Deep Learning-based Attacks on Masked AES Implementation

Daehyeon Bae[1], Jongbae Hwang[2], Jaecheol Ha[3*]

[1] School of Cybersecurity, Korea University, South Korea
[2] Next Network Security Provider (NNSP), South Korea
[3] Division of Computer Engineering, Hoseo University, South Korea
noeyheadb@gmail.com, qlghd98@gmail.com, jcha@hoseo.edu

Abstract

To ensure the confidentiality of the message, the AES (Advanced Encryption Standard) block cipher algorithm can be widely used. Furthermore, an implementation of masked AES is often used to resist side-channel attacks. To recover secret keys embedded in cryptographic devices with masked AES, we present some side-channel attacks based on deep learning models in profiling and non-profiling scenarios. The proposed method which applies the mask value profiling technique represents new approaches for extracting the secret key. To defeat the masked AES implementation, deep learning models such as multi-layer perceptron and convolutional neural networks are developed. In a non-profiling scenario, we adopt the DDLA (Differential Deep Learning Analysis) to extract sensitive information such as the secret key. The main idea of our method is that it is possible to adopt a new binary labeling method to conduct the DDLA based on the HW (Hamming Weight) model. We show several experiments using real power traces measured from the ChipWhisperer platform in profiling attacks and the ASCAD dataset in non-profiling attacks respectively. Whether we target naïve or masked AES implementation, the experimental results show the predominant key recovery accuracy.

Keywords: Internet of things, Side-channel attack, Masked AES implementation, Deep learning

1 Introduction

The AES (Advanced Encryption Standard) [1] known as the most representative block cipher can be used to provide confidentiality of communication messages between two peers. Nevertheless, several side-channel attacks have been proposed as an easy way for an adversary to find secret keys stored in hardware devices [2-3]. The power analysis attack, which observes power consumption as leakage information, is one of the most popular side-channel analyses. In particular, the side-channel attacks [4-5] on naïve implementation of block cipher algorithm are considered serious threats to ensuring confidentiality.

To defeat side-channel analysis, several countermeasures have been proposed. The masking countermeasures, in which we randomize the intermediate computation values, are widely used to thwart side-channel analysis. Furthermore, the masking technique can be also adopted to protect AES implementation.

A Boolean masking method is used to protect the vulnerable AES implementation. Here, the output of the S-Box is randomized using a Boolean function such as the XOR operation [6]. The security level of the masking technique can be evaluated as the number of secret shares. We call a countermeasure in which d masks are applied to one variable as a $d\text{-}th$ masking, and this countermeasure can provide robustness against $d\text{-}th$ order attacks.

In this paper, we present several power analysis attacks that can extract the secret key using the profiling and non-profiling based on Multi-Layer Perceptron (MLP) [7] and Convolutional Neural Network (CNN) [8] models. The practical research on side-channel attack has seen some interesting advancements since its introduction by Z. Martinasek *et al.* [9], in which was MLP shown to break naive AES. And R. Gilmore *et al.* demonstrated that the masked countermeasure of AES can be incapacitated by a neural network-based profiling attack [10] with 91.8% accuracy for recovering the mask values and 88.4% for secret keys respectively.

In the non-profiling scenario, we use the DDLA (Differential Deep Learning Analysis) [11] to defeat the first-order masking countermeasure. The DDLA is a deep learning-based non-profiling attack method. In the attack phase without the profiling phase, we train the deep learning models for each hypothetical key. If there was a validate key, this correct key can be extracted through the difference of metric values such as accuracy and cost (loss). One of the important factors that determine the success of non-profiling attacks is how to determine the label of deep learning models, including HW (Hamming Weight) labeling and binary labeling. We proposed a novel HW-based binary labeling method in consideration of performance and efficiency.

We present some experiments on masked AES implementation using real power traces measured from the ChipWhisperer platform in profiling attacks and the ASCAD dataset [12] in non-profiling attacks respectively. In the profiling attacks, we divided the deep learning-based analysis into two phases: mask value recovery and masked Sbox output recovery phase. We perform power analysis attacks using MLP and CNN models by directly implementing the AES-128 encryption on the XMEGA128 microcontroller [13].

Furthermore, we can break the masked AES using the MLP-based DDLA without any preprocessing or separating the attack steps. As a result, we confirmed that the secret key

*Corresponding Author: Jaecheol Ha; E-mail: jcha@hoseo.edu

DOI: 10.53106/160792642022072304024

of masked AES can be extracted with 81.0% accuracy using MLP and 75.4% accuracy using CNN. And the DDLA attack can extract perfectly the secret key of masked AES implementation when the epoch value is 10 or more.

2 Overview of Side-Channel Analysis

2.1 Power Analysis Attack

Side-channel signals, such as power, electromagnetic radiation, or operational time signal leaked hardware, can be used to extract sensitive information. We often divide these side-channel attacks into two categories. One is so-called profiling attacks such as the template attack, the stochastic attack, and machine learning-based attacks [14-18]. The other is non-profiling attacks such as DPA [2], CPA [3], and MIA [19].

In the case of 1st-order CPA, the attacker measures several side-channel signals and generates a leakage model of S-Box operation ($\mathrm{Sbox}(p \oplus k)$). Then, the attacker analyzes the correlation between the side-channel signal and modeled leakage to extract the secret key. Here, we denote p as the input message and k as a secret key.

In the profiling attack, before the attack on the victim device, we should measure the power traces at POI (Points Of Interest) of the profiling device in advance. After that, we can re-setting the embedded key of the profiling device for measuring side-channel leakage.

2.2 Countermeasure

To thwart the side-channel analysis, the masking countermeasure that unlinks the dependency between the side-channel signal and calculated (expected) leakage using some random numbers. Boolean masking, one of the several masking techniques, is usually used to protect naïve AES implementation consisting of the simplicity of Boolean operation. In Boolean masking, the sensitive information such as the Sbox operation, $\mathrm{Sbox}(p\oplus k)$ in AES algorithm, should be hidden by XORing it using a random mask value as follows:

$$S= \mathrm{Sbox}(p\oplus k) \oplus m_o = \mathrm{MSbox}(p\oplus m_i\oplus k) \qquad \textbf{(1)}$$

Our focus is 1st-order masking, where an original Sbox output is hidden by an input mask m_o such as $S= \mathrm{Sbox}(p\oplus k) \oplus m_o$. In the first-order countermeasure, the value of $(p \oplus k)$ can be denoted as a variable x. That is, $S= \mathrm{Sbox}(x)\oplus m_o = \mathrm{MSbox}(x \oplus m_i)$. To implement a first-order masking countermeasure, we generate MSbox using the following equation:

$$\mathrm{Sbox}[index]\oplus m_o = \mathrm{MSbox}[index \oplus m_i] \qquad \textbf{(2)}$$

For the second-order attack, the attacker has to combine side-channel leakages from two POIs using some mathematical methods. Then, the distinguisher such as Pearson's r is used to recover the secret key leaked from the masked intermediate values.

3 Deep Learning Techniques

Deep learning is a kind of machine learning technique that uses a neural network. A neural network is composed of perceptrons (neurons). To perform profiled SCAs, some previous publications were presented using different types of neural networks such as MLP and CNN [9-10, 16-18]. In [20], E. Cagli *et al*. showed that the CNN's translation-invariance property shows outstanding performance with misaligned signals. Therefore, we will intensively apply MLP and CNN models to perform power-based SCAs successfully.

The Deep learning model has been adopted to break some cryptographic algorithms. The process of DL-based side-channel analysis consists of the two steps:

- A training step: In this step, several side-channel signals measured from the profiling device are utilized to tune the weights to minimize error.

- A inference step: In this step, sensitive intermediate values can be extracted.

3.1 Multi-Layer Perceptron (MLP)

The MLP consists of perceptrons stacked in multi-layer as described in Figure 1. An MLP model is composed of an input layer, some hidden layers, and an output layer. The output of each node is forwarded to the next layer. The weights are tuned to minimize the error in the training step.

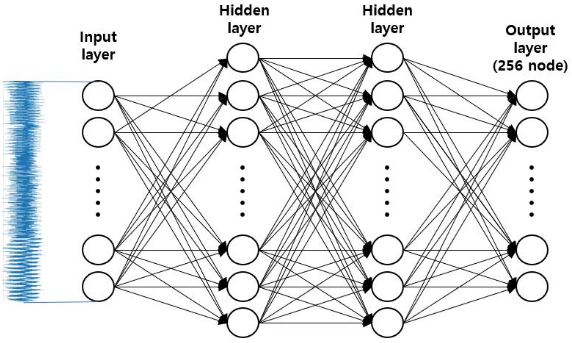


Figure 1. A brief structure of the Multi-Layer Perceptron model

3.2 Convolutional Neural Network (CNN)

The structure of CNN is composed of a feature extraction layer and classification layer as shown in Figure 2. The feature extraction layers have two types of sublayers: the convolutional layer, and the pooling layer. The convolutional layer conducts the convolution on the input data by shifting a set of kernels (filters) and passing its result to the pooling layer. The pooling layers are used to decrease the computing load by compressing the data. In the phase of training the CNN model, we optimize the weights of the filters to extract features.

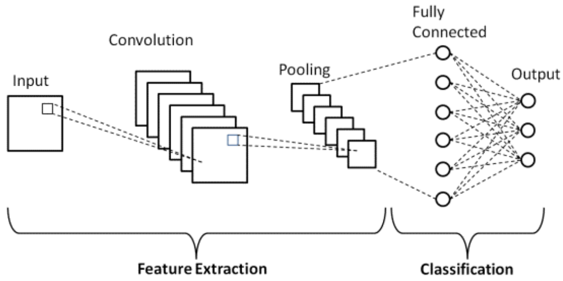


Figure 2. Convolutional Neural Network model

4 Side-Channel Analysis Based on the Deep Learning Models

In the next section, our side-channel attacks based on deep learning models are more effective than other previous attacks through experimental attacks on AES implementations.

4.1 Experimental Setup

For the profiling and non-profiling attack, we set up the experimental environment using a ChipWhisperer platform [13]. In the profiling attack scenario which has a profiling device, we measure the power consumption traces while AES encryption is being performed. Then, the adversary attempts to extract the secret key of AES by measuring a power trace of a victim device. On the other hand, the profiling scenario assumes that an adversary can only use measurements from the target device.

For experimental side-channel attacks, we implement an AES encryption scheme on a target board equipped with an XMEGA128 microcontroller. In addition, we supply the clock signal of 7.37MHz to the target board using the ChipWhisperer capture board. The traces measured from the target board were divided into a 3:1 ratio for the validation and test phase.

4.2 Attack on Naive AES Implementation

We only discuss the AES encryption with a 128-bit secret key as shown in Figure 3. First, the 16-byte plaintexts are fed into the AddRoundKey function, in which inputs are XORed with the initial round key. After initial AddRoundKey processing, the state composed of a 4x4 matrix of bytes is followed by round transformations such as SubBytes, ShiftRows, MixColumns, and AddRoundKey functions. The MixColumns operation is skipped in the final round.

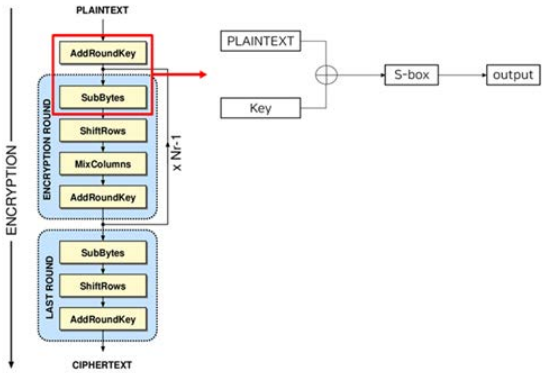


Figure 3. A structure of AES encryption

The following Figure 4 shows a particular magnification of the power signal measured from 1~3 rounds of AES. The small patterns for 16 bytes in a power signal can be identified.

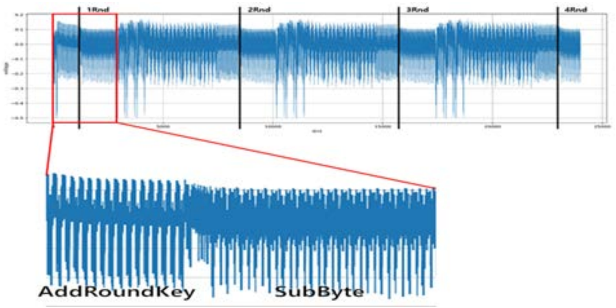


Figure 4. A Power signal of AES encryption

Thousands of training data concerning the Sbox output were characterized. Here, the sample value of each power signal is used as a feature of the training data, and the label is generated as the SBox output values. Therefore, the number of input neurons is equal to the length of the power signal.

The goal of the attack is to classify the input vector to the corresponding Sbox output. The output layer is consist of 256 perceptrons according to the range of SBox output values (0~255). After finding the Sbox output x, the secret key of the initial AddRoundKey can be extracted by computing k = (p ⊕Inverse Sbox(x)). The following Figure 5 shows two side-channel attacks on naive and first-order masked AES.

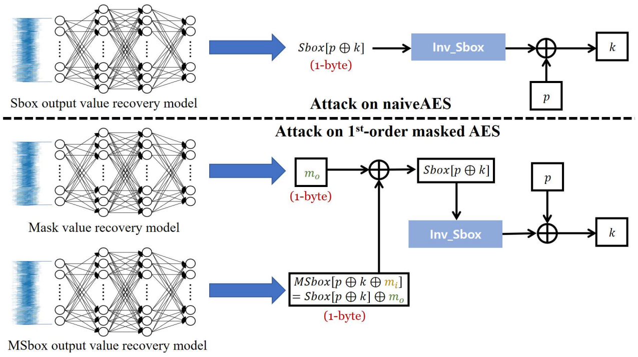

Sbox output value recovery model
$Sbox[p \oplus k]$
(1-byte)
Inv_Sbox
k
p
Attack on naiveAES
Attack on 1st-order masked AES
Mask value recovery model
m_o
(1-byte)
$Sbox[p \oplus k]$
p
Inv_Sbox
k
MSbox output value recovery model
$MSbox[p \oplus k \oplus m_i] = Sbox[p \oplus k] \oplus m_o$
(1-byte)

Figure 5. Attacks on naive and first-order masked AES

Our MLP model consists of two hidden layers (500 nodes per layer). We measure a total of 10 thousand power signals and split them into 3:1 for training and validation. The length of input nodes of MLP is equal to the sample points of a power trace. In this case, the power traces consist of 100 sample points containing the Sbox operation. And the length of the output layer is 256 (1 byte). The cost function and optimizer we used are cross-entropy and the Adam function.

In the CNN-based attack, we feed 100 points per power trace according to the Sbox outputs as the inputs of nodes. As those in the MLP model, the output is designed to have 256 nodes. We use two hidden layers that perform two convolutions using 16 kernels of sizes 8 and 4 respectively. However, we exclude the pooling layer to minimize the loss of features. And we adopt the ReLU activation function for each perceptrons. And we adopt dropout layers with a 0.25 dropout ratio.

4.3 Attack on Masked AES – Profiling Scenario

To apply a 2nd-order attack to 1st-order masked AES, we also implement it on an XMEGA128 board. The outputs of 16 S-boxes of AES are masked with a m_o. Our 2nd-order attack was performed against the following target points: the mask value and the output of the MSbox. The profiling process of mask values and MSbox output values is pre-completed in the same way as the bottom of Figure 5.

In the mask value profiling phase, the output mask m_o is labeled. In our experiments, the 24,400 samples of a power signal of the following instruction:

$$\text{Sbox[index]} \oplus m_o = \text{MSbox[index} \oplus m_i\text{]}. \quad \textbf{(3)}$$

The main difference between the 1st-order and 2nd-order attack models is the label used in the training phase. In the previous naive AES implementation, the label for extracting the secret key is Sbox output. On the other hand, since we first need to recover the mask, we use the masking value as a label of the DL model.

In the training phase on MSbox, we select the output value of MSbox as a label, that is, Sbox[x]$\oplus$ m_o =MSbox[x$\oplus$ m_i]. The power signals of SubBytes MSbox[x$\oplus$ m_i] are used as input values of DL models. Since m_o and MSbox[x$\oplus$ m_i] are extracted through our DL models, we can easily derive the output of Sbox by solving the following equation:

$$\text{Sbox(x)= (Sbox[x]}\oplus m_o\text{])} \ \oplus m_o = \text{MSbox[x}\oplus m_i\text{]} \ \oplus m_o \quad \textbf{(4)}$$

When Sbox[x] is known, the secret keys used in the initial AddRoundKey function could be extracted by computing k = (p $\oplus$ Sbox^{-1}(x)).

The accuracy of DL models according to the evolution of epochs is shown in Figure 6. In naïve AES implementation, the adversary succeeds the secret key recovery with more than 98% accuracy in the MLP and CNN models at 200 epochs. When applying the MLP and CNN models in first-order masked implementation, the attacker extracts the secret key with accuracies of 81.0% and 75.4%, respectively. Even though it is not shown in Figure 6, the recovery accuracy for mask value in each model is 100% at 10 epochs.

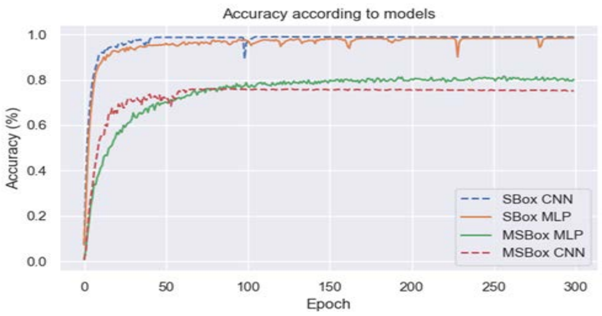


Figure 6. The accuracy of our attacks according to the evolution of epochs

We described the accuracy of side-channel attacks based on DL models on several AES implementations as shown in Table 1. First, we can observe that the secret key of naïve implementation can be extracted with an accuracy of over 98% by our MLP-based or CNN-based attack. Therefore, our practical result achieves a significant improvement compared to previous research.

Table 1. The comparison of DL-based attacks

DL-based Attack	Unprotected	Protected	
Model	(Sbox)	Mask	MSbox
MLP [9]	93.7%	-	-
MLP [16]	88.5%	-	-
CNN [17]	89.8%	-	-
SVM+CPA [18]	-	88.0%	-
ANN [10]	-	91.8%	88.4%
Our MLP	98.4%	100%	81.0%
Our CNN	98.9%	100%	75.4%

In a previous 2nd-order attack on a protected AES implementation proposed by Lerman et al. [15], the mask value was recovered with about 88% accuracy using a support vector machine, and the secret key could be extracted by performing the existing CPA attack. R. Gilmore *et al.* showed that their neural network-based attack on a masked AES can recover the mask value with 91.8% accuracy and masked Sbox output with 88.4%. As mentioned above, our DL model attack achieves 100% accuracy in the mask value recovery phase, and 75% and 85% accuracy in CNN, and MLP models, respectively. Therefore, our DL-based attacks show the overwhelming key recovery accuracy, whether it is an unmasked or masked implementation of AES.

4.4 Attack on Masked AES – Non-Profiling Scenario

As mentioned above, the non-profiling attacks statistically analyze traces measured from the only target board. The DDLA attack can surpass traditional non-profiling attack methods, such as differential power analysis and correlation power analysis. In particular, DDLA can perform high-order attacks and defeat masking countermeasures without signal combination. Note that DDLA can be performed without any leakage combination and knowledge about the specific secret sharing structure.

Since DDLA is carried out in a non-profiling scenario, we guess all keys in key space (1-byte in out attacks).

Whenever a hypothetical key is guessed, labels according to each power signal are created using known input (plaintext). Then, the training and validation steps are conducted. Here, the guessed key is equal to the correct key, and the PoI of the power signal is correlated to the label. Consequently, the model trained with the correct label shows surpasses other models in terms of metric values such as accuracy and loss. That is, the model trained with the correct label shows relatively high accuracy and low loss values rather than other models. Thus, the correct key can be extracted by examining the metric values.

It is important to select metrics and labeling to reveal the correct key during the DDLA attack. We applied MLP-DDLA with an accuracy metric over the epochs per guess on 10,000 power traces. That is, we adopt the MLP model as an underlying neural network and accuracy value as a metric. Here, we proposed a novel HW-based binary labeling method instead of the known methods such as HW, MSB, and LSB labeling. The HW-based binary labeling method assigns label '1' if the Hamming weight of the intermediate value *v*, which is generated by guessed key and input message, is greater than a certain value, and ‘0’ otherwise.

$$L = \begin{cases} 0 \ \ if \ HW(v) < 4 \\ 1 \ \ if \ \ HW(v) > 4 \end{cases} \quad \textbf{(5)}$$

Here, we use the open dataset ASCAD which adopted first-order masking. Then, we perform a DDLA attack phase using this dataset. The following Figure 7 shows that the correct key ‘0xE0’ is distinguished from the group of 255 wrong keys. Therefore, our DDLA attack can extract perfectly the secret key of masked AES implementation when the epoch value is 10 or more.

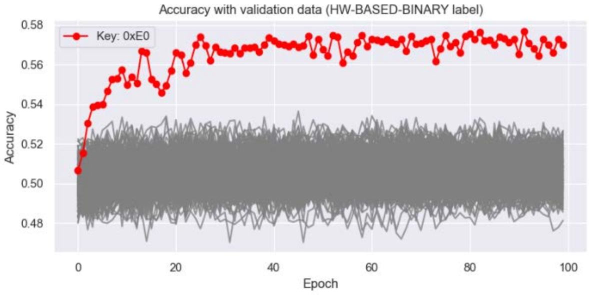


Figure 7. The result of a DDLA on first-order masked AES

5 Conclusions

In this paper, we evaluate the security level of unprotected and protected AES implementation using two DL-based side-channel attacks. Furthermore, the MLP and CNN models in the profiling attack on AES implementation are developed to break the first-order masking countermeasure. Our profiling attack on protected AES is composed of two phases, the first phase targets the mask value, and the second is the masked intermediate value.

To recover the secret key using a non-profiling attack, we adopt the MLP-DDLA. Here, we present a new HW-based binary labeling method to improve the performance of DDLA attacks. Our experimental results show the overwhelming key recovery accuracy when targeting the masked implementation of AES. Consequently, we found that an adversary could extract the secret key of first-order protected AES implementation with 81.0% accuracy using the ML-based profiling attack. And the non-profiling DDLA attack can extract perfectly the secret key of masked AES when the epoch value is 10 or more. Future work can include optimizing the DL models to minimize the number of traces required for SCAs and designing high-order countermeasures.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2020R1F1A1074358).

References

[1] J. Daemen, V. Rijmen, *The design of Rijndael*, Springer-verlag, 2002.

[2] P. C. Kocher, J. Jaffe, B. Jun, Differential power analysis, *The 19th Annual international cryptology conference*, Santa Barbara, CA, USA, 1999, pp. 388-397.

[3] E. Brier, C. Clavier, F. Olivier, Correlation Power Analysis with a Leakage Model, *International workshop on cryptographic hardware and embedded systems*, Cambridge, MA, USA, 2004, pp. 16-29.

[4] T. Messerges, Securing the AES finalists against power analysis attacks, *International Workshop on Fast Software Encryption*, New York, NY, USA, 2000, pp. 150-164.

[5] C. Herbst, E. Oswald, S. Mangard, An AES smart card implementation resistant to power analysis Attacks, *International Conference on Applied Cryptography and Network Security*, Singapore, 2006, pp. 239-252.

[6] E. Oswald, K. Schramm, An efficient masking scheme for AES software implementations, *International Workshop on Information Security Applications*, Jeju, South Korea, 2005, pp. 292-305.

[7] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, Vol. 65, No. 6, pp. 386-408, 1958.

[8] S. Albawi, T. A. Mohammed, S. Al-Zawi, Understanding of a Convolutional Neural Network, *International Conference on Engineering and Technology*, Antalya, Turkey, 2017, pp. 1-6.

[9] Z. Martinasek, V. Zeman, Innovative method of the power analysis, *Radioengineering*, Vol. 22, No. 2, pp. 589-594. June, 2013.

[10] R. Gilmore, N. Hanley, M. O’Neill, Neural network based attack on a masked implementation of AES, *IEEE International Symposium on Hardware Oriented Security and Trust*, Washington, DC, USA, 2015, pp. 106-111.

[11] B. Timon, Non-profiled deep learning-based side-channel attacks with sensitivity analysis, *IACR Transactions on Cryptographic Hardware and Embedded Systems*, Vol. 2019, No. 2, pp. 107-131, February, 2019.

[12] R. Benadjila, E. Prouff, R. Strullu, E. Cagli, C. Dumas, Study of deep learning techniques for side-channel analysis and introduction to ASCAD database, *Cryptology ePrint Archive*, Report 2018/053, 2018. https ://eprint.iacr.org/2018/053.

[13] C. O’Flynn, Z. D. Chen, Chipwhisperer: An open-source platform for hardware embedded security research, *International Workshop on Constructive Side-Channel Analysis and Secure Design*, Paris, France, 2014, pp. 243-260.

[14] S. Chari, J. R. Rao, P. Rohatgi, Template Attacks, *International Workshop on Cryptographic Hardware and Embedded Systems*, Redwood Shores, CA, USA, 2002, pp. 13-28.

[15] W. Schindler, K. Lemke, C. Paar, A Stochastic Model for Differential Side Channel Cryptanalysis, *International Workshop on Cryptographic Hardware and Embedded Systems*, Edinburgh, UK, 2005, pp. 30-46.

[16] H. Wang, M. Brisfors, S. Forsmark, E. Dubrova, How Diversity Affects Deep-Learning Side-Channel Attacks, *IEEE Nordic Circuits and Systems Conference*, Helsinki, Finland, 2019, pp. 1-7.

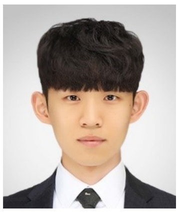
[17] L. Wei, B. Luo, Y. Li, Y. Liu, Q. Xu, I Know What You See: Power Side-Channel Attack on Convolutional Neural Network Accelerators, *34th Annual Computer Security Applications Conference*, San Juan, PR, USA, 2018, pp. 393-406.

[18] L. Lerman, S. F. Medeiros, G. Bontempi, O. Markowitch, A machine learning approach against a masked AES, *12th International Conference, on Smart Card Research and Advanced Applications,* Berlin, Germany, 2013, pp. 61-75.

[19] B. Gierlichs, L. Batina, P. Tuyls, B. Preneel, Mutual information Analysis, *International Workshop on Cryptographic Hardware and Embedded Systems*, Washington, DC, USA, 2008, pp. 426-442.

[20] E. Cagli, C. Dumas, E. Prouff, Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures, *International Workshop on Cryptographic Hardware and Embedded Systems*, Taipei, Taiwan, 2017, pp. 45-68.

Biographies



Daehyeon Bae received the BS and MS degrees in information security engineering from Hoseo University, Rep. of Korea, in 2021, and 2022, respectively. Currently, he is taking a doctor's course in information security at Korea University. His research interests include side-channel analysis, cryptography, hardware security, and machine learning.

Jongbae Hwang received the BS and MS degrees in information security engineering from Hoseo University, Rep. of Korea, in 2021, and 2022, respectively. Currently, he is an NNSP researcher. He is developing anomaly detection products using AI in OT systems.

Jaecheol Ha received the BE, ME, and Ph.D. in electronics engineering from Kyungpook National University, Rep. of Korea, in 1989, 1993, and 1998, respectively. He is currently a full professor in the division of computer engineering at Hoseo University, Asan, Rep. of Korea. From 1998 to 2006, he also worked as a professor in the department of information and communication at Korea Nazarene University, Cheonan, Korea. In 2014, he was a visiting researcher at Purdue University, USA. He is working as a vice-president of the Korea Institute of Information and Cryptography (KIISC). His research interests include post-quantum cryptography, network security, hardware security, and side-channel attacks.