Deep Learning Side-Channel Attacks against Hardware-Implemented Lightweight Cipher Midori 64

Madoka Sakou[1,*,†], Kunihiro Kuroda[1,†], Yuta Fukuda[1,†], Kota Yoshida[2,‡] and Takeshi Fujino[2,‡]

[1]*Graduate School of Science and Engineering, Ritsumeikan University 1–1–1 Nojihigashi, Kusatsu, Shiga, 525–8577 Japan*

[2]*Department of Science and Engineering, Ritsumeikan University 1–1–1 Nojihigashi, Kusatsu, Shiga, 525–8577 Japan*

Abstract

Lightweight ciphers are planned for implementation in IoT devices because of their small circuit size, low latency, and low power consumption. It is very important to verify tamper resistance on these devices, because an adversary can physically manipulate them. Side-channel attacks are a critical factor in tamper resistance evaluation. In particular, deep learning side-channel attacks (DL-SCAs), in which deep neural networks (DNNs) are used for analyzing side-channel information, demonstrate better performance than conventional side-channel attacks [e.g., correlation power analysis (CPA), template attack] in previous reports. In this work, we evaluated a profiling DL-SCA against Midori 64, which is a lightweight cipher. A conventional CPA was carried out on the last and 2nd rounds of the encryption process. In comparison, the last and 1st rounds were targeted in our attack. First, in the first attacks against the last round, a class missing problem occurred in the DNN training process. We show that it is necessary to connect waveforms from 16 kinds of encryption keys to mitigate the problem and train the DNN model. In the second, attacks against the 1st round, we claimed that the number of classes that the DNN classified (the complexity of the classification problem) was significantly reduced compared with the attack against the 2nd round. Our experimental results show that the DL-SCA successfully revealed all of the partial keys with 10,600 attack waveforms.

Keywords

Lightweight Cipher, Midori, DL-SCA, Hardware Security

1. Introduction

Recently, the number of IoT devices has increased. Data collected by sensors in IoT devices is sent to the cloud. Then, the cloud analyzes the data by using deep learning. Cryptography is an essential technology that ensures such data's authenticity and protects privacy. Furthermore, when deep learning applications are executed on IoT devices, it is necessary to implement cryptographic techniques to protect intellectual property, such as deep learning parameters. In this case, devices need to be equipped with lightweight ciphers with lower latency and lower

The 4th International Symposium on Advanced Technologies and Applications in the Internet of Things (ATAIT 2022), August 24–26, 2022, Ibaraki, Japan

ri0091hh@ed.ritsumei.ac.jp (M. SAKOU); ri0090ek@ed.ritsumei.ac.jp (K. KURODA); ri0073pi@ed.ritsumei.ac.jp (Y. FUKUDA); y0sh1d4@fc.ritsumei.ac.jp (K. YOSHIDA); fujino@se.ritsumei.ac.jp (T. FUJINO)

power consumption than AES, which is the standard symmetric key cipher.

It is also possible for attackers to physically manipulate IoT devices; then, physical attacks such as side-channel attacks are considered to be a threat. Side-channel attacks reveal secret information such as secret keys by analyzing power consumption and electromagnetic leakage during the operation of cryptographic circuits [1, 2, 3]. In recent years, deep learning side-channel attacks (DL-SCAs) have been actively studied as a more advanced attack method than conventional attacks, such as correlation power analysis (CPA). It has been reported that DL-SCAs are able to attack conventional countermeasure circuits through side-channel attacks [4].

DL-SCA requires that the class imbalance problem in training data be mitigated. Kubota et al. performed DL-SCA on hardware-implemented AES and reported that not all key bytes are revealed due to the class imbalance problem [5]. They avoided the class imbalance problem by mixing training labels with other bytes. Fukuda et al. also avoided the class imbalance problem by using random keys in the training of DNNs [6].

DL-SCA should also be studied for lightweight ciphers because of the differences in block length and S-box structure compared with AES. In this paper, Midori 64 [7], which is a lightweight cipher, was used as a target algorithm, and DL-SCA was evaluated against it. To the best of our knowledge, no studies on DL-SCA against Midori 64 have been reported. We evaluated DL-SCA against Midori 64 implemented with an FPGA and compared the attack performance between CPA [8] and template attacks, which were reported in the previous research.

2. Preliminary

2.1. Midori 64

The lightweight cipher Midori was presented at ASIACRYPT in 2015. It is a cryptographic algorithm oriented toward low power consumption in hardware implementations [7]. Midori has block lengths of 64 bits and 128 bits, which are called Midori 64 and Midori 128, respectively. The key length is 128 bits in both cases. This paper focuses on Midori 64, which has a different block length from AES. Figure 1 shows the encryption process of Midori 64, which has a loop architecture. The cryptographic process of Midori 64 consists of 17 rounds (0-16R), and each round is processed by using the round function. The round function includes SubCell, ShuffleCell, MixColumn, and AddKey. SubCell performs nonlinear substitution using the 4 bit S-Box shown in Table 1. The S-Box used for encryption and decryption is the same transformation. ShuffleCell performs the transposition of values in 4-bit units using the shuffle table shown in Table 2. MixColumn performs a matrix operation on 16-bit units with the constant matrix M [Equation (1)]. AddKey performs a XOR operation on the intermediate values of the encryption process and a round key, which is calculated from K_0, K_1, and round constant α. K_0 and K_1 are generated from a cryptographic key K using Equation (2). The operator || denotes the concatenation of the bit strings, and K_0 and K_1 are the first and second halves of a 128-bit key K divided into 64-bit lengths. In the 0th round operation, the XOR operation is performed for the plaintext and the key WK [Equation (3)] is performed. In the 16th round operation, the XOR operation for the key WK is performed after the SubCell operation.

Table 1
S-Box table for Midori 64

x	0	1	2	3	4	5	6	7
$S(x)$	C	A	D	3	E	B	F	7
x	8	9	A	B	C	D	E	F
$S(x)$	8	9	1	5	0	2	4	6

Table 2
Shuffle table for Midori 64

x	0	1	2	3	4	5	6	7
y	0	7	D	9	5	2	B	C
x	8	9	A	B	C	D	E	F
y	F	8	1	6	A	D	4	3

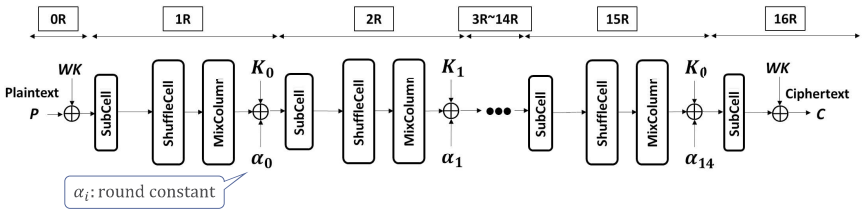

Figure 1: Midori 64 encryption architecture [9]

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \tag{1}$$

$$K = K_0 || K_1 \tag{2}$$

$$WK = K_0 \oplus K_1 \tag{3}$$

2.2. Profiling DL-SCA

DL-SCAs are new attack methods that combine side-channel attacks with deep learning techniques, and they have been actively studied recently. In DL-SCAs, attackers train a deep neural network (DNN) to classify the intermediate values of cryptographic operations on the basis of side-channel information such as power consumption waveforms. The features that are necessary for classifying these values are effectively extracted from power waveforms during the DNN training process. It is known that this allows attackers to attack without sufficient knowledge on the implementation of the target device. There are two types of DL-SCA scenarios: profiling attacks and non-profiling attacks. Profiling attacks are considered common in DL-SCA. In profiling attacks, it is assumed that the attacker possesses a freely controllable device (profiling device) of the same type as the target device. On the other hand, in non-profiling attacks, the attacker uses only the target device. In this paper, we focus on profiling attacks as initial study of DL-SCA against Midori.

The profiling DL-SCA consists of a profiling and an attack phase. In the profiling phase, the attacker sets a cryptographic key on the profiling device and acquires the power consumption waveforms during the cryptographic operation. The intermediate values during the cryptographic operation are calculated from known keys, and they are set as labels corresponding to

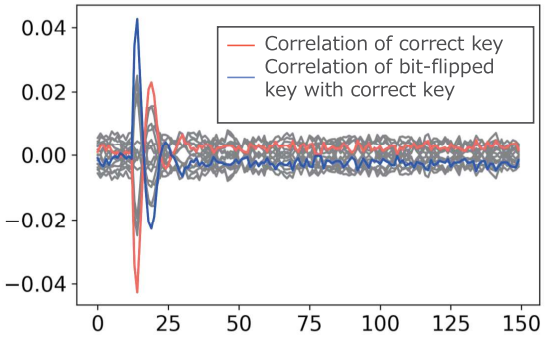

Figure 2: Results of correlation power analysis focusing on first round

the acquired power waveforms. Then, the DNN is trained to classify power waveforms into each label.

In the attack phase, the power waveform acquired from the target device is input to the trained DNN, and the conditional probability for each label is output. From these conditional probabilities, the probability of the target device's cryptographic key is calculated for each key candidate. These calculations are repeated multiple times for each encryption, and the calculated probabilities are aggregated using maximum likelihood estimation to reveal the cryptographic key.

3. CPA against Midori 64

Nozaki et al. reported that Midori 64 is vulnerable to correlation power analysis (CPA) [8]. They proposed an attack method that reveals the key through a two-stage analysis. The block size of Midori 64 is 64 bits, and the key size is 128 bits, so a single 64 bit key out of K_0, K_1, or WK is used in a single round as shown in Fig.1. Therefore, two kinds of 64 bit keys have to be revealed by attacking two rounds for the successful attack of the 128 bit key. Therefore, the first stage of analysis reveals only a part of the encryption key K, and the second stage of analysis reveals all of it. They succeeded in revealing WK by focusing on the final round (16R) as first-stage CPA and revealing K_0 by focusing on the second round as second-stage CPA. The attacker can calculate K_1 from the revealed WK and K_0 from Equations (2) and (3), and they can reveal the secret 128 bit key K.

However, due to the structure of Midori 64, the second stage of CPA includes a MixColumn process. Therefore, a total of $2^{44} \times 4$ calculations of correlation values are required to derive K_0, which is an extremely large number compared with the $2^8 \times 16$ calculations performed in CPA against AES-128 cryptographic circuits. Nozaki et al. roposed a method to reduce the calculation of these correlation values, and it was shown that K_0 can be revealed by calculating $(2^{12} + 2^4) \times 4$ correlation values.

In this paper, we focus on the first round in the second stage of attacks. In this case of an attack using CPA, the number of correlation calculations can be reduced to $2^4 \times 16 = 256$, but two types of correlation coefficients with the largest absolute value appear as shown in Fig. 2.

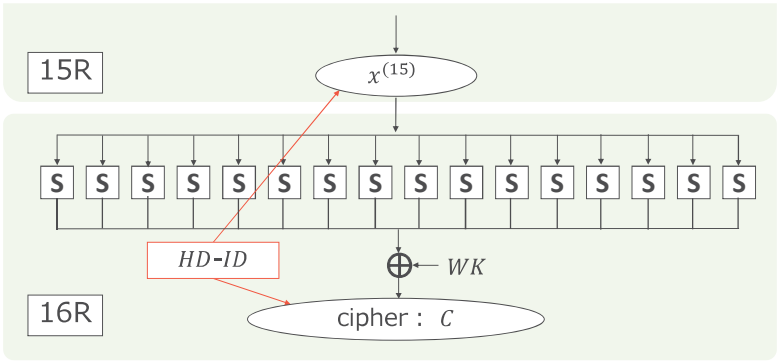

Figure 3: 1st stage attack focusing on 16th round

Table 3
Relationship of input x and intermediate value SubCell(x)

x	0	1	2	3	4	5	6	7
$S(x)$	C	A	D	3	E	B	F	7
$x \oplus S(x)$	C	B	F	0	A	E	9	0
x	8	9	A	B	C	D	E	F
$S(x)$	8	9	1	5	0	2	4	6
$x \oplus S(x)$	0	0	B	E	C	F	A	9

The reason is that the intermediate value for AddKey of the first round is the same, regardless of the candidate key. Therefore, a key and its bit-flipped counterpart have the same plus/minus correlation values.

4. DL-SCA against Midori 64

In the profiling DL-SCA against Midori 64, the secret key K is revealed by using a two-stage analysis the same as the conventional CPA. We selected 16th round for first stage analysis as same as previous work [8]. We also selected 1st round for second stage analysis to reduce calculation of intermediate value during the cryptograophic operation. The following sections explain the attack methods in the two selected rounds.

4.1. 1st stage of DL-SCA focused on 16th round

In the first stage of the DL-SCA, the attack focuses on the final (16th) round and estimates the key WK, as shown in Fig. 3. The intermediate value $x_t^{(15)}$ at the end of the 15th round is calculated from Equation (4).

$$x_t^{(15)} = S(C_t \oplus WK_t), \tag{4}$$

where C_t is a known ciphertext, $S()$ denotes a transformation by S-box, and t is the nibble (4-bit) of interest. The HD-ID calculated from $x_t^{(15)}$ and C_t according to Equation (5) is used as the

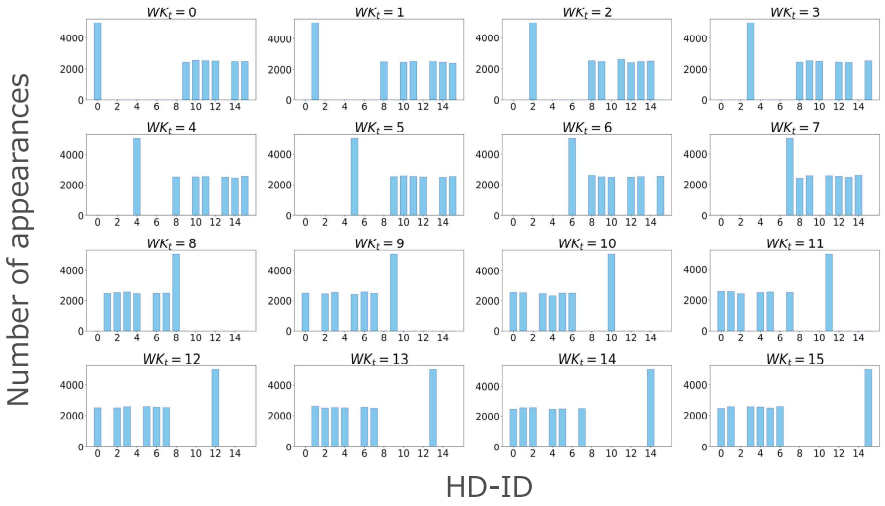

Figure 4: Frequency of training labels using each key WK_t

label when training the DNN.

$$\begin{aligned} \text{HD-ID} &= C_t \oplus x_t^{(15)} \\ &= C_t \oplus S(C_t \oplus WK_t) \end{aligned} \tag{5}$$

When a DNN is trained using HD-IDs [Equation (5)] with a single WK_t, a class missing problem occurs. Figure 4 shows a histogram for each of the HD-IDs when 20,000 encryption operations using random plaintexts and a fixed key set to 16 different values of WK_t. There were only seven kinds of HD-IDs observed for each WK_t. In addition, the number of HD-IDs with the same value as WK_t was about twice as much as the other HD-IDs. For example, in the case of $WK_t = 0$, only seven HD-IDs = $\{0, 9, 10, 11, 12, 14, 15\}$ appeared, and the frequency of HD-ID=0 was about twice as much as the others (HD-ID= $\{9, 10, 11, 12, 14, 15\}$). In addition, there was no frequency of training labels for nine kinds of classes, that is, HD-ID = $\{1, 2, 3, 4, 5, 6, 7, 8, 13\}$.

The reasons for such missing and imbalanced HD-IDs are as follows. As shown in Equations (6), the HD-ID values are the XORed values of x_t^{15}, $S(x_t^{15})$, and WK_t.

$$\begin{aligned} C_t &= S(x_t^{15}) \oplus WK_t \\ \text{HD-ID} &= x_t^{15} \oplus C_t = x_t^{15} \oplus S(x_t^{15}) \oplus WK_t \end{aligned} \tag{6}$$

Table 3 shows the values of $S(x)$ and $x \oplus S(x)$ when x was varied with 16 different values. There were only seven patterns of $x \oplus S(x)$ values ($\{0, 9, A, B, C, E, F\}$), and the count of $x \oplus S(x) = 0$ was twice that of the other patterns ($x \oplus S(x) = \{9, A, B, C, E, F\}$). Therefore, there were only seven HD-ID patterns for each value of WK_t, and the pattern HD-ID= WK_t appeared twice as much as the other patterns.

In DL-SCAs, it has been reported that DNN training is not effectively performed due to the class imbalance problem and class missing problem [5, 6]. In our approach for successful DNN training in the profiling phase, the waveforms from 16 different secret keys are equally mixed

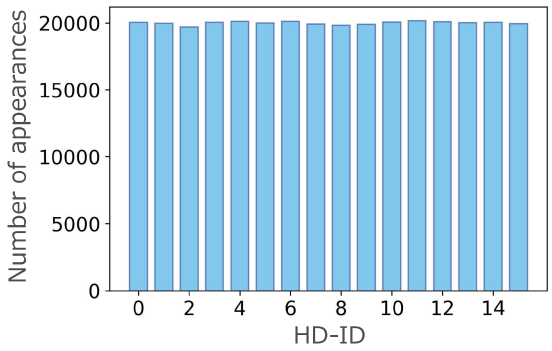

Figure 5: Frequency of training labels using 16 types of encryption keys

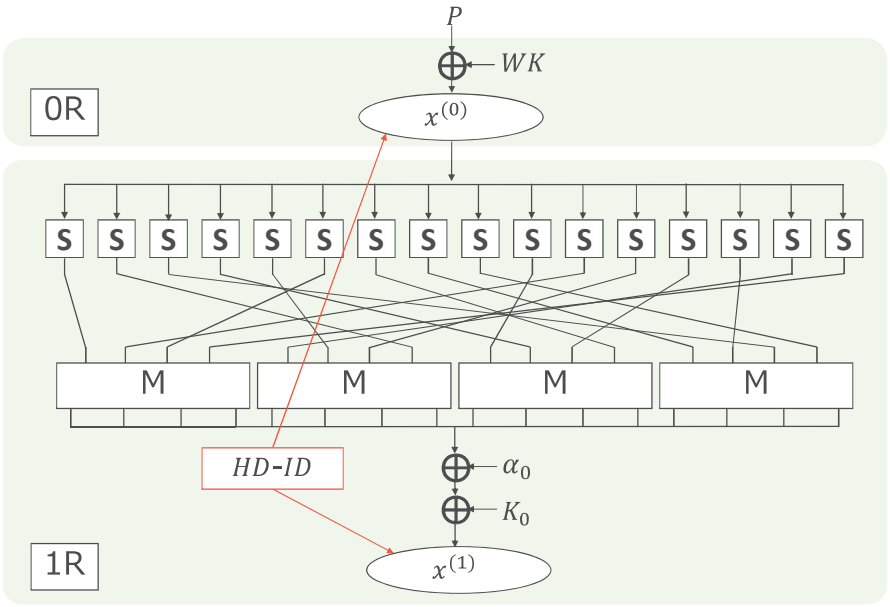

Figure 6: 2nd stage attack focusing on 1st round

in order to mitigate the class imbalance and the class missing problem. The frequency of the HD-IDs for the waveforms from the 16 keys has a uniform distribution, as shown in Fig. 5.

4.2. 2nd stage of DL-SCA focused on 1st round

In the second stage of DL-SCAs against Midori 64, the attacker focuses on the first round of the encryption process, and the partial key K_0 is revealed as shown in Fig. 6. The intermediate values at the end of the 0th and 1st rounds of Midori 64 are calculated by Equations (7) and (8), respectively.

$$x^{(0)} = P \oplus WK \tag{7}$$

$$x^{(1)} = M(Sh(SC(x^{(0)}))) \oplus \alpha_0 \oplus K_0, \tag{8}$$

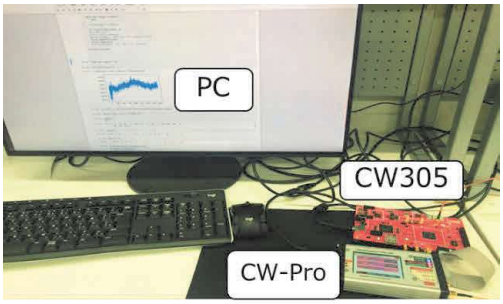

Figure 7: Experimental environment

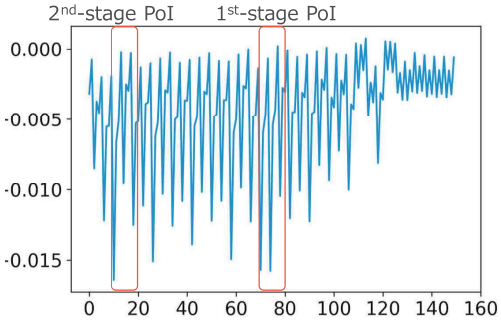

Figure 8: Acquired power waveforms and selected PoI

Table 4
DNN model architecture

Input layer	Convolution layer × 2	Fully connected layer × 2	Output layer
10	Size × filters = 3 × 10 Stride = 3, ReLU	32 neurons Tanh	16 neurons Softmax

where P is plaintext, the $M()$ function is MixColumn, the $Sh()$ function is ShuffleCell, the $SC()$ function is a SubCell, α_0 is the round constant in the 0th round, and t is the nibble (4-bit) of interest. The HD-IDs used as training labels are calculated by Equation (9).

$$\text{HD-ID} \quad = \quad x_t^{(0)} \oplus x_t^{(1)}, \tag{9}$$

where $x_t^{(0)}$ is the intermediate value at the end of the 0th round, and $x_t^{(1)}$ is the intermediate value at the end of the 1st round. The attacker trains DNN models by using the acquired power waveforms and calculated labels.

5. Evaluation of DL-SCA against Midori 64

5.1. Evaluation setup

Figure 7 shows the environment for acquiring waveforms. We used a CW305 (clock frequency: 10 MHz) developed by NewAE Technology as a target board. This board is equipped with Midori 64, and the VHDL source is published in [9]. We also used a CW-Pro (sampling rate: 40 MHz) to acquire power consumption waveforms from the CW305 during cryptographic operations. Figure 8 shows an example of the captured waveforms and the points of interest (PoI) used in the two stages of the attack. The structure of the DNN used in this evaluation is shown in Table 4. We adopted the DNN model architecture that was used in these previous works.[5, 6]. We also adjusted hyperparameters so that the DNN model would be optimized for our research target. Other hyperparameters were epochs = 30, batch_size = 1,024, and learning_rate = 0.0005, and the number of training datasets were 320,000.

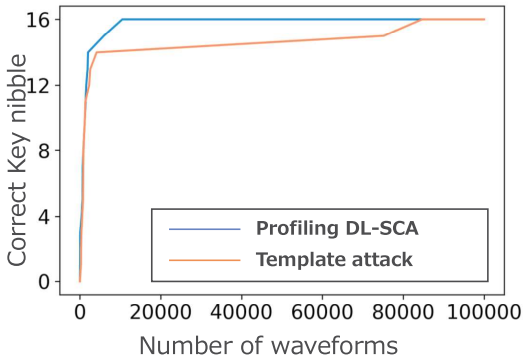

Figure 9: Required number of waveforms for each attack method (1st stage)

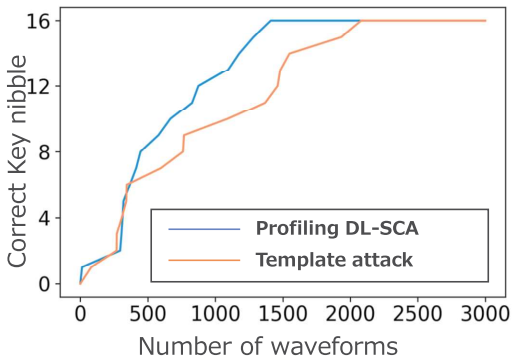

Figure 10: Required number of waveforms for each attack method (2nd stage)

5.2. Result for 1st stage of DL-SCA using 16 different keys

As mentioned in section 4.1, we reveal the key WK by focusing on the final (16th) round using waveforms acquired with 16 different secret keys. Figure 9 shows the results of the profiling DL-SCA and the template attack, which is a conventional profiling attack method. The profiling DL-SCA successfully revealed all of the nibbles (16 nibbles) of the key WK in about 10,600 waveforms. In comparison, the template attack required about 85,000 waveforms to reveal the key WK.

5.3. Result for 2nd stage of DL-SCA using 16 different keys

In the second stage of the profiling DL-SCA, the acquired waveforms were used to attack the first round and reveal the key K_0. Figure 10 shows the results of the profiling DL-SCA and the template attack. The profiling DL-SCAs successfully revealed all of the nibbles (16 nibbles) of the key K_0 in about 1,400 waveforms. In comparison, the template attack required about 2,000 waveforms to reveal the key K_0.

6. Conclusions

In this paper, we evaluated a two-stage profiling DL-SCA against Midori 64. In the first stage of the profiling DL-SCA, the frequency of the training labels was biased and missing when training waveforms were acquired with a single secret key. We confirmed that it is necessary to use 16 different secret keys in the acquisition of training waveforms. In the second stage of the profiling DL-SCA, we showed that it is possible to perform an attack by focusing on the first round, which is difficult to attack with the conventional CPA. We targeted Midori 64 as a basic study for profiling DL-SCAs against lightweight ciphers, but Midori 64 is not recommended in practice due to the existence of weak keys [10]. Therefore, we will continue to work with DL-SCAs against Midori 128 instead of Midori 64 or other lightweight ciphers. We plan to study SCA countermeasures for lightweight ciphers in the future.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP22H03593. We had discussions on Midori's side-channel attack with Prof. Masaya Yoshikawa and Assistant Prof. Yusuke Nozaki at Meijo University. We would like to express our gratitude to all of the people involved.

References

[1] P. Kocher, J. Jaffe, B. Jun, Differential power analysis, in: M. Wiener (Ed.), Advances in Cryptology — CRYPTO' 99, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, pp. 388–397.

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

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

[4] H. Maghrebi, T. Portigliatti, E. Prouff, Breaking cryptographic implementations using deep learning techniques, in: C. Carlet, M. A. Hasan, V. Saraswat (Eds.), Security, Privacy, and Applied Cryptography Engineering, Springer International Publishing, Cham, 2016, pp. 3–26.

[5] T. Kubota, K. Yoshida, M. Shiozaki, T. Fujino, Deep learning side-channel attack against hardware implementations of aes, in: 2019 22nd Euromicro Conference on Digital System Design (DSD), IEEE, 2019, pp. 261–268.

[6] Y. Fukuda, K. Yoshida, H. Hashimoto, T. Fujino, Deep learning side-channel attacks against lightweight sca countermeasure rsm-aes, in: 2021 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), IEEE, 2021, pp. 1–6.

[7] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, F. Regazzoni, Midori: A block cipher for low energy, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2015, pp. 411–436.

[8] Y. Nozaki, M. Yoshikawa, Hierarchical power analysis and evaluation for low power consumption lightweight cipher midori, The Journal of The Institute of Electrical Engineers of Japan C (Electronic / Information / Systems Division Magazine) 138 (2018) 1455–1463.

[9] C. A. Lara-Nino, A. Diaz-Perez, M. Morales-Sandoval, Fpga-based assessment of midori and gift lightweight block ciphers, in: International Conference on Information and Communications Security, Springer, 2018, pp. 745–755.

[10] J. Guo, J. Jean, I. Nikolić, K. Qiao, Y. Sasaki, S. M. Sim, Invariant subspace attack against full midori64, IACR Cryptology ePrint Archive, 2015:1189, 2015.