Performance Analysis of Non-Profiled Side Channel Attacks Based on Convolutional Neural Networks

Ngoc-Tuan Do[1], Van-Phuc Hoang[1*], Van-Sang Doan[2]
[1]Institute of System Integration, Le Quy Don Technical University, Hanoi, Vietnam.
[2]Vietnam Naval Academy, Nha Trang, Vietnam.
[*]Correspondence: phuchv@lqdtu.edu.vn

***Abstract*—There are emerging issues about side channel attacks (SCAs) on the cryptographic devices which are widely used today for securing secret information. Recently, the neural networks have been introduced as a new promising approach to perform SCA for hardware security evaluation of cryptographic algorithms. In this work, we present a non-profiled SCA using convolutional neural networks (CNNs) on an 8-bit AVR microcontroller device running the AES-128 cryptographic algorithm. We aim to point out the practical issues that occurs in CNN based SCA methods using the aligned power traces with a large number of samples. Furthermore, a method to build a suitable dataset for CNN training is introduced. Especially, practical experiment results of the CNN based SCA methods and a comprehensive investigation on the effect of noise are also presented. These experiments are performed with the original power traces and additive Gaussian noise. The results show that the CNN based SCA with our constructed dataset provides reliable results for non-profiled attacks. However, it is also shown that the Gaussian noise added on power traces becomes a serious problem.**

***Index Terms*—Non-profile side channel attack, AES, CNN**

I. Introduction

Nowadays, to ensure the communication between two electronic devices, the use of cryptographic algorithms is so popular. Although these mathematically secure algorithms cannot be broken by the brute-force attacks, there have been numerous accounts of breaking the confidential key by exploiting side-channel information such as power consumption, electromagnetic radiation, or acoustic vibrations captured from cryptographic devices. Side channel attack (SCA) is divided into two approaches, which are called profiled and non-profiled attacks. The most widely used profiled attack is the template attack. Traditionally, template attacks are implemented by generating templates for difference keys by utilizing a multivariate Gaussian distribution approximation of the pre-identified points of interest (POIs) [1], [2]. However, it is challenging for attackers to perform this attack in practice. Template attacks require the more complicated installation than other attacks; therefore, the attackers must have access to another copy of the protected device that they can fully control. Then, they must perform a great deal of pre-processing to create a template in practice, which may take a huge number of power traces. The advantage is that template attacks need only a small number of power traces from the victim to complete. In terms of non-profiled attacks, the most used attack is the correlation power analysis (CPA) proposed by Brier et al. [3]. It exploits the correlation between the power model and the real power consumption in order to extract the secret key of the cryptographic algorithm. Especially, CPA does not require a copy of the target device, so it is easy to perform this attack in practice.

Recently, the hardware security research community has focused the attention on the machine learning (ML) based SCAs. However, so far, their works have only focused on applying deep learning (DL) techniques to perform the profiled SCA. As mentioned above, implementing a profiling attack requires the access to a profiling device, which is strongly assumed that it can not always be met in practice. Therefore, a profiled attack may not be performed. However, non-profiled attacks such as CPA can still frighten the target. Indeed, applying ML to perform a non-profiled attack is a new approach for cryptographic analysis. In this paper, we focus on the DL based non-profiled attack and evaluate its efficiency on the attack with a large number of power traces and samples.

Our work is related to some previous ones as follows. Firstly, the data preparation is done by using correlation coefficients to extract the most relevant features. It exploits the correlation between the power model and real power consumption. The combination of correlation coefficient and convolutional neural network (CNN) for attacking was used in [4], but their works are only for the profiling attack. Most recently, in [5], the authors show that it is possible to exploit the advantages of DL and neural networks in the non-profiled scenario. They introduce metrics based on sensitivity analysis that can leak both secret key value as well as points of interest, such as leakages and masks locations in the traces. The new attack approach using those metrics is called the differential DL network (DDLA). The author presents the efficiency of this technique on both synchronized and non-synchronized power traces. However, in terms of synchronized traces, the author only uses a small number of power traces with only 500 samples containing the copy of S-box function in memory. This means that the attackers must know clearly about the Advanced Encryption Standard (AES) algorithm and be able to point out which samples of power traces are corresponding to S-box function of AES algorithm processed on chip. Especially, a technique called Hamming Weight (HW) labeling is mentioned in both [4] and [5]. However, the authors in [4], [5] applied the HW on intermediate values resulting in nine classes. In this paper, we assume that attackers do not know much about AES algorithm. They must record a large

978-1-7281-9396-0/20/$31.00 ©2020 IEEE

number of samples on each power trace. In this case, we take 10,000 samples that contain the whole process of the fist round and a part of the second round in the AES encryption. We also use HW labeling for CNN, but only three classes are used instead of nine classes. Furthermore, we aim to perform more experiments with different levels of Gaussian noise added to the power traces in order to illustrate the impact of noise on CNN based non-profiled techniques.

The rest of this paper is organized as follows. In Section II, data preparation is described in detail. Section III presents an overview of CNN and introduces our proposed CNN model to deal with the constructed dataset in the non-profiled scenario. Then in the next section, we will give detailed results from experiments implemented on power traces collected from the ChipWhisperer-Lite board. Finally, we conclude the paper in Section V.

II. Data Preparation

Power traces are vectors of voltage values recorded by the digital sampling oscilloscopes. They often contain thousands of samples. Consequently, the number of features is too large for CNN technique, and it is one of two factors comprising the time complexity for the algorithm. Therefore, we conduct the feature selection taking the 50 most correlation coefficient values. To select those features, we use Pearson correlation coefficients. This method was used in [4] but the novelty of our proposed approach is the use of only three HWs instead of nine HWs values. More interestingly, this method leads to approximately 30% reduce of number of power traces needed for the attack.

$$h_{n,k} = HW\left(S-box\left(Plaintext_n \oplus k\right)\right) \qquad (1)$$

We use N random plaintexts corresponding to N power traces in which each power trace has L samples. Note that $t_{i,j}$ is the value of j^{th} sample in the trace number i^{th} ($1 \leq j \leq L$, $1 \leq i \leq N$), $d_{i,B}$ is the byte value of byte B ($B \in [1;16]$) in the plain-text number i^{th}. In order to collect 50 features from the power traces, a half of number of power traces is used and denoted as $N_1 (N_1 = \frac{1}{2}N)$. In this case, S-box function of AES algorithm as shown in (1) is chosen as target for calculate HW. After that, (2) is applied to calculate the 50 highest samples from all guess keys ($Key = [0; 255]$).

$$\rho_{k,i} = \frac{\sum\limits_{n=1}^{N_1} (h_{n,k} - \overline{h}_k)(t_{n,i} - \overline{t}_i)}{\sqrt{\sum\limits_{n=1}^{N_1} (h_{n,k} - \overline{h}_k)^2 \sum\limits_{n=1}^{N_1} (t_{n,i} - \overline{t}_i)^2}} \qquad (2)$$

where $\overline{h}_k$ and $\overline{t}_i$ are the average of the power consumption model and real power consumption at instant i, respectively.

Consequently, we have 16 folders corresponding to 16 subbytes of a secret key. Each folder contains 256 subfolders which were created from 256 values of guess key. In a subfolder, three folders named HW3, HW4, HW5 were used as three labels for the CNN. Finally, for each label folder, the power traces were partitioned, which have the same intermediate value (HW = 3, 4, 5) for each hypothesis key. Fig. 1 illustrates the dataset in detail.

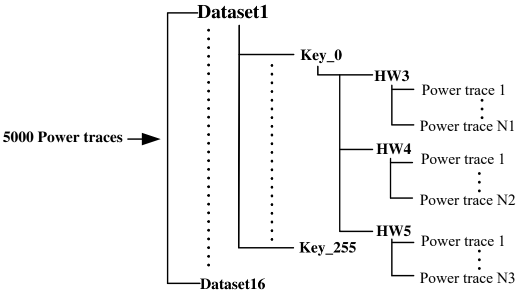

Fig. 1. The dataset construction: 5000 original power traces (10,000 samples/trace) are calculated and partitioned in three groups (HW3, HW4 and HW5). Each group contains N_1, N_2, N_3 power traces and each power trace has 50 samples which are highest correlation values.

III. Proposed CNN Model for Non-Profiled Attack

One of the most widely used neural networks is CNN which is mainly used for image recognition [6]. However, they have demonstrated to be a powerful classifiers for time series data [7]. CNNs have a natural translation-invariance property and suitable models for detailed features extraction and tackling complicated data classification models. Therefore, CNNs are particularly interesting for our work to perform attacking the side-channel data. However, one disadvantage of the CNN technique is that it is necessary to perform a CNN training for each key hypothesis. In our scenario, we use 8-bit key guesses, it means that 256 trainings are needed.

CNNs usually combine two types of layers called convolutional layer and pooling layer. These layers are today often completed with a so-called Batch Normalization Layer. In [8], Ioffe Szegedy has introduced Batch normalization to reduce the so-called internal covariate shift in the neural network. The authors also show that it allow for the usage of higher learning rate. In this work, we investigate the performance of CNN with a large number of aligned power traces. However, since the sample of one power trace is too big, we cannot use them as input features for CNN. Therefore, we use the correlation coefficient as a pre-processing technique to extract the main features of these power traces. The datasets used in CNN are usually divided into three distinct sets: training set, as the name implies, to train the network, validation set to validate the performance of the trained network on previously unseen data which are kept separately, and a test set, which is used to test the prediction or classification accuracy at the end. Next, all the details for the proposed network architecture will be shown.

Our neural network models were implemented on MATLAB software for evaluation. The network is composed of three convolutional layers and three pooling layers in the middle, followed by the fully connected and classification layers. For the first convolution layer, we use 16 filters with the size

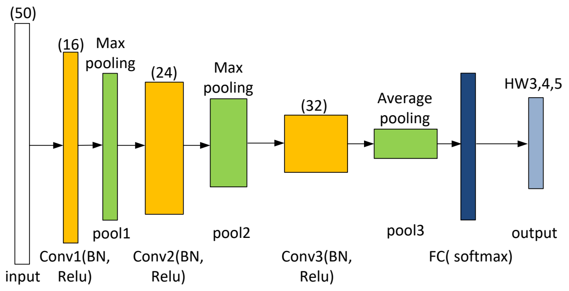

Fig. 2. The proposed CNN architecture.

of [1 3], stride of [1 2], and the output layer has the same size as the input. The numbers of filters in the next two convolutional layers are 24 and 32, respectively and have the same size as the first one. Convolutional layers perform the convolution operations to the input by sliding a set of filters along the preceeding layer. The filter weights allow CNN to learn translation-invariant features, and they are adjusted in order to minimize the loss function. In terms of the SCA, filters will be slid along the power traces. For pooling, we use Max-pooling on the two first pooling layers and Average pooling on the last pooling layer, using kernel size of [1 2] and stride [1 2]. The pooling layers, such as max-pooling or average pooling, are non-linear layers and used to reduce the number of dimensions. Max-pooling provides the maximum value of each area, and average-pooling produces the average of each area. In our model, we use Rectified linear activation function (RELU) for all convolutional layers. It is a piecewise linear function so that the output will be equal to the input if the input value is positive. Otherwise, it will provide the zero output. For classification, we use Softmax function in the output layer. The details our CNN architecture are described in Fig. 2 and Table I.

TABLE I
DEVELOPED CNN ARCHITECTURE.

Layer	Weight shape	Stride	Activation
Convolutional(1)	1 x 3 x 16	-	-
Batch Normalization(1)	-	-	ReLu
MaxPooling(1)	-	[1 2]	-
Convolutional(2)	1 x 3 x 24	-	-
Batch Normalization(2)	-	-	ReLu
MaxPooling(2)	-	[1 2]	-
Convolutional(3)	1 x 3 x 24	-	-
Batch Normalization(3)	-	-	ReLu
Average-pooling (1)	-	[1 2]	-
FC-output	-	-	Softmax

IV. Experimental Results

We have performed the experiments on CW1173 ChipWhisperer board [9]. This SCA platform includes a target board with an 8-bit Atmel AVR Xmega128 microcontroller running AES-128 algorithm. ChipWhisperer provides a capture setup CW Lite capture using an onboard ADC. This setup allows us to send program, plaintext, and the key to Xmega board and record captured traces directly from a laptop. We have collected 5000 power traces for experiments. Each power trace contains 10,000 samples corresponding to Round 1 and apart of Round 2 of the AES algorithm. In non-profiled attack scenario, we keep the key fixed and 5000 plaintexts random, which is opposite to the profiled attack. Only the first round of AES is attacked in all of our results.

Then, we perform a CNN training using 80% of the dataset as the training data, and the rest for testing. It can be seen that, by taking the highest correlation values among all values for i and k, we can take the features which are useful for classifying HW labels in CNN model. For the correct key value k, the series of intermediate HW values will be correctly guessed, and therefore the labels used for CNN training will be accurate. If the CNN is able to learn the targeted features from the correct key, this should lead to a successful training and good training metrics such as a decreasing loss and increasing accuracy over the epochs. On the other hand, for all the other key candidates, the series of intermediate values will be incorrect, and this should lead to unsuccessful training. The attacker can discriminate the correct key value from the other candidates by selecting the key leading to the best training metrics. The details of experimental results are described as follows.

To demonstrate the efficiency of applying CNN for a non-profiled attack, the validation accuracy of the training network is used as a criterion. As presented above, the two main metrics that can be used to monitor the CNN training process are the loss and the accuracy of the training over epochs. In this paper, we pointed out that the accuracy can be used to discriminate the correct sub-byte key. As an illustration, we present in Fig. 3a the validation accuracy obtained when performing an attack with our dataset with $n_e = 30$ epochs per guess. In the graph the vertical axis represents the validation accuracy of training network, the horizontal axis show the number of training epochs. It is clear that our training using the correct sub-byte key value leads to a higher validation accuracy compared to the others. Furthermore, we can see that the attack is possible to find the confidential key after first ten epochs. From the accuracy of each sub-byte key guess, it is easy to take the highest accuracy value corresponding to the correct key guess. More interestingly, we can also use confusion matrix to discriminate the distribution of three HW labels, the incorrect candidates will be classified in HW4, but the correct one is different, as show in Fig. 3b, c.

After obtaining these results, we decided to further investigate the effect of noise in power traces on the accuracy of the proposed CNN. Three different levels of Gaussian noise are added to original power traces. Then, three different datasets were created. The results of the training process are shown in Fig. 4. It is clear that with the small level of noise, our proposed CNN still shows good performance in detecting the correct key after the first ten epochs, as shown in Fig. 4a. However, when the deviation of noise increases, the validation

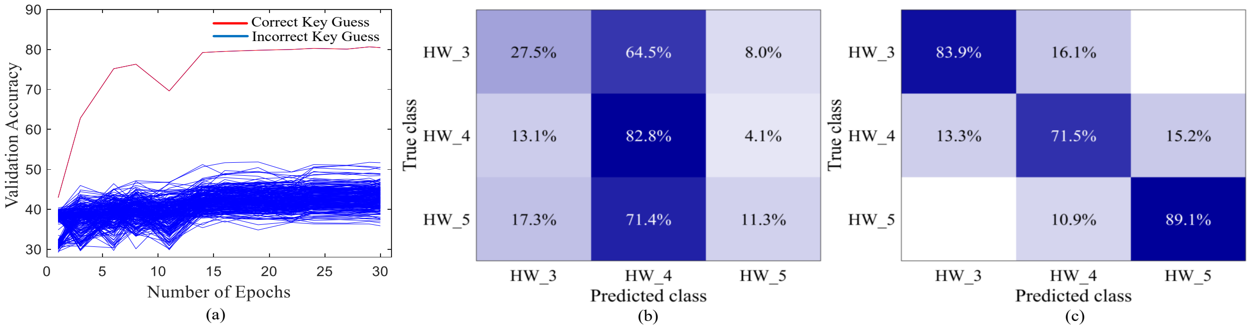

Fig. 3. Results of validation accuracy and confusion matrix with clean power traces. a) Validation accuracy; b) Incorrect key guess; c) Correct key guess.

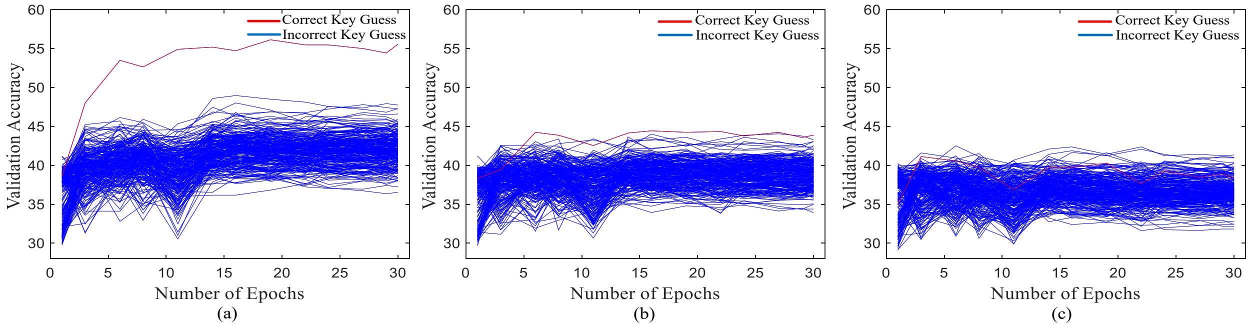

Fig. 4. Results of validation accuracy with three different deviations of Gaussian noise added: a) 0.025, b) 0.05, c) 0.075.

accuracy decreases significantly, and after all epochs using more than 3000 power traces, the correct key is hidden inside the remaining key byte hypothesis.

V. Conclusion

In this paper, we have shown the practical issues that occur in SCA using CNN for aligned power traces which have a large number of samples. Then, a data preparation method for CNN training was introduced. Our experiments were performed with both the original power traces and the power traces with the added Gaussian noise. The results show that our data preparation technique is capable to extract the good features for the non-profiled SCA based on CNNs. Since the power traces are partitioned in only three groups, our technique helps to reduce the number of power traces used for attacks. Furthermore, our proposed CNN architecture provides the reliable results for non-profiled attacks, but they also show that the Gaussian noise added on power traces is a serious problem. In the future work, we will investigate some pre-processing methods to reduce the effect of noise in power traces to increase the performance of neural networks for non-profiled attacks.

Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.02-2020.14.

References

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

[2] M. O. Choudary and M. G. Kuhn, "Efficient, portable template attacks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 490–501, 2018.

[3] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems - CHES 2004*, M. Joye and J.-J. Quisquater, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 16–29.

[4] S. Picek, I. P. Samiotis, J. Kim, A. Heuser, S. Bhasin, and A. Legay, "On the performance of convolutional neural networks for side-channel analysis," in *Security, Privacy, and Applied Cryptography Engineering*, A. Chattopadhyay, C. Rebeiro, and Y. Yarom, Eds. Cham: Springer International Publishing, 2018, pp. 157–176.

[5] B. Timon, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 2, pp. 107–131, Feb. 2019.

[6] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *ArXiv e-prints*, 11 2015.

[7] A. oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 09 2016.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015, pp. 448–456.

[9] C. O'Flynn and Z. Chen, "Chipwhisperer: An open-source platform for hardware embedded security research," vol. 8622, 04 2014.