Deep Learning-based Side-Channel Analysis on PIPO*

Ji-Eun Woo[1][0000−0003−2139−7804], Jaeseung Han[1][0000−0001−7111−2315], Yeon-Jae Kim[1][0000−0003−0405−8026], Hye-Won Mun[1][0000−0002−8608−2128], SeongHyuck Lim[1][0000−0002−9166−3402], Tae-Ho Lee[1][0000−0003−0892−5414], Seong-Hyun An[1][0000−0002−0325−1214], Soo-Jin Kim[1][0000−0001−5001−1574], and Dong-Guk Han[1,2][0000−0003−1695−5103]

[1] Department of Financial Information Security, Kookmin University, Seoul, Republic of Korea
[2] Department of Information Security, Cryptology, and Mathematics, Kookmin University, Seoul, Republic of Korea
{dnwldms928,jae1115,duswo0024,qwerty25879,seonghyeck16, 20141932,ashtree,suzin22,christa}@kookmin.ac.kr

Abstract. As the global IoT market increases, the importance of security in the IoT environment is growing. So, studies on lightweight cipher techniques are actively underway for limited environments. In ICISC 2020, PIPO, a bitslice lightweight cipher that can effectively apply a countermeasure considering side-channel analysis, was proposed. In this paper, we propose Deep Learning-based profiled and non-profiled Side-Channel Analysis for PIPO. In profiled attack, we use an 8-bit model instead of 1-bit model that considered the bitslice characteristic of S-Box output. Although an each bit of S-Box output is distributed across the power trace, the 8-bit model has shown high training performance with 98% accuracy, and was able to derive right key successfully. In non-profiled attack, we propose a labeling technique suitable for the bitslice characteristic and show the excellence of our proposed labeling through experiments. Also, we expect that these characteristics will apply to other bitslice block ciphers as well as PIPO.

Keywords: Side-Channel Analysis · Deep Learning · PIPO · Block Cipher.

1 Introduction

Recently, with the development of the Internet of Things (IoT), IoT devices, such as wearable biometric sensors and smart home devices, have increasingly

* This work was supported by Institute for Information communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00520, Development of SCR-Friendly Symmetric Key Cryptosystem and Its Application Modes).

become common in daily life and are widely used worldwide. However, IoT devices have limitations, such as the size of the hardware and power consumption. To implement encryption in such a limited environment, studies on lightweight block ciphers are increasing. However, these lightweight block ciphers may be vulnerable to Side-Channel Analysis.

Side-Channel Analysis (SCA) is an technique that recovers secret information by using side-channel information(e.g., power consumption, sound, and electromagnetic leaks) generated while encryption is performed on the target devices. In addition, SCA that uses the power consumption can be classified as profiled SCA and non-profiled SCA.

Profiled SCA creates a profile using a device that is very similar to a target device and finds secret information by matching the power traces obtained from the target device with the created profile, e.g., Template Attacks [1]. Non-profiled SCA derives secret information through statistical analysis of the power traces obtained when encrypting with the same secret keys and random plaintexts on the target device, e.g. Correlation Power Analysis (CPA), Differential power analysis (DPA) [3].

Furthermore, as deep learning techniques advance, studies on deep learning-based SCA are also increasing. Deep learning-based SCA derives secret information by training neural networks such as Multi-Layer Perceptron (MLP) and Convolution Neural Network (CNN) with intermediate values corresponding to side-channel information. In the case of profiling SCA, the intermediate values of the attack traces on target devices are derived using the trained neural network. In non-profiling SCA, the secret key is determined by the training performance of the neural network for each guessed key.

In this work, we propose deep learning-based profiled and non-profiled SCA on bitslice lightweight block cipher PIPO-64/128. In profiled SCA, we show that the neural network trained well by extracting features of power traces when we use an ID leakage model as label. Although the 1-bit model of S-Box output is considered when analyzing bitslice block cipher, we show that features are well extracted even using our model. In non-profiled SCA, we propose the improved labeling method on bitslice block cipher. And we demonstrate the excellence of our proposed labeling by comparing it to existing.

2 Background

2.1 PIPO: Bitslice Lightweight Block Cipher

PIPO (Plug-In Plug-Out) is a bitslice lightweight block cipher with a Substitution Permutation Network (SPN) structure proposed in 2020 [2]. It provides excellent performance in 8-bit AVR software with a bitsliced implementation. PIPO supports a 64-bit block size and uses an S-Box with 8-bit input and output. There are two variants of PIPO based on different key sizes (PIPO-64/128, PIPO-64/256), as shown in Table 1.

Table 2 shows the definition of notations used throughout this paper. For PIPO-64/128, the master key K is split into two 64-bit subkeys K_0 and K_1,

Table 1: Types of PIPO

	Master Key size	The number of rounds
PIPO-64/128	128-bit	13
PIPO-64/256	256-bit	17

i.e., $K = K_1||K_0$. The round keys are then defined as $RK_r = K_{r \bmod 2}$ where $r = 0, 1, \cdots, 13$. Similarly, for PIPO-64/256, the master key K is split into four 64-bit subkeys K_0, K_1, K_2, and K_3, i.e., $K = K_3||K_2||K_1||K_0$, and the round keys are $RK_r = K_{r \bmod 4}$ where $r = 0, 1, \cdots, 17$.

Table 2: Definition of notations

Notation	Definition
$\oplus$	XOR(eXclusive OR) operator
$\gg$	right shift operator
&	AND operator
$\|\|$	bitwise concatenation operator
$m \bmod n$	the remainder when m is divided by n
K	master key
RK_r	r-th Round Key
S_i	i-th byte of S-Layer output
$s_{i,j}$	j-th bit of S_i

Each round consists of non-linear S-Layer, linear R-Layer, and Key Addition. The structure of PIPO is shown in Figure 1.

Specifically, the non-linear S-Layer applies a bitslice implementation. Block ciphers in a bitslice structure can process multiple S-Box operations in parallel without table reference, since S-Box is implemented as a bitwise logical operation. Therefore, for 8-bit input and output of S-Box both are stored in different registers and operated in parallel, so as shown in Figure 2, the operation is performed by storing the i-th input bits of each S-Box in the i-th byte ($0 \leq i \leq 7$).

2.2 Supervised Learning using Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) is a feed-forward neural network with multiple layers, and each layer has multiple perceptrons [4]. It forms a fully connected structure in which each perceptron of the previous layer and the perceptrons of the next layer are all connected. As shown in Figure 3, MLP consists of an input layer, hidden layers, and an output layer. MLP can be learned through a data set given a large amount of data and corresponding labels (correct answer), which is called supervised learning [5]. The learning process is the following:

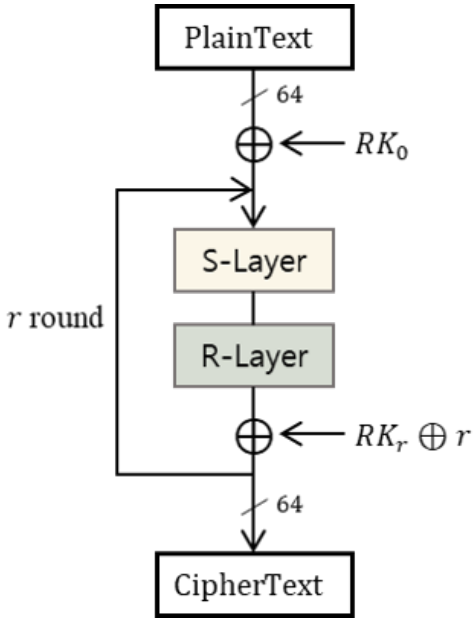


Fig. 1: PIPO structure

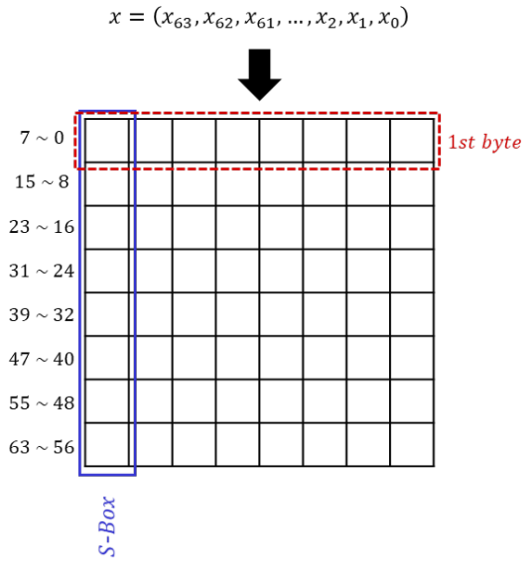


Fig. 2: S-Layer structure

Calculate the loss value between the label for the training data and the output value of the MLP, and update the weights and bias of the neural network through a backpropagation algorithm in the direction of reducing the loss value. In this process, an overfitting problem that cannot predict well new data may occur, since a neural network model can be trained to fit only the training data. Therefore, to evaluate the predictive performance of the MLP, we monitor the overfitting using validation data which is independent of the training data [6]. If overfitting occurred, we then modify the hyper-parameters to improve the generalization performance of the MLP.

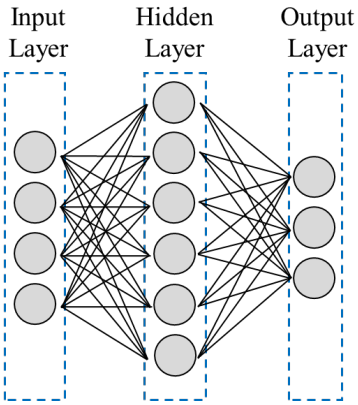


Fig. 3: The structure of MLP

2.3 Deep Learning-based profiled Side-Channel Attack

Profiled SCA is a technique of generating the profile through a profiling device which is similar the target device, then analyzing the secret information by matching the side-channel information of the attack target device with the profile of profiling device. In 2013, Martinasek et al. proposed Deep Learning-based profiled SCA (DL-based profiled SCA) [9]. The analysis process of DL-based profiled SCA is divided into profiling and attack phases. In the profiling phase, the neural network is trained by setting the input as the training traces and the output as the label about the target operation intermediate value. In the attack phase, the attack trace is input to the trained neural network and the intermediate value is restored through the output of the neural network. Martinasek et al. were set the label through the one-hot encoding of an intermediate value and performed DL-based profiled SCA for AES [9]. As above, in DL-based profiled SCA, typically, the one-hot encoding value of the intermediate value is used as a label.

2.4 Deep Learning-based non-profiled Attack

Non-profiled SCA is a technique for analyzing secret information using multiple side-channel information collected during the encrypting for the random plaintexts with the same secret key. In 2019, Benjamin proposed a Deep Learning-based non-profiled SCA (DL-based non-profiled SCA) [7]. The analysis process of DL-based non-profiled SCA is as follows. First, for each guess key, the neural network is trained by setting the input as the traces and the output as the label value for the target intermediate value. Then, since the label calculated by the right key is a value related to the traces, the neural network is well trained. However, since the label calculated by the wrong key is a value unrelated to the traces, the neural network is not well trained. We analyze the secret key by judging the guess key that trained the neural network best as the right key. In DL-based non-profiled SCA, we also prefer that the intermediate value of wrong keys have a low correlation with the intermediate value of the right key like the traditional non-profiled SCA, so it takes generally the output of the non-linear operation to the intermediate value. Timon was set the LSB or MSB of the S-Box output as the label and performed DL-based non-profiled SCA for AES [7].

3 DL-based profiled SCA on PIPO

3.1 Attack Scenario

Profiling Phase

The profiling phase is the process of training neural networks using power consumption collected during the encryption of PIPO-64/128. In this paper, a neural network is constructed using an MLP model. The PIPO S-Layer is implemented in parallel so that each byte of S-Box output is not all stored in the same register, but power consumption information is distributed. However, since an MLP has a fully connected layer structure, it is expected that the features of the corresponding byte can be extracted even if the information is distributed. Therefore, to recover one byte of the first round key of PIPO-64/128, the power consumption should be set as the input of the neural network, and the output of the PIPO S-Layer is set as the label. In order to recover the entire first round key RK_0, we have to repeat this for all bytes.

Attack Phase

The attack phase is the process of finally recovering the key of PIPO-64/128 by putting the attack trace as an input into the trained neural network. By putting attack power consumption as input into one neural network obtained from the training phase, one byte of the output of the PIPO S-Box can be recovered. As shown in Figure 4, the result of performing an inverse S-Box operation on the recovered S-Box output and an XOR operation with the plaintext is considered as the correct key. This process can be done on all bytes to recover the entire first round key RK_0, and the master key K can finally be recovered when both first and second round keys are found.

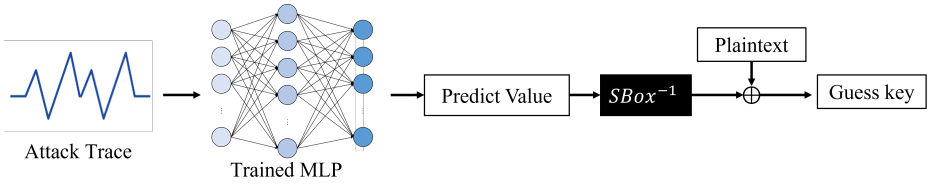


Fig. 4: Attack phase

Performance metric

In order to evaluate whether the attack stage was well performed, the ratio of the number of correct guess keys among a total of 1,000 attack traces collected with random plaintexts and random keys is used as a performance metric. For example, if the attack phase is performed on 1,000 attack traces and 500 keys are matched, the success rate of the attack is 50%.

3.2 Experimental Result

In this section, we present our experiment results that apply DL-based profiled SCA for PIPO-64/128.

Experimental Environment

For the experiment, we obtain the power traces from PIPO-64/128 1st round S-Layer to R-Layer when encrypted 10,000 times with a random key and a random plaintext in the experimental environment as Table 3. Power traces were divided into 9,000 profiling dataset and 1,000 attack dataset, and 10% of the profiling dataset was used as the validation data set for verification in the profiling phase.

We target S_i of first round, the output of S-Layer, and use the identity (ID leakage) model as our power model, so there are 256 classes. Thus, we set the number of the last node in neural network to 256. Also, label is target value's one-hot encoding value.

Table 3: Experimental Environment

Target Board	ChipWhisperer-Lite
Target Chip	Atmel XMEGA 128 (8-bit processor)
Sampling Rate	29,538 MS/s
Tensorflow version	1.15.0
Keras version	2.3.1

MLP Architecture

This section describes the our MLP model used in the experiment. It has three hidden layers with 150, 100 and 50 nodes. also, activation function of the hidden layer used "ReLU" and the output layer used "Softmax". The number of nodes in the input layer is 1,260, which is the number of points in the power traces, and the number of nodes in the output layer is 256, the number of possible target values. Each hidden layer and input layer include a batch normalization and a dropout to prevent overfitting.

Table 4 shows our MLP architecture, and detail of hyperparameters are shown in Table 5.

Table 4: MLP on DL-based profiled SCA

Layer	Node (in, out)	Kernel initializer
Input	(1260,1260)	-
Dense	(1260,150)	he uniform
ReLU	(150,150)	-
Dense	(150,100)	he uniform
ReLU	(100,100)	-
Dense	(100,50)	he uniform
ReLU	(50,50)	-
Dense	(50,256)	he uniform
Softmax	(256,256)	-

Table 5: Hyperparameters

Label	One-hot encoding (8-bit) of S-Layer output
Optimizer	Adam (learning rate = 0.001)
Batch Size	32
Epochs	100
Loss function	Categorical cross entropy
Dropout	0.1

Experimental Result

This section presents the experimental results when each byte of RK_0 is recovered. Figure 5 shows the training, validation loss(left) and the training, validation accuracy(right) of first byte of RK_0. In Figure 5, The x-axis of the graph represents the number of epochs, and the y-axis represents the loss or accuracy. Also, the blue line is result of training and orange line is result of validation.

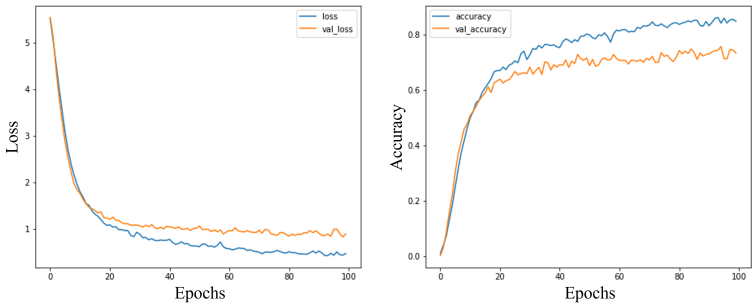


Fig. 5: Result of the first byte of RK_0

The maximum validation accuracy is 75%, and the correct guess rate when performing an attack on 1,000 attack data set is 72.8%. Table 6 shows the maximum validation accuracy and correct guess rate for RK_0. We can see that every byte has more than 60% validation accuracy, and the correct guess rate is similar. Therefore, RK_1 is recovered in the same way, the master key K for PIPO-64/128 can be recovered using RK_0, RK_1.

Table 6: Results of all bytes of RK_0

Byte	0	1	2	3	4	5	6	7
Maximum validation accuracy(%)	75	66	77	75	90	60	79	98
Correct guess rate(%)	72.8	67.4	76.8	71.8	87.1	56.4	78.3	97.1

Due to the bitslice structure characteristics of the PIPO S-Layer, each bit of the S-Box output is distributed across the power traces. Nevertheless, it can be seen that the MLP model learns by extracting features by itself, even if the label is composed using the 8-bit model of the S-Box output value rather than the 1-bit model. We expect to be applicable to not only PIPO but also other bitslice block ciphers.

4 DL-based non-profiled SCA on PIPO

4.1 Attack Scenario

Since PIPO-64/128 is also a block cipher using an S-Box, DL-based non-profiled SCA can be performed by similarly applying Timon's analysis [7]. The DL-based non-profiled SCA process is as follows.

– Set the input of the MLP to the power traces and the output to the label value (MSB or LSB, etc.) of the S-Box 1 byte output about the arbitrary guess key.

– After training is performed for all guess keys, the best-trained guess key is determined as the right key.

The learning metric uses the training loss value. That is, the guess key with the lowest final loss value is judged as the right key. Timon used the MSB or LSB value as the label value of DL-based non-profiled SCA for AES [7]. However, unlike AES, PIPO-64/128 is a bitslice block cipher. Therefore, in this paper, we propose a new labeling method considering the bitslice structure. Figure 6 shows

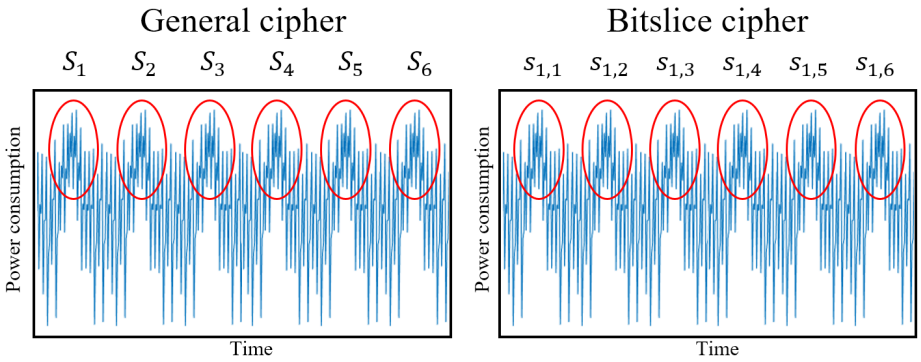


Fig. 6: General cipher's trace vs Bitslice cipher's trace

the difference in the power trace of the general cipher and the bitslice cipher. In the general cipher, all bits of the S-Box output are exposed at a single time point.

Therefore, in DL-based non-profiled SCA for general cipher, single-bit labeling or HW labeling in which multiple bits overlap is appropriate. However, in the traces of the bitslice block cipher, only a single-bit of the S-Box output is exposed at a single time point. In this case, the power information about each bit of the S-Layer output is data in an independent time point. Therefore, we propose binary encoding labeling that each single-bit is encoded as an independent label in DL-based non-profiled SCA for bitslice block cipher.

Algorithm 1 shows the proposed binary encoded labeling algorithm.

4.2 Experimental Result

In this section, we present our experimental results that apply DL-based non-profiled SCA for PIPO-64/128. For each labeling method, we derived the minimum number of traces required to DL-based non-profiled SCA and compared the performance of the methods.

Algorithm 1 Binary encoded labeling algorithm

Input : a 8-bit value x

Output : a binary encoded label $y = (x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7), x_i \in \{0, 1\}$

1: **for** $i = 0$ to 7 **do**

2: $\quad x_i \leftarrow (x \gg i)\ \&\ 1$

3: **end for**

4: **Return** $y = (x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)$

Experimental Environment

We analyzed using the PIPO-64/128 non-profiled analysis open data set[8]. The library version used in the experiment is 2.6.0 for Tensorflow and 2.6.0 for Keras.

MLP Architecture

This section describes the our MLP model used in the experiment. It has one hidden layers, with the number of layer nodes is 200. Activation function of the hidden layer used "ReLU" and the output layer used "Sigmoid". The number of nodes in the input layer is 2,200, which is the number of points in the power traces, and x (the number of nodes in the output layer) is 1 (MSB labeling) or 8 (binary encoding labeling), the number of possible target values. Each binary encoding value is not independent of the other. For example, when let $y_1 = (0, 0, 0, 1), y_2 = (0, 0, 1, 1), y_3 = (1, 1, 1, 0)$, y_1 is closer with y_2 than y_3 because y_1, y_2 differ only one element but y_1, y_3 differ all element. Therefore, we did not apply one-hot encoding about labels in non-profiled SCA and the target of our MLP model is the regression problem. So we choose mean squared error as the loss function. Table 7 shows our MLP architecture, and detail of hyperparameters are shown in Table 8.

In our experiments, since the performance of the MSB labeling was better than LSB labeling, we focused on comparing MSB labeling with our proposed labeling method.

Table 7: MLP on DL-based non-profiled SCA

Layer	Node (in, out)	Kernel initailizer
Input	(2200,2200)	-
Dense	(2200,200)	he normal
ReLU	(200,200)	-
Dense	(200,x)	he normal
Sigmoid	(x,x)	-

Table 8: Hyperparameters

Label	Binary encoding (8-bit), MSB of S-Layer output
Optimizer	Adam (learning rate = 0.0001)
Batch Size	100
Epochs	1000
Loss function	Mean Squared Error

Experimental Result

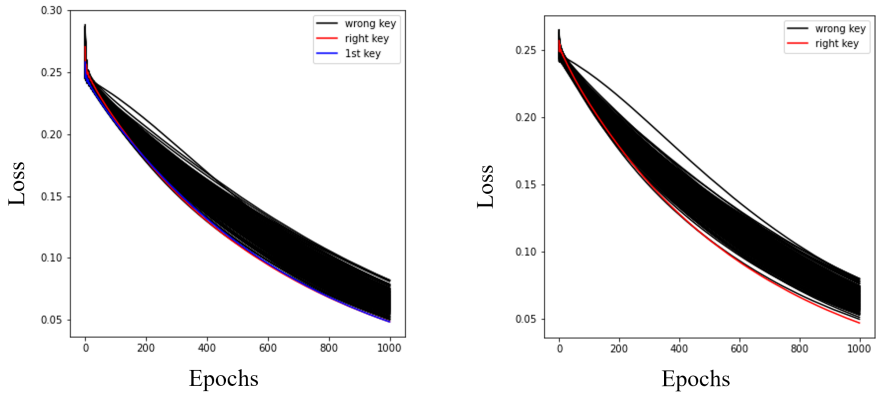
We defined "number of traces required for analysis" is the number of traces when the right key has the lowest loss value. Figure 7 is the learning result of first byte of RK_0 on the binary encoding labeling neural network, and it is the result of learning with 60, 70, and 80 traces, respectively. The x-axis of the graph represents the number of epochs and the y-axis represents the loss value. In the graph, the black line is the wrong keys, the red line is the right key, and the blue line is some wrong key that has lower loss than the right key. At 60 traces, the analysis failed because there was some wrong key with a lower final loss than the right key, but at 70 traces, the analysis succeeded because the final loss of the right key was the lowest. Additionally, at 80 traces, the final loss of the right key decreased comparing a result of 70 traces. As such, we repeated the analysis with a trace of 10 units, and we find a minimum number of traces required for analysis by each labeling method. Table 9 shows the minimum number of traces by each binary encoding, and MSB labeling method in the first round key analysis. As a result, the average minimum number of traces for binary encoding is 86.75 and that of MSB is 153.75. Consequently, binary encoding labeling has minimum analysis traces that about 56.4% of MSB labeling.

Table 9: Minimum number of traces required for analysis by each labeling method in RK_0

Label\Byte	0	1	2	3	4	5	6	7	Average
Binary encoding	70	90	90	100	100	100	70	70	86.75
MSB	170	160	190	160	130	140	120	160	153.75

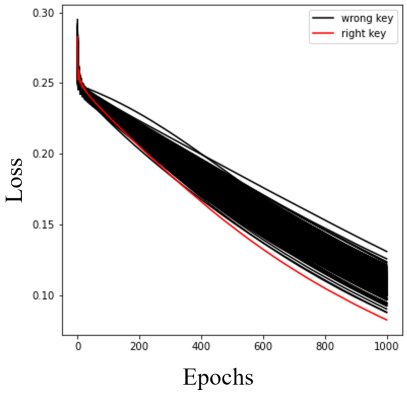
5 Conclusion

In this paper, we propose deep learning-based profiled and non-profiled side-channel analysis for PIPO-64/128. For DL-based profiled SCA, the PIPO S-Layer output's one-hot encoding value is used as a label. As a result, all bytes of first round key show a validation accuracy greater than 60% and up to 98%. In addition, the attack phase uses the ratio of the number of correct guess keys



(a) Result at 60 traces

(b) Result at 70 traces



(c) Result at 80 traces

Fig. 7: Result of first byte of RK_0 on DL-based non-profiled SCA by binary encoding labeling

among a total of attack traces to evaluate attack performance. The ratio of correct guessing on attack dataset is up to 97%, which is similar to the validation accuracy. Thus, the right key could be recovered with an 8-bit model even though the S-Layer output was distributed across the power trace by bit slicing. On the other hand, for DL-based non-profiled SCA, we use binary encoding of PIPO S-Box output as the label. And the training loss value was used to evaluate the training performance. We show that when using binary encoding as the label, all bytes can be recovered with 100 power traces less than when using MSB label.

We expect that our experimental results can also be applied to (high-order) analysis on other block ciphers which are either bitslice-based or not. Thus, in future work, we plan to analyze other block ciphers using the proposed DL-based profiled SCA and DL-based non-profiled SCA.

References

1. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Jr., B.S.K., Koç, Ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2523, pp. 13–28. Springer (2002). https://doi.org/10.1007/3-540-36400-5_3, https://doi.org/10.1007/3-540-36400-5_3
2. Kim, H., Jeon, Y., Kim, G., Kim, J., Sim, B., Han, D., Seo, H., Kim, S., Hong, S., Sung, J., Hong, D.: PIPO: A lightweight block cipher with efficient higher-order masking software implementations. In: Hong, D. (ed.) Information Security and Cryptology - ICISC 2020 - 23rd International Conference, Seoul, South Korea, December 2-4, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12593, pp. 99–122. Springer (2020). https://doi.org/10.1007/978-3-030-68890-5_6, https://doi.org/10.1007/978-3-030-68890-5_6
3. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1666, pp. 388–397. Springer (1999). https://doi.org/10.1007/3-540-48405-1_25, https://doi.org/10.1007/3-540-48405-1_25
4. Priddy, K., Keller, P.: Artificial Neural Networks: An Introduction. SPIE tutorial texts, SPIE Press (2005), https://books.google.co.kr/books?id=BrnHR7esWmkC
5. Reed, R.D., Marks, R.J.: Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. MIT Press, Cambridge, MA, USA (1998)
6. Schaffer, C.: Selecting a classification method by cross-validation. Mach. Learn. **13**, 135–143 (1993). https://doi.org/10.1007/BF00993106, https://doi.org/10.1007/BF00993106
7. Timon, B.: Non-profiled deep learning-based side-channel attacks with sensitivity analysis. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**(2), 107–131 (2019). https://doi.org/10.13154/tches.v2019.i2.107-131, https://doi.org/10.13154/tches.v2019.i2.107-131
8. TrusThingz: PIPO data set. https://trusthingz.org/index.php/pipo-data-set
9. Zeman, V., Martinasek, Z.: Innovative method of the power analysis. Radioengineering **22**, 586–594 (2013)