes the output confidence, y denotes the predicted labels (also known as pseudo labels), θ denotes the parameters (e.g., weights) of DL models. To select a set of x', we evaluate four types of AL strategies as well as their combination are considered in this paper (Due to the vast majority of works on AL strategies, we focus only on the most representative ones here.)

Random Sampling. For reference, we consider an extreme scenario where an adversary randomly samples x from the related domain and queries the oracle DL model to generate the synthetic dataset. In this case, an adversary can use all available SCA traces to build the synthetic dataset and obtain the resulting DL model. Nevertheless, using such a massive amount of traces to train the model often incurs high computation costs, which makes it less efficient and even impractical as the amount of traces that an adversary can record is usually very limited in real-life scenarios.

Uncertainty Sampling. The authors in [37] introduce a novel sampling method for active learning, which is quantified in terms of predictive uncertainty. Mathematically, given the probability vectors $\mathbf{Y}$ predicted by a DL model, the corresponding cross-entropy function $\mathcal{H}_i$ can be defined as follows:

$$\mathcal{H}_i = -\sum_j \mathbf{Y}_{i,j} \log \mathbf{Y}_{i,j} \tag{2}$$

where the parameters i and j are label indexes. By maximizing the entropy values $\mathcal{H}_i$ in Equation (2), we can select a subset of "useful" samples with which the DL model is maximally uncertain. These samples would be further used to build the synthetic dataset for fine-tuning the DL model for efficient side-channel attacks.

K-center Sampling. In this paper, we utilize the well-known greedy k-center algorithm to choose the subset of informative samples which are expected to yield the best prediction results in the tested scenarios [38]. Specifically, as an active learning algorithm, the K-center strategy first chooses the most distant samples from existing centers and then presents them to an oracle DL model for labeling:

$$\underset{(x_i,y_i)\in D_i}{\arg\max}\ \underset{(x_j,y_j)\in D_{i-1}}{\min}\ \left\|y_i - f\left(x_j\right)\right\|_2^2 \tag{3}$$

where $f(x)$ denotes the output of the DL model while given the input trace x, such as power or EM. It is worth noting that we iterate this sampling process until all the representative traces are chosen from a very huge collection.

DFAL Sampling. DeepFool-based active learning (DFAL) algorithm can be used to choose informative samples for label assignment from a large-scale data set [36]. As mentioned in [39], the DeepFool algorithm crafts adversarial examples by adding particular noise to the original inputs. In the algorithm, the authors generate the perturbation η_i by solving the following box-constraint optimization problem:

$$\begin{aligned} &\underset{\eta}{\arg\min} && \|\eta\|_2 \\ &\text{s.t.} && f\left(x\right) + \nabla f\left(x\right)^T \eta = 0 \end{aligned} \tag{4}$$

Unlike existing works on adversarial attacks, this paper presents an opposite perspective by exploring how these particular perturbations (i.e., noises) can be applied to augment the dataset for training the DL model. Since the generated examples are close to the decision boundary, the DL model trained with these particular examples often achieves higher classification performance than the model trained with original examples, as suggested in [40], [41].

Ensemble Strategies. Although existing active learning algorithms have progressed, none of them guarantee that the resulting traces are informative and diverse at the same time. For example, the uncertainty strategy routinely tends to suffer from the problem of the chosen traces being overly similar (i.e., less diversity), leading to poor classification performance while adversaries consider such traces as an ideal training set. After conducting a thorough analysis, we also observe that utilizing solely the K-center sampling and DFAL sampling methods fails to generate an efficient dataset for training a DL model (The result is consistent with the finding in [31]). In this paper, we implement a fusion of these AL algorithms to effectively address this challenge, thereby ensuring both the informativeness and diversity of the selected traces simultaneously. To conduct a more comprehensive evaluation of AL algorithms for side-channel attacks, we examine three distinct ensemble strategies within this study: Uncertainty + K-center, Uncertainty + DFAL, and DFAL + K-center. Take the DFAL + K-center strategy as an example, the DFAL strategy is first used to choose an initial subset of informative traces from the unlabeled active pool. Then, we utilize the K-center strategy to eliminate further the redundancy for increasing the diversity of valuable traces lying on the classification boundary of the DL model. As a result, the model trained on such traces is competitive over the remaining traces.

B. Deep Unsupervised Active Learning for SCA

Our Dual-Leak method utilizes an unsupervised active learning algorithm to effectively transfer knowledge from a label-rich source domain to a fully unlabeled target domain. It is worth noting that we use AES implementations as sample cryptographic implementations in this paper while the proposed attack applies to other cryptographic algorithms as well. Given input traces $x \in \mathbf{X}$, the corresponding output labels $y \in \mathbf{Y}$ and the statistical data distribution $\mathcal{D}$, we formally define the objective function of DL models as follows:

$$\arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_u} \left[\mathcal{L}(\theta, x, y)\right] \tag{5}$$

Where $\mathbb{E}$ denotes the population risk, θ and $\mathcal{L}$ denote network parameters and the loss function, respectively. During the training stage, the network's parameters, such as weights or biases, are iteratively optimized via gradient descent algorithms such as the Stochastic Gradient Descent (SGD). However, the model trained in this manner often fails to generalize well across different domains (i.e., devices) due to its overfitting problem. To tackle this challenge, this paper utilizes a novel loss function proposed in [40] to train the deep learning model for an efficient side-channel analysis. Mathematically, the loss function of our DL model can be defined as follows:

$$\arg\min_{\theta} \left[\mathbb{E}_{(x,y)\sim\mathcal{D}} \left(\mathcal{L}(\theta, x, y) + \max_{x'\in\mathcal{D}_u} \mathcal{L}(\theta, x', y')\right)\right] \tag{6}$$

where $\mathcal{L}(\theta, x, y)$ and $\mathcal{L}(\theta, x', y')$ denotes the classification loss calculated on the labeled profiling traces (i.e., source domain) and the fully unlabeled attacking traces (i.e., target domain), respectively. Moreover, the informative traces x' can be selected by utilizing various AL algorithms, as suggested in Equation (1). We query the neural network model with these side-channel traces x' to obtain the corresponding pseudo-labels y', which will be treated as ground truth in our methodology. Further, we then optimize the network's parameters by minimizing the loss function in Equation (6) using SGD algorithms. As a result, our DL model can effectively distill valuable features from both the source and target domains concurrently and thus converge faster while avoiding the overfitting problem.

Step 1: In our attack, the deep learning model is first pre-trained on the labeled profiling dataset X_0 to obtain its initial network parameters. In particular, such pre-trained parameters will be viewed as a starting point for the neural network while transferring general features from the source domain to the target domain.

Step 2: We then build the active pool D_u with a set of the unlabelled side-channel information which are collected from the target device. By using the AL algorithms as mentioned in the previous sections, we select the most representative traces from the unlabeled active pool and present to the pre-trained model for labeling. The resulting input-output pairs can be viewed as the synthetic dataset during the training stage.

Step 3: Finally, we re-train the DL model on the synthetic dataset to optimize its parameters (e.g., weights). We also iterate the training process by treating the fine-tuned model as the pre-trained model and re-generate the particular synthetic subset with the AL algorithms for fine-tuning the DL model. In the our attack, the number of iterations is set to a fixed value of 3 as it can help us to achieve a good trade-off between the model's performance and the computation cost.

Algorithm 1 Dual-Leak: For the DL model F_t with hyper-parameters (e.g., learning rate, kernel size, etc), a maximum number of iterations n, AL strategies $\mathcal{S}$, a labeled dataset $\mathcal{D}_t(x)$, unlabeled active pool $\mathcal{D}_u$ and the initial fine-tuning dataset $(\mathcal{X}_0^s, \mathcal{Y}_0)$.

Input: F_t, $\mathcal{S}$, $\mathcal{D}_t(x)$
Output: Student model F_s
  Pre-trained a DL model F_t with dataset $D_t(x)$
  Initialize $i \leftarrow 0$, $\mathcal{X}_0^s \leftarrow X_0, \mathcal{Y}_0 \leftarrow \mathcal{Y}_0$
  while $i < n$ **do**
    Select the useful subset $\Delta\mathcal{X}_i^s$ in $\mathcal{D}_u$ with strategies $\mathcal{S}$
    Query F_t and obtain labels $\Delta\mathcal{Y}_i$ for all traces in $\Delta\mathcal{X}_i^s$
    $\mathcal{X}_i^s \leftarrow \mathcal{X}_i^s \cup \Delta\mathcal{X}_i^s, \mathcal{Y}_i \leftarrow \mathcal{Y}_i \cup \Delta\mathcal{Y}_i$
    Fine-tune the DL model F_i^S with $(\mathcal{X}_i^s, \mathcal{Y}_i)$
    Update the active pool $\mathcal{X}_u \leftarrow \mathcal{D}_u - \Delta\mathcal{X}_i^s$
  end while

C. Evaluation Metric

In this paper, we utilize guessing entropy (GE) to assess the effectiveness of our proposed Dual-Leak attack [7], [42]. GE determines how many traces an attacker needs to recover a secret key from the target device while performing side-channel attacks. Given the input vectors $V = [v_1, v_2, ..., v_m]$ in our attacking stage, the predicted probability $\hat{p}_{ij}$ for key candidates, an attacker outputs a key guessing vector $g = [g_1, g_2, ..., g_{|\mathcal{K}|}]$, where $|\mathcal{K}|$ is the size of the key space and g_i is the log-likelihood principle that can be formally described as follows:

$$g_i = \sum_{i=1}^{m} \log\left(\hat{p}_{ij}\right) \tag{7}$$

Given the secret keys, the GE can be generated via calculating the average values of these keys over m testing traces. When GE reaches 0, the true key is the most likely guess and thus recovered.

V. Evaluation

A. Experimental Setup

To evaluate the performance of our proposed Dual-Leak attack method, we implement extensive experiments on the SCA measurement platform as shown in Figure 2. Specifically, an oscilloscope (Keysight MSOX4154A, 1.5GHz, 5GSa/s) is connected to a computer to collect side-channel traces. The computer sends random plaintext and a random key to the SCA evaluation board running the 128-bit AES algorithm. We use a popular AES software algorithm implementation called tiny-AES-c[1] to perform the encryption process on different microprocessors. We also apply a low noise Keysight

[1]https://github.com/kokke/tiny-AES-c

N7020A power rail probe and a shunt resister to measure the power consumption of the development board. A Langer near field probe is utilized to collect the EM leakage from the microprocessors under test in conjunction with PA303 low noise amplifier. The power and EM traces are captured at the same time, while the development board runs the encryption algorithm. The development board also toggles specific General Purpose I/O (GPIO) pins as the trigger signal right before the encryption process. Each encryption iteration is repeated 32 times with the same key-text pair. In addition to this collection setup, a customized 3D printer is deployed to automatically localize the maximum leakage source while collecting EM traces. In particular, the data acquisition is performed while the microprocessor performing the sbox lookup operation on the first byte of key and plaintext, $sbox(key[0] \oplus plaintext[0])$, to achieve the maximum sampling rate on the oscilloscope.

With the SCA measurement platform, we implement the software version of AES algorithms on various microprocessors, including four different ARM development boards and an ATXMEGA development board. More specifically, our test boards include STM32F0, STM32F1, STM32F3, STM32F4 series microprocessors which allow us to study the side channel attack difference among ARM Cortex-M0, Cortex-M3, Cortex-M4 architecture respectively. We further run our data collection on the ATXMEGA platform to show how well our DL models work across designs with completely different microarchitectures. For each target platform, we collect 100,000 traces while the AES algorithm performs encryptions with randomly generated plaintexts and keys.

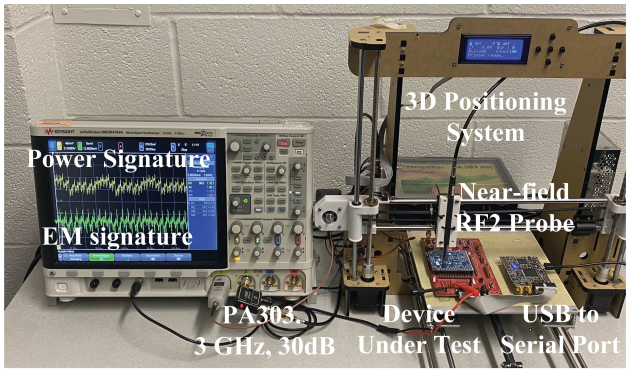


Fig. 2: Overview of the SCA measurement platform.

We further explore whether the proposed attack can be applied to reveal the AES encryption key used in the ASCAD dataset. We determine the ASCAD[2] dataset to be a good general test as it represents a de-facto benchmark for DL based SCA. For this datatset, all side-channel traces are collected from the software AES algorithm running on a 8-bit AVR microcontroller (ATmega8515). Masking countermeasures are also implemented to further protect the key used in encryption from side-channel attacks. In this experiment, we utilize 30,000 traces to train and 10,000 traces to test our DL models.

[2]https://github.com/ANSSI-FR/ASCAD

In this paper, we conduct all experiments on a data server, which is equipped with Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60GHz, 128GB memory, Ubuntu 18.04 system and the NVIDIA Tesla V100 GPU. We apply a simple but efficient DL model for SCA which consists of two convolution layers, two max-pooling layers, and four fully-connected layers. The Rectified Linear Unit (ReLU) is also used as the activation function for the fully-connected hidden layers (20 neurons) of our DL model. During the evaluation, we set the learning rate and the epoch of DL models to the fixed values of $1 \times e^{-5}$ and 50, respectively.

B. Device Variations

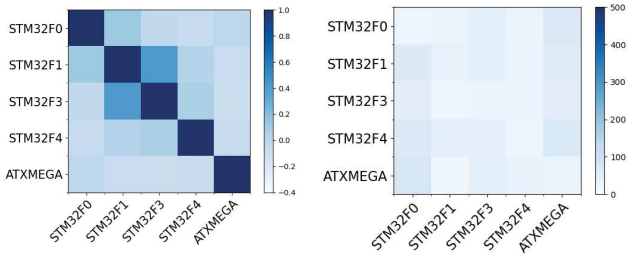
In this section, we use a popular metric, known as *Pearson Product-Moment Correlation Coefficient (PPMCC)*, to evaluate the device variation across different circuit designs. Formally, the *PPMCC* metric can be defined as follows:

$$\text{Pearson}(x, y) = \frac{\sum_{i=1}^{N} \left((x_i - \bar{x})\left(y_i - \bar{y}\right)\right)}{\sqrt{\sum_{i=1}^{N} \left(x_i - \bar{x}\right)^2} \sqrt{\sum_{i=1}^{N} \left(y_i - \bar{y}\right)^2}} \quad (8)$$

where N is the sample size, x_i and y_i are sample points indexed with i. In particular, the $\bar{x}$ and $\bar{y}$ are average values over these points:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i. \quad (9)$$

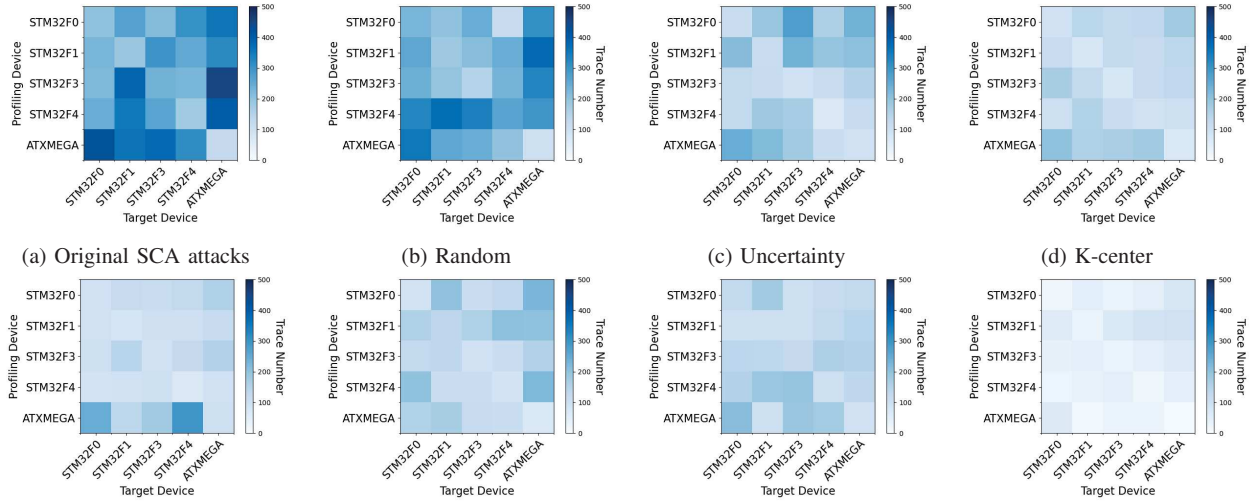
During the experiments, we attack the victim devices, capture their physical SCA measurements and evaluate their device variations using the *PPMCC* metric. Specifically, we collect 20,000 power traces for each cryptographic device running the AES algorithm to compute correlation coefficients. As shown in Figure 3, we have the following findings. First, variations for the devices with the same instruction set architectures (i.e., identical devices) are small, which means an attacker can easily recover the confidential information from the victim device. Second, the correlation coefficients between devices with different instruction set architectures (i.e., non-identical devices) such as ATXMEGA and STM32Fx are quite large, demonstrating that it is challenging to deploy side-channel analysis in such a scenario. It is worth noting that the similar



(a) Device Variations (*PPMCC*) (b) Cross-Device Attacks (Nt_{GE})

Fig. 3: Evaluation of device variations using PPMCC and SCA attacks on target devices. Nt_{GE} represents the number of traces required for GE to reach 0.

trend is also observed while we utilize the *PPMCC* metric to evaluate device variations using the measured EM traces.



(a) Original SCA attacks (b) Random (c) Uncertainty (d) K-center

(e) DFAL (f) Uncertainty + K-center (g) Uncertainty + DFAL (h) DFAL + K-center

Fig. 4: Evaluation of SCA attacks while using different AL strategies.

C. Case Study 1: Cross-Device Single-Domain Dual-Leak

Cross-Device Power Dual-Leak. During the evaluation, we start with comparisons of cross-device single-domain attacks. That is, the side-channel traces collected from the profiling devices and the victim devices are all power traces from different ISAs. In our implementations, we keep the DL model architecture fixed, meaning that only one unified model is trained on the profiling device. This model is then applied to the victim device to recover its confidential information, such as secret keys. In our attack, we first capture 15,000 labeled traces from the profiling device for pre-training the DL model to obtain suitable initial parameters. It is worth noting that we iterate this pre-training process multiple times and make sure the trained model is good enough to be transferred to other devices. Then, we fine-tune the neural network model on the synthetic dataset generated by querying the pre-trained model (i.e., oracle) with 30,000 unlabeled traces selected by various AL strategies from a very large collection (i.e., active pool) of 100,000 unlabeled attacking traces. For reference, the remaining 70,000 unlabeled attacking traces from the active pool is also presented to the oracle for labeling. The resulting trace-prediction pairs will be viewed as the reference dataset for re-training the DL model and the corresponding testing results are also shown in Figure 4 (a)). In our experiments, 10,000 traces are collected from the victim cryptographic algorithm to validate the performance of our proposed attack. We also evaluate how the effectiveness varies with different synthetic dataset generation methods. Specifically, we consider seven types of generation strategies: random, uncertainty, K-center, DFAL, Uncertainty + K-center, Uncertainty + DFAL and DFAL + K-center. These strategies will be used for generating the synthetic dataset to fine-tune the neural network model for side-channel analysis. Figure 4 summarizes the influence of different dataset generation strategies on the attack performance. From this figure, we can see that:

- The model trained by the synthetic dataset always achieves higher accuracy than the model trained by the reference dataset (see Figure 4 (a)-(h)), meaning that only a few traces from the unlabeled active pool are useful for determining the separating surface (i.e., decision boundary) of the DL model and all the rest of the other traces are superfluous to the model. The DL model trained with these useful traces would show remarkable results for local datasets even in the case where the device variation between the profiling device and the target device is quite large.
- Further, we also observe that the synthetic dataset generated by various separate AL algorithms can help an adversary optimize the parameters of DL models in a more efficient way, leading to a lower guessing entropy while attacking the target device of the same and different circuit designs. For the sampling time, although the random strategy obtains the traces using less time (i.e., 60s), the resulting SCA attack performance is much worse than the result achieved by using the other AL algorithms (i.e., $4\times$), clearly indicating that the randomly sampled traces are not informative.
- Among all the AL strategies, the ensemble of DFAL and K-center strategies attempts to vary the contribution of the traces that lie approximately on the decision boundary on the DL model, which helps us increase the diversity and informativeness of the useful traces at the same time. This observation is consistent with the findings in [31]. Specifically, as shown in Figure 4 (b)-(h), the model trained on the DFAL + K-center synthetic dataset can reveal the secret keys from the victim device using as few as 8 traces, which is much better than the results achieved by other sampling strategies, such as random, uncertainty, DFAL, and K-center. In some cases, although the ensemble strategies Uncertainty + K-center and Uncertainty + DFAL

outperform separate AL strategies such as random or DFAL, none of them ensure that the selected traces are informative and diverse simultaneously, leading to worse attack effectiveness than the combined strategy of DFAL + K-center.

Cross-Device EM Dual-Leak. In this paper, we also implement cross-device EM based SCA attacks to explore whether the proposed Dual-Leak method can be used in the low signal-to-noise scenario. Specifically, we assume that the attacker can only capture noisy EM traces from the profiling devices and the victim devices. In the evaluation stage, we first pre-train the neural network model on the profiling dataset which contains 15,000 labeled EM traces, and then fine-tune the model on the synthetic dataset including 30,000 unlabeled EM traces. We find that such a DL model is unable to recover the secret keys from the victim device even using 2,000 EM traces. Then, we enlarge the profiling data set and use 60,000 EM traces to train the DL model. The extensive results demonstrate that our well-trained DL model can converge towards guessing entropy 0 within 500 traces, which is much better than the results achieved by the current state-of-the-art DL based SCA attacks. These results further show that the proposed Dual-Leak attack method can still break the victim devices even using low signal-to-noise EM traces as long as enough signal information is collected.

Comparison to Existing Attacks. To further evaluate the efficacy of our proposed Dual-Leak attack method on the device of the same and different circuit designs, we compare with the existing state-of-the-art attack methods, including DL-SCA [7], SSL-SCA [16], AL-PA [17], N2C-SCA [12], FL-SCA [14] and MTL-SCA [15]. In our implementations, we train the DL model on the profiling devices of different ARM architectures and test on the attacking device equipped with the ATXMEGA microprocessor. When compared, we reproduce the setting of these previous works reported in their papers and then modified them to make such attacks more suitable for specific tasks. As discussed in the previous sections, the combination of DFAL + K-center shows remarkable results for all the considered SCA datasets and is also the only one that can perform better than other AL strategies in all setups. Therefore, in this section, we utilize such a combination strategy to choose 30,000 informative attacking traces from the unlabeled active pool for creating the synthetic dataset to fine-tune the neural network model. During the attacking phase, we then deploy the well-trained model to perform SCA attacks on the target device.

The comparison results are shown in Table I, from which we observe that:

- Our Dual-Leak is able to reveal the confidential information from the victim device with 16 ± 5 traces (i.e., mean $\pm$ standard deviation), which outperforms all the current state-of-the-art works. The same trend also appears while we utilize ATXMEGA as the profiling device to train the model and then apply it to attack the target device with different ARM architectures, e.g, STM32Fx. During the evaluation, we also observe that the proposed attack requires much less SCA traces from the profiling device for pre-training (by 30%) when compared to existing works. Although these works (e.g., [7], [16], [12], [14] and [15]) use fewer attacking traces to fine-tune the neural network model for side-channel attacks, in their threat models they assume that adversaries have full access to the label information from the target device, which can not be easily obtained in the realistic attack. Moreover, the authors in [12] and [14] require the target-specific pre-processing for side-channel analysis on the victim device, which incurs higher computation costs than our proposed Dual-Leak attack. Similar to our method, Cao *et al.* [17] applied the unlabeled attacking traces to adjust the parameters of the pre-trained model. However, the inherent drawback of the adversarial learning model used in their attack usually makes the training stage unstable, resulting in the model's parameters hardly being optimized. This would significantly degrade the attack's effectiveness especially in the real-world scenario where the device variation between the profiling device and the target device are large.

These experimental results further demonstrate that, by combining the advantages of unsupervised learning and active learning, we are able to create a successful DL model for SCA in a data-efficient manner. In comparison to existing works, our Dual-Leak could break victim devices with fewer side-channel traces during the attacking stage and lower training costs during the profiling stage, while no data pre-processing and labeled traces are required from victim devices.

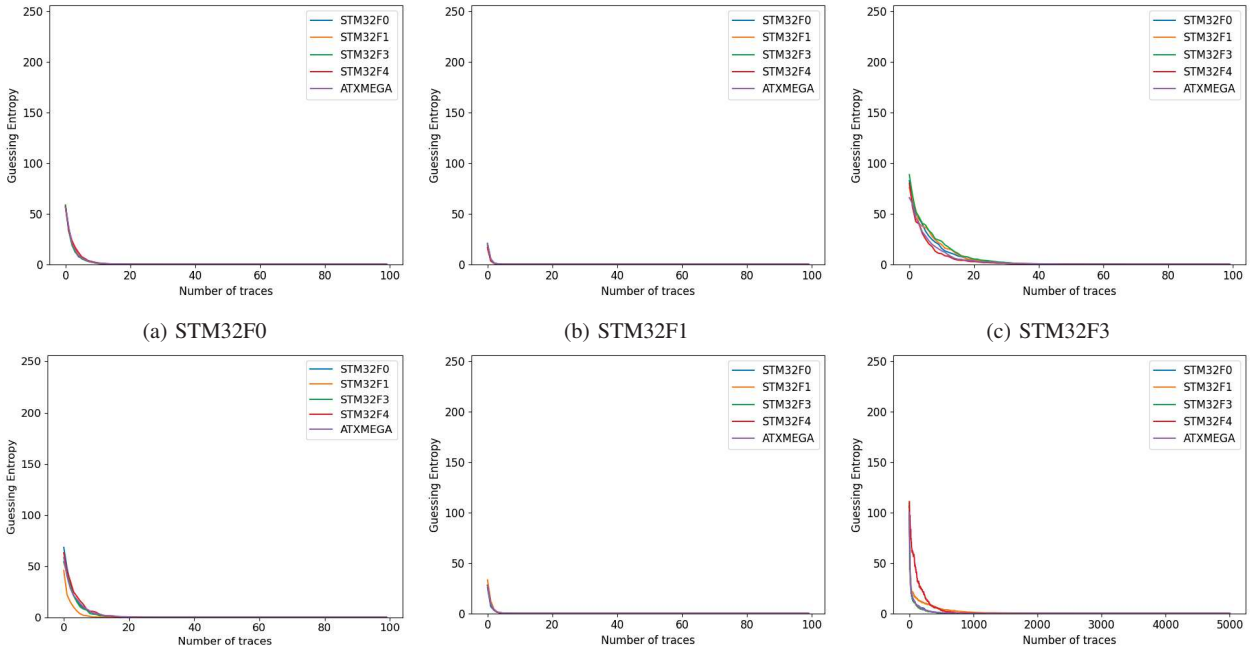
D. Case Study 2: Cross-Device Cross-Domain Dual-Leak

Local datasets. Similar to recent work in [15], we evaluate the proposed Dual-Leak attack on local datasets in cross-domain scenarios, i.e., side-channel traces from the profiling devices and victim devices are from different physical domains. In this scenario, we assume that the adversary is able to collect labeled power side-channel traces from profiling devices. In the meantime, the adversary can only capture EM traces, often more noisy than power traces, from victim devices without knowing any precise intermediate variable (i.e., unlabeled EM traces).

During the experiments, we collect 20,000 labeled power traces to build the dataset for pre-training the neural network model. We also utilize the ensemble strategy of DFAL + K-center to select 30,000 unlabeled EM traces from a very large collection to generate the synthetic dataset for fine-tuning the pre-trained model. The experimental results of local datasets are demonstrated in Figure 5 (a)-(e). As shown in these figures, the DL model fine-tuned by the synthetic dataset can break AES implementations using fewer than 40 unlabeled EM traces. Further, we also consider two variants of the proposed attacks: Dual-Leak-V1 and **Dual-Leak-V2**, which use the loss functions in Equation (5) and **Equation (6)** respectively to optimize the model's parameters (e.g., weights) during the training stage. The experimental results are shown in Table II. We report the performance of our attack method in two forms: the average

TABLE I: A detailed performance comparison of our attack to current state-of-the-art works. Nt_{GE} represents the number of traces required for GE to reach 0. We report the mean and standard deviation of Nt_{GE} for each method as a metric to evaluate its attack performance. It is worth noting that we round all these values to their nearest integers in order to simplify the calculation. FFT - Fast Fourier Transform.

Device Variation	Method	Profiling Labels	Attacking Labels	Preprocessing	Main Property	$\boldsymbol{Nt_{GE}}$
Identical Devices	DL-SCA [7]	✓	✓	✗	✗	325 ± 40
	SSL-SCA [16]	✓	✓	✗	Semi-Supervised Learning	271 ± 121
	AL-PA [17]	✓	✗	✗	Adversarial Learning	51 ± 10
Non-Identical Devices	N2C-SCA [12]	✓	✓	U-Net	Inductive-Transfer Learning	73 ± 38
	FL-SCA [14]	✓	✓	FFT	✗	151 ± 53
	MTL-SCA [15]	✓	✓	✗	Meta-Transfer Learning	61 ± 12
	Our Dual-Leak	✓	✗	✗	Active Learning	$\mathbf{16 \pm 5}$



(a) STM32F0 (b) STM32F1 (c) STM32F3

(d) STM32F4 (e) ATXMEGA (f) ASCAD

Fig. 5: Evaluation of the proposed SCA attack methods on both local datasets and publicly available ASCAD dataset.

TABLE II: A comparison to current state-of-the-art cross-device/cross-domain SCA. Dual-Leak-V1 and Dual-Leak-V2 are two variants of our attack. MTL - Meta-Transfer Learning, UAL - Unsupervised Active Learning.

Method	Pre-Train	Fine-Tune	Accuracy	$\boldsymbol{Nt_{GE}}$
MTL-SCA [15]	✓	MTL	29.80%	350
Dual-Leak-V1	✓	UAL	56.49%	74
Dual-Leak-V2	✓	UAL	**60.25%**	**39**

testing accuracy of the DL model and the minimum values of $\boldsymbol{Nt_{GE}}$ (Similar trends can also be found while we use the mean as well as standard deviations of $\boldsymbol{Nt_{GE}}$ to evaluate the effectiveness of our Dual-Leak). These forms of representation would offer a comprehensive and informative analysis of the attack performance, allowing for a clear understanding of the effectiveness of our proposed Dual-Leak. From this table, we can see that the DL model trained using the loss function in Equation (6) can achieve better attack performance, which is much better than the results achieved by the the model trained using the loss function in Equation (5). With the attack Dual-Leak-V2, an adversary can use as few as 39 traces to reveal confidential information from the target device. The result is also much better than the result (350 traces) achieved by current state-of-the-art cross-device/cross-domain SCA in [15]. Further, these experimental results show that, with the novel loss function as mentioned in Equation (6), the DL model can effectively distill valuable features from both the source and

target domains, thus reducing the probability of overfitting, which is consistent with the results in [40]. Consequently, the DL model trained with a mixture of side-channel traces can generalize well across different domains.

ASCAD and local datasets. To further evaluate the effectiveness of the proposed Dual-Leak attack in cross-domain situations, we also train the DL model on the local dataset and then test the model on the public ASCAD dataset. To simulate a real-world situation where traces from different datasets may be protected by some countermeasures, we also perform masking operations on the ASCAD dataset. During the profiling stage, we keep the architecture of DL models fixed as we only train one model on both local and public datasets. The DL model is first pre-trained with 20,000 labeled power traces from the local dataset and then fine-tuned with 30,000 unlabeled EM traces from the ASCAD dataset. Note that these EM traces are selected by the combined AL algorithm (i.e., DFAL + K-center) from the active pool. During the attacking phase, we utilize the well-trained neural network model to recover confidential information from the victim device using the measured EM traces. The experimental results of our attack method is shown in Figure 5 (f). With the help of the AL algorithm, our attack can reveal confidential information from victim devices using as few as 500 traces, which is much better than the previous SCA attacks in [15], [16]. Even with masking protection, our attack is still able to break the target device using as few as 600 traces. Note that we evaluate the state-of-the-art cross-device/cross-domain attack in [15] with the same experimental setting and find that their DL models cannot converge towards $GE = 0$ even using 2,000 traces. These results demonstrate that, by combining the advantages of active learning and unsupervised learning, our attack can extract confidential information (e.g., secret keys) from the target device with lower computation costs and fewer side-channel traces, while having no requirements of label information and specific data pre-processing methods.

VI. Discussions and Potential Solutions

In this paper, we propose a novel SCA attack method that utilizes unsupervised active learning to build a DL model to effectively break encryption running on a victim device. Still, there are some limitations that we may address in the future. For example, during the evaluation, we found that the DNN model's performance would degrade as the number of unlabelled side-channel traces increases in some cases. We believe this is because pseudo-labels returned by an oracle are not accurate thus cannot be treated as the reference labels during the training stage. We plan to address this challenge by using other advanced DL training schemes or DL architectures to build the profiled models. As a result, we can obtain more accurate pseudo-labels while querying oracle models with unlabelled side-channel traces. The DL models trained by the resulting dataset (i.e., synthetic dataset) would achieve higher performance while recovering the secret keys from victim devices.

As shown in the experimental results, the proposed Dual-Leak attack can effectively recover the confidential information from the victim devices. As our attacks have posed a serious threat to these devices, the corresponding defense methodologies should be carefully considered in the future to protect the AES implementations from DL based side-channel attacks. One possible direction is to craft adversarial examples against DL models. It is worth noting that DL models are vulnerable to these examples generated by adding special noises/perturbations to original examples. Since there are huge amounts of adversarial example generation methods available in the AI domain, we could borrow some ideas from these existing works and further develop adversarial examples against DL models in the SCA domain. The main challenge of this direction is how we can generate special noises/perturbations for original side-channel traces. One way is to design novel optimization algorithms to search for adversarial noises for these traces. Another possible method is to directly generate the adversarial traces by developing particular hardware circuits for cryptographic implementations. As a result, vendors or users who want to keep their information confidential could deploy such special circuits on the cryptographic implementations to defense against DL-based SCA attacks.

VII. Conclusions

Deep Learning based side-channel analysis has posed a serious privacy and security threat to cryptographic implementations. Using this method, an attacker can determine the secret keys of target devices with the measured physical traces. In this paper, we propose an effective and efficient SCA attack that utilizes unsupervised active learning to build the DL model for breaking cryptographic implementations. Our experimental results demonstrate that the proposed attack method can infer the secret key from the victim device with fewer side-channel traces and lower computation costs when compared to current state-of-the-art works. In the future, we will develop novel defense mechanisms that can counter DL-based SCA attacks and thus improve the robustness of cryptographic implementations.

Acknowledgments

This research was partially supported by National Science Foundation under award numbers 1801599 and 1818500.

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] E. Oswald and S. Mangard, "Template attacks on masking—resistance is futile," in *Proceedings of the 7th Cryptographers' Track at the RSA Conference on Topics in Cryptology*, ser. CT-RSA'07. Berlin, Heidelberg: Springer-Verlag, 2007, p. 243–256.

[3] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, ser. Lecture Notes in Computer Science, vol. 2162. Springer, 2001, pp. 251–261.

[4] J.-J. Quisquater and D. Samyde, "Electromagnetic analysis (ema): Measures and counter-measures for smart cards," in *Proceedings of the International Conference on Research in Smart Cards: Smart Card Programming and Security*, ser. E-SMART '01. Berlin, Heidelberg: Springer-Verlag, 2001, p. 200–210.

[5] L. Zhang, X. Xing, J. Fan, Z. Wang, and S. Wang, "Multilabel deep learning-based side-channel attack," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 6, pp. 1207–1216, 2021.

[6] B. Hettwer, S. Gehrer, and T. Güneysu, "Deep neural network attribution methods for leakage analysis and symmetric key recovery," in *Selected Areas in Cryptography–SAC 2019: 26th International Conference, Waterloo, ON, Canada, August 12–16, 2019, Revised Selected Papers 26*. Springer, 2020, pp. 645–666.

[7] 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, pp. 1–36, Nov. 2019.

[8] L. Masure, C. Dumas, and E. Prouff, "A comprehensive study of deep learning for side-channel analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 1, pp. 348–375, Nov. 2019.

[9] L. Wouters, V. Arribas, B. Gierlichs, and B. Preneel, "Revisiting a methodology for efficient cnn architectures in profiling attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 147–168, 2020.

[10] J. Kim, S. Picek, A. Heuser, S. Bhasin, and A. Hanjalic, "Make some noise. unleashing the power of convolutional neural networks for profiled side-channel analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 3, pp. 148–179, May 2019.

[11] P. Wang, P. Chen, Z. Luo, G. Dong, M. Zheng, N. Yu, and H. Hu, "Enhancing the performance of practical profiling side-channel attacks using conditional generative adversarial networks," *arXiv preprint arXiv:2007.05285*, 2020.

[12] H. Yu, M. Wang, X. Song, H. Shan, H. Qiu, J. Wang, and K. Yang, "Noise2clean: Cross-device side-channel traces denoising with unsupervised deep learning," *Electronics*, vol. 12, no. 4, p. 1054, 2023.

[13] D. Das, A. Golder, J. Danial, S. Ghosh, A. Raychowdhury, and S. Sen, "X-deepsca: Cross-device deep learning side channel attack*," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.

[14] F. Zhang, B. Shao, G. Xu, B. Yang, Z. Yang, Z. Qin, and K. Ren, "From homogeneous to heterogeneous: Leveraging deep learning based power analysis across devices," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[15] H. Yu, H. Shan, M. Panoff, and Y. Jin, "Cross-device profiled side-channel attacks using meta-transfer learning," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*, 2021, pp. 703–708.

[16] S. Picek, A. Heuser, A. Jovic, K. Knezevic, and T. Richmond, "Improving Side-Channel Analysis through Semi-Supervised Learning," in *CARDIS 2018 - 17th Smart Card Research and Advanced Application Conference*, Montpellier, France, Nov. 2018.

[17] P. Cao, H. Zhang, D. Gu, Y. Lu, and Y. Yuan, "Al-pa: Cross-device profiled side-channel attack using adversarial learning," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, ser. DAC '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 691–696. [Online]. Available: https://doi.org/10.1145/3489517.3530517

[18] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *2018 European Conference on Computer Vision (ECCV)*, Sep 2018, pp. 580–596.

[19] K. Lee, K. Lee, K. Min, Y. Zhang, J. Shin, and H. Lee, "Hierarchical novelty detection for visual object recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[20] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3d object detection leveraging accurate proposals and shape reconstruction," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[21] A.-T. Hoang, N. Hanley, and M. O'Neill, "Plaintext: A missing feature for enhancing the power of deep learning in side-channel analysis? breaking multiple layers of side-channel countermeasures," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 4, pp. 49–85, Aug. 2020.

[22] B. Hettwer, D. Fennes, S. Leger, J. Richter-Brockmann, S. Gehrer, and T. Güneysu, "Deep learning multi-channel fusion attack against side-channel protected hardware," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[23] L. Wu, G. Perin, and S. Picek, "The best of two worlds: Deep learning-assisted template attack," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2022, pp. 413–437, 2021.

[24] G. Zaid, L. Bossuet, F. Dassance, A. Habrard, and A. Venelli, "Ranking loss: Maximizing the success rate in deep learning side-channel analysis," *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 872, 2020.

[25] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '08, 2008, pp. 1070–1079.

[26] N. Papernot, P. D. McDaniel, and I. J. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," vol. abs/1605.07277, 2016.

[27] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[28] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds., 2008, pp. 1289–1296.

[29] D. Wang, Y. Li, L. Wang, and B. Gong, "Neural networks are more productive teachers than human raters: Active mixup for data-efficient knowledge distillation from a blackbox model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1498–1507.

[30] W. Li, G. Dasarathy, K. Natesan Ramamurthy, and V. Berisha, "Finding the homology of decision boundaries with active learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 8355–8365. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/5f14615696649541a025d3d0f8e0447f-Paper.pdf

[31] S. Pal, Y. Gupta, A. Shukla, A. Kanade, S. K. Shevade, and V. Ganapathy, "Activethief: Model extraction using active learning and unannotated public data," in *AAAI Conference on Artificial Intelligence*, 2020.

[32] C. Li, K. Mao, L. Liang, D. Ren, W. Zhang, Y. Yuan, and G. Wang, "Unsupervised active learning via subspace learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, pp. 8332–8339, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17013

[33] K.-P. Ning, X. Zhao, Y. Li, and S.-J. Huang, "Active learning for open-set annotation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 41–49.

[34] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 1183–1192.

[35] W. H. Beluch, T. Genewein, A. Nurnberger, and J. M. Kohler, "The power of ensembles for active learning in image classification," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9368–9377.

[36] M. Ducoffe and F. Precioso, "Adversarial active learning for deep networks: a margin based approach," *CoRR*, vol. abs/1802.09841, 2018.

[37] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *SIGIR '94*, B. W. Croft and C. J. van Rijsbergen, Eds. London: Springer London, 1994, pp. 3–12.

[38] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *International Conference on Learning Representations*, 2018.

[39] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2574–2582.

[40] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[41] H. Yu, K. Yang, T. Zhang, Y.-Y. Tsai, T.-Y. Ho, and Y. Jin, "Cloudleak: Large-scale deep learning models stealing through adversarial examples." in *NDSS*, 2020.

[42] F.-X. Standaert, T. G. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *Advances in Cryptology - EUROCRYPT 2009*, A. Joux, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 443–461.