IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 June 2022, accepted 21 June 2022, date of publication 29 June 2022, date of current version 5 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3187201

RESEARCH ARTICLE

Deep-Learning-Based Side-Channel Analysis of Block Cipher PIPO With Bitslice Implementation

JI-EUN WOO[1], JAESEUNG HAN[1], AND DONG-GUK HAN[1,2]
[1]Department of Financial Information Security, Kookmin University, Seoul 02727, 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 International Collaborative Research and Development Programme through the Ministry of Trade, Industry and Energy (MOTIE), South Korea.

ABSTRACT Recently, as deep learning has been applied to various fields, deep-learning-based side-channel analysis (SCA) has been widely investigated. Unlike traditional SCA, it can perform well independently of the attacker's ability. In this paper, we propose deep-learning-based profiled and non-profiled SCA of PIPO, (Plug-In Plug-Out), which is a bitslice block cipher that can effectively apply a countermeasure for SCA. Our datasets were captured from three different boards (XMEGA128D4, MSP430F2618, STM32F303) running PIPO-64/128. For profiled SCA, the identity (ID) labeling method exhibited better performance than the most significant bit (MSB) and hamming weight (HW) labeling methods. That is, even if each bit of the S-Box output was distributed in the power traces by the bitslice implementation, the neural network trained well each bit of the S-Box output by itself. For non-profiled SCA, we proposed a novel labeling technique that considers bitslice characteristics. We compared our proposed labeling method to MSB and HW labeling by analyzing the three aforementioned datasets. For non-profiled SCA, the proposed labeling method was more effective than the MSB and HW labeling methods on all datasets.

INDEX TERMS Side-channel analysis, deep learning, bitslice implementation, block cipher, PIPO, profiled SCA, non-profiled SCA.

I. INTRODUCTION

Recently, with development of IoT devices, studies of light-weight block cipher that can be used in a limited environment are growing, and then traditional analysis has been observed accordingly [1]. Also, A light-weight block cipher used in an embedded environment such as the IoT devices may have weaknesses in the side-channel analysis. Side-channel analysis (SCA) was proposed by Kocher in 1996 [2] and obtains secret information (e.g., secret keys) by exploiting the side-channel information (e.g., sound, electromagnetic leaks, power consumption) generated when encryption is performed on the target device. Power consumption-based SCA is divided into two categories: profiled SCA and non-profiled SCA.

Profiled SCA creates a profile using a controllable profiling device similar to the target device, and obtains secret information by matching the power traces obtained from the target device with the profile, e.g. template attacks (TA) [3]. Non-profiled SCA obtains secret information through statistical analysis on power traces generated when a fixed secret key and random plaintext are encrypted on the target device, e.g. correlation power analysis, differential power analysis [4].

In traditional SCA, effectiveness is largely dependent on the attacker's ability, such as selecting PoI (Point of Interesting) related to the secret information or the application of additional preprocessing techniques. Therefore, recnetly, SCA techniques using neural networks such as multi layer perceptron (MLP) and convolutional neural network, have recently been proposed to reduce the dependency on the attacker's ability [5]–[7]. Deep-learning (DL)-based profiled SCA predicts the secret key from attack power traces using trained neural networks, while DL-based non-profiled SCA

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen.

determines the secret key using the learning level (e.g., accuracy, loss) of neural networks for each candidate key.

DL-based profiled SCA uses the intermediate value to which the labeling method is applied as the label of the profiling dataset. Labeling methods are generally divided into the hamming weight (HW) [8] and identity (ID) methods [7], [9], [10]. In [11], [12], the performance of the two labeling methods (HW, ID) was compard. The results indicated that HW performed better than ID for fewer layers. However, HW had the problem of imbalanced data for each class, and its performance was lower on datasets with countermeasures.

DL-based non-profiled SCA was proposed by Timon in 2019, and analysis was performed on the AES ASCAD dataset using the most significant bit (MSB), least significant bit (LSB), and HW labeling of the S-Box output [13]. In addition, in other studies, DL-based non-profiled SCA was also performed using the MSB, LSB, or HW label of the target block cipher's S-Box output [14]–[18]. Unlike traditional non-profiled SCA, in DL-based non-profiled SCA, bit models such as MSB and LSB exhibited better performance than HW models.

Existing related studies have focused on the lookup table (LUT) implementation of block ciphers [7]–[18]. So, the labeling method in related studies considers the data characteristics of the LUT implementation, may not be suitable for the bitslice implementation. To the best of our knowledge, there have been no studies on DL-based SCA for the bitslice implemented block ciphers. Therefore, we propose DL-based profiled and non-profiled SCA suitable for the bitslice implementation of block ciphers through analysis of the latest bitslice block cipher, PIPO (Plug-In Plug-Out).

We apply our analysis to several datasets to evaluate the soundness of our experimental results. We use open-source of PIPO code [19] and three datasets collected from AVR, MSP, and ARM-based microcontroller units (MCUs), respectively. Our contributions are as follows:

- DL-based profiled SCA
  1) **Investigating the labeling method that is most suitable for bitslice block ciphers.**
     The previous works on DL-based profiled SCA have focused on the LUT implementation. But the data characteristics of the LUT and bitslice implementation are different, we investigate the suitable labeling method for bitslice block cihpers. Also, We compare the ID, MSB, and HW labeling by analyzing three datasets on the latest bitslice block cipher PIPO. Our results indicate that ID labeling requires approximately 22 times fewer attack traces to derive the secret key than MSB and HW labeling.
- DL-based non-profiled SCA
  1) **Proposing a novel labeling method considering the structure of bitslice block ciphers.**
     To characteristic of the bitslice implementation, 8 bits of the S-Box output is computed at different times. So, we propose binary encoding labeling that is considering these characteristic. Also, we compare our proposed labeling to traditional MSB and HW labeling methods by analyzing three datasets. Our results indicate that the proposed labeling method requires approximately 2.5 times fewer traces to derive the secret key than the MSB labeling method and approximately 3.7 times fewer traces than the HW labeling method.

The remainder of this paper is organized as follows. Section 2 explains the background of our paper, and Section 3 explains the datasets of our experimental. In Section 4 and 5, we propose the DL-based profiled SCA and DL-based non-profiled SCA on the bitslice implementation, respectively, and verify the performance of ID labeling method (profiled SCA) and proposed labeling method (non-profiled SCA) by compare with other labeling methods. Finally, we provide a conclusion and our future works in section 6.

II. BACKGROUND

A. BITSLICE BLOCK CIPHER PIPO

PIPO is a bitslice lightweight block cipher considering the bitslice implementation that was proposed in 2020 [20]. Since it has fewer non-linear operations than other block ciphers, it can be used as an effective countermeasure against SCA (e.g., higher-order masking). In addition, PIPO provides excellent performance in 8-bit AVR software with the bitslice implementation [21].

The block size is 64-bit, and the number of rounds varies depending on the key size. PIPO-64/128 with a 128-bit key size has 13 rounds, and PIPO-64/256 with a 256-bit key size has 17 rounds. The notations used in this paper are shown in Table 1.

TABLE 1. Notations used in this paper.

Notation	Definition
$\oplus$	XOR(eXclusive OR) operator
$\gg$	Right shift operator
$\wedge$	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

The key schedule is simple. The master key K is divided into 64-bit subkeys. For PIPO-64/128, it is divided into $K = K_1||K_0$, and each round key is $RK_r = K_{r \bmod 2}$ where $r = 0, 1, \cdots, 13$. Similarly, for PIPO-64/256, $K = K_3||K_2||K_1||K_0$, and $RK_r = K_{r \bmod 4}$ where $r = 0, 1, \cdots, 17$. Each round consists of the S-Layer, P-Layer, and Key-Addition, and the structure of PIPO is illustrated in Figure 1.

Since the S-Layer uses a bitslice implementation, each S-Box output is computed in parallel using bitwise operations

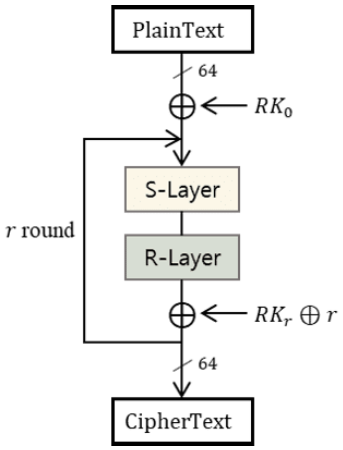


FIGURE 1. PIPO structure.

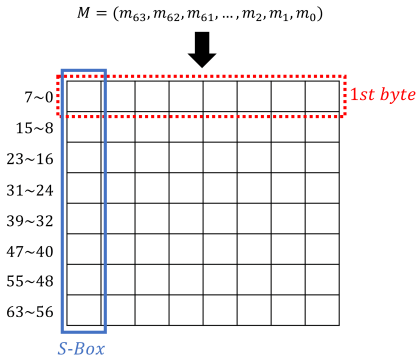


FIGURE 2. S-Layer structure.

without a LUT. Accordingly, as shown in Figure 2, the S-Box output is stored in different registers and consists of operations of the i-th bits of each S-Box output in the j-th byte ($0 \leq i, j \leq 7$).

B. DL-BASED SCA

1) DL-BASED PROFILED SCA

Profiled SCA is a technique that recovers the secret key by matching the power trace on target device to the profile generated from a controllable profiling device. DL-based profiled SCA uses neural networks for this process [9], which is divided into two phases as follows:

- **Profiling phase (training phase)**
  The neural network is trained using profiling traces as the input and intermediate values corresponding to the profiling traces as the labels.
- **Attack phase**
  The attack trace is used as the input of the trained neural network, and the output of the neural network is obtained. Then, the secret key is derived by calculation with a known value (e.g., plaintext, ciphertext) and output.

That is, the neural network is trained on profiling traces with a label (intermediate value), and the label is predicted by inputting unlabeled attack traces into the trained neural network. Similar to traditional SCA, intermediate value is generally the output of a non-linear function (e.g., S-Box). The performance of DL-based SCA is determined according to the neural network model, labeling method, and key determination metric. In this paper, we focus on the performance of different labeling methods.

Most of previous studies focused on the LUT implementation of the block cipher AES [7]–[11] and bitslice cipher PRESENT [12], the performance of attack was compared using the ID or HW labeling method. However, in [22], since there are fewer classes in HW than in the ID, data imbalance occurs in the class, and there is less information about the secret key because different intermediate values are mapped to the same HW class. Thus, the HW is less discriminant than the ID. However, the HW (with 9 classes) is more resistance to noise than the ID (with 256 classes), and there are thus fewer misclassifications. Therefore, it is necessary to select a suitable labeling method according to the training data.

2) DL-BASED NON-PROFILED SCA

Non-profiled SCA is a technique that recovers the secret key by analyzing multiple power traces collected during encryption of random plaintexts with a fixed secret key on target device. The analysis process of DL-based non-profiled SCA is as follows:

- **Training phase**
  The neural network is trained using attack traces as the input and intermediate values (e.g., S-Box output of first round) for the arbitrary guessed key as a label. By repeating this step for each guessed key, the neural network is trained for each key.
- **Attack phase**
  The learning level of each neural network is determined by a metric (e.g., training loss). The guessed key of the neural network with the highest learning level is considered the secret key.

A label calculated by the right key is actually a value related to the attack traces; therefore, the neural network is trained well. However, a label calculated by an wrong key is a value unrelated to the attack traces; therefore, the neural network is not trained well. Thus, we assume that the guessed key training the neural network as the highest learning level is the right key. In DL-based non-profiled SCA, the intermediate values of wrong keys should have a low correlation with the intermediate value of the right key, as in traditional non-profiled SCA. Therefore, we take the output of a non-linear function, such as S-Box, as the intermediate value. As with DL-based profiled SCA, we focus on the difference in performance due to different labeling methods.

In [13], Timon set the LSB, MSB, or HW of the S-Box output as the label and performed DL-based non-profiled SCA for AES. The results indicated that binary labeling (MSB, LSB) led to better performance than HW labeling. In addition, according to [13], if the ID of the S-Box output is used as the classification label, then the learning levels of all

TABLE 2. Related work vs. our paper.

Paper	Profiled/Non-profiled	Target cipher	Implementation method	Labeling method
[7]	Profiled	AES	LUT	ID
[8]	Profiled	AES	LUT	HW
[9]	Profiled	Masked AES	LUT	ID
[10]	Profiled	AES, Masked AES	LUT	ID
[11]	Profiled	AES	LUT	HW, ID
[12]	Profiled	AES, PRESENT	LUT	HW, ID
[13]	Non-profiled	AES	LUT	MSB, LSB, HW
[14]	Non-profiled	Masked AES	LUT	MSB, LSB
[15]	Non-profiled	AES, Masked AES	LUT	LSB
[16]	Non-profiled	Masked AES	LUT	LSB, HW-based binary model
[17]	Non-profiled	Masked AES	LUT	LSB
[18]	Non-profiled	DES, SM4	LUT	HW, One-bit model
Our paper	Profiled, Non-profiled	PIPO	Bitslice	MSB, HW, ID, BE (our proposed)

guessed keys' neural networks are equivalent. Thus, ID labeling is not used in DL-based non-profiled SCA. Later studies also performed DL-based non-profiled SCA using MSB or LSB labeling on the LUT implementation of AES [14]–[17]. In [18], Xiangliang *et al.* performed DL-based non-profiled SCA on SM4 and DES using the MSB, HW labeling, and they got the result that the MSB labeling has better performance than HW labeling.

Table 2. shows summary of related works and our paper, all of related works is analyzed the LUT implementation of block ciphers. In this paper, we propose the DL-based SCA on the bitslice implementation of block cipher. Furthermore, we compare the performance of the MSB and HW labeling methods with that of the ID (DL-based profiled SCA) and proposed (DL-based non-profiled SCA) labeling method by analysis on bitslice block cipher PIPO.

III. EXPERIMETAL OF DATASETS

We obtained power traces when PIPO's first-round S-Layer and P-Layer operated on three MCUs XMEGA128D4, MSP430F2618, and STM32F303 with open-source of PIPO code [19]. For the profiling phase, 50,000 profiling traces of random keys and plaintexts were collected, and 10% of the profiling traces were used for validation in the profiling phase. For the attack phase, 5,000 attack traces of a fixed key and random plaintexts were collected, and DL-based profiled and non-profiled SCA used 1,000 of them as attack traces. The experimental environment is shown in Table 3.

A. 8-BIT MCU XMEGA128D4

ChipWhisperer-Lite [23] was used as the capture board with a sampling rate of 29.538 MS/s, and AVR XMEGA128D4 (8-bit MCU) was used as the target board as Figure 3. The XMEGA128D4 traces are illustrated in Figure 4.

B. 16-BIT MCU MSP430F2618

As shown in Figure 5, SCARF-MSP430 V1.5 board provided by ETRI [24] was used and MSP430F2618 (16-bit MCU) was used as the target board. Traces were collected using the Lecroy HDO610 oscilloscope with a sampling rate of 500 MS/s. The MSP430F2618 traces are illustrated in Figure 6.

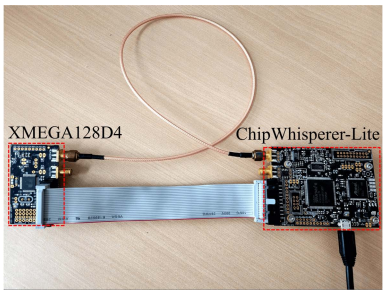


FIGURE 3. ChipWhisperer-Lite and 8-bit MCU XMEGA128D4.

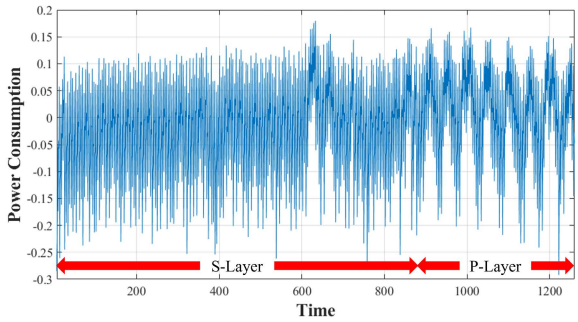


FIGURE 4. XMEGA128D4 trace.

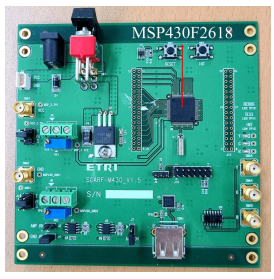


FIGURE 5. SCARF-MSP430 V1.5 board with 16-bit MCU MSP430F2618.

C. 32-BIT MCU STM32F303

ChipWhisperer-Lite was used as the capture board with a sampling rate of 29.538 MS/s, and ARM STM32F303RCT7

TABLE 3. Experimental environment.

Dataset	XMEGA128D4	MSP430F2618	STM32F303
Capture Board	ChipWhisperer-Lite	Lecroy Oscilloscope HDO610	ChipWhisperer-Lite
Target Board	8-bit MCU XMEGA128D4	16-bit MCU MSP430F2618	32-bit MCU STM32F303
Sampling Rate	29.538 MS/s	500 MS/s	29.538 MS/s
Number of points	1,260	1,250	1,152

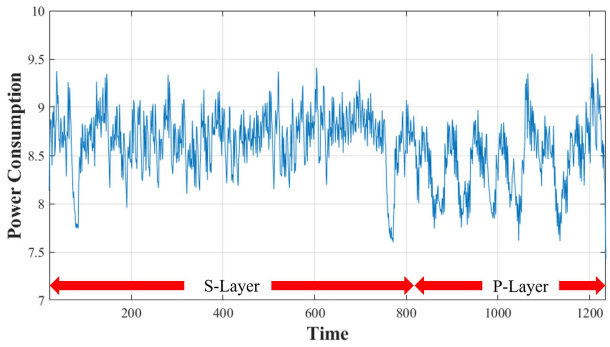


FIGURE 6. MSP430F2618 trace.

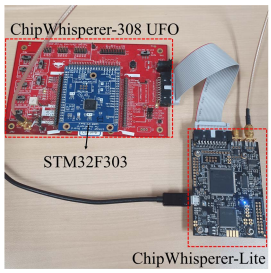


FIGURE 7. ChipWhisperer-Lite and ChipWhisperer-308 UFO board with 32-bit MCU STM32F303.

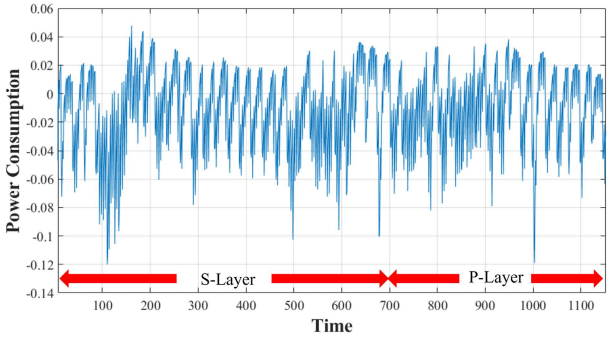


FIGURE 8. STM32F303 trace.

(32-bit MCU) was used as the target board combined with a ChipWhisperer-308 UFO board [25], as displayed in Figure 7. The STM32F3 traces are illustrated in Figure 8.

IV. DL-BASED PROFILED SCA ON PIPO

A. ATTACK SCENARIO

1) PROFILING PHASE

The profiling phase is the process of generating a profile by training the neural network using power traces obtained from the profiling device. In this paper, the neural network was constructed using an MLP model. In the PIPO S-Layer, the S-Box output was not stored in the same register because each byte was implemented in parallel due to the bitslice structure. Therefore, an MLP with a fully-connected layer was used to train each bit of the S-Box output distributed in the power traces. The power traces were used as the input of the neural network. In addition, we used not only the ID and HW labeling, but also the MSB labeling considering the 1-bit model. Thus, the ID, MSB, or HW value of the PIPO S-Box output was used as the label. The experimental results of the three labeling methods were then compared.

The ID labeling method used the intermediate value as the label; thus, it had 256 classes for 8-bit. The MSB labeling method uses the MSB of the intermediate value; thus, it has two classes (0, 1). Finally, the HW labeling method used the number 1 for binary representation of the intermediate value. In the case of 8-bit, there were nine classes.

2) ATTACK PHASE

The attack phase is the process of recovering a secret key by inputting the power traces collected from the attack device into the trained neural network in the profiling phase. Since the output of the neural network is the S-Box output, the secret key is recovered by performing an inverse S-Box and an XOR operation with the plaintext.

B. EXPERIMENTAL RESULTS

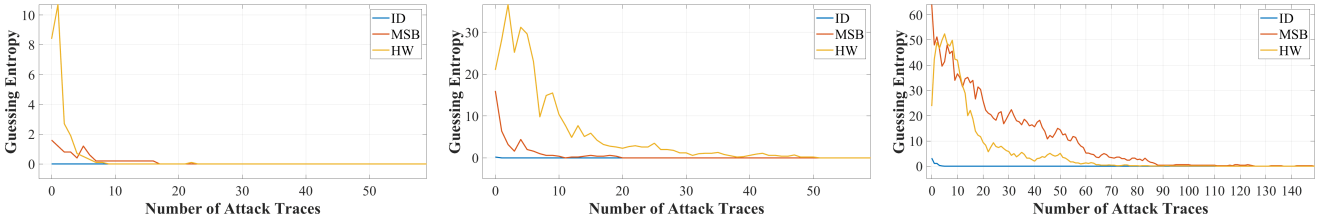
1) MLP ARCHITECTURE

This section describes the MLP model used in DL-based profiled SCA. It had two hidden layers consisting of 100 and 50 nodes, and "Leaky ReLU" was used as the activation function of the hidden layer. The output layer consisted of 2, 9, or 256 nodes according to the labeling method and used "Softmax" as the activation function. Each hidden layer and input layer include batch normalization and dropout to prevent over-fitting.

Figure 4 shows our MLP architecture, where x is the number of points in the power traces, and y is the number of possible intermediate values (2, 9, or 256). The details of the hyperparameters are presented in Figure 5.

2) EXPERIMENTAL RESULTS

We used guessing entropy (GE) to evaluate the attack performance in SCA [26]. GE is the average rank for the right key, and the faster GE converges to 0, the more successful the



(a) Result of XMEGA128D4 (b) Result of MSP430F2618 (c) Result of STM32F303

FIGURE 9. Profiled SCA results of first byte of three labeling methods.

TABLE 4. MLP on DL-based profiled SCA.

Layer	Node (in, out)	Kernel initializer
Input	(x, x)	-
Dense	$(x, 100)$	he uniform
Leaky ReLU	(100, 100)	-
Dense	(100, 50)	he uniform
Leaky ReLU	(50, 50)	-
Dense	$(50, y)$	he uniform
Softmax	(y, y)	-

TABLE 5. Hyperparameters of DL-based profiled SCA.

Label	ID, HW, MSB of S-Box output
Optimizer	Adam (learning rate = 0.0001)
Batch Size	32
Epochs	100
Loss function	Categorical cross entropy
Dropout	0.2

attack is. fig:1 byte analysis of three leakage models shows the experimental results of the first byte of three labeling methods for three datasets (XMEGA128D4, MSP430F2618, and STM32F303). In the graph, the x-axis represents the number of attack traces, while the y-axis represents the GE. Table 6 presents the number of traces required for GE to converge to 0 in each labeling method on three datasets.

TABLE 6. Number of traces required for GE to converge to 0 in DL-based profiled SCA.

Labeling	XMEGA128	MSP430	STM32F3
ID	≥ 1	≥ 2	≥ 6
MSB	> 25	≥ 23	> 150
HW	> 25	≥ 53	> 120

On XMEGA128, MSB and HW required more than 25 traces for GE to converge to 0, whereas ID required only one trace. On MSP430, GE converged to 0 using 23 or more traces for MSB and 53 or more traces for HW, whereas only two traces were required for ID. Finally, on STM32F3, GE converged to 0 using more than 150 traces for MSB and 120 traces for HW, whereas only six traces were required for ID.

On all three datasets, the ID labeling method exhibited the best performance compared to the other labeling methods. In particular, STM32F3 was analyzed with more than 150 traces by the HW and MSB labeling methods, but only six traces by the ID labeling method. Thus, the neural network effectively extracts and trains each bit of the S-Box output distributed in the power traces.

V. DL-BASED NON-PROFILED SCA ON PIPO

A. ATTACK SCENARIO

1) BINARY ENCODING LABELING METHOD

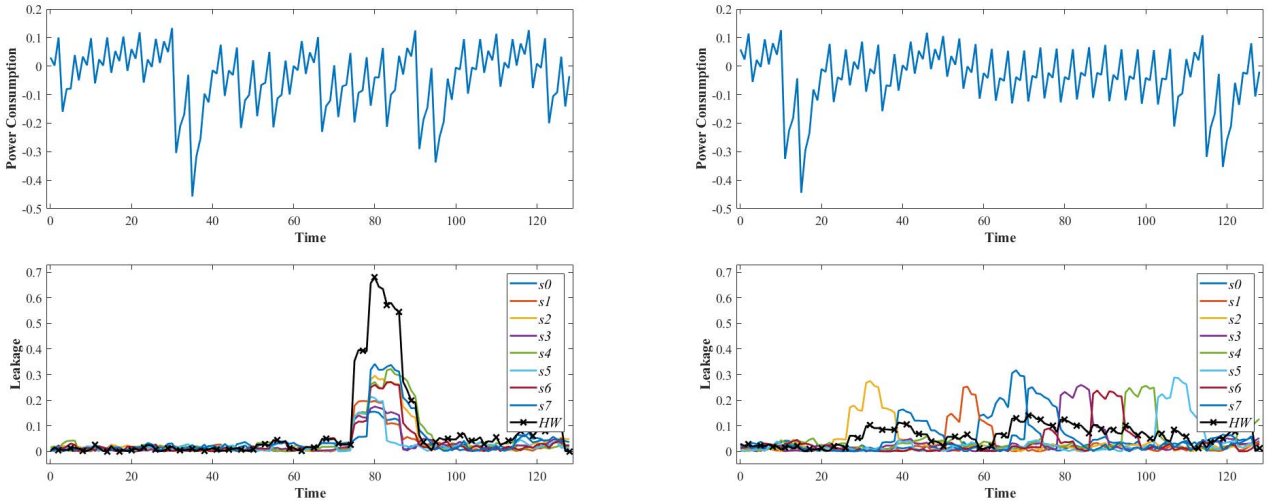
In this section, we propose a new labeling method for DL-based non-profiled SCA on bitslice block ciphers. Figure 10 illustrates the difference in leakage in the LUT and bitslice implementation. In order to illustrate the difference in leakage, 5,000 traces of the first S-Box (LUT implementation) and of the S-layer (bitslice implementation) about PIPO were collected, respectively. Above implementations was run on XMEGA128D4 and using correlation power analysis, we analyzed the leakage of each bit of the S-Box output in each implementation. The upper graph in Figure 10 illustrates the power traces of each implementation, while the lower graph indicates which bit leaked at which point in time. Here *s0, s1, s2, s3, s4, s5, s6,* and *s7* represent each bit of the first S-Box output of the first round, where *s0* is the LSB and *s7* is the MSB. And *HW* is the HW of S-Box output $(s7||s6||s5||s4||s3||s2||s1||s0)_2$. Let T_{si} be the time point existing the highest leakage about the value *si* ($i \in \{0, 1, 2, \ldots, 7\}$). Then, leakage of each implementation have below properties, respectively.

Property 1 (LUT Implementation): $T_{s0} = T_{s1} = \ldots = T_{s7}$ and the HW has the **highest** leakage in 9 models (*s0*, *s1*, *s2*, *s3*, *s4*, *s5*, *s6*, *s7*, and *HW* model).

Property 2 (Bitslice Implementation): $T_{s0} \neq T_{s1} \neq \ldots \neq T_{s7}$ and the HW has the **lowest** leakage in 9 models (*s0*, *s1*, *s2*, *s3*, *s4*, *s5*, *s6*, *s7*, and *HW* model).

Actually, in Figure 10 (a), since 8 bits of the S-Box output were computed all at once, $T_{s0} = T_{s1} = \ldots = T_{s7}$ and 8 bits HW has the highest leakage. On the other hand, in Figure 10 (b), since 8 bits of the S-Box output were computed at different time points, $T_{s0} \neq T_{s1} \neq \ldots \neq T_{s7}$ and 8 bits HW has the lowest leakage.

According to the above Property 1, Property 2, it can be predicted that the HW labeling will have relatively higher



(a) LUT implementation (b) Bitslice implementation

FIGURE 10. Leakage of the LUT and bitslice implementation traces (*s0*, *s1*, ..., *s7* are an S-Box output bit, *HW* is the HW of the S-Box output).

performance in the LUT implementation and, conversely, lower performance in the bitslice implementation. In the case of the single-bit labeling, it can be expected that similar performance in both LUT and bitslice implementation, because the size of leakages is similar. However, single-bit labeling on the bitslice implementation has a limitation that the power information about the only one time point of $T_{s0} \neq T_{s1} \neq \ldots \neq T_{s7}$ is used. Thus, considering these characteristics of the bitslice implementation, we propose binary encoding (BE) labeling, which uses all bits of S-Box output and constructs each bit independently. Algorithm 1 is the proposed BE labeling algorithm.

Algorithm 1 BE Labeling Algorithm

Input: 8-bit value a

Output: binary encoded label $b = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7), a_i \in \{0, 1\}$

1: **for** $k = 0$ to 7 **do**
2: $\quad a_k \leftarrow (a \gg k) \wedge 1$
3: **end for**
4: **Return** $b = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$

2) TRAINING PHASE

The training phase is the process of training the neural network for each guessed key. Since PIPO-64/128 is a block cipher using an S-Box, DL-based non-profiled SCA can be performed by applying the analysis described by Timon [13]. For DL-based non-profiled SCA, we set the neural network model as the MLP, the intermediate value to be used for the label as the output of the S-Box, and the metric and

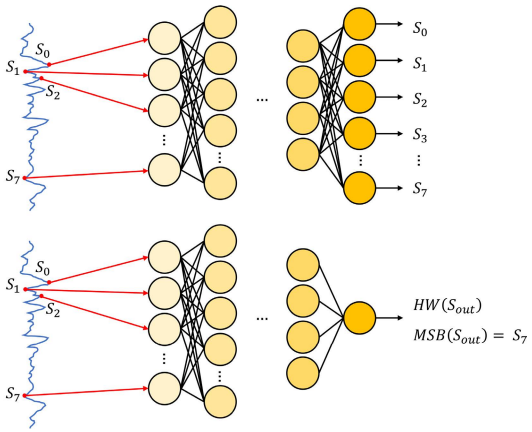


FIGURE 11. MLP architecture of the HW, MSB, and BE labeling.

loss function as the mean squared error. Related studies used the LSB, MSB, or HW value as the label in DL-based non-profiled SCA [13]–[18]. We then compare MSB, HW labeling to BE labeling.

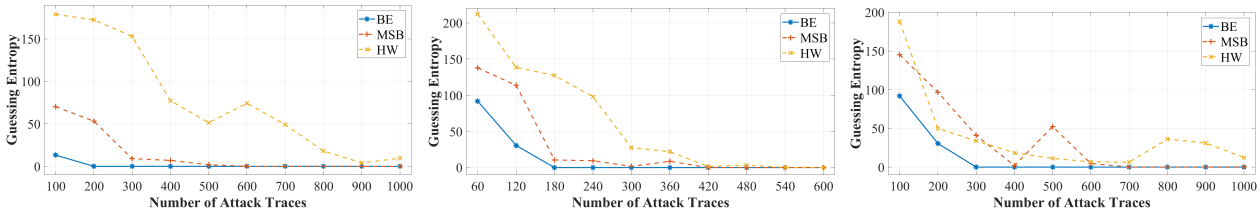
3) ATTACK PHASE

The attack phase is the process of determining the right key by judging the learning level of the trained neural networks. In this paper, we use the training loss of the last epoch as the learning level. We sort the training loss of the neural networks for each guessed key in ascending order and measure the rank of the right key.

B. EXPERIMENTAL RESULTS

1) MLP ARCHITECTURE

This section describes the MLP model used in DL-based non-profiled SCA. It had one hidden layer, where the number of layer nodes was 200. The output layer consisted of 1 or 8 nodes according to the labeling method and the activation function of the hidden layer was "ReLU", while that of the



(a) Result of XMEGA128D4 (b) Result of MSP430F2618 (c) Result of STM32F303

FIGURE 12. Non-profiled SCA results of first byte of three labeling methods.

output layer was "Sigmoid". We used the mean squared error as the loss function and key guessing metric. To maintain the relevance of each class in BE and HW labeling, we did not apply one-hot encoding in DL-based non-profiled SCA.

Table 7 presents our MLP architecture, where x is the number of points in the power traces, and y is 1 (MSB and HW labeling) or 8 (BE labeling). Figure 11 shows MLP architecture of the HW, MSB, and BE labeling. In the case of the BE labeling, there are 8 output nodes and whole loss value is calculated by summation of each loss value of the output nodes. The details of the hyperparameters are presented in Table 8.

TABLE 7. MLP on DL-based non-profiled SCA.

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

TABLE 8. Hyperparameters of DL-based non-profiled SCA.

Label	BE, MSB, HW of S-Box output
Optimizer	Adam (learning rate = 0.0005)
Batch Size	100
Epochs	100
Loss function	Mean squared error

2) EXPERIMENTAL RESULTS

As in DL-based profiled SCA, we used GE to evaluate the attack performance. We obtained the GE in 10 iterations of the analysis for each number of attack traces in each dataset. Figure 12 presents the DL-based non-profiled SCA results of the first byte of the three labeling methods for three datasets. In the graph, the x-axis represents the number of attack traces, while the y-axis represents GE. Analysis was performed on 100 unit traces on the XMEGA128 and STM32F3 datasets, and analysis was performed on 60 unit traces on the MSP430 dataset. Table 9 presents the number of traces required for GE to converge to 0 in each labeling method on three datasets. On XMEGA128, MSB required 600 traces or more for GE to converge to 0, whereas HW required more than 1,000 traces. In contrast, BE required only 200 traces. On MSP430, MSB required 420 traces or more for GE to converge to 0, whereas HW required 540 traces or more. In contrast, BE required only 180 traces. Finally, on STM32F3, MSB required 700 traces or more for GE to converge to 0, whereas HW required more than 1000 traces. However, BE required only 300 traces. On all datasets, the proposed BE labeling method had the best performance in DL-based non-profiled SCA.

TABLE 9. Number of traces required for GE to converge to 0 in DL-based non-profiled SCA.

Labeling	XMEGA128	MSP430	STM32F3
BE	≥ 200	≥ 180	≥ 300
MSB	≥ 600	≥ 420	≥ 700
HW	> 1000	≥ 540	> 1000

VI. CONCLUSION

In this paper, we propose DL-based profiled and non-profiled SCA for three datasets (XMEGA128D4, MSP430F2618, and STM32F303) of PIPO-64/128. For DL-based profiled SCA, the experimental results for the three labeling methods (ID, MSB, and HW) were compared. For our proposed BE labeling method, the number of traces require for GE converge to 0 was more than 200, which was greater than that of existing labeling methods; therefore, the BE labeling method was excluded from profiled SCA. In summary, the ID labeling method requires approximately 22 times fewer attack traces than the MSB and the HW labeling method. This signifies that since the neural network effectively extracts and trains each bit of the S-Box output from the power traces, the performance will be high even if the S-Box output is labeled without considering the bitslice characteristics of the bitslice block cipher.

For DL-based non-profiled SCA, we propose the BE labeling method considering the structure of bitslice block ciphers. We compared our proposed BE labeling method with the HW and MSB labeling methods. In summary, the proposed labeling method requires approximately 2.5 times fewer traces than the MSB labeling method and approximately 3.7 times fewer traces than the HW labeling method on average. Furthermore, BE labeling is more effective than MSB and HW labeling on all datasets. The results demonstrate that the proposed BE labeling is more effective than MSB and HW labeling regardless of the target board.

We expect that our investigation and proposed method to also be applied to (high-order) analysis on other bitslice block ciphers. In future work, we plan to analyze other bitslice block ciphers applying SCA countermeasures using the proposed DL-based profiled and non-profiled SCA. Also, we will apply a combination of our proposed labeling method and the proposed methodologies in DL-based SCA (e.g., custom activation function, loss function…).

REFERENCES

[1] A. D. Dwivedi, "Security analysis of lightweight IoT cipher: Chaskey," *Cryptography*, vol. 4, no. 3, p. 22, Aug. 2020.

[2] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Proc. 16th Annu. Int. Cryptol. Conf. (CRYPTO)* (Lecture Notes in Computer Science), vol. 1109, N. Koblitz, Ed. Santa Barbara, CA, USA: Springer, Aug. 1996, pp. 104–113.

[3] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Proc. 4th Int. Workshop Cryptograph. Hardware Embedded Syst. (CHES)* (Lecture Notes in Computer Science), vol. 2523, B. S. Kaliski, Ç. K. Koc, and C. Paar, Eds. Redwood Shores, CA, USA: Springer, Aug. 2002, pp. 13–28.

[4] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. 19th Annu. Int. Cryptol. Conf. (CRYPTO)* (Lecture Notes in Computer Science), vol. 1666, M. J. Wiener, Ed. Santa Barbara, CA, USA: Springer, Aug. 1999, pp. 388–397.

[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.

[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. 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)*, Apr. 2021, pp. 395–400.

[8] 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 *Proc. Int. Conf. Secur., Privacy, Appl. Cryptogr. Eng.* Springer, 2018, pp. 157–176.

[9] 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, 2020.

[10] J. Edmonds and T. Moon, "Machine learning-based side-channel analysis on the advanced encryption standard," Tech. Rep., 2021.

[11] L. Weissbart, "Performance analysis of multilayer perceptron in profiling side-channel analysis," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Springer, 2020, pp. 198–216.

[12] A. Heuser, S. Picek, S. Guilley, and N. Mentens, "Side-channel analysis of lightweight ciphers: Does lightweight equal easy?" in *Proc. Int. Workshop Radio Freq. Identificat., Secur. Privacy Issues*. Springer, 2016, pp. 91–104.

[13] 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, Feb. 2019.

[14] 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.

[15] 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.

[16] D. Bae and J. Ha, "Performance metric for differential deep learning analysis," *J. Internet Services Inf. Secur.*, vol. 11, no. 2, pp. 22–33, 2021.

[17] 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. (ASHES@CCS)*, C. Chang, U. Rührmair, S. Katzenbeisser, and D. Mukhopadhyay, Eds. New York, NY, USA: ACM Press, Nov. 2021, pp. 29–40.

[18] M. Xiangliang, L. Bing, W. Hong, W. Di, Z. Lizhen, H. Kezhen, and D. Xiaoyi, "Non-profiled deep-learning-based power analysis of the SM4 and DES algorithms," *Chin. J. Electron.*, vol. 30, no. 3, pp. 500–507, 2021.

[19] H. Kim, Y. Jeon, G. Kim, J. Kim, B. Sim, D. Han, H. Seo, S. Kim, S. Hong, J. Sung, and D. Hong. *PIPO-Blockcipher*. Accessed: Jun. 2, 2022. [Online]. Available: https://github.com/PIPO-Blockcipher/PIPO-Blockcipher

[20] H. Kim, Y. Jeon, G. Kim, J. Kim, B. Sim, D. Han, H. Seo, S. Kim, S. Hong, J. Sung, and D. Hong, "PIPO: A lightweight block cipher with efficient higher-order masking software implementations," in *Proc. 23rd Int. Conf. Inf. Secur. Cryptol. (ICISC)* (Lecture Notes in Computer Science), vol. 12593, D. Hong, Ed. Seoul, South Korea: Springer, 2020, pp. 99–122.

[21] H. Kim, Y. Jeon, G. Kim, J. Kim, B. Y. Sim, and D. G. Han, "A new method for designing lightweight S-boxes with high differential and linear branch numbers, and its application," *IEEE Access*, vol. 9, pp. 150592–150607, 2021.

[22] S. Picek, A. Heuser, A. Jovic, S. Bhasin, and F. Regazzoni, "The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2019, no. 1, pp. 1–29, 2019.

[23] NewAE Technology. *ChipWhisperer-Lite (CW1173) Two-Part Version*. Accessed: Jun. 2, 2022. [Online]. Available: https://rtfm.newae.com/Capture/ChipWhisperer-Lite/

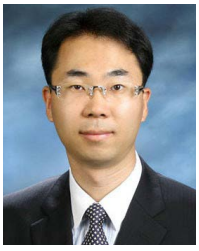
[24] Y. Choi, D. Cho, and J. Ryou, "Implementing side channel analysis evaluation boards of KLA-SCARF system," *J. Korea Inst. Inf. Secur. Cryptol.*, vol. 24, no. 1, pp. 229–240, 2014.

[25] NewAE Technology. *ChipWhisperer-308 UFO Board*. Accessed: Jun. 2, 2022. [Online]. Available: https://rtfm.newae.com/Targets/CW308%20UFO/

[26] F.-X. Standaert, T. G. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Springer, 2009, pp. 443–461.

JI-EUN WOO received the B.S. degree in information security, cryptology, and mathematics from Kookmin University, Seoul, Republic of Korea, in 2021, where she is currently pursuing the master's degree in financial information security. Her research interests include side-channel attacks, symmetric key cryptography, and deep-learning-based analysis.

JAESEUNG HAN 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 side-channel attacks, symmetric key cryptography, and lattice-based cryptography, and deep-learning-based analysis.



DONG-GUK HAN received the B.S. and M.S. degrees in mathematics and the Ph.D. degree in information security engineering from Korea University, Seoul, Republic of Korea, in 1999, 2002, and 2005, respectively. He was a Postdoctoral Researcher at Future University Hakodate, Hokkaido, Japan. After finishing his doctoral course, he was then 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 at the Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea. He is currently working as a Professor with the Department of Information Security, Cryptology, Mathematics, Kookmin University, Seoul. He is a member of KIISC, IEEK, and IACR.