IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 August 2023, accepted 24 August 2023, date of publication 28 August 2023, date of current version 7 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3309422

RESEARCH ARTICLE

Deep-Learning Based Nonprofiling Side-Channel Attack on Mask Leakage-Free Environments Using Broadcast Operation

SEONGHYUCK LIM[1], HYE-WON MUN[1], AND DONG-GUK HAN[1,2]
[1]Department of Financial Information Security, Kookmin University, Seoul 02707, Republic of Korea
[2]Department of Information Security, Cryptology, and Mathematics, Kookmin University, Seoul 02707, Republic of Korea

Corresponding author: Dong-Guk Han (christa@kookmin.ac.kr)

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) funded by the Korean Government (MSIT) through the Development of SCR-Friendly Symmetric Key Cryptosystem and Its Application Modes under Grant 2017-0-00520.

ABSTRACT With the recent development of artificial intelligence (AI), efforts to apply related technologies to various fields are rapidly increasing. In the field of cryptanalysis, research utilizing deep learning is continuously being published in order to keep up with this trend. Side-channel analysis is a a type of cryptanalysis that uses physical information and can be classified into profiling and nonprofiling analyses. Nonprofiling attacks using deep learning take advantage of the fact that training is performed relatively well when the right key is compared to the wrong key. Masking countermeasures are applied to design a secure cipher against side-channel analysis. The traditional second-order attack for analyzing masked ciphers is used by preprocessing the side channel information to remove the mask value. However, deep learning has the advantage of being able to omit this process. Related works proposed so far attempted to analyze the masked cipher, but focused only on 1-byte analysis using the masking information itself. In reality, grasping the time-points, in which only the masking information is revealed, is difficult and far from the secret key analysis area. In this study, we attempt to analyze the case of combining masked 2-byte information, not only using the masking information. We also propose a neural network design scheme to perform more effective attacks. The proposed method highlights the relative difference between the right and wrong keys. Previous research on analysis evaluation criteria has been lacking. Therefore, we propose herein new evaluation metrics that can be easily used and demonstrate their validity using simulation and actually collected data. As a result of the experiment, the proposed methods based on the loss metric improved by approximately 228.59% in the simulation dataset and 739.46% in the real dataset compared to the binary labeling. And it reduced the minimum number of analytical traces by approximately 78.95% and 72.5%, respectively.

INDEX TERMS Side-channel analysis, deep-learning, nonprofiling attack, masked block cipher, second-order analysis.

I. INTRODUCTION

Side-channel analysis (SCA) is a technique that analyzes secret values using additional physical information such as power consumption and electromagnetic emissions generated when algorithms operate in cryptographic devices [1]. The SCA is classified into profiling (P-SCA) and nonprofiling analyses (NP-SCA) according to the analysis environment and the attacker's assumption. The P-SCA is a method of generating a profile based on the side-channel information obtained in advance from devices in the same environment as the target and analyzing the secret values by matching them with the side-channel information acquired from the target device. A representative P-SCA includes a template

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

attack (TA) [2]. Meanwhile, the NP-SCA is a method of repeatedly performing the encryption of random plaintexts on a target device that operates as a fixed key and analyzing the secret values through the statistical relation between the side-channel information and the intermediate values. The representative NP-SCA includes differential power analysis (DPA) [3] and correlation power analysis (CPA) [4].

Deep learning (DL) is a kind of machine learning that is an important field in the Fourth Industrial Revolution. It is used in a wide range of fields, including image and voice recognition and natural language processing. Studies on the application of DL are being continued in the cryptanalysis field (e.g., SCA). The DL-based SCA is classified into profiling and nonprofiling, with active research being conducted on this topic. The DL-based P-SCA is a method that allows an artificial neural network (ANN) to predict the secret key related to the target traces after generating DL-based profiles by learning the side-channel information according to the secret keys [5], [6], [7], [8]. Meanwhile, the DL-based NP-SCA is a method that takes advantage of the characteristic of the correct key having a relatively better learning performance (accuracy, loss, etc.) when learning associations with all key candidates for the target traces [9].

Hiding and masking techniques are studied as countermeasures to prevent the SCA [10], [11], [12]. The hiding technique increases the SCA complexity by randomizing the operation time through dummy operations and shuffling techniques, among others. The masking technique randomizes the intermediate value to eliminate the relationship between the side-channel information and the intermediate value, making it statistically independent. Masked ciphers cannot be key-guessed with traditional statistical analysis.

Second-order SCA (SOSCA) is a technique that attacks masking countermeasures by generating statistical relationships with intermediate values through the preprocessing of the side-channel information of two-time points using the same mask value [13], [14]. This means that the attacker must explore and preprocess sensitive points in the trace. The DL is one of the important technologies that can alleviate these challenges. The SOSCA is divided into two approaches. It is classified according to whether the side-channel information for the mask value is leaked. The mask leakage-free environments have a wide key search area, making analysis difficult.

Unlike the traditional SOSCA, the DL has the advantage of not requiring preprocessing because it can learn the side-channel information related to the combination of 2-byte of secret keys. The DL-based SCA on cryptographic algorithm-applied countermeasures is steadily being studied, but research on the NP-SCA compared to the P-SCA is lacking. In the case of the previously studied DL-based NP-SCA, research mainly focused on guessing 1-byte by targeting the mask value and the masked intermediate value [9], [15], [16], [17], [18]. This is mainly because the ASCAD dataset [19] is used for verification. From a realistic point of view, however, the timing of the mask value generation is often far from the actual sensitive side-channel leakage timing, and the collected data often do not include it. We define the above environment as a mask leakage-free environment. In this environment, learning about a combination of two bytes is required, which is a different problem than previous studies. Therefore, we will conduct herein a DL-based second-order NP-SCA (SONP-SCA) study in realistic environments requiring a combination of 2-byte of secret keys. Unlike previous studies, there are no open datasets of mask leakage-free environments that can be compared. Therefore, we show the results on simulation traces and directly collected traces. In previous studies on DL-based NP-SCA, various methods for performance improvement have been proposed. However, there are difficulties in determining hyperparameters due to the diversity of neural networks and targets in this field. To mitigate these challenges, we explore methods to maximize the concept of DL-based NP-SCA to show meaningful performance while using the basic hyperparameters.

The primary contributions of this study are as follows:

- **Conducting DL-based SONP-SCA intensive study in a mask-free environment.**
  We investigated more realistic environments (e.g., analysis of two nearing bytes containing the same mask value), which was not addressed in the previous studies. Through this, we show the advantage of a DL-based analysis using the side-channel information as it is without preprocessing.
- **Proposal and validation of methods to maximize attack mechanisms of DL-based NP-SCA.**
  We proposed a broadcast output DNN utilizing broadcast operation and the usage of a stretch sigmoid as the last layer activation function. This showed the effect of maximizing the difference when observing the learning index of the right and wrong keys. We demonstrated that our proposed technique is effective by analyzing the simulation traces generated based on the HW of the intermediate value, and the power traces of the masked AES collected from the Chipwhisperer-Lite capture board [20].
- **Suggestion of new performance evaluation metrics for the DL-based NP-SCA.**
  Considering the lack of research on the analytical criteria for the NP-SCA in previous studies, we propose herein new metrics using outlier search techniques. We also prove the effectiveness of the proposed methods based on the proposed metrics.

Section II begins with descriptions of DL, SOSCA, and boxplot and introduces the concept of the DL-based NP-SCA. Section III describes the attacker's assumption that performs the DL-based NP-SCA and proposes an attack performance improvement method. Section IV proposes a new lightweight evaluation metric by applying a method to detect outliers in the quartiles. Section V-A1 demonstrates the validity of this work through experiments on simulation and actually collected traces. Section VI describes the limitations and

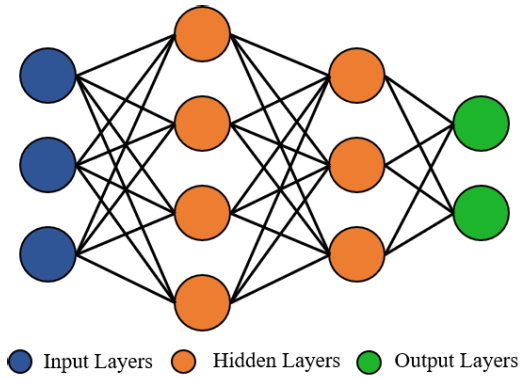


FIGURE 1. Example of the MLP.

scalability of this study. Finally, Section VII presents the conclusion and future works.

II. BACKGROUNDS

A. DEEP LEARNING

DL means performing ML using ANNs with multiple layers. It is also called deep structured and hierarchical learning. In traditional ML, researchers must analyze and determine which characteristics would be extracted from among the various target data characteristics. In DL, the difference is that machines automatically extract the features they want to learn. An ANN is a statistical learning algorithm that implements biological neural networks in computer science. It can be classified into multi-layer perceptron (MLP), convolutional neural network (CNN), and recursive neural network (RNN) according to the form. New mechanisms of neural network structures, such as attention, are steadily studied. DL can be divided into regression and classification problems depending on the type of value to be predicted. In regression, the result value has continuity by predicting through real number variables. Classification is a problem in which the target values are categorical and have discrete rather than continuous values.

The MLP used in this work is a feed-forward neural network, in which perceptrons are connected in multiple layers (Figure 1). It consists of an input layer, hidden layers, and an output layer, and is a fully-connected structure in which each perceptron of the current layer is connected to all perceptrons of the next layer. Supervised learning is performed while calculating the loss value through the real and predicted output, and updating the ANN weight (W). The layer output (Out) of the MLP is calculated through the inner product of the weight ($W = [w_1, \ldots, w_n]$) and input vectors ($X = [x_1, \ldots, x_n]$), addition with bias (b), and activation function (f) as follows:

$$Out = f(w_1x_1 + w_2x_2 + \ldots + w_nx_n + b) \tag{1}$$

Nonlinear functions like tanh, sigmoid, ReLU, and softmax are used as activation functions in DL construction.

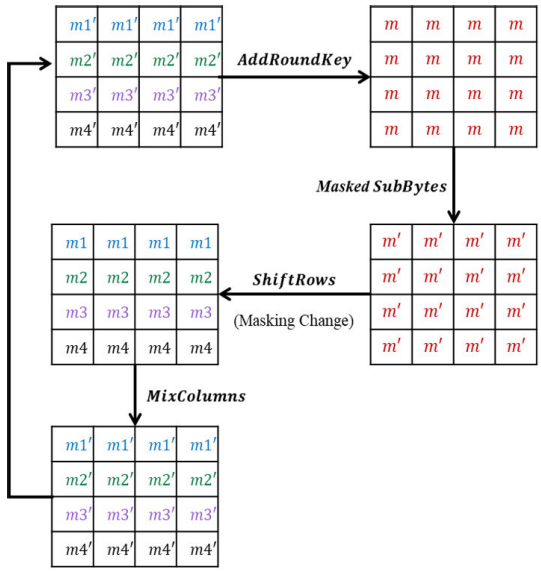


FIGURE 2. Masking values for each layer of the AES boolean masking design.

B. SECOND-ORDER SIDE-CHANNEL ANALYSIS

Masking was proposed to safely design encryption algorithms from SCAs (e.g., CPA). This technique removes the relationship with the side-channel information by hiding sensitive intermediate values as mask values [11]. Figure 2 depicts a typical AES boolean masking design and the mask values for each layer. The SOCPA is a method for analyzing masked ciphers. When performing it, multiple points of the side-channel information with the same mask value are preprocessed to generate information related to the intermediate value in which the mask value has been removed. The sensitive time-points can be determined through a simple power analysis (SPA). A common preprocess technique is the *absolute difference* (AD) combining function proposed in [13]. When two different point sets at which each length of the side-channel information is x, y are defined as (t_x, t_y), the AD for all (x, y) pairs is calculated as follows:

$$AD(t_x, t_y) = |t_x - t_y|, \forall(x, y) \tag{2}$$

Two main approaches can be used to perform the SOCPA. The first approach targets the mask value (M) and the masked S-Box ($S(p_i \oplus k_i) \oplus M$). Only 1-byte is targeted; hence, the candidate keys are treated in $\{0, 1\}^8$. However, the timing of loading or generating the mask value must be included in the target information, which is a problem. The second approach considers two distinct masked S-Boxes, namely, $S(p_i \oplus k_i) \oplus M$ and $S(p_j \oplus k_j) \oplus M$ with $i \neq j$. In this case, the candidate keys are handled within $\{0, 1\}^{16}$ as a guess for 2-byte, which takes a relatively long analysis time. This is a more realistic approach because the two point sets are close, making it easy to search for the sensitive area. In addition, the mask value can be utilized even when the mask generative area is excluded.

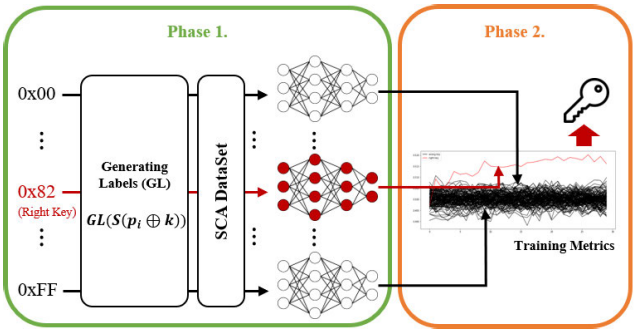


FIGURE 3. Overall process of the DL-based NP-SCA when analyzing a 1-byte key.

C. DIFFERENTIAL DEEP LEARNING ANALYSIS

The DL-based NP-SCA was proposed by Timon in 2019 under the name of differential deep learning analysis (DDLA) [9]. The attack process is explained in two phases as follows:

- **Phase 1: Training by candidate key**
  In this phase, the side-channel information is collected when encrypting random plaintexts on cryptographic devices that use fixed secret keys. Training according to each guess key is repeated for the input data after setting labels (e.g., LSB, MSB, HW, etc.) for the intermediate value of the target operation (e.g., S-Box output).
- **Phase 2: Results analysis**
  In this phase, the attacker compares the performance (e.g., accuracy, loss, etc.) for each training after the training of all guess keys is completed. The right key is relatively good at predicting labels compared to the wrong ones, resulting in high accuracy and low loss.

Figure 3 shows the DL-based NP-SCA process when analyzing a 1-byte key. When performing the NP-SCA, Timon stated that identity (ID) labeling is unsuitable because the learning levels of the guess keys become equivalent when using ID labels. He performed the attack by adopting the LSB, MSB, and HW models of the AES S-Box output as labels, consequently proving that binary labeling (i.e., LSB or MSB) has a better performance than HW labeling. Later studies also performed the DL-based NP-SCA using MSB or LSB labeling on the AES [15], [16], [17], [18]. All these previous studies investigated the second-order analysis, but aimed only at 1-byte analysis using the mask values. A study suggested using binary encoding (BE) labeling on bit-sliced block cipher [21]. This has the advantage of deriving generalization performance while utilizing the advantage of binary labeling even for commonly implemented ciphers. Table 1. compares previous studies described above. Most of them target the ASCAD dataset and analyze the environment described in the first approach. These also show that labeling, which was used mostly in the past, is being used. We emphasize that the proposed attack was conducted in a more realistic environment by applying the second approach that was not targeted in previous studies. Ref. [15] presented

TABLE 1. Comparison with previous studies.

Ref.	Target	Labeling	Key Range
[15]	ASCAD & CW Masked AES	LSB, HW	$\{0,1\}^8$
[16]	Simulated & CW & ASCAD Masked AES	MSB	$\{0,1\}^8$
[17]	ASCAD Masked AES	LSB, HW	$\{0,1\}^8$
[18]	ASCAD Masked AES (Known Mask)	LSB	$\{0,1\}^8$
[21]	Normal PIPO	BE	$\{0,1\}^8$
This paper	Simulated & CW Masked AES	BO-DNN	$\{0,1\}^{16}$

evaluation metrics using reliability; however, showing relativity is limited because the learning index for the wrong keys may also exceed the standard. Meanwhile, Ref. [17] proposed evaluation metrics using the normalized maximum margin (NMM). It requires a normalized margin calculation according to all key candidates and epochs. We believe that a study on lightweight evaluation metrics was needed because we performed a 2-byte estimation here. Therefore, we propose evaluation metrics that can accurately and easily determine the right key.

D. QUARTILE & OUTLIER

The boxplot is a data visualization method that can easily compare different data groups while simultaneously showing the data distribution and outliers. It does not use raw data as they are, but processes and visualizes them with a statistical concept called a five-number summary. During this time, the concept of quartile and outlier is used. As shown in Figure 4, the data were divided into five points: maximum (Max), upper quartile ($Q3$), median ($Q2$), Lower quartile ($Q1$), and minimum (Min). The $Q1$, $Q2$, and $Q3$ meant the data at 25%, 50%, and 75% locations when the entire data were sorted in ascending order. The quartile is widely used as a method of searching for outliers for univariate data. The difference between $Q3$ and $Q1$ is defined as the IQR, and the outliers are defined as follows:

$$(x < Q1 - 1.5 \cdot IQR) \vee (x > Q3 + 1.5 \cdot IQR) \longleftrightarrow x =^{def} Outlier \quad (3)$$

As an evaluation of the DL-based NP-SCA, a method of searching for outliers using quartiles was proposed here. The evaluation method will be described in detail in Section III-B.

III. PROPOSED DNN STRUCTURES IN SONP-SCA

This section describes the attacker's assumptions and attack targets prior to performing the DL-based SONP-SCA. We also propose performance improvement methods, DNN structure, and new criteria for the performance evaluation.

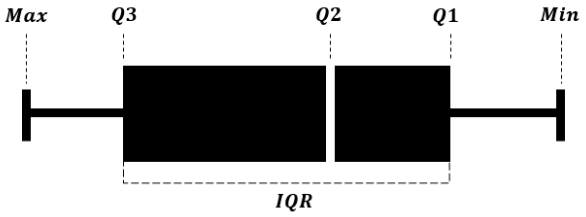


FIGURE 4. Five-number summary in the boxplot.

A. ATTACKER's ASSUMPTION

The attacker of the nonprofiling attack can collect the side-channel information emitted when a cryptographic device that uses a fixed secret key operates. During this time, the attacker can input random plaintexts into the device. In the traditional SOCPA, two sensitive time-points are identified by first performing the SPA, and then preprocessing the traces. Considering the advantages of DL-based attacks, we assumed an environment where it is difficult for the attacker to apply the SPA. That is, when performing an attack, part of the collected traces is used without preprocessing. We considered a more realistic target that does not include the times for generating and using the masking information or is far from the analysis area. Therefore, an attacker must search for a key candidate of $\{0, 1\}^{16}$ with $S(p_i \oplus k_i) \oplus S(p_j \oplus k_j)$, $i \neq j$ as the label to succeed in the attack.

B. PROPOSED METHODS

Our proposed method focused on maximizing the relative difference in learning metrics when learning with the right and wrong keys.

1) BROADCAST OUTPUT DNN

When performing operations between a multi-dimensional vector ($v = (v_0, \ldots, v_k)$) and a uni-dimensional vector (w), broadcast is defined as operations between scalar w and each element of v as follows:

$$(w) \odot (v_0, v_1, \ldots, v_k) = (w \odot v_0, w \odot v_1, \ldots, w \odot v_k) \tag{4}$$

We define the broadcast output DNN (BO-DNN) herein as a mechanism for predicting the DNN output to a single value when training multi-dimensional labels. Previous studies proved that using binary labels is more effective than using HW labels when performing the DL-based NP-SCA. We used the BE for the HW as a label to integrally take advantage of the information of each bit because the training performance was different for each bit. BE is an encoding method in which a binary representation of the corresponding value is constructed as a vector as follows:

$$Y = (y_0 y_1 \ldots y_7)_2 \longrightarrow Y_{label} = (y_0, y_1, \ldots, y_7) \tag{5}$$

We also considered the NP-SCA as a regression problem and adopted sigmoid as the activation function of the output node.

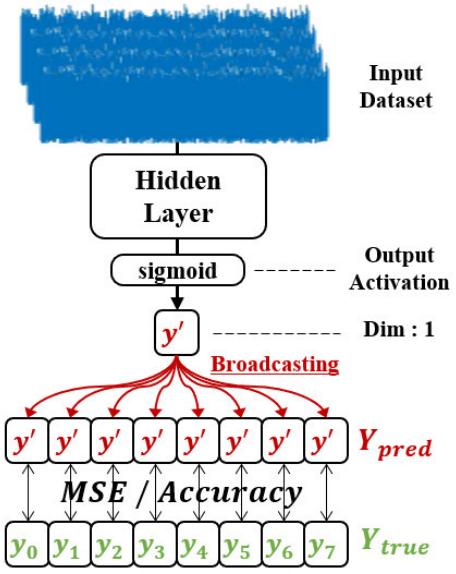


FIGURE 5. Overall structure of the BO-DNN.

We used the mean squared error (MSE) as the loss function. Figure 5 depicts the final constructed DNN.

In the proposed BO-DNN, eight-dimensional labels are learned as one average value. The average of each calculation result between each element of the eight-dimensional label and the uni-dimensional output of the DNN is returned when computing accuracy and loss. Binary accuracy is particularly used for accuracy. 0 or 1 is judged according to rounding. The binary accuracy is determined by $0.125 \times (\#\ of\ correct\ bits)$ for one trace. The average for the total number of traces is then calculated. The BO-DNN learns the average value of the labels. Therefore, if the HW of the intermediate values is greater than 4, it is more likely to return 0.5 or higher. Conversely, less than 4 is more likely to return an output of less than 0.5. 4 0.5. If the HW is large, 1 is judged as the correct bit. By contrast, if the HW is small, 0 is determined as the correct bit. That is, when there are many specific bit values, the corresponding bits are all determined as the correct label. Hence, a high weight is given to the accuracy if it is a definite key. The previous explanation is expressed as Equation 6 where $n(= a + b + \ldots + i)$ is the total number of training traces, and $\{a, b, \ldots, i\}$ denotes the number of cases with the HW of 0, 1, …, and 8, respectively. When $m \in \{a, \ldots, i\}$, m' is defined as the number of correct predictions among the results of learning m times.

$$\begin{aligned} Binary\_Accuracy &= \frac{S}{n}, \\ S &= a' + (b'(\frac{7}{8}) + (b - b')(\frac{1}{8})) + (c'(\frac{6}{8}) + (c - c')(\frac{2}{8})) + \\ &\quad (d'(\frac{5}{8}) + (d - d')(\frac{3}{8})) + (e'(\frac{4}{8}) + (e - e')(\frac{4}{8})) + \\ &\quad (f'(\frac{5}{8}) + (f - f')(\frac{3}{8})) + (g'(\frac{6}{8}) + (g - g')(\frac{2}{8})) + \\ &\quad (h'(\frac{7}{8}) + (h - h')(\frac{1}{8})) + i' \end{aligned} \tag{6}$$

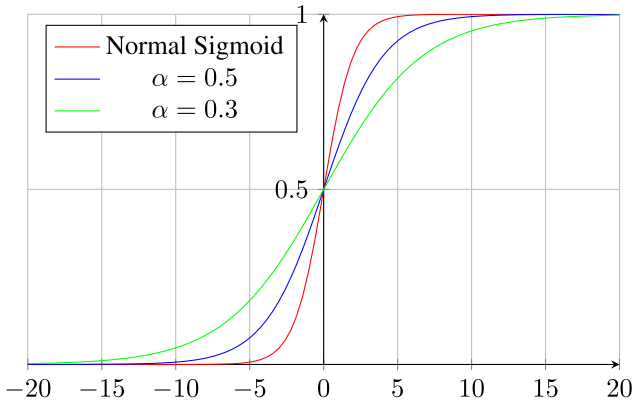


FIGURE 6. Sigmoid graphs according to the stretch coefficient.

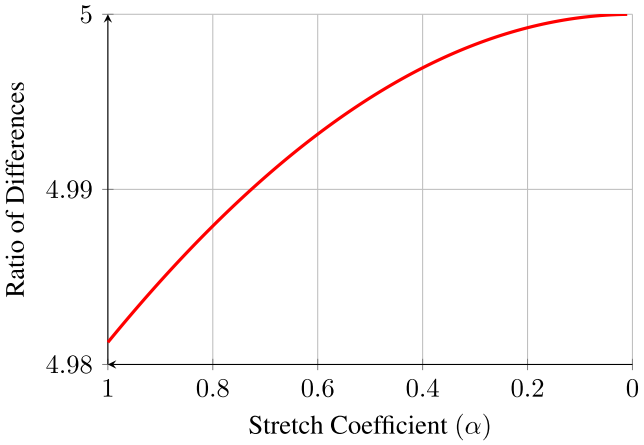


FIGURE 7. Relative difference between the right and wrong keys according to the stretch coefficient of the sigmoid.

Compared to that when learning with the wrong key, m' increases and $(m - m')$ decreases when training with the right key. In this case, the constants associated with m' are a value greater than $\frac{4}{8}$, and the average accuracy increases. Conversely, when learning with the wrong key, $(m - m')$ increases, such that constants less than $\frac{4}{8}$ are multiplied, which has a relatively low average accuracy. The BO-DNN can maximize the relative difference between the right and wrong keys by increasing the training metric values of the right key and decreasing those of the wrong keys. Dimension of the output layer of the BO-DNN is 1, such as binary and HW labeling. The overhead generated in DNN learning occurs only in loss and accuracy calculation between Y_{pred} and Y_{true}. This has a relatively small time complexity compared to the feed-forward operation of the DNN, so it does not significantly affect the analysis time.

2) STRETCH SIGMOID

BE for the intermediate values was employed as the labels. Therefore, sigmoid was used as an active function of the output layer as a regression problem. We propose the stretch sigmoid as a method of maximizing the difference between the output values for the right and wrong keys. The stretch sigmoid was multiplied by the sigmoid input and the stretch coefficient (α) as follows:

$$sigmoid_{stretch}(x, \alpha) = \frac{1}{1 + e^{-\alpha x}} \tag{7}$$

Depending on the α, the graph changes to a stretched form as shown in Figure 6. As α decreases, the ratio between the cases, where the differences in training results are large and small, increases. When O_{key} is the output of the stretch sigmoid as a result of training with the candidate keys, the ratio of difference is defined as follows:

$$Ratio_{difference} = \frac{O_{right} - O_{wrong_1}}{O_{wrong_1} - O_{wrong_2}} \tag{8}$$

Figure 7 shows the change in the ratio of difference according to the decrease in α when $O_{right} = 0.53, O_{wrong_1} = 0.505$, and $O_{wrong_2} = 0.5$. Regardless of the size of O_{key}, if the right key deviates from the wrong key distribution, it will have an upward-sloping shape as shown in Figure 7. Thus, through the change in α, we can maximize the relative difference between the right and wrong keys in the NP-SCA.

IV. PROPOSED EVALUATION METRICS

We proposed the NP-SCA evaluation metrics using an outlier search method using quartiles. As mentioned in Section II-D, x satisfying Equation 3 were judged as outliers. The outlier coefficient is defined as 1.5, and if it is greater than 3, it is defined as a strong outlier. When analyzing the training metrics, the right key in the DL-based NP-SCA showed relatively good performance, whereas the wrong keys formed a dense distribution. Accordingly, the attack performance was evaluated by considering the right key metrics as an outlier. Through the experiments, we confirmed that even the basic coefficient of 1.5 can sufficiently filter the correct key candidates. Equation 9 represents the criterion for the right key determination. The upper-limit outlier is defined herein as the criterion for determining the accuracy, and the lower-limit outlier is defined as the criterion for determining the loss. If multiple values are considered outliers, all of them are considered key candidates.

$$\begin{aligned} Criteria_{acc} &= Q3 + 1.5 \cdot IQR, \\ Criteria_{loss} &= Q1 - 1.5 \cdot IQR \end{aligned} \tag{9}$$

If the analysis is successful, the right key will be equal to *Max* of accuracy and *Min* of loss. *Max* and *Min* can be expressed as quartile points and *IQR* as follows:

$$\begin{aligned} Max &= Q3 + k \cdot IQR, \\ Min &= Q1 - k \cdot IQR, \quad k \in R \end{aligned} \tag{10}$$

Based on the above equation, we proposed $Ratio_{acc}$ and $Ratio_{loss}$ as the metrics of the DL-based NP-SCA evaluation as follows:

$$Ratio_{acc} = \frac{Max - Q3}{1.5 \cdot IQR},$$

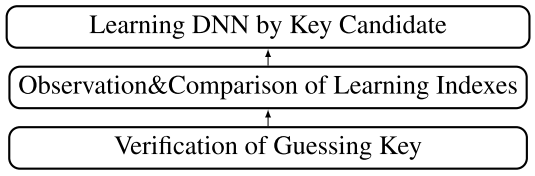


FIGURE 8. Basic process of the experiments.

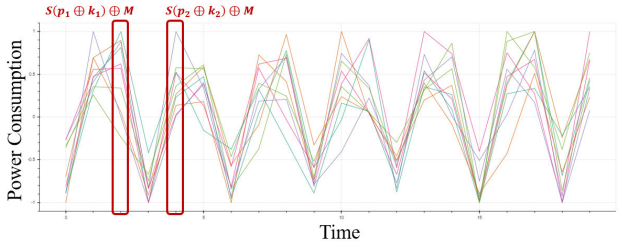


FIGURE 9. Simulation power traces based on HW model.

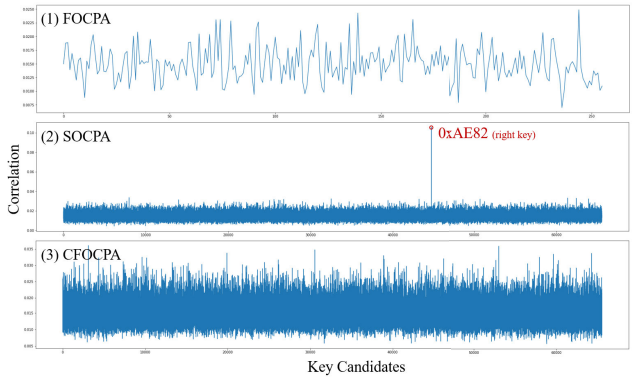


FIGURE 10. Statistical SCA results for the simulation dataset.

$$Ratio_{loss} = \frac{Q1 - Min}{1.5 \cdot IQR} \tag{11}$$

They mean how large relative to the outlier criterion (1.5) for the outlier coefficient k of the *Max* or *Min* value determined by the correct guess key. Each is an evaluation metric for accuracy and loss. The higher the two values, the clearer the distribution of the right and wrong keys, so it can be judged as the more efficient attack. We define the value corresponding to the outlier coefficient 1.5 as the threshold and an example of this can be found in Figure 13 shown in Section V-B Experiments.

V. EXPERIMENT

This section verifies the effectiveness of the proposed methods on two datasets. Compared to the BE and LSB labeling, the proposed method BO-DNN exhibits performance improvement. The LSB is the most commonly used labeling, and the BE has the advantage of binary labeling and shows generalization performance, so they were selected as a comparison target. In addition, the performance improvement when using the stretch sigmoid according to the stretch

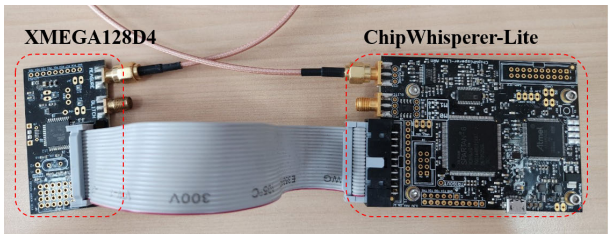


FIGURE 11. Chipwhisperer-Lite and AVR XMEGA128D4.

coefficient is shown here. Performing training on a certain number of traces in an experiment means that all corresponding traces are used for training and the verification set is not configured separately. This is because our experiment is the mechanism for observing relative differences in training. The experimental process is as follows. First, an intermediate value according to all key candidates is calculated, and deep learning is performed by using these values as labels of the traces. The learning indexes are compared when all key candidates are completed. At this time, the key candidate with the best performance is judged as the right key. Finally, the success of the attack is confirmed by determining whether the guessed key matches the actual key. A summary of the experimental process is shown in Figure 8.

A. ENVIRONMENT

1) SIMULATION DATASET

First, simulation traces were generated according to the HW model for the intermediate value. They contained the HW information of two masked AES S-Box output bytes $(S(p_1 \oplus k_1) \oplus M, S(p_2 \oplus k_2) \oplus M)$. The remaining information comprised meaningless information. Subsequently, noise insertion and normalization were performed. The generated simulation trace had a length of 20 (Figure 9). The dataset consisted of 20,000 traces. The red highlight in the figure depicts the masked sensitive point. Figure 10 shows the results of applying the traditional CPA to the generated simulation dataset. The AD for sections 0 to 3 and 4 to 7 point of traces was applied when the SOCPA was performed. We confirmed that the simulation waveform was secure for the first-order CPA (FOCPA), and that the SOCPA must be performed for the analysis. We also confirmed that it was secure for the combined FOCPA (CFOCPA). The CFOCPA is an attack using vulnerability, where 2-byte combined information exists in the raw-data [22]. By analyzing waveforms without this vulnerability, we show herein that the DNN generates combined information well.

2) CHIPWHISPERER DATASET

The second target is the Chipwhisperer dataset collected through Chipwhisperer-Lite, a capture board with a sampling rate of 29.538 MS/s. The target board was an 8-bit MCU-based AVR XMEGA128D4. Figure 11 illustrates the overall configuration. The power traces were collected by operating the masked AES according to Figure 2. The analysis

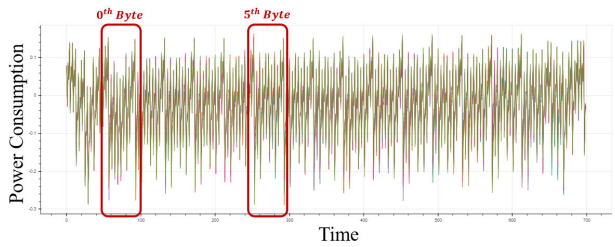


FIGURE 12. Collected Chipwhisperer dataset.

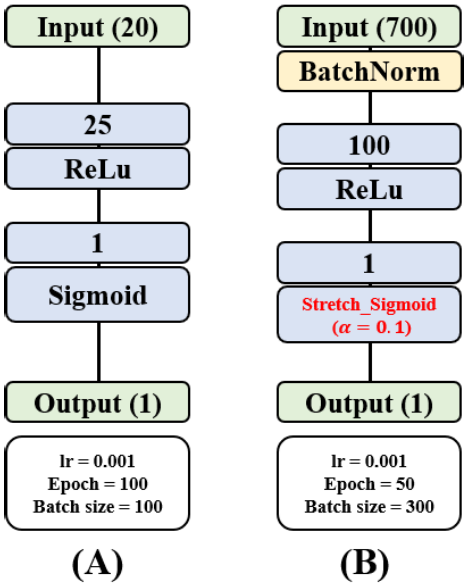


FIGURE 13. DNN structures used in the experiment. (A) Simulation dataset, (B) Chipswhisperer dataset.

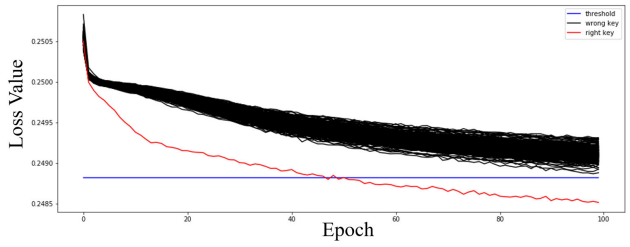


FIGURE 14. Pproposed DL-based SONP-SCA results on the simulation dataset. (loss metrics).

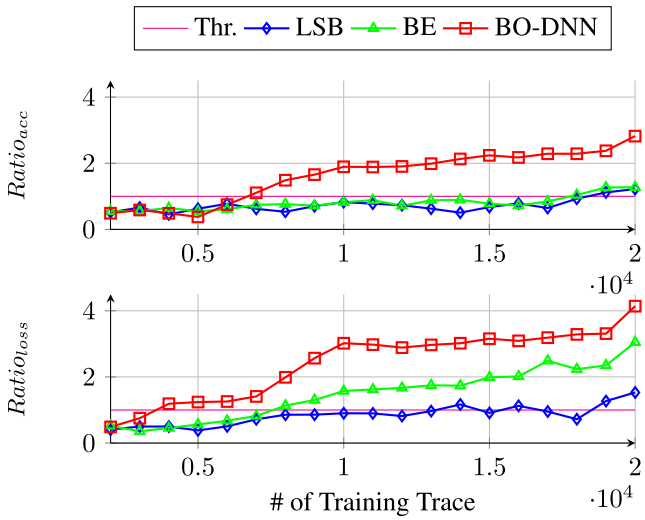


FIGURE 15. Comparison of the experimental results according to labeling in the simulation dataset.

was conducted on the entire area of the SubBytes function. Figure 12 shows the collected dataset, which like the simulation dataset, requires a 2-byte analysis. We selected the 0^{th} and 5^{th}-byte combinations as the targets to perform attacks against combinations with high robustness to vulnerability [22]. We confirmed that the target combination can be analyzed only through the SOCPA and was secure for the FOCPA and the CFOCPA.

Finally, we performed a DL analysis using the Keras platform [23]. Elements other than the hyperparameters shown in Figure 13 are in accordance with the Keras default setup, and all experiments used Xavier initialization. The proposed BO-DNN was constructed by utilizing the operation broadcasting function of the tensor. We provide experimental code examples for CW [24].

B. EXPERIMENTAL RESULTS

1) SIMULATION DATASET

We will first show here the performance improvements according to the proposed methods for targeting the simulation dataset. This experiment was conducted with the MLP consisting of one hidden layer (Figure 13-A). For the BE labeling, eight nodes were used in the output layer. Figure 14 presents the loss metrics of the BO-DNN learning results. Only the correct key exceeded the threshold. Figure 15 displays the minimum number of analysis traces according to each labeling.

When the proposed BO-DNN was used, $Ratio_{metric}$ increased compared to BE and LSB labeling. In particular, the analysis performance was better when the loss metric was used. The analysis was successful with approximately 5,000 traces. The BO-DNN also showed a positive effect on the accuracy metric performance compared to other labeling approaches. In other words, the DNN effectively maximized the relative difference between the right and wrong keys. Therefore, the proposed technique is expected to maximize the analysis effect and fully use the accuracy metrics.

We confirmed that the BO-DNN provides good performance at small epochs. Figure 16 compares the labeling-specific performances according to the epoch. In the mask leakage-free environment where two bytes need to be analyzed at once, it takes 256x more learning time. However, in the above experiments, the BO-DNN showed overwhelming performance on 10 epochs and had approximately 3x and

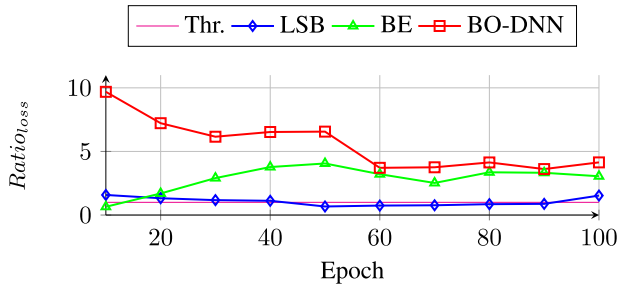


FIGURE 16. Comparison of the experimental results according to epoch in the simulation dataset (BO-DNN).

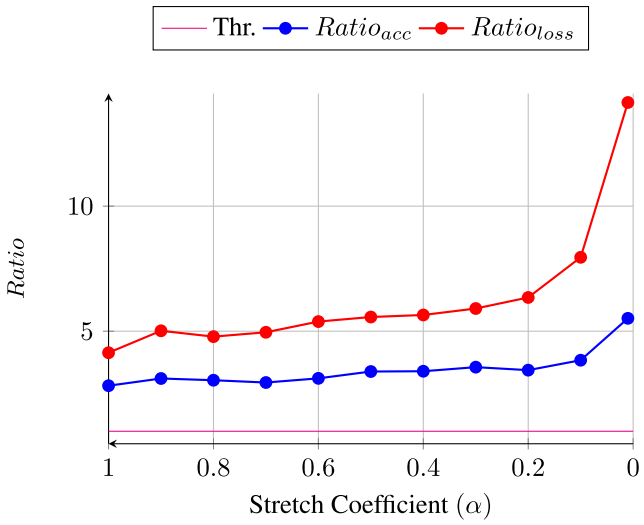


FIGURE 17. Comparison of the experimental results according to the stretch coefficient in the simulation dataset.

approximately 10x analysis time mitigation effects compared to LSB and BE, respectively. Through this, we show that the BO-DNN can be a merit in the DL-based SONP-SCA that requires a long analytical time.

Figure 17 shows the performance improvement according to the stretch coefficient size. This was the performance result using 20,000 traces for the BO-DNN used in the previous experiment. The analysis performance improved as the stretch coefficient decreased. The effect was highlighted in the loss metrics. Therefore, we demonstrated that the use of the stretch sigmoid was effective in maximizing the learning difference between the wrong and right keys.

2) CHIPWHISPERER DATASET

The experimental results on the Chipwhisperer dataset will be elaborated here. Using batch normalization was effective when analyzing the real-world datasets. The number of hidden layer nodes was increased according to the increased trace length. Figure 13-B exhibits the DNN used in the experiment. We set the parameters based on the experimental results on the simulation dataset. Stretch sigmoids with a 0.1 coefficient were used. Metrics of 50 epoch learning results employed. Figure 18 shows the experimental results according to labeling. Similar to the previous experimental results, the BO-DNN yielded the best performance. Analysis was possible with approximately 10,000 - 15,000 traces. Using of accuracy metrics was also possible.

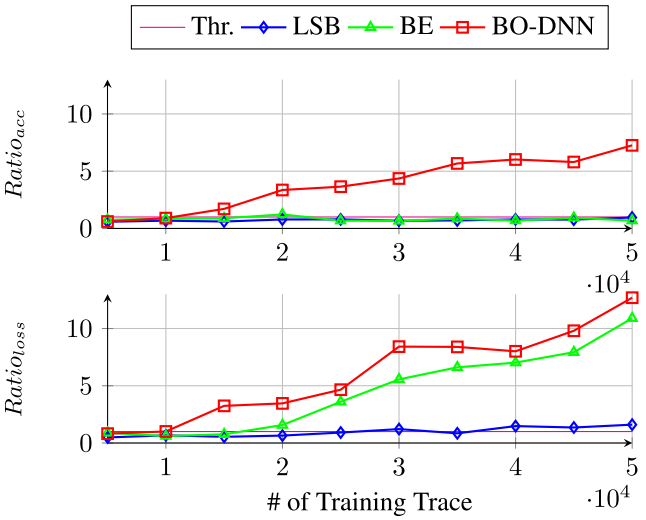


FIGURE 18. Comparison of the experimental results according to labeling in the Chipwhisperer dataset.

TABLE 2. Analytical efficiency of the BO-DNN compared to the LSB labeling.

(A) Simulation dataset

Metrics	Maximum *Ratio* Increase Rate	Minimum # of Analysis Traces Decrease Rate
Accuracy	$\approx 131.01\%$	$\approx 63.16\%$
Loss	$\approx 228.59\%$	$\approx 78.95\%$

(B) Chipwhisperer dataset

Metrics	Maximum *Ratio* Increase Rate	Minimum # of Analysis Traces Decrease Rate
Accuracy	$\approx 625.79\%$	$\approx 72.73\%$
Loss	$\approx 739.46\%$	$\approx 72.50\%$

3) SUMMARY OF THE EXPERIMENT RESULTS

Table 2 shows the degree of performance improvement according to the experiment results. Compared to the LSB labeling, which was studied a lot before, the increase rates of $Ratio_{loss}$ and $Ratio_{acc}$ of the proposed BO-DNN were calculated. When the *Ratio* does not exceed threshold, the increase rate is calculated based on the threshold. As a result of the experiment, the accuracy improved up to 131.01% and the loss up to 228.59% for the simulation dataset. In case of Chipwhisperer dataset, the accuracy improved up to 625.79% and the loss up to 739.46%. The BO-DNN contributed to reducing the number of traces required for analysis. For all datasets, both accuracy and loss showed a decrease rate of more than 60%, proving that efficient attacks were possible.

VI. DISSCUSION

Another important aspect of scientific research is the ability to repeat experiments or studies in different environments and reuse or apply results. First, this study verified the simulation trace of the HW model, which is a power consumption model mainly used in SCA. We were able to find the difference in learning degrees between the right and wrong keys about the simulation waveform. In addition, it was verified whether the results of the study could be identified equally for the actual environment. In other words, this study verified not only the theoretical environment but also the actual environment. Experiments show that although there may be differences depending on the environment, our method can increase efficiency with a simple DNN structure. Second, this study targeted the most realistic attack environment. The reason why we considered $\{0, 1\}^8$ even though it is valid in the previously studied $\{0, 1\}^{16}$ environment is that most of the cases do not know the location of the mask value in the actual attack environment. The problematic part of the $\{0, 1\}^{16}$ environment is the DNN learning time. Although some contributions have been made to reduce the analysis time, learning 65,536 times is still very overhead. However, in two-byte analysis, this is considered an inevitable problem. We expect that the parallelization technology of deep learning will solve this problem.

VII. CONCLUSION

This study dealt with the DL-based SONP-SCA is dealt with. In particular. Different from previous works, we targeted a situation in which a 2-byte estimation using the same mask value was required. We targeted a more realistic environment, in which the mask value is often far from the actual analysis timing, and the timing of the mask value generation is often excluded from the collected dataset. We proposed the usage of the BO-DNN and the stretch sigmoid as the DNN structure and parameter tuning methods that can maximize the difference in the learning performance between the wrong and right keys in the DL-based NP-SCA. We also proposed lightweight evaluation metrics for situations where more key candidates must be predicted. In addition, we utilized the outlier search technique using quartiles using the characteristic of the right key being separated from the learning distribution of the relatively wrong key. The experiment was evaluated through the proposed metrics. We validated not only the simulation dataset but also the power traces that occurred when operating on a real device. The experiment results showed that the proposed methods improved the performance compared to other labeling approaches (e.g., BE and LSB). In particular, the loss metrics showed a better effect. The loss metric improved by approximately 228.59% in the simulation dataset and 739.46% in the real dataset compared to the binary labeling. And it reduced the minimum number of analytical traces by approximately 78.95% and 72.5%, respectively. We also found its effect of producing a good performance in a small epoch. One of the disadvantages of the NP-SCA using DL is its long attack time. Effective attacks that can be performed with only a small epoch are expected to contribute to compensating for this shortcoming. Although we focused on SONP-SCA, the proposed technique is valuable because it can be sufficiently applied to the general NP-SCA. Our future work will include a study on the logic that can develop the proposed methods. When combined with methods from previous studies, we will observe how much the proposed methods can improve performance. Additionally, we will conduct a systematic study on hyperparameter selection in DL-based NP-SCA. We will apply this to higher-order analysis to increase the candidate key range and determine if it can have positive effects on more difficult problems. Finally, Studies on parallelization will be conducted to compensate for the shortcoming of the analysis time, which is more prominent when guessing 2-byte.

REFERENCES

[1] P. Kocher, "Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems," in *Proc. IACR CRYPTO* (Lecture Notes in Computer Science), vol. 1109. Santa Barbara, California, USA: Springer, Aug. 1996, pp. 104–113, doi: 10.1007/3-540-68697-5_9.

[2] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems—CHES*, vol. 2523. Redwood Shores, CA, USA: Springer, 2002, pp. 13–28, doi: 10.1007/3-540-36400-5_3.

[3] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology*, vol. 1666. Santa Barbara, CA, USA: Springer-Verlag, 1999, pp. 388–397, doi: 10.1007/3-540-48405-1_25.

[4] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems—CHES*, vol. 3156. Cambridge, MA, USA: Springer, 2004, pp. 16–29, doi: 10.1007/978-3-540-28632-5_2.

[5] G. Hospodar, B. Gierlichs, E. De Mulder, I. Verbauwhede, and J. Vandewalle, "Machine learning in side-channel analysis: A first study," *J. Cryptograph. Eng.*, vol. 1, no. 4, pp. 293–302, Dec. 2011, doi: 10.1007/s13389-011-0023-x.

[6] L. Lerman, G. Bontempi, and O. Markowitch, "A machine learning approach against a masked AES," *J. Cryptograph. Eng.*, vol. 5, no. 2, pp. 123–139, Jun. 2015.

[7] 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*, vol. 11348. Kanpur, India: Springer, 2018, pp. 157–176, doi: 10.1007/978-3-030-05072-6_10.

[8] S. Ghandali, S. Ghandali, and S. Tehranipoor, "Profiled power-analysis attacks by an efficient architectural extension of a CNN implementation," in *Proc. 22nd Int. Symp. Quality Electron. Design (ISQED)*, Santa Clara, CA, USA, Apr. 2021, pp. 395–400, doi: 10.1109/ISQED51717.2021.9424361.

[9] B. Timon, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2019, no. 2, pp. 107–131, 2019, doi: 10.13154/tches.v2019.i2.107-131.

[10] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound approaches to counteract power-analysis attacks," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1666. Santa Barbara, CA, USA: Springer, 2000, pp. 398–412, doi: 10.1007/3-540-48405-1_26.

[11] K. Schramm and C. Paar, "Higher order masking of the AES," in *Topics in Cryptology* (Lecture Notes in Computer Science), vol. 3860. San Jose, CA, USA: Springer, Feb. 2006, pp. 208–225, doi: 10.1007/11605805_14.

[12] M. Rivain, E. Prouff, and J. Doget, "Higher-order masking and shuffling for software implementations of block ciphers," in *Cryptographic Hardware and Embedded Systems—CHES*, vol. 5747. Lausanne, Switzerland: Springer, 2009, pp. 171–188, doi: 10.1007/978-3-642-04138-9_13.

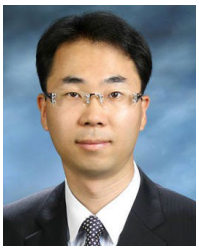
[13] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Berlin, Germany: Springer, 2007.

[14] E. Prouff, M. Rivain, and R. Bevan, "Statistical analysis of second order differential power analysis," *IACR Cryptol. ePrint Arch.*, 2010. [Online]. Available: http://eprint.iacr.org/2010/646

[15] Y.-S. Won, D.-G. Han, D. Jap, S. Bhasin, and J.-Y. Park, "Non-profiled side-channel attack based on deep learning using picture trace," *IEEE Access*, vol. 9, pp. 22480–22492, 2021, doi: 10.1109/ACCESS.2021.3055833.
[16] X. Lu, C. Zhang, and D. Gu, "Attention-based non-profiled side-channel attack," in *Proc. Asian Hardw. Oriented Secur. Trust Symp. (AsianHOST)*, Shanghai, China, Dec. 2021, pp. 1–6, doi: 10.1109/AsianHOST53231.2021.9699481.
[17] D. Bae and J. Ha, "Performance metric for differential deep learning analysis," *J. Internet Serv. Inf. Secur.*, vol. 11, no. 2, pp. 22–33, 2021, doi: 10.22667/JISIS.2021.05.31.022.
[18] K. Kuroda, Y. Fukuda, K. Yoshida, and T. Fujino, "Practical aspects on non-profiled deep-learning side-channel attacks against AES software implementation with two types of masking countermeasures including RSM," in *Proc. 5th Workshop Attacks Solutions Hardw. Secur.*, Nov. 2021, pp. 29–40, doi: 10.1145/3474376.3487285.
[19] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, and C. Dumas, "Deep learning for side-channel analysis and introduction to ASCAD database," *J. Cryptograph. Eng.*, vol. 10, no. 2, pp. 163–188, Jun. 2020, doi: 10.1007/s13389-019-00220-8.
[20] C. O'Flynn and Z. D. Chen, "Chipwhisperer: An open-source platform for hardware embedded security research," in *Constructive Side-Channel Analysis and Secure Design*, vol. 8622. Paris, France: Springer, 2014, pp. 243–260, doi: 10.1007/978-3-319-10175-0_17.
[21] J.-E. Woo, J. Han, and D.-G. Han, "Deep-Learning-Based side-channel analysis of block cipher PIPO with bitslice implementation," *IEEE Access*, vol. 10, pp. 69303–69311, 2022, doi: 10.1109/ACCESS.2022.3187201.
[22] J. Coron, C. Giraud, E. Prouff, S. Renner, M. Rivain, and P. K. Vadnala, "Conversion of security proofs from one leakage model to another: A new issue," in *Constructive Side-Channel Analysis and Secure Design*, vol. 7275. Darmstadt, Germany: Springer, 2012, pp. 69–81, doi: 10.1007/978-3-642-29912-4_6.
[23] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io
[24] S. Lim. (2023). *Github*. [Online]. Available: https://github.com/SeongHyuckLim/DLNPSOCPA.git

SEONGHYUCK LIM received the M.S. degree in financial information security from Kookmin University, Seoul, Republic of Korea, in 2022, where he is currently pursuing the Ph.D. degree in financial information security. His research interests include symmetric key cryptography, AI-based side-channel analysis, fault injection attacks, and security of financial IC cards.

HYE-WON MUN received the M.S. degree in financial information security from Kookmin University, Seoul, Republic of Korea, in 2023. She is currently with COONTEC, South Korea. Her research interests include symmetric key cryptography, AI-based side-channel analysis, fault injection attacks, and security of financial IC cards.



DONG-GUK HAN received the B.S. and M.S. degrees in mathematics and the Ph.D. degree in engineering and in information security from Korea University, Seoul, Republic of Korea, in 1999, 2002, and 2005, respectively. He was a Postdoctoral Researcher with Future University Hakodate, Hokkaido, Japan. After finishing the Ph.D. course, he was an Exchange Student with the Department of Computer Science and Communication Engineering, Kyushu University, Japan, from April 2004 to March 2005. From 2006 to 2009, he was a Senior Researcher with the Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea. He is currently a Professor with the Department of Information Security, Cryptology, Mathematics, Kookmin University, Seoul. He is a member of KIISC, IEEK, and IACR.