PAPER *Special Section on Cryptography and Information Security*

Model Reverse-Engineering Attack against Systolic-Array-Based DNN Accelerator Using Correlation Power Analysis

Kota YOSHIDA[†a)], *Student Member*, Mitsuru SHIOZAKI[††], Shunsuke OKURA[†††], Takaya KUBOTA[††], *and* Takeshi FUJINO[†††], *Members*

SUMMARY A model extraction attack is a security issue in deep neural networks (DNNs). Information on a trained DNN model is an attractive target for an adversary not only in terms of intellectual property but also of security. Thus, an adversary tries to reveal the sensitive information contained in the trained DNN model from machine-learning services. Previous studies on model extraction attacks assumed that the victim provides a machine-learning cloud service and the adversary accesses the service through formal queries. However, when a DNN model is implemented on an edge device, adversaries can physically access the device and try to reveal the sensitive information contained in the implemented DNN model. We call these physical model extraction attacks model reverse-engineering (MRE) attacks to distinguish them from attacks on cloud services. Power side-channel analyses are often used in MRE attacks to reveal the internal operation from power consumption or electromagnetic leakage. Previous studies, including ours, evaluated MRE attacks against several types of DNN processors with power side-channel analyses. In this paper, information leakage from a systolic array which is used for the matrix multiplication unit in the DNN processors is evaluated. We utilized correlation power analysis (CPA) for the MRE attack and reveal weight parameters of a DNN model from the systolic array. Two types of the systolic array were implemented on field-programmable gate array (FPGA) to demonstrate that CPA reveals weight parameters from those systolic arrays. In addition, we applied an extended analysis approach called "chain CPA" for robust CPA analysis against the systolic arrays. Our experimental results indicate that an adversary can reveal trained model parameters from a DNN accelerator even if the DNN model parameters in the off-chip bus are protected with data encryption. Countermeasures against side-channel leaks will be important for implementing a DNN accelerator on a FPGA or application-specific integrated circuit (ASIC).

key words: *model extraction attack, deep neural networks, correlation power analysis, systolic array*

1. Introduction

A deep neural network (DNN) has been applied to various machine-learning services. A DNN training requires a large dataset, computation resources, and expertise. Also, information on the trained DNN model provides a method of revealing sensitive training data [1], [2] and deceiving the inference [3], [4]. Therefore, information on a trained DNN model is an attractive target for an adversary not only in terms of intellectual property but also of security.

Model extraction attacks are security issues in DNNs. In such attacks, an adversary tries to train a local model that achieves equivalent accuracy to the target model or to reveal information of the target model, such as the model architecture and hyperparameters. Previous studies on model extraction attacks assumed that the victim provides a machine learning cloud service, and the adversary accesses the service through formal queries [5], [6]. On the other hand, some DNN execution environments are transitioning to edge devices due to privacy protection demand and real-time processing.

When a DNN is executed on edge devices, the adversary can physically access the device and try an invasive or non-invasive physical attack to reveal the DNN model information. We call these physical model extraction attacks model reverse-engineering (MRE) attacks to distinguish them from attacks on cloud services.

Memory bus tapping is one of the most straightforward of MRE attacks, but the DNN model can be protected by model parameter encryption. Hua et al. proposed an MRE attack for revealing the DNN structure by exploiting the memory and timing side-channels, even with data encryption [7]. Wang et al. proposed an architecture of a secure DNN accelerator that contains model parameter and processor instruction encryption [8]. It provides secure off-chip memory access to the DNN accelerator chip and is a countermeasure against attacks based on the memory access pattern.

In the previous studies, including ours, MRE attacks against several types of DNN processors have been evaluated with power side-channel analyses. Batina et al. highlighted the potential vulnerabilities of software embedded neural networks [9]. They measured information leakage with power side-channel analysis against an ARM 8-bit MCU. They revealed a target neural network model structure by simple electromagnetic (EM) analysis and the model parameters by correlation EM analysis (CEMA). We measured information leakage from an 8-bit DNN processing element (PE) which is implemented on field-programmable gate array (FPGA) [10]. The PE consists of one multiplier and adder, and some registers, and calculates matrix multiplication serially. We attacked the PE with CEMA and indicated information leakage from a register that stores an intermediate sum of matrix multiplication. In the practical DNN accelerators, many PEs are usually implemented

Manuscript received March 16, 2020.
Manuscript revised August 7, 2020.

[†]The author is with the Graduate School of Science and Technology, Ritsumeikan University, Kusatsu-shi, 525-8577 Japan.
[††]The authors are with Research Organization of Science and Engineering, Ritsumeikan University, Kusatsu-shi, 525-8577 Japan.
[†††]The authors are with Department of Science and Engineering, Ritsumeikan University, Kusatsu-shi, 525-8577 Japan.
a) E-mail: ri0044ep@ed.ritsumei.ac.jp

DOI: 10.1587/transfun.2020CIP0024

and run in parallel. Power analysis against target PE among multiple PEs get difficult because the noise from other PEs decrease the signal to noise ratio of information leakage from the target PE. We have already started to investigate power analysis against practical DNN accelerator architecture called systolic array [11]. The systolic array is one of the matrix multiplication circuits architectures and is utilized in DNN accelerators. There are multiple types of systolic arrays such as a wavefront array and a tensor processing unit (TPU). We implemented a wavefront array and attack with correlation power analysis (CPA) for revealing the weight parameters of a DNN model. In other studies about side-channel analysis against practical DNN accelerator, Dubey et al. applied power-based side-channel analysis to binarized neural networks implemented on FPGA and proposed the first side-channel countermeasure [12].

In this paper, we implemented two types of systolic arrays (wavefront array and TPU) as CPA target devices. We found that the hamming distance CPA attack upon multiply-accumulate operation strongly depends on the previous value of the register as our CPA simulation results. In both architectures of the systolic arrays, weight parameters are repeatedly used in matrix calculation, so multiple CPA attack results can be obtained. We applied an extended method called "chain CPA" which relies on the CPA results when the non-zero previous value is stored on the register. It is noted that the "chain CPA" means a post-processing method of 1st order CPA. In the CPA attack experiments on both architectures, more weight parameters were revealed in chain CPA than the conventional CPA.

The main contributions of this work are as follows.

- We evaluated an MRE attack against two types of systolic array architecture [13], [15] which was implemented on FPGA. To the best of our knowledge, our work is the first successful attack that DNN model parameters are revealed on the systolic array.
- We simulated CPA against a multiply-accumulate operation executed in the PEs. Our simulation results suggested that hamming-distance CPA strongly depends on the value of registers. In the case of systolic array operation, the same weight parameter can be repeatedly attacked on different values. We apply the post-processing technique called "chain CPA" for selecting the correct parameter from multiple CPA results.
- An MRE attack using CPA is experimentally performed against two types of systolic arrays. In the case of wavefront array, chain CPA succeeded to attack to all PEs of the wavefront array and revealed all weight parameters. In the case of TPU, chain CPA succeeded to attack seven of nine multiply-accumulations. It means that chain CPA narrowed the candidates for three of nine weight parameters down to two patterns and uniquely revealed the other six of nine weight parameters.

2. Structure of DNN Accelerator

Matrix multiplication is frequently used in DNN algorithms. Thus, systolic arrays are designed to execute matrix multiplication with high performance and at low power.

We evaluated two types of systolic arrays that calculate the three-by-three matrix dot product, which is shown in Eq. (1). Where, for example, the calculation of c_{11} is done using Eq. (2). When systolic arrays are used as DNN accelerators, matrix a is either input or activation, matrix b is the weight parameters of the DNN model, which is the adversary's target, and matrix c is the intermediate value of the inference process. We assume a typical DNN inference accelerator for artificial intelligence (AI) edge devices, and a and b are represented by an 8-bit integer. The calculation result c is represented by an 18-bit integer to prevent bit overflow.

$$\begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \tag{1}$$

$$c_{11} = a_{11} \times b_{11} + a_{12} \times b_{21} + a_{13} \times b_{31} \tag{2}$$

2.1 Attack Target (1): Wavefront Array

The wavefront array proposed by Kung [15] is a systolic array architecture and is used as a DNN accelerator [14]. The architecture is illustrated in Fig. 1, where a PE is placed as an array and inputs and outputs of each PE are connected to adjacent PEs. A PE is composed of an adder, multiplier, and registers. The PE receives an a and b from the upper and left PEs, respectively. These matrix elements are used in the multiplication and transferred to the right and bottom PEs through registers a_{reg} and b_{reg}. A register c_{reg} accumulates the multiplication result. Each PE performs a multiply-accumulate operation in the corresponding position. For example, PE_{11} calculates Eq. (2) sequentially by using three clocks, as shown in Eq. (3). The PEs share the same weight parameter in each column. For example, the weight parameter b_{11} is used in PE_{11}, PE_{21}, and PE_{31} and is multiplied by a_{11}, a_{21}, and a_{31}.

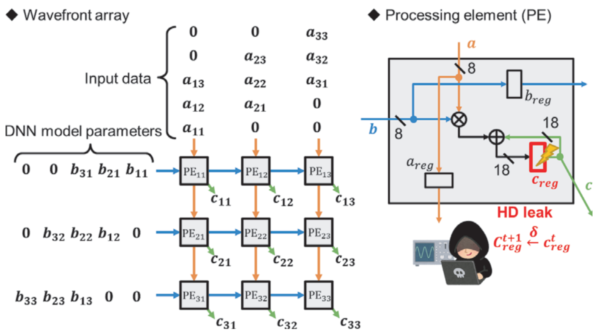

◆ Wavefront array
Input data
DNN model parameters
0 0 a_{33}
0 a_{23} a_{32}
a_{13} a_{22} a_{31}
a_{12} a_{21} 0
a_{11} 0 0
0 0 b_{31} b_{21} b_{11}
0 b_{32} b_{22} b_{12} 0
b_{33} b_{23} b_{13} 0 0
PE_{11} PE_{12} PE_{13}
PE_{21} PE_{22} PE_{23}
PE_{31} PE_{32} PE_{33}
c_{11} c_{12} c_{13}
c_{21} c_{22} c_{23}
c_{31} c_{32} c_{33}
◆ Processing element (PE)
a
8
b_{reg}
b
8
18
a_{reg}
18
c_{reg}
c
HD leak
$c_{reg}^{t+1} \overset{\delta}{\leftarrow} c_{reg}^{t}$

Fig. 1 Architecture of wavefront array.

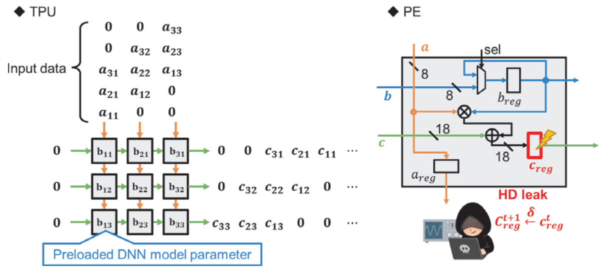


Fig. 2 Architecture of TPU.

$$
\begin{aligned}
c_{reg}^{t+1}(PE_{11}) &= 0 && (t=0)\\
c_{reg}^{t+1}(PE_{11}) &= a_{11}\times b_{11} + c_{reg}^{t}(PE_{11}) &&\\
&= a_{11}\times b_{11} + 0 && (t=1)\\
c_{reg}^{t+1}(PE_{11}) &= a_{12}\times b_{21} + c_{reg}^{t}(PE_{11}) &&\\
&= a_{12}\times b_{21} + a_{11}\times b_{11} && (t=2)\\
c_{reg}^{t+1}(PE_{11}) &= a_{13}\times b_{31} + c_{reg}^{t}(PE_{11}) &&\\
&= a_{13}\times b_{31} + a_{12}\times b_{21} + a_{11}\times b_{11} && (t=3)
\end{aligned}
\tag{3}
$$

2.2 Attack Target (2): Tensor Processing Unit

The tensor processing unit (TPU) proposed by Jouppi et al. is a systolic array architecture [13]. The architecture is illustrated in Fig. 2, where a PE is placed as an array, and inputs and outputs of each PE are connected to adjacent PEs. Each PE receives and holds the corresponding b into b_{reg} before calculation. A PE receives a partial sum c and an input a from the left and upper PEs, respectively. The input a is used in the calculation and transferred to the bottom PEs through register a_{reg}. A register c_{reg} transfers the partial sum $a \times b + c$ to the right PE. The PE in each row performs the multiply-accumulate operation on the corresponding row. For example, PE_{11}, PE_{12}, and PE_{13} are used to calculate Eq. (2) sequentially by using three clocks, as shown in Eq. (4). The weight parameters are not transferred when calculating but are reused for sequentially multiplied by a. For example, the weight parameter b_{11}, which is stored in PE_{11}, is sequentially multiplied by a_{11}, a_{21}, and a_{31}.

$$
\begin{aligned}
c_{reg}(PE_{11}) &= a_{11}\times b_{11} + 0 && (t=1)\\
c_{reg}(PE_{12}) &= a_{12}\times b_{21} + c_{reg}(PE_{11}) &&\\
&= a_{12}\times b_{21} + a_{11}\times b_{11} && (t=2)\\
c_{reg}(PE_{13}) &= a_{13}\times b_{31} + c_{reg}(PE_{12}) &&\\
&= a_{13}\times b_{31} + a_{12}\times b_{21} + a_{11}\times b_{11} && (t=3)
\end{aligned}
\tag{4}
$$

3. Threat Model

3.1 Scenario

We assume that the adversary's target is an AI edge device. The device is equipped with a systolic array as a DNN accelerator. Figure 3 shows the scenario of an MRE attack.

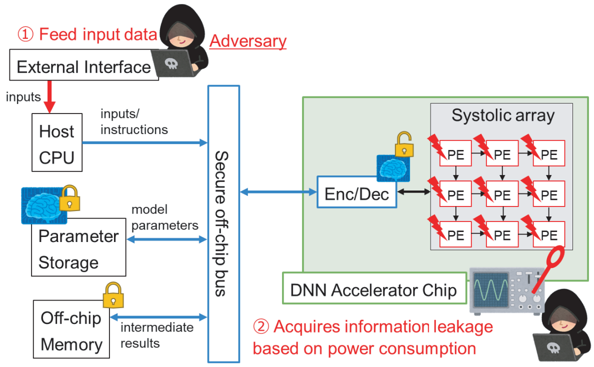


Fig. 3 Scenario of MRE attack.

We assume that the trained DNN model is encrypted and stored in the parameter storage before shipping. The encrypted DNN model in the parameter storage is decrypted in the DNN accelerator chip. Thus, the adversary cannot reveal the DNN model parameters by reading the parameter storage directly or by memory bus snooping [8]. However, the DNN model parameters are decrypted during the operation and are vulnerable to side-channel attacks against the DNN accelerator chip.

3.2 Adversary's Capability

- **An adversary can input any data into the DNN accelerator:** Note that the adversary does not need to know any output data and/or output probability.
- **The adversary knows the target DNN accelerator architecture:** The adversary needs to be able to calculate the register values of each PE. The architecture will be known if the open-source or standard hardware architecture is used in the DNN accelerator.
- **The adversary knows the DNN model architecture:** The adversary knows the DNN model architecture other than weight parameters. It includes the number of layers and nodes, type of activations, batch normalization and bias parameters depending on the DNN models. These conditions are advantageous for the attacker, however, some of the parameters can be known when the well-known architectures are applied as a DNN model.

4. Attack Methodology

4.1 Correlation Power Analysis

CPA was proposed by Brier et al. [16] and is a typical and powerful method of revealing a cryptographic key by using the power consumption during the cryptographic circuit operation. We use CPA to reveal weight parameters from the target DNN model. An adversary uses the correlation between the power consumption and intermediate value or transition on the circuit node. For example, a register consumes power when the value transitions from 0 to 1 or 1 to

0 at the clock edge. Thus, the register's power consumption depends on the hamming distance (HD) of the bit transition. CPA for MRE attacks uses the bit transitions of the register, which stores intermediate results when the result of the multiply-accumulate operation is updated.

In this study, we applied CPA to the c_{reg} register shown in Figs. 1 and 2 for revealing each DNN parameter b. These systolic arrays have different PE architecture but the adversary can attack with the same procedure when they focus on the c_{reg} register. The attack procedure is as follows. The adversary chooses a target PE and focuses on its c_{reg} as a target register. First, the adversary inputs random numbers $a_n (0 \leq n \leq N-1)$ into the DNN accelerator N-times and observes power consumption $P_n (0 \leq n \leq N-1)$. The adversary assumes 256 types (0x00 to 0xff) of 8-bit integer b_i candidates and predicts all patterns of the transition of the target register from c^{t}_{reg} to c^{t+1}_{reg}. The transition of c_{reg} is represented by Eqs. (3), (4). The adversary calculates $H\hat{D}_{n,b_i}$ for all a_n and b_i (Eq. (5)). Finally, the adversary calculates the correlation coefficient between these $H\hat{D}_{n,b_i}$ and power trace P_n (Eq. (6)). The estimated parameter $\hat{b}$ is an argument of the maximum $|\rho(b_i)|$ (Eq. (7)).

As shown in Eqs. (3) and (4), the register transition is dependent on the previous calculation result. The adversary knows that the register is initialized by zero, and there is one unknown parameter when $t = 1$. If the adversary identifies the weight parameter at $t = 1$, there is one unknown parameter when $t = 2$. Therefore, the adversary can reveal the unknown parameters used in the multiply-accumulate operation in order from the first value.

$$H\hat{D}_{n,b_i} = HD(c^{t+1}_{reg}, c^{t}_{reg}) \tag{5}$$

$$\rho(b_i) = \frac{\Sigma_{n=0}^{N-1}(P_n - \bar{P})(H\hat{D}_{n,b_i} - H\bar{D}_{b_i})}{\sqrt{\Sigma_{n=0}^{N-1}(P_n - \bar{P})^2}\sqrt{\Sigma_{n=0}^{N-1}(H\hat{D}_{n,b_i} - H\bar{D}_{n,b_i})^2}} \tag{6}$$

$$\hat{b} = arg\max_{b_i}(|\rho(b_i)|) \tag{7}$$

4.2 CPA Simulation for Multiply-Accumulate Operation

CPA against multiplication is not only used for MRE attacks. Side-channel attacks against pairing-based cryptography use hamming-weight-based CPA against multiplication for revealing secret keys [17], [18]. However, there are differences between CPA against multiplication on pairing-based cryptography and on the DNN inference. The significant differences are that the DNN inference consists of arithmetic multiplication rather than modular multiplication and involves arithmetic addition. These differences provide different results from existing attacks on cryptographic circuits. For example, the CPA against the DNN inference is very sensitive to the noise contained in the signals. We simulated CPA against the multiply-accumulate operation for this reason.

The simulation supposes that a PE has a multiplier, an adder, and a register which stores partial sum. This assumption is common for both systolic arrays. The procedures assume wavefront array but similar results are obtained in TPU.

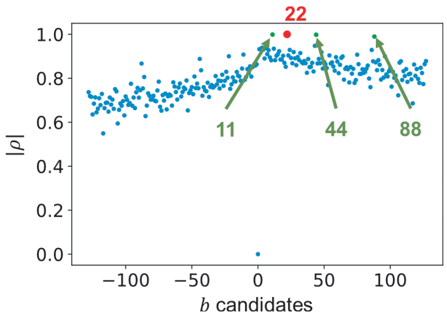


Fig. 4 CPA simulation results when $HD(a \times b + 0, 0)$.

This simulation calculates a correlation between $HD(a \times b_{true} + c, c)$ and $HD(a \times b_{candidates} + c, c)$. Where a is an input value, b is a weight parameter value, c is a stored value in c_{reg}. The b_{true} is a target value which assumes the DNN model parameters, $b_{candidates}$ are candidate values in 8-bit integer. The range of a and b is an 8-bit integer.

The simulation procedure is as follows. First, we set the target value by assuming the DNN model parameters. Second, we calculate the HD for all patterns of the input a and candidate value b_i. The b_i is one of the candidate values from $b_{candidates}$.

Third, we calculate the correlation coefficient between the HD distribution of the target values b_{true} and b_i. Finally, we evaluate the CPA simulation results using the difference between the correlation coefficient of each candidate b_i and target value b_{true}. When $b_i = b_{true}$, the correlation coefficient is obviously 1.0.

CPA is successful in deriving the target value when $arg\max_{b_i}(|\rho(b_i)|) = b_{true}$. The CPA result is robust against noise when the differences between the correlation coefficients are significant.

Figure 4 shows the CPA simulation results when the $c = 0$ and the target value is 22. For the wavefront array, this is satisfied when $t = 1$ in Eq. (3).

CPA is successful in deriving the target value because the correlation coefficient value of 22 is 1.0, which is the highest value in all candidates. However, the CPA result is not robust because the difference between the correlation coefficient is insignificant. Therefore, if in the noisy experimental environment, the correlation coefficient value of 22 may become lower than that of other candidates. In particular, a positive 8-bit integer obtained by bit-shifting $b_{true} = 22$ (e.g. 11,44,88) has a high correlation value. These HD distributions of the n-bit shifted candidates are the same as the b_{true} when the input is $a \geq 0$, as shown in the following equation.

$$\begin{aligned} HW(a \times b) &= HW(a \times (b << n)) \\ &= HW((a \times b) << n) \end{aligned} \tag{8}$$

where function $HW(\cdot)$ calculates a hamming weight.

For instance of Eq. (9), the following calculations have the same HW, respectively. Where $a = 5$ and $b =$

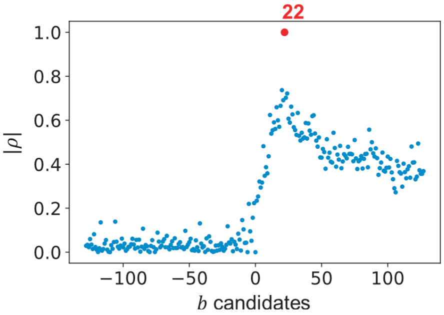


Fig. 5 CPA simulation results when $HD(a \times b + a' \times b', a' \times b')$, a' = 8-bit random number, $b' = -73$.

$11, 22, 44, 88$. The same results are obtained for other $a \geq 0$.

$$\begin{array}{rcccl} (5 \times 11)_{10} & = & (55)_{10} & = & (000000110111)_2 \\ (5 \times 22)_{10} & = & (110)_{10} & = & (000001101110)_2 \\ (5 \times 44)_{10} & = & (220)_{10} & = & (000011011100)_2 \\ (5 \times 88)_{10} & = & (440)_{10} & = & (000110111000)_2 \end{array} \quad (9)$$

Figure 5 shows the CPA simulation results when $c = a' \times -73$, a' = 8-bit random number. For the wavefront array, this is satisfied when $t > 1$ in Eq. (3).

CPA is successful in deriving the target value because the correlation coefficient value of 22 is 1.0, and it is the highest value. Moreover, the CPA result is robust because the differences between the highest correlation coefficient and the others are significant.

These simulations revealed that CPA is very sensitive to the noise contained in signals when the target operation is composed of only multiplication due to the difference between the correlation coefficient of b_{ture} and b_i being insignificant. In contrast, CPA is robust when the target operation is composed of multiplication and addition. The target operation of an MRE attack is multiply-accumulation, but the first operation consists of multiplication and zero addition. Thus, adversaries should note that they may predict the wrong candidate when the attack is on the first value (e.g., $t = 1$ Eqs. (3) and (4)). Also, the intermediate result of the latter operations is dependent on the result of the first operation. The adversary predicts the wrong candidate at latter operations when they predicts the wrong candidate at the first operation.

4.3 Chain CPA

In this section, we discuss the CPA against systolic arrays using a wavefront array as an example. Weight parameters b are repeatedly used in matrix calculation on the systolic array, so an adversary can attack multiple times against each PE.

Algorithm 1 Attack procedure of chain CPA against PE_{11}

```
Input:      $a_{11}, a_{12}, a_{13}, P_n, J$
Output:     $b_{11}, b_{21}, b_{31}$
Initialize: $b_i \in \{0, \ldots, 255\}, \quad j \in \{1, \ldots, J\}$
    $c_{reg} \leftarrow 0$
    $c_{reg} \leftarrow a_{11_n} \times b_i + c_{reg} \quad for\ all\ n, i$
    1st CPA: Calculate $|\rho(b_i)| \quad for\ all\ i$ by equation(5-7) with $P_n$
    $\hat{b}_{11}^{(j)} \leftarrow best\ J\ candidates\ of\ higher\ |\rho(b_i)|$
    $c_{reg}^{(j)} \leftarrow a_{12_n} \times b_i + a_{11_n} \times \widehat{b_{21}^{(j)}} \quad for\ all\ n, i, j$
    2nd CPA: Calculate $|\rho(b_i)| \quad for\ all\ i$ by equation(5-7) with $P_n$
    $\hat{b}_{21}^{(j)} \leftarrow arg\max_i |\rho(b_i)| \quad for\ all\ j$
    $c_{reg}^{(j)} \leftarrow a_{13_n} \times b_i + a_{12_n} \times \widehat{b_{21}^{(j)}} + a_{11_n} \times \widehat{b_{11}^{(j)}}\ for\ all\ j$
    3rd CPA: Calculate $|\rho(b_i)| \quad for\ all\ i$ by equation(5-7) with $P_n$
    $\hat{b}_{31}^{(j)} \leftarrow arg\max_{b_i} |\rho(b_i)| \quad for\ all\ j$
    $j_d \leftarrow arg\max_j |\rho(\hat{b}_{31}^{(j)})|$
    $b_{11}, b_{21}, b_{31} \leftarrow \hat{b}_{11}^{(j_d)}, \hat{b}_{21}^{(j_d)}, \hat{b}_{31}^{(j_d)}$
```

In the CPA against the first operation of a PE, 2^8 candidate is given as a weight parameter b_i, and the candidate with the largest correlation is selected. Unfortunately, the correctness of simple CPA is low because the multiplication and zero addition are operated in the first operation of the PE. In the case of the second operation of the PE, the result of the first operation is stored in the register and is added by the current multiplication result, so the confidence of CPA results increases. However, the number of second CPA attack candidates increases as much as 2^8 times, because the previous value on the register has 2^8 variation depending on the results of the first CPA attack. Considering the third CPA attack, the number of candidates increases $2^8 \times 2^8$ times. Hence, the naive multiple CPA calculates $2^8 \times 2^8 \times 2^8$ patterns of the combination b_{11}, b_{21}, b_{31} at attacking against Eq. (2). There are many combinations, which increase exponentially depending on the size of the matrix. We applied a post-processing approach called "chain CPA" for efficiently reduces the combinations by using the structure of the multiply-accumulate operation.

In a simple CPA, the candidate with the highest correlation is selected as a correct parameter. As explained in the previous section, multiple candidates which have the shifted value of correct one have high correlation when t=1, so the highest candidate may be a false positive parameter. Then, multiple candidates which have first to Jth highest correlation are selected in the chain CPA. For example, in the case of Fig. 4, the adversary sets the $J = 4$ and chooses candidates $11, 22, 44, 88$. When attacking the operation after $t = 1$, the adversary calculates CPA assuming each of the J previous candidates. The adversary chooses one candidate that has the highest correlation coefficient for each CPA. After CPA is used against the series of calculations based Eq. (3) or (4), the combination of candidates that achieves the highest correlation coefficient at the last operation is selected as the estimated values. Chain CPA calculates $2^8 + J \times 2^8 \times (3 - 1)$ patterns of the combination b_{11}, b_{21}, b_{31} at attacking against Eq. (2). It means that the chain CPA calculates correlation against 2^8 number of candidates and reduces the candidates to J, and calculates correlation against 2^8 number of candidates for each J number of first value candidates in each remaining calculations.

These combinations depend on J, do not increase exponentially depending on the size of the matrix. The details of this attack procedure against PE_{11} from wavefront array are given in Algorithm 1.

5. Experiment

5.1 Setup

Figure 6 shows our experimental environment. We implement two systolic arrays shown in Figs. 1 and 2 to an FPGA to evaluate CPA and our chain CPA. The target platform is ZUIHO, which is the side-channel attack standard evaluation board developed by the National Institute of Advanced Industrial Science and Technology (AIST) of Japan. The target FPGA chip is Xilinx spartan3-A. An oscilloscope (Agilent Technologies DSO6104A) is used for acquiring power traces. The power traces were acquired with AC coupling. The goal of our experiment is deriving nine secret DNN model parameters $(b_{11}, b_{12}, \dots, b_{33})$. We configured the target model parameters as follows.

$$b = \begin{pmatrix} -92 & 122 & 22 \\ -20 & -16 & 46 \\ -104 & -73 & 73 \end{pmatrix} \tag{10}$$

We input nine individual 8-bit random numbers (input a) into the DNN accelerator and acquire 20,000 power traces from wavefront array and 50,000 power traces from TPU.

Figure 7 shows the mean waveform of these traces from the wavefront array. Figure 8 shows the mean waveform of these traces from the TPU. There are power consumption peaks due to each PE operation. For example, PE_{11} operates $t = 1, t = 2$, and $t = 3$ at times (1), (2), and (3), as shown in Fig. 7, referring to the calculation sequence in Eq. (3). Similarly, $PE_{11}, PE_{12}, PE_{13}$ operates $t = 1,\ t = 2,\ and\ t = 3$ at times (1), (2), and (3), as shown in Fig. 8, referring to the calculation sequence in Eq. (4).

5.2 MRE Attack with CPA

In this section, we discuss our evaluation of CPA against two types of the systolic arrays, i.e., wavefront array (Fig. 1) and TPU (Fig. 2).

Table 1 shows the predicted weight values by CPA against the wavefront array. The shaded area shows that the predicted weight value is correct. The adversary succeed to reveal weight parameters when attacks to $PE_{11}, PE_{12}, PE_{21}, PE_{23}, PE_{31}, PE_{32}$ and PE_{33} but revealed wrong parameters when attacks to the other targets. These wrong parameters are close to shifted values from the correct parameters. As described in Sect. 4.2, the measurement noise may cause incorrect parameters having the highest correlation coefficient when $t = 1$, and the strong candidates are values that shifted from the correct parameter.

Figures 9 and 11 shows the results of CPA with 20,000 power traces for the wavefront array when b_{11} and b_{21} at PE_{11}, and these figures correspond to simulation results in Figs. 4, 5. Figures 10 and 12 shows the results of CPA for the wavefront array against the number of traces untill 2,000 power traces when b_{11} and b_{21} are targeted at PE_{11}. Figures 9 and 10 shows the CPA evaluation results when the $t = 1$ at Eq. (3) and the target value was -92. The correlation coefficient of the b_{true} and the others are antagonizing. The reason was introduced in Sect. 4.2.

Figure 10 represent how many traces are need for CPA. The solid red line which represents correlation coefficient value of b_{true} achieves the highest rank when the number of traces is more than 200 traces, but the red line is close to other candidates that are represented by gray solid lines

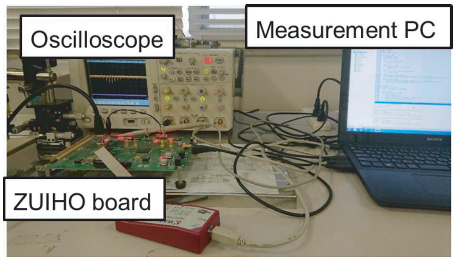


Fig. 6 Image of our experimental environment.

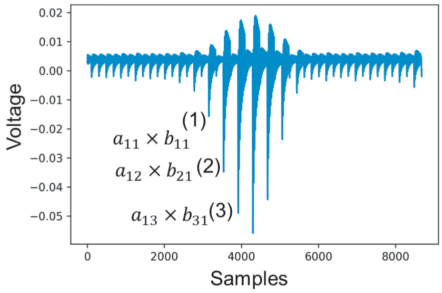


Fig. 7 Mean waveform of power traces from wavefront array.

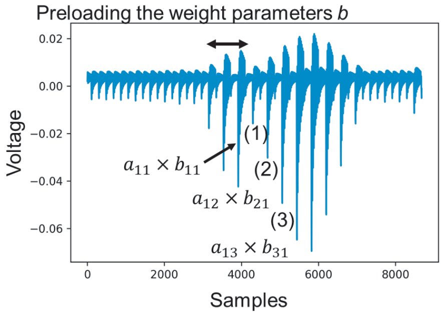


Fig. 8 Mean waveform of power traces from TPU.

Table 1 The CPA results for wavefront array.

	b_{11}	b_{21}	b_{31}	b_{12}	b_{22}	b_{32}	b_{13}	b_{23}	b_{33}
Correct	-92	-20	-104	122	-16	-73	22	46	73
		PE_{11}			PE_{12}			PE_{13}	
Predicts	-92	-20	-104	122	-16	-73	11	23	36
		PE_{21}			PE_{22}			PE_{23}	
Predicts	-92	-20	-104	61	-8	-36	22	46	73
		PE_{31}			PE_{32}			PE_{33}	
Predicts	-92	-20	-104	122	-16	-73	22	46	73

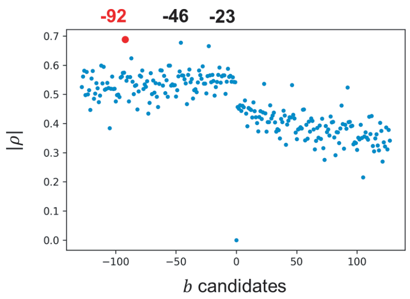


Fig. 9 Results of CPA against first parameter b_{11} at PE_{11}. Where $HD(a\times b,\ 0)$, a =8-bit random number, $b_{true} = -92$. They correspond to simulation results in Fig. 4.

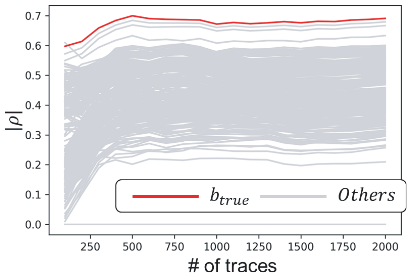


Fig. 10 CPA evaluation results against wavefront array. Where $HD(a \times b,\ 0)$, a =8-bit random number, $b_{true} = -92$.

even if the number of traces increases. It suggests that an adversary can reveal b_{ture} with 200 traces but the CPA result is sensitive against measurement noises though a large number of traces are acquired.

Figure 11 shows the CPA evaluation results when $t = 2$ in Eq. (3) and the target value was -20. The target calculation consisted of multiplication and addition where $c \neq 0$, so CPA was robust due to the significant differences between the highest correlation coefficient and the others.

In Fig. 12, the solid red line was achieved the highest rank before 100 traces and the difference between the red line and gray lines is wide. It shows that an adversary can reveal b_{true} by less than 100 traces and the CPA result is robust against measurement noises.

If the adversary selects the incorrect value at $t = 1$, they also predicts the incorrect values at $t > 1$ because the multiply-accumulate process depends on the previously selected parameter b (Eq. (3)). For example, as shown in the PE_{13} cells of Table 1, the target value was $b_{31} = 22$ but the wrong candidate 11 achieved the highest correlation coefficient value.

Table 2 shows the predicted weight values by CPA against the TPU. The shaded area shows that the predicted weight value is correct. The adversary succeed to reveal weight parameters when attacks to $PE_{11-13}, PE_{21-23}, PE_{31-33}$ with a_{11-13}, PE_{21-23} with a_{21-23} and PE_{11-13} with a_{31-33} but revealed wrong parameters when attacks to the other targets. These wrong parameters are close to shifted values from the correct parameters.The correlation coefficient graph of TPU were similar to that of wavefront array which was shown in Figs. 9–12.

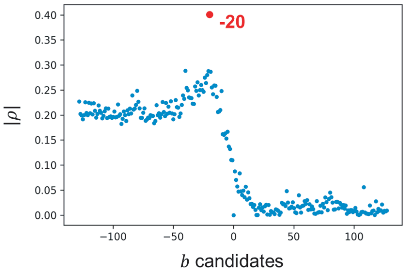


Fig. 11 Results of CPA against second parameter b_{21} at PE_{11}. Where $HD(a \times b + c,\ c)$, a, a' =8-bit random numbers, $b_{true} = -20$, $c = a' \times -92$. They correspond to simulation results in Fig. 5.

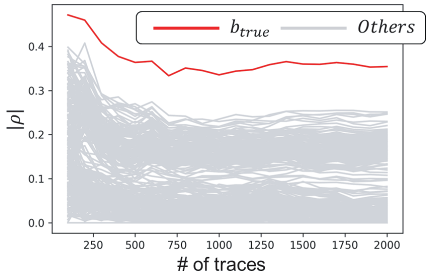


Fig. 12 CPA evaluation results against wavefront array. Where $HD(a \times b + c,\ c)$, a, a' =8-bit random numbers, $b_{true} = -20$, $c = a' \times -92$.

5.3 MRE Attack with Chain CPA

In this section, we discuss our evaluation of chain CPA against two types of the systolic arrays, i.e., wavefront array (Fig. 1) and TPU (Fig. 2).

Table 3 shows the predicted weight values by chain CPA against the wavefront array. The shaded area shows that the predicted weight value is correct. The adversary was able to reveal all nine of the target weight parameters with chain CPA. As shown in Table 1, the reason why CPA predicts incorrect (shifted) weight parameters is that the adversary selects the incorrect value at $t = 1$. Comparing the results in the Table 3 and Table 1, chain CPA can predict correct weight parameters even if the incorrect candidate achieved the highest correlation coefficient at $t = 1$.

Table 4 shows the predicted weight values by CPA against the TPU. The shaded area shows that the predicted weight value is correct. The adversary succeed to reveal weight parameters when attacks to $PE_{11-13}, PE_{21-23}, PE_{31-33}$ with a_{11-13}, PE_{21-23}, PE_{31-33} with a_{21-23} or a_{31-33} but revealed wrong parameters when attacks to the other targets. These wrong parameters are close to shifted values from the correct parameters.

The chain CPA succeeded in attacking with more targets than CPA, it shows that the chain CPA mitigates the effect of measurement noises in the calculation $t = 1$. However, the cell that attacked PE_{11-13} with a_{21-23} is succeeded by CPA but failed by chain CPA.

Table 2 The CPA results for TPU.

	b_{11}	b_{21}	b_{31}	b_{12}	b_{22}	b_{32}	b_{13}	b_{23}	b_{33}
Correct	-92	-20	-104	122	-16	-73	22	46	73
Predicts		PE_{11-13}			PE_{21-23}			PE_{31-33}	
with a_{11-13}	-92	-20	-104	122	-16	-73	22	46	73
Predicts		PE_{11-13}			PE_{21-23}			PE_{31-33}	
with a_{21-23}	-46	-10	-52	122	-16	-72	11	23	37
Predicts		PE_{11-13}			PE_{21-23}			PE_{31-33}	
with a_{31-33}	-92	-20	-104	61	-8	-36	11	23	36

Table 3 The chain CPA results for wavefront array.

	b_{11}	b_{21}	b_{31}	b_{12}	b_{22}	b_{32}	b_{13}	b_{23}	b_{33}
Correct	-92	-20	-104	122	-16	-73	22	46	73
		PE_{11}			PE_{12}			PE_{13}	
Predicts	-92	-20	-104	122	-16	73	22	46	73
		PE_{21}			PE_{22}			PE_{23}	
Predicts	-92	-20	-104	122	-16	-73	22	46	73
		PE_{31}			PE_{32}			PE_{33}	
Predicts	-92	-20	-104	122	-16	-73	22	46	73

Table 4 The chain CPA results for TPU.

	b_{11}	b_{21}	b_{31}	b_{12}	b_{22}	b_{32}	b_{13}	b_{23}	b_{33}
Correct	-92	-20	-104	122	-16	-73	22	46	73
Predicts		PE_{11-13}			PE_{21-23}			PE_{31-33}	
with a_{11-13}	-92	-20	-104	122	-16	-73	22	46	73
Predicts		PE_{11-13}			PE_{21-23}			PE_{31-33}	
with a_{21-23}	-23	-5	-26	122	-16	-72	22	46	73
Predicts		PE_{11-13}			PE_{21-23}			PE_{31-33}	
with a_{31-33}	-23	-5	-26	122	-16	-73	22	46	73

5.4 Discussion

Our experimental results indicate that the adversary can derive all the secret DNN model parameters through CPA and chain CPA against systolic arrays. We indicated the register c_{reg}, which stores the intermediate result of the calculation, has information leakage about the secret weight parameter. In principle, the adversary can attack a larger systolic array with a similar procedure. A systolic array has various derivations, but the adversary can attack in a similar procedure if these architectures have registers that store accumulated results such as c_{reg} and the adversary can calculate the register transitions. When the acquired trace is too noisy, the adversary can improve the signal-to-noise (S/N) ratio by acquiring more traces or using EM analysis. In particular, EM analysis can focus on the power consumption of a specific PE and may have advantages for a larger systolic array.

In the MRE attack scenario, the adversary has the edge AI with secret weight parameters, and he try to reveal parameters by the MRE using CPA. In order to accomplish practical MRE attacks, the adversary have to verify the correctness of weight parameters obtained by CPA. It is an important and a difficult challenge, because the adversary may get incorrect parameters, or want to distinguish which of two candidates of revealed weight parameters are correct as shown in our experimental results. In principle, the verification process can be carried out as follows. At first, the adversary set the obtained parameters on another edge AI device as a test device. Secondly, he should compare the inference-output results from the target device and that from the test device. Lastly, if the identical results are output from these two devices when any inputs are supplied, the parameters set on the test device may be correct. Although, to the best of our knowledge, there have been no studies to confirm whether each weight parameter on two devices are identical from input-output pair data. These are important future research topics to establish the verification method of the candidate parameters.

It is necessary to introduce countermeasures for preventing an adversary from using the correlation between the power consumption of the circuit and register transition to protect DNN model parameters. The main idea of a countermeasure is to make it difficult for the adversary to predict the intermediate value of the operation or observe the correlation between the power consumption of the circuit and intermediate value. The simple idea is that the c_{reg} of each PE is initialized by a random value through a dedicated path. It is easy to apply to the TPU since the calculation result is not dependent on the initial value of c_{reg}. However, ingenuity is required to apply such a countermeasure to the wavefront array due to the calculation result changes depending on the initial value of c_{reg}.

Batina et al. mentioned the shuffling technique as a countermeasure [9]. In a multiply-accumulate operation, the result of the operation does not change even if the order of addition is changed. The operation of each element of the matrix is also an independent. The shuffling can reduce the threat of CPA, but the adversary can attack even if the countermeasure is applied when the adversary has enough power traces.

Dubey et al. introduced a countermeasure against CPA to a binarized DNN accelerator [12]. The countermeasure is roughly divided masking and hiding. The masking technique separates the input a from the share $a - r$ and r by a random number r. The operation result of the share is summed after two multiply-accumulate operations for each and the effect of r disappears. The adversary cannot predict the intermediate value of the multiply-accumulate operation due to the unknown r. However, the countermeasure requires two calculations and summations, and the latency increases more than doubles. The hiding technique applies a complementary circuit, such as a wave dynamic differential logic (WDDL) [21], to the leaky operation. The WDDL breaks the link between the power consumption of the devices and processed data values, and the adversary cannot observe the correlation. However, the countermeasure requires a larger circuit than the original.

There are countermeasures to protect the parameters by applying the homomorphic encryption scheme [19], [20]. However, these schemes require an extremely high processing performance and are unsuitable for an edge device (low-power and low-cost device).

These countermeasures have pros and cons, and we should carefully evaluate the effect of a countermeasure and the implementation cost. The tamper resistance of a DNN accelerator may be more improved by combining multiple countermeasures.

6. Conclusion

A DNN accelerator is important for an AI application that is executed on an edge device. AI edge devices should be robust against hardware-oriented attacks. Thus, the study of tamper-resistant DNN accelerator hardware is required for protecting the DNN model, which is important intellectual property.

In this paper, we measured information leakage from two types of systolic arrays that are used for the matrix multiplication unit in DNN processors. We demonstrated that an adversary can apply correlation power analysis (CPA) to MRE attack which reveals weight parameters of a DNN model from the systolic array.

We simulated CPA against PEs, which are elements of a systolic array. CPA is very sensitive to the noise contained in signals when the target operation is composed of only multiplication. However, it is robust when the target operation is composed of multiplication and addition. The intermediate result of the latter operations is dependent on the result of the first operation. We found that CPA against the first calculation is sensitive to the measurement noises by the results of simulations. Thus, an adversary predicts the wrong candidate during latter operations when they predicts the wrong candidate during the first operation.

We applied an extended method of CPA called "chain CPA" for mitigating the problem in the normal CPA. Chain CPA efficiently reduces the combinations of the brute force CPA by using the structure of the multiply-accumulate operation in systolic arrays. While the computational cost of the brute force CPA increases exponentially depending on the size of the matrix, the computational cost of chain CPA is several times that of the simple CPA. The adversary can mitigate the noise sensitivity of CPA against the first operation by using chain CPA.

From the experimental results of normal CPA against systolic arrays, the attack estimates the correct parameter on seven of nine PEs on the wavefront array, and five of nine multiply-accumulations on the TPU. The reason why CPA predicted the wrong candidates was that the adversary predicts the wrong (shifted) candidate during the first operation. Since the second calculation depends on the first calculation, if the adversary estimates the wrong weight parameter at the first calculation, the adversary estimates the wrong parameters at the subsequent calculations.

In the result of chain CPA against the wavefront array, the adversary succeeded and revealed correct parameters on all PEs. The chain CPA revealed correct weight parameters even if the wrong candidate achieved the highest correlation coefficient at $t = 1$. In the result of chain CPA against the TPU, the adversary succeeded to attack seven of nine multiply-accumulations. The adversary narrowed the candidates for three of nine weight parameters down to two patterns and revealed the other six of nine weight parameters. These results are improved compared to the normal CPA, which indicates that chain CPA mitigates the problem of CPA against systolic arrays.

Our experimental results show that an adversary can reveal trained model parameters from a DNN accelerator even if the DNN model parameters in the off-chip bus are protected with data encryption. This suggests that countermeasures against side-channel leaks are important for implementing a DNN accelerator on an FPGA or ASIC.

Acknowledgments

This work was supported by JST-Mirai Program Grant Number JPMJMI19B6, Japan.

References

[1] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," Proc. 2015 ACM SIGSAC Conference on Computer and Communications Security (CCS), pp.1322–1333, 2015.

[2] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models," arXiv, arXiv:1806.01246, 2018.

[3] I.J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv, arXiv:1412.6572, 2014.

[4] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning models," arXiv, arXiv:1707.08945, 2017.

[5] F. Tramer, F. Zhang, A. Juels, M.K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," SEC'16: Proc. 25th USENIX Conference on Security Symposium, pp.601–618, 2016.

[6] B. Wang and N.Z. Gong, "Stealing hyperparameters in machine learning," arXiv, arXiv:1802.05351, 2018.

[7] W. Hua, Z. Zhang, and G.E. Suh, "Reverse engineering convolutional neural networks through side-channel information leaks," Proc. 55th Annual Design Automation Conference, 2018.

[8] X. Wang, R. Hou, Y. Zhu, J. Zhang, and D. Meng, "NPUFort: A secure architecture of DNN accelerator against model inversion attack," Proc. 16th ACM International Conference on Computing Frontiers, pp.190–196, 2019.

[9] L. Batina, S. Bhasin, D. Jap, and S. Picek, "CSI neural network: Using side-channels to recover your artificial neural network information," IACR Cryptology ePrint Archive, vol.2018, p.477, 2018.

[10] K. Yoshida, T. Kubota, M. Shiozaki, and T. Fujino, "Model-extraction attack against FPGA-DNN accelerator utilizing correlation electromagnetic analysis," 27th IEEE International Symposium On Field-Programmable Custom Computing Machines, 2019.

[11] K. Yoshida, S. Okura, M. Shiozaki, T. Kubota, and T. Fujino, "Model reverse-engineering attack using correlation power analysis against systolic array based neural network accelerator," IEEE International Symposium on Circuits and Systems, 2020.

[12] A. Dubey, R. Cammarota, and A. Aysu, "MaskedNet: The first hardware inference engine aiming power side-channel protection," arXiv, arXiv:1910.13063, 2019.

[13] N.P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T.V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C.R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin,

G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, and D.H. Yoon, "In-datacenter performance analysis of a tensor processing unit," arXiv, arXiv:1704.04760, 2017.

[14] Z. Yang, L. Wang, D. Ding, X. Zhang, Y. Deng, S. Li, and Q. Dou, "Systolic array based accelerator and algorithm mapping for deep learning algorithms," IFIP International Conference on Network and Parallel Computing, pp.153–158, 2018.

[15] H.T. Kung "Why systolic architectures?," IEEE Computer 15.1, pp.37–46, 1982.

[16] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," Conference on Cryptographic Hardware and Embedded Systems, LNCS, vol.3156, pp.16–29, 2004.

[17] T. Unterluggauer and E. Wenger, "Practical attack on bilinear pairings to disclose the secrets of embedded devices," 9th International Conference on Availability, Reliability and Security, 2014.

[18] D. Jauvart, J.J.A. Fournier, N. El Mrabet, and L. Goubin, "Improving side-channel attacks against pairing-based cryptography," Risks and Security of Internet and Systems, LNCS, vol.10158, pp.199–213, 2017.

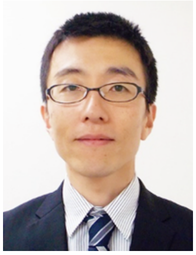
[19] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy," International Conference on Machine Learning, pp.201–210, 2016.

[20] B. Reagen, W. Choi, Y. Ko, V. Lee, G.-Y. Wei, H.-H.S. Lee and D. Brooks, "Cheetah: Optimizations and methods for PrivacyPreserving inference via homomorphic encryption," arXiv, arXiv:2006.00505, 2020.

[21] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," 2004 Design, Automation and Test in Europe Conference and Exposition (DATE2004), vol.1, pp.246–251, IEEE Computer Society, 2004.

Kota Yoshida received his B.E. and M.E. in electronic engineering from Ritsumeikan University in 2017 and 2019. He is currently a doctoral student at the Graduate School of Science and Technology, Ritsumeikan University. His research interests include machine learning and hardware security. He is a member of IEICE, IEEE.

Mitsuru Shiozaki received his B.E. and M.E. in electronic engineering from Ritsumeikan University in 1998 and 2000 and received a Ph.D. in electronics engineering from Hiroshima University in 2004. He is currently an associate professor with the Research Organization of Science & Engineering at Ritsumeikan University. His research interests include hardware security and physically unclonable functions. He is a member of ACM, IEEE, IEICE.



Shunsuke Okura received his B.S., M.S., and Ph.D. from Osaka University, Osaka, Japan, in 2003, 2005, and 2010. He was with the Rosnes Corporation from 2007, Renesas Electronics Corporation from 2010, and Brillnics Incorporated from 2014, where he has been engaged in the research and development of CMOS image sensors. Since 2019, he has been the associate professor in Electrical and Computer Engineering at Ritsumeikan University. He is a member of the Institute of Image Information and Television Engineers in Japan and IEEE.

Takaya Kubota joined NTT Software Corporation in 1991, and was involved in the development of network software. From 2005 to 2012 he worked on the development of java distributed objects running on embedded systems at the National Institute for Advanced Industrial Science and Technology (AIST) in Japan. He also developed a side-channel testing environment for cryptographic modules. He is currently a researcher at Ritsumeikan University. He is engaged in side-channel analysis for anti-tamper cryptographic modules.

Takeshi Fujino was born in Osaka, Japan, on March 17, 1962. He received his B.E. and M.E., and Ph.D. in electronic engineering from Kyoto University, Kyoto, Japan, in 1984, 1986, and 1994. He joined the LSI Research and Development center, Mitsubishi Electric Corp. in 1986. Since then, he had been engaged in the development of micro-fabrication processes, such as electron beam lithography, and embedded DRAM circuit design. He has been a professor at Ritsumeikan University since 2003. His research interests include hardware security such as side-channel attacks and physically unclonable functions. He is a member of IEICE, IPSJ, JSAP, IEEE.