Leveraging Deep CNN and Transfer Learning for Side-Channel Attack

Amit Garg and Nima Karimian  
Department of Computer Engineering  
San Jose State University  
San Jose, California, USA

Abstract—The adoption of deep neural networks for profiled side-channel attacks provides powerful options for leakage detection and key retrieval of secure products. Although deep learning is being widely adopted for computer vision, less research has been prominent in template-based profiling power SCA attacks. In addition, most of the existing works fall into a one-dimensional (1-D) CNN technique rather than two-dimensional (2-D) CNN methods. Training a deep 2-D CNN from scratch is computationally expensive and requires a large amount of training data. To overcome these challenges, we adopt deep 2-D CNNs, GoogLeNet, InceptionV3, VGG16, and MobileNetV2 pre-trained to identify all possible AES key bytes. In order to use 2-D CNN and transfer learning, we also propose a novel multi-scale continuous wavelet transform of the power traces and generate scalograms from the wavelet coefficients. Moreover, we propose ResNet 1-D CNN architecture using power traces signal to break AES-128 implementation. To evaluate our proposed work, a key rank metric with the ASCAD dataset is utilized. Our proposed deep CNN framework achieves ≥ 99 accuracy when key rank is less than 10.

I. INTRODUCTION

In recent years, technologies related to healthcare and self-driving car have been significant advanced with high-quality artificial intelligence and sensor techniques. This technology is designed to provide more personal and private forms of an assistant. Moreover, self deriving cars and self-driving healthcare system has been already deployed such as Tesla Autopilot system [2] and Apple Watch healthcare [1]. While aforementioned consumer technology trends provide many exciting applications, they become more challenging to protect from physical attacks and more attractive to adversaries. For instance, an adversary is able to compromise the model file in a self-driving car [20] by allowing a computing system to sense and control the physical system using physical side-channel attacks (SCAs). In addition, SCAs can retrieve sensitive patient’s healthcare information from medical devices [7]. The goal of physical side-channel attack is to observe the CPU’s power consumption during a cryptographic operation to retrieve sensitive information.

Within the area of side-channel attacks of cryptographic implementation, there are various analysis methods proposed and successfully demonstrated over the past two decades, such as Simple Power Analysis (SPA) [18], Differential Power Analysis (DPA) [18], Template Attacks (TA) [6], Correlation Power Analysis (CPA) [5], Mutual Information Analysis (MIA) [9], Partitioning Power Analysis (PPA) [19], and Test Vector Leakage Assessment (TVLA) [10] to derive the secret key of the cryptosystem. SPA is a method that probes a device’s power consumption over a period of time. Because different operations show different power profiles, it is possible to determine what type of function is performed at a given time [18]. DPA uses a more advanced statistical analysis than SPA by modeling theoretic power consumption for each secret key of the cryptosystem. In this technique, multiple power traces from two sets of data will be collected, then computes the likelihood of these traces. CPA is a statistical type of attack based on modeling power consumption such as Hamming distance model and uses the Pearson correlation coefficient to correlate data. Template attacks (TAs) are a subset of profiling attacks, where an attacker creates a “profile” of trace for each key from the target device and applies this profile and to quickly find a victim’s secret key by estimating the conditional probability between the target device and the victim’s one. In summary, the difference between the aforementioned techniques is the requirement of the number of power traces that will be collected during the cryptographic operation (encryption or decryption). With the advent of deep learning, numerous machine-learning (ML)-based side-channel techniques using profiling power (TAs) have been performed.

Gilmore et al. [11] deployed neural network for key recovery mask AES in differential power analysis (DPA) technique. This attack demonstrated that masking based countermeasures are vulnerable to neural networks. Das et al. [8] proposed Deep Neural Network (DNN) with 256 classes. The proposed architecture consists of one layer followed by two hidden layers with 500 neurons for each layer and the output layer has 256 neurons for predicting the correct key byte. They used 200K total traces for the training to obtain a 99.9% attack success rate. This architecture requires huge number of power traces for training to achieve a high success rate.

Golder et al. [12] proposed three methods to recover key bytes from AES-128. First, they utilized multilayer perceptron (MLP) without any preprocessing to break AES-128 implemented on an 8-bit microcontroller where the averaging at 61.98% accuracy has been obtained across the 30 devices. The second approach was 1-D Convolutional Neural Network (CNN) where the accuracy has been degraded to 29.97% compared to MLP. Then they used dynamic time warping (DTW) as a pre-processing to align the power traces along...
with principal component analysis (PCA) followed by the 256-class MLP classifier. PCA-MLP test accuracy outperforms by > 10% compared to CNN and MLP in the cross-device attack performance. However, DTW can align if the reference and test patterns are obtained, and sometimes it may fail to find obvious sequence alignments. Moreover, DTW suffers from the speed and may not be the best solution for SCAs.

Shivam et al. [4] illustrated profiled side-channel attack under different scenario including Same device and same key, Same device and different key, Different device and same key, Different device and different key and conduct experiments with MLP and CNN. The MLP architecture has two hidden layers where each hidden layer has 100 neurons. Moreover, the CNN architecture is comprised of five convolutional blocks and two fully-connected layers. However, it is not clear the CNN is 2-D or 1-D. In addition, it is not clear how an image is constructed if 2-D CNN has been utilized in this paper. Moreover, the dataset has 50,000 traces where each trace has 500 features. Among the datasets, 40,000 have been used for training and 10,000 for testing followed by 3,000 traces for validation. However, to avoid overfitting, the validations set must contain the ‘unseen’ power traces, where that setup is missing in this paper. Kim et al. [16] used 1-D CNN architecture to break AES-128 with four different datasets including, ASCAD dataset, DPAcontest v4 dataset, AES\textsubscript{H/D} dataset, and Random Delay dataset. Similar to [4], the aforementioned paper has the same limitation.

To address the aforementioned limitations, we propose a 2-D CNN architecture to take the most advantage of the computer vision field. However, the power consumption leakage from AES implementation is a signal rather be an image. To tackle this issue, we have converted the power trace signals into an image structure using continuous wavelet transform. The example of the transformed signal into the image has been illustrated in Fig. 1. By converting the power trace signal into an image, not only timing information will be analyzed, but also it provides frequency information. In addition, it can get benefits from state-of-the-art deep learning techniques. Moreover, we can use transfer learning to decrease the training model as it is crucial for the side-channel attack.

In short, the novelty and contributions of the paper are as follows:

A novel image representation of power trace: We propose a novel multi-scale continuous wavelet transform to convert power trace signals to 2-D image structure to accurately obtain micro-texture feature characteristics.

Residual neural architecture: We propose novel residual neural network architecture that allows power traces to propagate well in the deep neural networks. The proposed architecture consists of 6 residual blocks with 2 convolutional layers, each having 64 filters of length 16, per block.

Image based power traces using transfer learning: We employed deep CNN architecture such as GoogLeNet, VGG16, mobilenetv2, Inception-V3 architecture to evaluate the profiled side-channel attack using the converted 2-D representation of power trace signals. To evaluate the effectiveness of our proposed new CNN architecture, public ASCAD datasets have been examined.

II. NOVEL RESIDUAL NEURAL ARCHITECTURE

We propose a novel end-to-end Residual Neural architecture to classify each 256 sub-key of the device under attack using a side-channel power trace. The architecture has 13 layers of convolution followed by a fully connected layer and a Softmax, which is inspired from [14]. Our convolutional neural networks (CNN) architecture contains shortcut connections that are similar to Residual Neural architecture [13]. The proposed architecture allows power traces collected from devices under attack to propagate better in the deep neural networks. The network consists of 6 residual blocks with two convolutional layers. Each convolutional layers contain 64 filters of length 16 per block as shown in Figure 2. When the block reduced the dimensions of the input, the corresponding shortcut connections also reduced the dimensions by applying Max Pooling with the same factor. We also employed batch normalization [15] and a rectified linear activation before each convolutional layer.

In order to avoid over-fitting, a dropout Layer [24] was added in between a rectified linear activation and a convolutional layer. The first layer of the network is special-cased due to this pre-activation block structure. The final layer of the network is a fully connected layer with softmax. The network accepts power trace from each byte as input (700 samples) and outputs a prediction of one out of 256 possible byte classes every 700 input samples. Since the model was trained from scratch, we used He initialization. Adam optimizer [17], with the default parameters, was used to update the weights of the model. The learning rate was decreased by a factor of 10 when the validation loss stops improving. Residual connections and batch-normalization are used to make the optimization tractable. Increasing the depth helps to add the non-linearity as well as increases the capability to capture more of the meaning of the data. The classification model is trained on the 50,000 power traces which consists of 256 classes.

![Image 1](https://via.placeholder.com/150)

Fig. 1: Converting power trace into an image. (a) ASCAD power traces collected from a single byte of AES-128 [3], (b) converted the same power traces into an image using scalogram of continuous wavelet transform.
Fig. 2: Our proposed Residual Neural architecture for predicting the correct key byte utilizing the softmax function. Our deep neural network consisted of 13 convolutional layers followed by a fully connected layer into a softmax. The network accepts power trace from each byte as input (700 samples) and outputs a prediction of one out of 256 possible byte classes every 700 input samples.

III. 2-D CNN ARCHITECTURE

In recent years, CNN is applied to select prominent features from the image processing domain in which various filters for feature extraction will be learned automatically and classify them. However, due to the signal characteristic of power traces, most of the existing works are not taking the advantage of deeper CNN architecture. In this paper, we examined the possibility of using the frequency image as input to deep 2-D CNN. To the best of our knowledge, this is the first work that shows the performance of deep CNN models using converted power traces into a 2-D matrix using wavelet transform.

The most highlighted problem with CNN is that they need large data set to be trained well. Usually, training large data is time-consuming and it is most likely to suffer from the overfitting problem. Thus, training the deep CNN from scratch may not be the best solution for the side-channel attack. Thereby, hence transfer learning techniques can be beneficial in such cases. This motivated us to use transfer learning [21] to deal with the aforementioned drawbacks. In other words, the weights pre-trained on ImageNet are employed to fine-tune our train model to take advantage of transfer learning. Thus, we performed transfer learning with popular CNN architectures such as Inception-V3 [26], VGG16 [23], GoogleNet [25], and MobileNetV2 [22].

Fig. 3 shows a side-channel attack using the image representation of power traces under varying models of deep learning. The following subsection details how we implemented deep CNN models for SCA.

A. Two-Dimensional Representation of Power Traces

A 2-D representation of power traces instead of the 1-D signal provides more opportunity to integrate many properties from the computer vision field into side-channel attack domains including 2-D convolution and max-pooling, transfer learning, larger filter sizes. To generate a 2-D representation of one byte of power traces, a continuous wavelet transformation function with Morse wavelets is applied to each byte of power traces collected during profiling. The resulting coefficients

$$\Psi_{P,\gamma}(\omega) = U(\omega) a_{P,\gamma} \frac{p^2}{\pi} e^{-\omega^2\gamma}$$  \hspace{1cm} (1)$$

where $U(\omega)$ is the unit step, a $a_{P,\gamma}$ is a normalizing constant, $p^2$ is the time-bandwidth product, and $\gamma$ characterizes the symmetry of the Morse wavelet. After converted 2-D representation of the power signal, we then resized into the $224 \times 224$ or $256 \times 256$ dimension to get uniformity in the shape of input. In summary, converting power trace signal from time state to frequency state utilizing the wavelet transform technique, the 1-D signal becomes a 2-D image matrix, and it can be evaluated on multi-resolution.

B. Convolutional Neural Network Models

VGG16: VGG16 is a structure of deep learning to improve AlexNet architecture, with 16 convolutional layers, five pooling layers, three fully connected layers, and an output layer [23]. Then the image with the size of $224 \times 224$ is
passed through a stack of convolutional layers with very small \((3 \times 3)\) kernel-sized filters in all layers. The network also consists of five max-pooling layers, which follow some of the convolutional layers. Please noted that not all the convolutional layers are followed by max-pooling. Moreover, all the concatenated features from the last layer of pooling are connected to three Fully-Connected (FC) layers. From the FC layer, the first two layers have 4096 channels each, the third contains 1000 channels. The final layer is the soft-max layer where it can classify 256 bytes. Our proposed VGG16 model consists of five convolutional layers, followed by a non-linear activation function (ReLU) and a max-pooling layer on each layer. We selected \(3 \times 3\) for the kernel size of the convolutional layers. \(3 \times 3\) kernel size has been selected for the pooling layer with stride 2.

**GoogLeNet:** The GoogLeNet architecture comprises 22 layers where 9 layers of inception module are part of the overall architecture [25]. The inception module is used to reduce the dimension of the area of the image prior to computer convolutions with a larger area of the image. Thereby, convolutions are implemented at various scales sizes simultaneously, from \(1 \times 1\), which is the most accurate, to \(5 \times 5\). This would allow to process of the convolutions at different sizes and then aggregated to the next phase that can extract features from the different scales in parallel. Moreover, the architecture consists of max-pooling layers with stride 2 where all the features are concatenated on the next layer. In order to avoid vanishing gradient, two auxiliary classifiers were used in the middle of the GoogLeNet structure.

**Inception-v3:** While VGG16 architecture is simple in terms of feature extraction, the evaluation of the network comes at a high computation [26]. Moreover, GoogLeNet is also designed to perform well under memory and computational resource constraints. Inception-v3 is another type of Inception family that has several advantages in label smoothing to make improvements by factorized \(7 \times 7\) convolutions. In order to make a model with low computation cost, transfer learning is used to avoid training a new model from scratch. The Inception-v3 architecture consists of 42 layers with auxiliary classifiers to avoid overfitting [26]. To use Inception-v3 as reconstructing AES-128 key bytes, we freeze the convolution layers to be the feature extractor. Then, a fully connected layer is used as a classifier, which is fine-tuned using our labeled power trace images via backpropagation. For optimizer, gradient descent is used during training with batches size of 128. The learning rate began at 0.1 and the weight decay was 0.7 for every 10 steps.

**MobileNetV2:** MobileNetV2 comprises one residual block with a stride of one and one downsizing block with a stride of 2. Each block has 3 layers in which the first layer consists of \(1 \times 1\) convolution with ReLU, the second layer has the depthwise convolution and the third layer has another \(1 \times 1\) convolution without any non-linearity [22]. In order to reduce the computation and increase the robustness of the system, ReLU has been replaced as the non-linearity function. Moreover, a filter with a kernel size of \(3 \times 3\), dropout, and batch normalization is utilized during training. To train the power trace images, pre-trained model MobileNet v2 with a depth multiplier of 1.0, the learning rate of 0.1, batch size of 32 were used. The weight decay is reduced until it reaches \(5e^{-4}\). We also use SGD with a momentum of 0.9 to optimize models.

### IV. Experimental setup

#### A. Benchmark

To evaluate our proposed CNN model, a recently available ASCAD database is used [3]. The power traces of ASCAD were recorded from an 8-bit ATMega8515 board while running a masked AES-128 implementation. To acquire the power trace, emitted electromagnetic emanation (EM) from the target device while the AES-128 platform was running. The meta-data ASCAD.h5 is composed of profiling power traces and the attack power traces where a profiling power traces contains 50,000 traces as a source of training data for deep learning architecture and another set of 10,000 traces from the attack test to validate the training architecture. Moreover, metadata ASCAD.h5 is composed of plaintext, ciphertext, key and mask which all are arrays of 16 unsigned 8-bit integers. Each power trace consists of 700 feature sets. Among 16 bytes of AES-128 \((i = \{k_1, k_2, \ldots, k_{16}\})\), first two elements of the mask array correspond to the unmask \(k_1\) and \(k_2\). Mask array is started from third byte, \(Z = SBox(p[i] \oplus k[i]) \oplus r[i]\) where \(p[i], k[i]\), and \(r[i]\) respectively denote the plain-text, secret key of \(i^{th}\) byte, and mask value. For instance, third byte of the mask state is represented by \(SBox[\text{state}0[3]] = SBox(p[3] \oplus k[3]) \oplus r_{out}\) with \(i = 3\) during the first round of AES-128 implementation. Thus, the value of the byte \(SBox[p[i] \oplus k[i]]\) is the label of deep learning, leading to 256 (number of bytes) possible classes. Moreover, possible classes (label) can be compressed by 9 classes using the Hamming Weight (HW) of \(SBox[p[i] \oplus k[i]]\). In other word, after calculating HW, each byte (class) will be converted into one of nine possible classes \((0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1)\). However, as can be seen in Fig 4, the distribution of each class is not equal. Among 256 bytes, 70 bytes have the hamming weight of 0.5; 56 bytes have the HW of 0.6250 and another 56 bytes’ HW is 0.375; 28 bytes have the HW of 0.75 and another 28 bytes’ HW is 0.25; 8 bytes have the HW of 0.875 and another 8 bytes’ HW is 0.125; bytes 0xFF and 0x00 have the HW of 1 and 0, respectively. Thereby, due to the uneven distribution of the Hamming weight classes, we will be using 256 classes in this paper.

#### B. Evaluation Metrics

The key rank is a common metric used as an assessment of SCA [3]. To evaluate the performance of our deep neural network for both image and signal-based, key rank is employed. Here we assume \(g\) is our deep learning model for evaluation and \(c\) is a metric, training set \(D_{train}\), which the deep learning is trained, and of the test set \(D_{test}\) that the model will be tested. Moreover, to evaluate model on unseen power traces,
Fig. 4: Hamming Weight calculation of entire 256 bytes. The 256 classes are converted into a total of 9 labels. Each label represents one class of deep learning that may be comprised of several bytes.

$k$-fold cross-validation is applied. The performance on $D_{test}$ is computed as follow:

$$\frac{1}{k} \sum_{i=1}^{k} c_i (\hat{g}, D_{test}) \tag{2}$$

Now, let assume sub-key ($k^*$ used during the side-channel measurement) denote by $k^* \in K$. The key rank corresponding to a trained model $\hat{g}$ with the profiled dataset $D_{train}$ and tested with the dataset $D_{test}$ is defined by:

$$\text{rank}(\hat{g}, D_{train}, D_{test}, n) = \left\{ k \in K \mid \overrightarrow{d}_n[k] > \overrightarrow{d}_n[k^*] \right\} \tag{3}$$

where $\overrightarrow{d}_n[k]$ is the score for the candidate $k$ as defined $\overrightarrow{d}_n[k] = \prod_{i=1}^{n} \bar{y}_i[k]$ with following equation:

$$\bar{y}_i[k] = \hat{g}_{\vec{l},p} \left( \vec{\ell}_i, p_i \right) [k] \tag{4}$$

where $\vec{\ell}_i$ represent new leakage. Thereby by replacing $D_{test}$ and $n$ with $D_{attack}$ and $N_a$, respectively, probability of model output will be obtained.

V. EXPERIMENTAL RESULTS

A. Training and Testing Data

Overall we have 50,000 profiling power traces and 10,000 from attack power traces. For fair evaluation, We performed 10-fold cross-validation for each deep learning model and measure the average key rank as the evaluation metric. Specifically, 80% of profiling power traces are considered as training, and 20% of profiling power traces have been utilized for validation. Note that we randomly select training and validation power trace sets out of the entire 50,000 trace pool for the experiments. Among 10,000 from attack power traces, 2,000 random power traces have been selected for testing our proposed deep learning model. Finally, we evaluate our model by measuring the average key rank.

B. Results and Discussion

Here we present all the results obtained from different deep learning architectures. As we discussed in Section IV-B, we use key rank as the metric for comparison. In other words, we compute the average key rank from the number of power traces. To compute Key rank, the probability of a successful target key byte recovery in the ID classification has been measured. Our goal is to recover the first four AES-128 SBox. An adversary can break the AES implementation if the key rank $\leq 10$ with the minimum required power trace. To further illustrate the generalization of our proposed deep CNN framework, the rank results from published ASCAD CNN and MLP best model are compared.
Fig. 5a is demonstrated the key rank based on 6 layers MLP, CNN, and our proposed ResNet architecture using power traces signal from ASCAD public dataset. We first compute the key rank using our proposed ResNet model where we need 240 traces to reach a key rank less than 10. We also compare our proposed CNN model with published ASCAD MLP with 6 layers and CNN with 5 layers of convolution. As can be seen in Fig. 5a, 5 layer ASCAD CNN model requires > 2,000 power traces to reach key rank less than 10 while the key rank of MLP model at the best scenario is ≈ 50. Our proposed deep learning-based attacks outperform in terms of key rank and number of power traces. Our proposed ResNet results surpass by far compared with ASCAD.

We also perform attacks using 2-D CNN using state-of-the-art models including VGG16, MobileNet, GoogLeNet, and Inception-V3. Fig. 5b shows key rank according to an increasing number of traces based on transformed power traces into an image structure. For reaching key rank<10 using 2-D CNN, an adversary needs respectively 240, 320, and 370 traces (image) using Inception-V3, GoogLeNet, and MobileNet. VGG16 is outperformed in key rank in the earlier of the stage (less number of power trace). However, by increasing the trace, the speed of reaching the rank<10 is much slower compared to other architecture. To the best of our knowledge, this is the first study that examines the 2-D CNN model against masked implementation.

One can conclude that our 2-D deep CNN architecture performs well against masked implementation compared to 1-D CNN. In other words, the number of traces needed to break the AES-128 implementation is less than the 1-D CNN model.

VI. CONCLUSION

In this study, we show that it is indeed possible to break the AES-128 implementation. First, we develop a novel 1-D ResNet CNN architecture model, where consists of 6 residual blocks with two convolutional layers for better propagation in the deep neural networks. Second, we explore the 2-D CNN model where the power trace signal (leakage from emitted electromagnetic emanation) is converted into the image using a continuous wavelet transformation function. Specifically, we adopt deep 2-D CNNs, GoogLeNet, InceptionV3, VGG16, VGG19, and MobileNetV2 pre-trained to identify all possible AES key bytes. To evaluate our proposed work, a key rank metric with the ASCAD dataset is utilized. Our proposed deep CNN framework achieves ≥ 99 accuracy when key rank<10.

REFERENCES