IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 8, 2022, accepted March 26, 2022, date of publication March 31, 2022, date of current version April 18, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3163852

Improved Correlation Power Analysis on Bitslice Block Ciphers

JAESEUNG HAN[1], YEON-JAE KIM[1], SOO-JIN KIM[1], BO-YEON SIM[2], AND DONG-GUK HAN[1,3]

[1]Department of Financial Information Security, Kookmin University, Seoul 02707, Republic of Korea
[2]Department of Intelligent Convergence Research Laboratory, Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Republic of Korea
[3]Department of Mathematics, Kookmin University, Seoul 02707, Republic of Korea

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

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

ABSTRACT Bitslice block ciphers have the advantage of allowing parallel computation using bitwise logical operations, and Boolean masking can be applied efficiently. Thus, various bitslice block ciphers, such as Robin, Fantomas, RECTANGLE, RoadRunneR, PRIDE, and CRAFT, have been proposed previously. Additionally, a bitslice implementation for AES, National Institute of Standards and Technology (NIST) standard block cipher, has been proposed. These ciphers construct an S-Box using only bitwise logical operators. They perform operations by storing the *i-th* bits of each S-Box input/output value in one register, *i.e.*, they have a feature that each bit of an S-Box output is stored in a different register. Because of this feature, in correlation power analysis (CPA) for bitslice block ciphers, a single-bit of the S-Box output should be selected as an intermediate value. Moreover, depending on which bit is selected as the intermediate value, there are differences in analysis performance. Consequently, we propose an algorithm that predicts the CPA performance of each single-bit and we describe the theoretical basis of this algorithm. The effectiveness of the proposed algorithm is verified experimentally by comparing actual CPA results and predicted results on various bitslice block ciphers.

INDEX TERMS Bitslice block cipher, correlation power analysis, side-channel analysis, AES, Robin, Fantomas, RECTANGLE, RoadRunneR, PRIDE, CRAFT.

I. INTRODUCTION

Side-channel analysis (SCA) is a technique that analyzes secret information (*e.g.*, a secret key) using side-channel information, such as power consumptions, electromagnetics, and sounds. Paul Kocher [1] first described SCA and he demonstrated that it could be applied against mathematically secure cryptographic algorithms. Analyzing power consumption is a representative example of SCA that includes simple power analysis [1], differential power analysis [2], and correlation power analysis (CPA) [3]. Among these, CPA is a technique that calculates Pearson's correlation coefficient between power consumption traces and intermediate values of cryptographic algorithm to analyze secret key [3].

A block cipher is an algorithm that encrypts data in a given block unit and comprises linear and nonlinear functions. Generally, S-Box operation is used as nonlinear function that affects all output bits when even 1-bit of input changes. Because of these characteristics, the S-Box output is typically used as the intermediate value when performing CPA. For block ciphers implemented without SCA countermeasures, secret information can be extracted using CPA. Among countermeasures against CPA, the most commonly used technique is Boolean masking, which is operated by dividing intermediate values into several shares.

The bitslice technique implements algorithms using bitwise logical operators. Bitslice block ciphers are designed to optimize a bitslice implementation. Therefore, bitslice block cipher algorithms mostly comprise bitwise logical operators. Particularly, the S-Box operations comprise logical operators that can perform operations in parallel. Because of these features, Boolean masking can be applied efficiently to bitslice block ciphers, resulting in the proposition of various bitslice block ciphers, such as Robin & Fantomas [4], RoadRunneR [5], and CRAFT [6].

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

In these implementations, each bit of the S-Box input/output values is stored in a different register. Considering this, Balasch et al [7] performed CPA using a single-bit of S-Box output as an intermediate value of the bitslice implementation of AES [8]. However, there is no study about which single-bit is more effective for CPA. In this paper, we propose an algorithm that predicts CPA performance by each single-bit of the S-Box output.

A. CONTRIBUTIONS

First, we propose a method that calculates CPA predictive performance determined hypothetical power consumption. We also explain the mathematical basis for this method. Using the proposed method, we propose an algorithm to calculate the CPA predictive performance for each single-bit intermediate value. Finally, we experimentally verify the proposed algorithm by comparing the actual single-bit CPA results of various bitslice block ciphers and the predictive results of the proposed algorithm.

B. ORGANIZATION

The remainder of the paper is organized as follows. In Section II, we describe CPA, bitslice implementation, CPA on bitslice block ciphers, and our challenge. In Section III, we describe the proposed method and algorithm. In Section IV, we compare the CPA results for the Robin cipher and the predictive values obtained using the proposed algorithm. In Section V, we compare the CPA results and predictive values for bitslice implementation of AES, Fantomas, RECTANGLE [9], RoadRunneR, PRIDE [10], and CRAFT. In Section VI, we conclude this paper.

II. PRELIMINARIES

A. SYMBOLS AND NOTATIONS

Table 1 lists the notations used in this paper.

TABLE 1. Definition.

Symbol	Definition
$\oplus$	eXclusive OR operation
$\wedge$	AND operation
$\vee$	OR operation
n	Input, output bit size of S-Box
d	eXclusive OR difference between S-Box inputs
$Cov(X, Y)$	Covariance between X and Y
σ_X	Standard deviation of X
$r_{X,Y}$	Pearson's correlation coefficient between X and Y
$\|x\|$	Absolute value of x
rk (*Right key*)	Right key, $rk \in \{0,1\}^n$
gk (*Guessed key*)	Guessed key, $gk \in \{0, 1, 2, \cdots, 2^{n-1}\}$
sk (*Second key*)	gk having the highest absolute value of Pearson's correlation coefficient except rk
$\mathbb{A}$	A sequence of actual power consumption values
$\mathbb{H}_K$	Hypothetical power consumption values when the key is set to K
$Ratio$	$\|r_{\mathbb{A},\mathbb{H}_{rk}}\|/\|r_{\mathbb{A},\mathbb{H}_{sk}}\|$
$S(x)$	S-Box output for input x
$HW(x)$	Hamming weight value of x
$s_i(x)$	*i-th* bit of S-Box output for input x $(0 \le i < n)$

B. CORRELATION POWER ANALYSIS

CPA is an SCA method that uses Pearson's correlation coefficient (PCC) between the hypothetical power consumption values $\mathbb{H}_K$ and the actual power consumption values $\mathbb{A}$ [3]. The hypothetical power consumption values $\mathbb{H}_K$ are determined by key-dependent intermediate values and a power consumption model. It generally takes the output of a nonlinear function, such as S-Box, as the intermediate value. Example of power consumption models are Hamming weight (HW) model and Hamming distance model. For instance, if the intermediate value is the S-Box output and the power consumption model is the HW model, then the hypothetical power consumption $\mathbb{H}_K$ is the HW of the S-Box output when the key is K. If the key K and power consumption model are correct, there is a linear relationship between hypothetical and actual power consumption values. The formula for the PCC between $\mathbb{A}$ and $\mathbb{H}_K$ is

$$-1 \le r_{\mathbb{A},\mathbb{H}_K} = \frac{Cov\,(\mathbb{A}, \mathbb{H}_K)}{\sigma_{\mathbb{A}}\sigma_{\mathbb{H}_K}} \le 1.$$

In CPA, we compute the absolute value of PCC $|r_{\mathbb{A},\mathbb{H}_{gk}}|$ for every *Guessed key gk*. We expect that the absolute value of PCC $|r_{\mathbb{A},\mathbb{H}_{rk}}|$ of the right key *rk* is highest. The CPA process is as follows:

- Determine the intermediate value
- Determine the power consumption model
- Calculate the absolute value of PCC $|r_{\mathbb{A},\mathbb{H}_{gk}}|$ of every *Guessed key gk*

We expect the *Guessed key* having the highest PCC to be the *Right key*. From this perspective, *Ratio* is used as a CPA performance indicator; it denotes a value that divides $|r_{\mathbb{A},\mathbb{H}_{rk}}|$ by $|r_{\mathbb{A},\mathbb{H}_{sk}}|$, which is the absolute values of PCC about the *Right key rk* and *Second key sk*, respectively. There is another indicator known as *guessing entropy*, which is the average rank position of the *Right key*. However, *guessing entropy* cannot distinguish between cases where the *Right key* is the first rank key. Conversely, the *Ratio* indicates how well the *Right key* can be analyzed as the first rank key. Therefore, we choose *Ratio* as the CPA performance indicator in this paper.

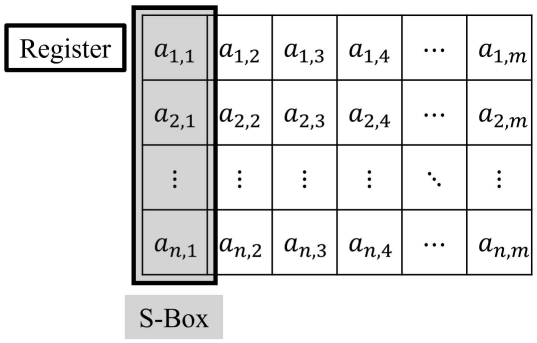


FIGURE 1. Structure of lookup table implementation.

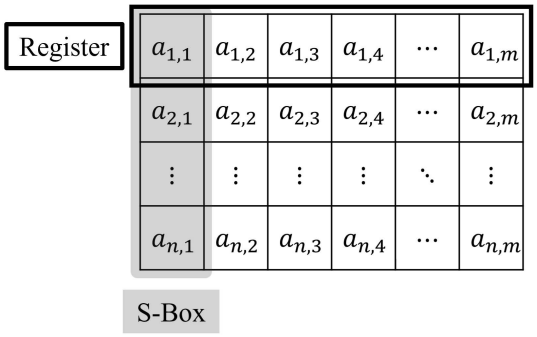


FIGURE 2. Structure of bitslice implementation.

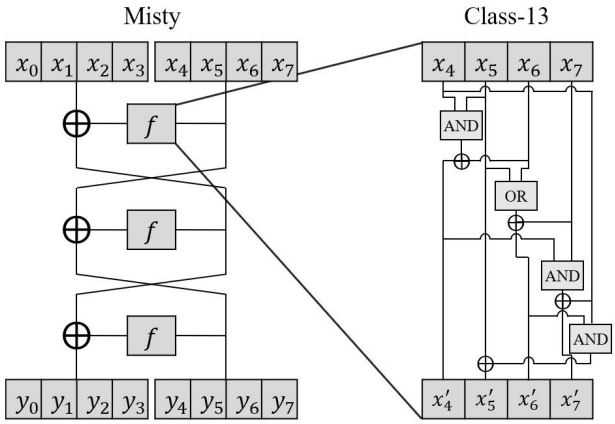


FIGURE 3. Structure of Robin's S-box.

Algorithm 1 **Robin**'s S-Box: Class-13 Algorithm

Input: x_4, x_5, x_6, x_7 (x_7 is MSB)
Output: x'_4, x'_5, x'_6, x'_7

1: $x'_4 = (x_4 \wedge x_5) \oplus x_6$
2: $x'_6 = (x_5 \vee x_6) \oplus x_7$
3: $x'_7 = (x'_4 \wedge x_7) \oplus x_4$
4: $x'_5 = (x'_6 \wedge x_4) \oplus x_5$
5: **Return** x'_4, x'_5, x'_6, x'_7

C. BITSLICE IMPLEMENTATION OF CIPHERS

The basic idea behind bitslicing is expressing a function using only bitwise logical operators, *e.g.*, the AND, XOR, OR, and NOT operators, to perform multiple instances of the function in parallel. Therefore, S-Box operations are parallelly operated in bitslice implementation. Conversely, implementation techniques such as the lookup table perform serial operations in units of S-Box input/output. Figure 1 and Figure 2 depict the register usage structure of the lookup table implementations and bitslice implementations, respectively. In the lookup table implementation, an n-bit input/output value of the S-Box is stored in a single register, whereas each bit of the n-bit input/output value of the S-Box is stored in a different register in bitslice implementation.

An example of bitslice block ciphers is **Robin**, which uses an 8-bit S-Box. Figure 3 is the S-Box structure of **Robin**, where x_7 is the most significant bit (MSB), and x_0 is the least significant bit (LSB). Algorithm 1 shows the pseudocode for the Class-13 function of **Robin**'s S-Box. In this S-Box structure, $n \times m$ bits S-Box outputs can be calculated in parallel because it only comprises the bitwise operations.

D. CPA ON BITSLICE BLOCK CIPHER

Before CPA, we should determine the hypothetical power consumption using the intermediate value and power consumption model. Generally, to improve CPA performance, the output of a nonlinear function, such as S-Box, is preferred as an intermediate value, and the HW power consumption model is mainly employed in a software implementation. However, in the bitslice implementation (Figure 2), it is necessary to guess the $n \times m$ bits key to calculate the hypothetical power consumption of the register's value after the S-Box operation.

Because of the key guessing complexity, Adomnicai *et al.* [11] performed CPA for a bitslice block cipher **PRIDE** using the key XOR output as the intermediate value. In this case, the hypothetical power consumption of the register can be calculated by guessing the m-bit key. However, because of the feature of key XOR, which is a linear operation, the difference in the PCC between the *Right key* and other *Guessed key*s is not large, so the CPA performance decreases, and even specific *Guessed key* that has the same PCC to the *Right key* exists.

Conversely, Balasch *et al.* [7] performed CPA using the single-bit of S-Box output as the intermediate value for the bitslice implementation of **AES**. They used only the single-bit hypothetical power consumption of the register value; however, because the nonlinear function's output was used as the intermediate value, the CPA performance was relatively guaranteed, and the hypothetical power consumption can be calculated by guessing the n-bit key. Thus, we focus on a single-bit of the S-Box output as the intermediate values in CPA.

E. CHALLENGE

Related work shows that effective CPA can be performed using a single-bit of the S-Box output as an intermediate value

for bitslice block ciphers. However, there is no study about the difference in the CPA performance for each single-bit of the S-Box output and the selection of a single-bit that is suitable for CPA as an intermediate value. Thus, we intend to propose an algorithm that receives an S-Box structure as an input and returns each single-bit CPA predictive performance as an output.

III. EACH SINGLE-BIT CPA PERFORMANCE PREDICTION ALGORITHM

In this section, we propose a method that predicts CPA performance and describe a mathematical basis of this method. Further, through the method, we propose an algorithm that predicts the CPA performance of each single-bit of a target cipher's S-Box.

A. CPA PERFORMANCE PREDICTION METHOD

As mentioned in Section II, we choose *Ratio* as the indicator of CPA performance. In other words, we predict the CPA performance by predicting the *Ratio* value. For simplicity, let $\mathbb{B} = \mathbb{H}_{rk}, \mathbb{C} = \mathbb{H}_{sk}$, and $\mathbb{C}_d = \mathbb{H}_{rk\oplus d}$. The CPA process calculates $|r_{\mathbb{A},\mathbb{C}_d}|$ values about every d, $(0 \le d \le 2^n - 1)$, and *Ratio* value is defined as $|r_{\mathbb{A},\mathbb{B}}|/|r_{\mathbb{A},\mathbb{C}}|$. Consequently, we propose a method that predicts $|r_{\mathbb{A},\mathbb{B}}|/|r_{\mathbb{A},\mathbb{C}}|$ without calculating $|r_{\mathbb{A},\mathbb{B}}|, |r_{\mathbb{A},\mathbb{C}}|$.

Proposition 1 suggests that it is possible to predict *Ratio* using $r_{\mathbb{B},\mathbb{C}}$ without the CPA results $r_{\mathbb{A},\mathbb{B}}$ and $r_{\mathbb{A},\mathbb{C}}$. To prove **Proposition** 1, we assume the following.

1) The power consumption model is linear [12], [13], which allows $\mathbb{A} = \{\epsilon b + \delta : b \in \mathbb{B}\}$, where ϵ is a constant and δ is noise random variable.
2) Random variable δ is independent of $\mathbb{B}$ and $\mathbb{C}$ because the noise is unrelated to the process of predicting hypothetical power consumptions $\mathbb{B}$ and $\mathbb{C}$.

Since δ is independent of $\mathbb{B}$ and $\mathbb{C}$, $Cov(\delta, \mathbb{B}) = Cov(\delta, \mathbb{C}) = 0$. The relational formula between $r_{\mathbb{A},\mathbb{B}}$, $r_{\mathbb{B},\mathbb{C}}$ and $r_{\mathbb{A},\mathbb{C}}$ is as expressed as follows.

Proposition 1: Let $\mathbb{A}, \mathbb{B}$ and $\mathbb{C}$ be random variables and $\mathbb{A} = \epsilon\mathbb{B} + \delta$ (where ϵ is a constant and δ is a random variable of noise). Let δ be independent of $\mathbb{B}$ and $\mathbb{C}$. Then $\left|\frac{r_{\mathbb{A},\mathbb{B}}}{r_{\mathbb{A},\mathbb{C}}}\right| = \left|\frac{1}{r_{\mathbb{B},\mathbb{C}}}\right|$.

Proof of Proposition 1:

$\mathbb{A}, \mathbb{B}, \mathbb{C}, \delta, X, Y, Z$: Random variables

a, b, c, ϵ : Constants

Theorem 1) $Cov(aX + bY, cX) = ac(\sigma_X)^2 + bcCov(X, Y)$

Theorem 2) $Cov(aX + bY, cZ) = acCov(X, Z) + bcCov(Y, Z)$

$$\left|\frac{r_{\mathbb{A},\mathbb{B}}}{r_{\mathbb{A},\mathbb{C}}}\right| = \left|\frac{\frac{Cov(\mathbb{A},\mathbb{B})}{\sigma_{\mathbb{A}}\sigma_{\mathbb{B}}}}{\frac{Cov(\mathbb{A},\mathbb{C})}{\sigma_{\mathbb{A}}\sigma_{\mathbb{C}}}}\right| = \left|\frac{\sigma_{\mathbb{C}}\, Cov(\mathbb{A},\mathbb{B})}{\sigma_{\mathbb{B}}\, Cov(\mathbb{A},\mathbb{C})}\right|$$

$$= \left|\frac{\sigma_{\mathbb{C}}\, Cov(\epsilon\mathbb{B}+\delta,\mathbb{B})}{\sigma_{\mathbb{B}}\, Cov(\epsilon\mathbb{B}+\delta,\mathbb{C})}\right| (\because \mathbb{A} = \epsilon\,\mathbb{B} + \delta)$$

$$= \left|\frac{\sigma_{\mathbb{C}}\left(\epsilon(\sigma_{\mathbb{B}})^2 + Cov(\delta,\mathbb{B})\right)}{\sigma_{\mathbb{B}}\, Cov(\epsilon\mathbb{B}+\delta,\mathbb{C})}\right| \text{(by Theorem 1)}$$

$$= \left|\frac{\sigma_{\mathbb{C}}\left(\epsilon(\sigma_{\mathbb{B}})^2 + Cov(\delta,\mathbb{B})\right)}{\sigma_{\mathbb{B}}\left(\epsilon Cov(\mathbb{B},\mathbb{C}) + Cov(\delta,\mathbb{C})\right)}\right| \text{(by Theorem 2)}$$

$$= \left|\frac{\sigma_{\mathbb{C}}\,\epsilon(\sigma_{\mathbb{B}})^2}{\sigma_{\mathbb{B}}\,\epsilon\, Cov(\mathbb{B},\mathbb{C})}\right| (\because Cov(\delta,\mathbb{B}) = Cov(\delta,\mathbb{C}) = 0)$$

$$= \left|\frac{\sigma_{\mathbb{C}}\,\sigma_{\mathbb{B}}}{Cov(\mathbb{B},\mathbb{C})}\right| = \left|\frac{1}{r_{\mathbb{B},\mathbb{C}}}\right|$$

□

Proposition 1 shows that the *Ratio* value $|r_{\mathbb{A},\mathbb{B}}/r_{\mathbb{A},\mathbb{C}}|$ can be calculated as $|1/r_{\mathbb{B},\mathbb{C}}|$ by some assumptions. Using **Proposition** 1, we can predict the *Ratio* value using the hypothetical power consumptions $\mathbb{B}$, $\mathbb{C}$ without the actual power consumptions $\mathbb{A}$. However, there is a limitation that we cannot know which of $\mathbb{C}_d$ is $\mathbb{B}$ or $\mathbb{C}$ without actual CPA, and we cannot calculate $|r_{\mathbb{B},\mathbb{C}}|$ without the actual plaintexts. Therefore, we transform $|r_{\mathbb{B},\mathbb{C}}|$ into simulation value $|r_{\mathbb{B}',\mathbb{C}'}|$.

First, we assume that actual plaintexts are determined by the uniform distribution. Then we can define alternative sequences $\hat{\mathbb{B}}$ and $\hat{\mathbb{C}}_d$ of hypothetical power consumption sequences $\mathbb{B}$ and $\mathbb{C}$ about the actual plaintexts. For example, if the intermediate value is the n-bit S-Box output, and the power consumption model is HW, then since possible plaintext is one of 0 to $2^n - 1$, the alternative sequences $\hat{\mathbb{B}}$ and $\hat{\mathbb{C}}_d$ are defined as follows:

$$\hat{\mathbb{B}} := \{HW(S(0 \oplus rk)),\ HW(S(1 \oplus rk)),\ \ldots,\ HW(S(2^n - 1 \oplus rk))\}$$
$$\hat{\mathbb{C}}_d := \{HW(S(0 \oplus rk \oplus d)),\ HW(S(1 \oplus rk \oplus d)),\ \ldots,\ HW(S(2^n - 1 \oplus rk \oplus d))\}$$

Since $\hat{\mathbb{B}}$, $\hat{\mathbb{C}}_d$ have elements of each $\mathbb{B}$, $\mathbb{C}_d$ by the uniform distribution, we expect that $r_{\mathbb{B},\mathbb{C}_d} \approx r_{\hat{\mathbb{B}},\hat{\mathbb{C}}_d}$. Additionally, by rearranging sequences $\hat{\mathbb{B}}$ and $\hat{\mathbb{C}}_d$ based on rk can be simplified as follows:

$$\mathbb{B}' := \{HW(S(0)),\ HW(S(1)),\ \ldots,\ HW(S(2^n - 1))\}$$
$$\mathbb{C}'_d := \{HW(S(0 \oplus d)),\ HW(S(1 \oplus d)),\ \ldots,\ HW(S(2^n - 1 \oplus d))\}$$

Since both $\hat{\mathbb{B}}$ and $\hat{\mathbb{C}}_d$ are rearranged on the same basis, we have $r_{\hat{\mathbb{B}},\hat{\mathbb{C}}_d} = r_{\mathbb{B}',\mathbb{C}'_d}$. We can calculate $r_{\mathbb{B}',\mathbb{C}'_d}$ about every d $(1 \le d \le 2^n - 1)$ using only the target cipher's structure (in the case of the example, the S-Box). Next, we define that $\mathbb{C}'$ is the $\mathbb{C}_d$ that has the highest value of $|r_{\mathbb{B}',\mathbb{C}'_d}|$. Consequently, $r_{\mathbb{B}',\mathbb{C}'}$ has the following relationship with the *Ratio* value.

$$Ratio = \left|\frac{r_{\mathbb{A},\mathbb{B}}}{r_{\mathbb{A},\mathbb{C}}}\right| = \left|\frac{1}{r_{\mathbb{B},\mathbb{C}}}\right| \approx \left|\frac{1}{r_{\hat{\mathbb{B}},\hat{\mathbb{C}}}}\right| = \left|\frac{1}{r_{\mathbb{B}',\mathbb{C}'}}\right|$$

By calculating $\left|1/r_{\mathbb{B}',\mathbb{C}'}\right|$, we can predict the *Ratio* value without hypothetical power consumption $\mathbb{B}$ and $\mathbb{C}$ about actual plaintexts and actual power consumption $\mathbb{A}$.

B. EACH SINGLE-BIT CPA PERFORMANCE PREDICTION ALGORITHM

In this section, we propose an algorithm that predicts each single-bit CPA performance on an n-bit S-Box using the proposed CPA performance prediction method. First, we define $\mathbb{B}'$ and $\mathbb{C}'_d$ for a single-bit of S-Box output and get the $r_{\mathbb{B}',\mathbb{C}'}$ value by calculating every $r_{\mathbb{B}',\mathbb{C}'_d}$. Then, the performance of this single-bit is predicted by $|1/r_{\mathbb{B}',\mathbb{C}'}|$. This process is repeated to predict the CPA performance for every single-bit. Assuming that the hypothetical power consumption is the *i-th* bit of the n-bit S-Box output, $\mathbb{B}'$ and $\mathbb{C}'_d$ of the *i-th* bit are defined as $\mathbb{S}_i$, $\mathbb{S}_i(d)$, respectively, as follows:

$$\mathbb{S}_i := \{s_i(0),\ s_i(1),\ \ldots,\ s_i(2^n-1)\}$$
$$\mathbb{S}_i(d) := \{s_i(0\oplus d),\ s_i(1\oplus d),\ \ldots,\ s_i((2^n-1)\oplus d)\},\quad (0<d<2^n).$$

Algorithm 2 is an algorithm that receives an n-bit S-Box as an input and returns each single-bit CPA performance as an output. In Algorithm 2, c_i^d in line 3 means $r_{\mathbb{B}',\mathbb{C}'_d}$ of the *i-th* bit, *tmp* in line 5 means $r_{\mathbb{B}',\mathbb{C}'}$, and R_i in line 6 means $|1/r_{\mathbb{B}',\mathbb{C}'}|$ the CPA performance prediction value of the *i-th* bit. Algorithm 2 repeats this process for all i and returns the CPA performance predictive values of each *i-th* bit.

Algorithm 2 Each Single-bit CPA Performance Prediction Algorithm on n-bit S-Box

Input: An n-bit S-Box
Output: Each single-bit CPA performance R_i

1: **for** $i = 0$ to $n-1$ **do**
2: **for** $d = 1$ to 2^n-1 **do**
3: $c_i^d \leftarrow \left|r_{\mathbb{S}_i,\mathbb{S}_i(d)}\right|$
4: **end for**
5: $tmp \leftarrow \max(c_i^1, c_i^2, \ldots, c_i^{2^n-1})$
6: $R_i \leftarrow 1/tmp$
7: **end for**
8: **Return** $(R_0, R_1, \ldots, R_{n-1})$

IV. EXPERIMENTAL RESULTS FOR BITSLICE BLOCK CIPHER ROBIN

In this section, we discuss the difference in CPA performance caused by a single-bit intermediate value through experiments on Robin and verify the effectiveness of the proposed algorithm.

Table 2 describes the experimental environment. As the target chip, we chose the Atmel XMEGA128, an 8-bit low power microcontroller unit (MCU). Additionally, using the ChipWhisperer-Lite board, we collected power traces and sampled the S-Box operation part of the first round while Robin encryption was performed on a total of 2,000 random plaintexts.

TABLE 2. Experimental environment.

Target board	ChipWhisperer-Lite [14]
Target Chip	Atmel XMEGA128 (8-bit MCU)
Number of traces	2,000
Sampling rate	29,538 MS/s
Target cipher	Robin
Analysis point	1 round S-Box
Power model	Single-bit

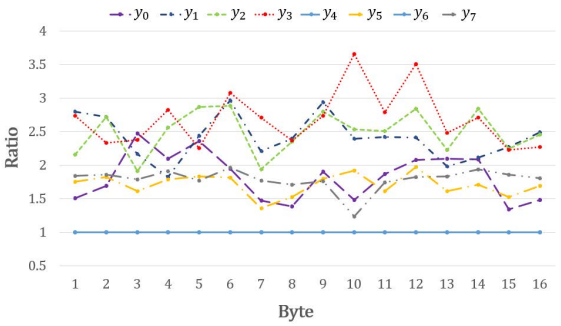


FIGURE 4. ***Ratio*** **of Robin CPA by each single-bit.**

A. COMPARING BETWEEN PREDICTION OF CPA PERFORMANCE AND ACTUAL CPA PERFORMANCE ABOUT EACH SINGLE-BIT INTERMEDIATE VALUE

We performed a 16 byte CPA by selecting each single-bit of the S-Box output as the intermediate value. Figure 4 shows that the *Ratio* of Robin CPA result differed depending on which single-bit is selected. We subsequently applied Algorithm 2 to Robin's S-Box and compared the results shown in Figure 4 and the predictive performance. First, we calculated R_i of Robin using Algorithm 2, and then we compared R_i with the results of actual CPA.

Algorithm 2 returns predictive CPA performance $R_i (0 \le i \le 7)$ shown in Table 3. Assuming that the 8-bit S-Box output of Robin is represented by $Y = (y_7y_6y_5y_4y_3y_2y_1y_0)_2$ (y_0 is the LSB), R_i means the predictive *Ratio* value of CPA that uses the *i-th* bit of S-Box output y_i as the intermediate value. Robin CPA results can be divided into the following three sets depending on the size of R_i.

$$Y_1 = \{y_4,\ y_6\},\quad Y_2 = \{y_0,\ y_5,\ y_7\},\ Y_3 = \{y_1,\ y_2,\ y_3\}$$

1) SELECT INTERMEDIATE VALUE FROM $Y_1 = \{y_4, y_6\}$

Table 3 shows that R_4 is 1.0. Furthermore, in Algorithm 2 about Robin's S-Box, c_4^d has the maximum value when the XOR difference d is 0×04, 0×08, or $0\times0C$. This suggests that if y_4 is selected as the intermediate value, the *Ratio* value will be $R_4 = 1.0$ and the XOR differences between the

TABLE 3. R_i of robin.

i	0	1	2	3	4	5	6	7
R_i	2.0	3.57	4.0	4.0	1.0	2.0	1.0	2.0

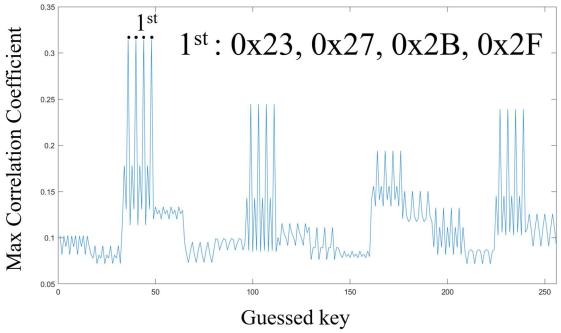


FIGURE 5. Max correlation coefficient by *Guessed key* (intermediate value: y_4).

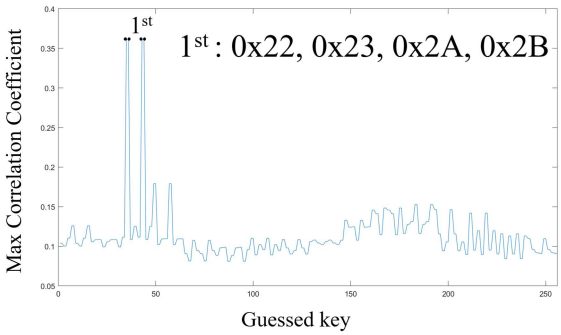


FIGURE 6. Max correlation coefficient by *Guessed key* (intermediate value: y_6).

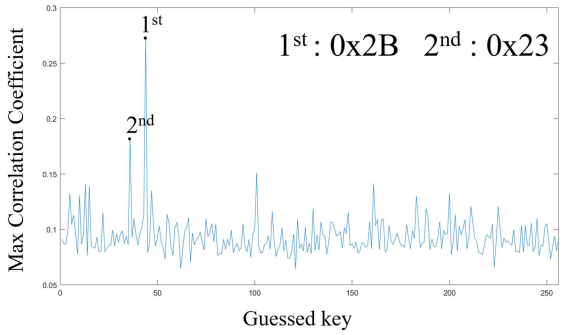


FIGURE 7. Max correlation coefficient by *Guessed key* (intermediate value: y_0).

*Second key*s and the *Right key* will be 0×04, 0×08, and $0 \times 0C$. In other words, if y_4 is selected as the intermediate value, we can predict that the three *Second key*s will have the same absolute value of PCC as the *Right key*; also, the *Second key*s will have XOR differences of 0×04, 0×08, and $0 \times 0C$ from the *Right key*.

Figure 5 shows the result of performing the first-byte CPA with y_4 as the intermediate value. In Figure 5, the max correlation coefficient means the maximum value of the corresponding *Guessed key*'s absolute value of PCC about each sample point. Thus, the *Ratio* value is calculated as the *Right key*'s max correlation coefficient divided by the *Second key*'s max correlation coefficient. Figure 5 shows the max correlation coefficients of each *Guessed key*. In practice, Figure 5 shows that keys $0 \times 2F$ ($= 0 \times 2B \oplus 0 \times 04$), 0×23 ($= 0 \times 2B \oplus 0 \times 08$), and 0×27 ($= 0 \times 2B \oplus 0 \times 0C$) have the same absolute value of PCC to the *Right key* $0 \times 2B$.

Similarly, $R_6 = 1.0$ and in Algorithm 2 about Robin's S-Box, c_6^d has the maximum value when XOR difference d is 0×01, 0×08, or 0×09. Thus, if y_6 is selected as the intermediate value, we can predict that $0 \times 2A$ ($= 0 \times 2B \oplus 0 \times 01$), 0×23 ($= 0 \times 2B \oplus 0 \times 08$), and 0×22 ($= 0 \times 2B \oplus 0 \times 09$) will be the *Second key*s with the same absolute value of PCC to the *Right key* $0 \times 2B$.

Figure 6 shows the result of performing the first-byte CPA with y_6 as the intermediate value. As we predicted, Figure 6 shows that keys 0×22, 0×23, and $0 \times 2A$ have the same absolute value of PCC to the *Right key* $0 \times 2B$.

2) SELECT INTERMEDIATE VALUE FROM $Y_2 = \{y_0, y_5, y_7\}$

Table 3 shows $R_0 = 2.0$. Furthermore, in Algorithm 2 about Robin's S-Box, c_0^d has the maximum value when d is 0×08. Thus, we can predict that the *Ratio* value is close to 2.0 and the XOR difference between the *Second key* and the *Right key* will be 0×08.

Figure 7 shows the result of performing CPA with y_0 as the intermediate value. The absolute value of PCC 0.272 for the *Right key* $0 \times 2B$ is approximately 1.51 times greater than 0.18 for the *Second key* 0×23 ($= 0 \times 2B \oplus 0 \times 08$). Because of the effect of practical noise, the *Ratio* value 1.51 is less than the predictive value $R_0 = 2.0$.

Similarly, R_5 is 2.0 and c_5^d has the maximum value when d is 0×02, 0×04, 0×06, $0 \times 0A$, $0 \times 0C$, or $0 \times 0E$. As shown in Figure 8, the XOR difference between *Second key* 0×29 and *Right key* $0 \times 2B$ is 0×02. Furthermore, the *Ratio* value is approximately 1.75, which is close to $R_5 = 2$.

We observed similar results for y_7. c_7^d has the maximum value when d is 0×01, 0×02, 0×03, 0×05, 0×06 and 0×07. Figure 9 shows that the XOR difference between the *Second key* 0×29 and *Right key* $0 \times 2B$ is 0×02. In Figure 9, the *Ratio* value is approximately 1.84, which is close to $R_7 = 2$.

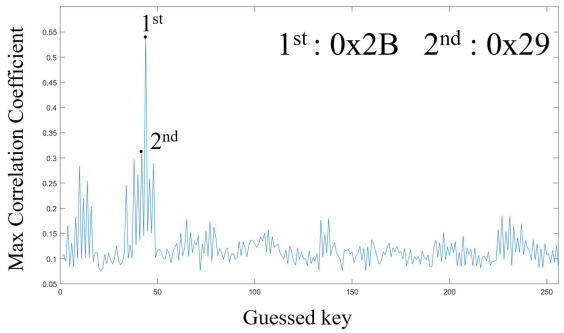


FIGURE 8. Max correlation coefficient by *Guessed key* (intermediate value: y_5).

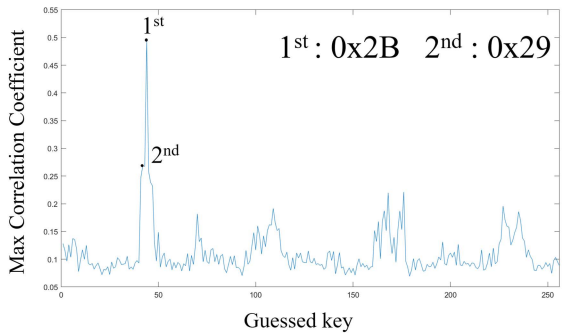


FIGURE 9. Max correlation coefficient by *Guessed key* (intermediate value: y_7).

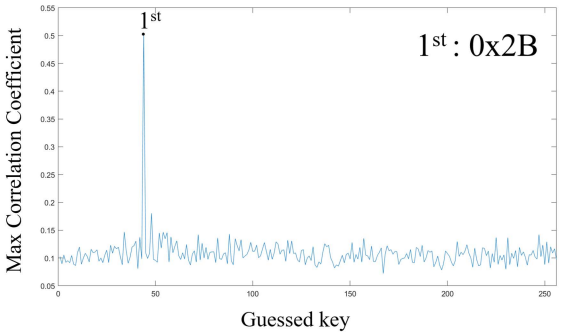


FIGURE 10. Max correlation coefficient by *Guessed key* (intermediate value: y_1).

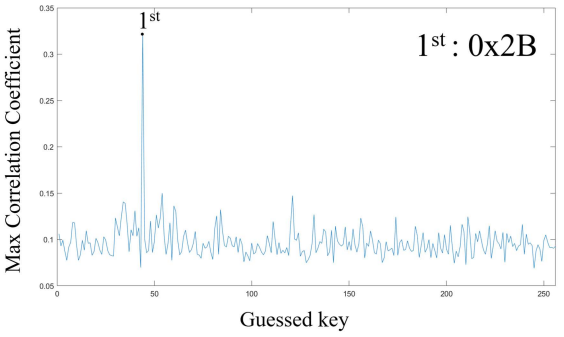


FIGURE 11. Max correlation coefficient by *Guessed key* (intermediate value: y_2).

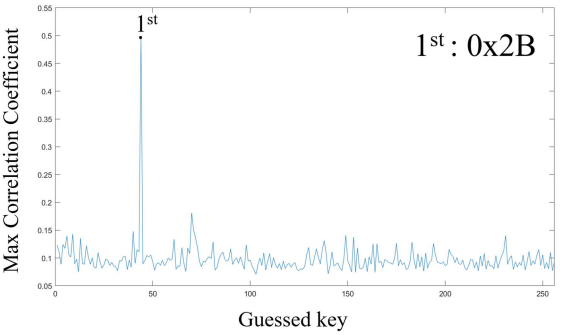


FIGURE 12. Max correlation coefficient by *Guessed key* (intermediate value: y_3).

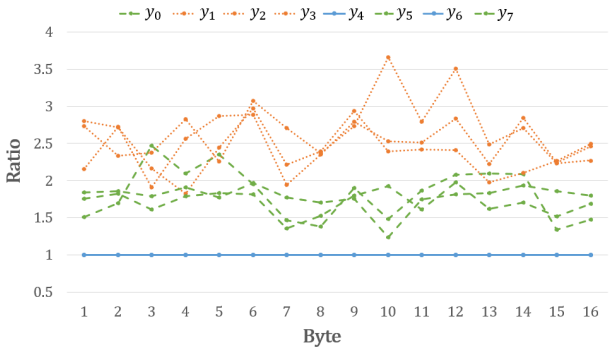


FIGURE 13. CPA results for Robin (Y_1, Y_2, Y_3).

3) SELECT INTERMEDIATE VALUE FROM $Y_3 = \{y_1, y_2, y_3\}$

For Y_3, Table 3 shows that R_1 is 3.57, and R_2 and R_3 are both 4.0.

Figure 10 shows that when y_1 is selected as the intermediate value, the *Ratio* value is approximately 2.80. Figure 11 shows that when y_2 is selected, the *Ratio* value is approximately 2.15. Additionally, Figure 12 shows that when y_3 is selected, the *Ratio* is approximately 2.73.

In the case of Y_3, because the *Guessed key*s, except for the *Right key*, have absolute value of PCC of a similar level, we do not predict which key is the *Second key*.

Consequently, as shown in Figure 10, Figure 11, and Figure 12, Y_3 has the larger *Ratio* value than Y_1 and Y_2. Therefore, selecting $y_1, y_2, y_3 (\in Y_3)$ as the intermediate values improves CPA performance over other single-bits.

B. RESULT OF TOTAL BYTES

Figure 13 shows the 16 byte CPA results for Robin when the bits are grouped according to Y_1, Y_2, and Y_3; blue, green, and orange graphs are for the Y_1, Y_2, and Y_3 groups, respectively. Thus, the CPA results for Robin show that CPA performance improves when Y_3 with relatively high R_i values is selected as the intermediate value.

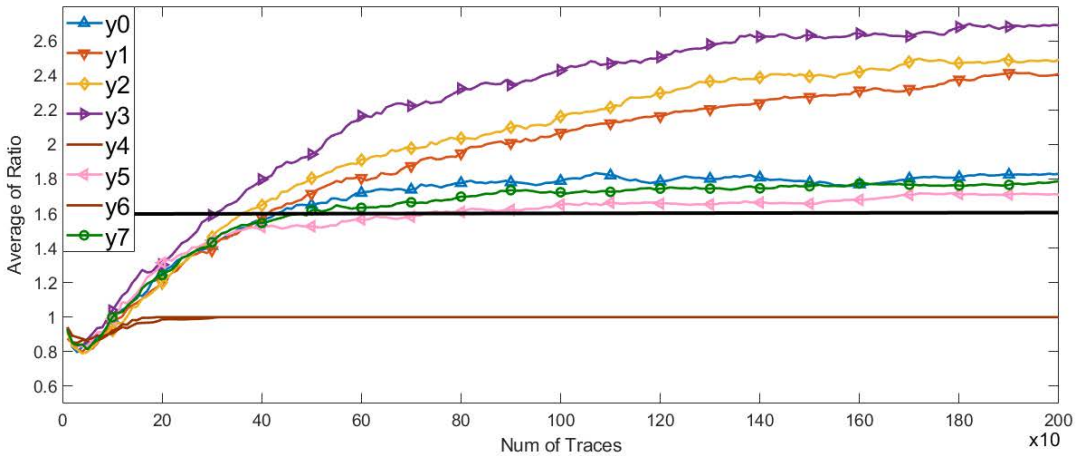


FIGURE 14. *Ratio* by bit according to the number of traces, the CPA results for Robin.

TABLE 4. *Ratio* of robin.

i	0	1	2	3	4	5	6	7
$Ratio$ of y_i	1.83	2.41	2.49	2.69	1.0	1.71	1.0	1.79

TABLE 5. R_i of other ciphers.

Cipher	i	0	1	2	3	4	5	6	7
AES	R_i	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0
Fantomas	R_i	2.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0
RECTANGLE	R_i	1.0	1.0	2.0	2.0	-	-	-	-
RoadRunneR	R_i	1.0	2.0	1.0	2.0	-	-	-	-
PRIDE	R_i	2.0	2.0	1.0	1.0	-	-	-	-
CRAFT	R_i	2.0	1.0	2.0	2.0	-	-	-	-

Figure 14 shows a graph of the average *Ratio* for each intermediate value y_i, the *i-th* bit of S-Box output, based on the number of traces when CPA is performed for Robin. In this paper, the average *Ratio* means the average value of *Ratio* on each key byte (or nibble if S-Box size is 4-bit) CPA of the first round key. The black line in Figure 14 indicates that the *Ratio* value is 1.6. At 2,000 traces, Y_1 does not satisfy the *Ratio* value of 1.6 or greater. By contrast, when selecting the bit from Y_3, the minimum number of attack traces satisfying the *Ratio* value of 1.6 or greater is approximately 360, and it is 510 when selecting the bit from Y_2. Consequently, when selecting bits from Y_3, the analysis performance is 1.41 times better than when selecting bits from Y_2.

Table 4 summarizes the *Ratio* results of each single-bit CPA using 2,000 traces for Robin.

V. EXPERIMENTAL RESULTS FOR OTHER CIPHERS

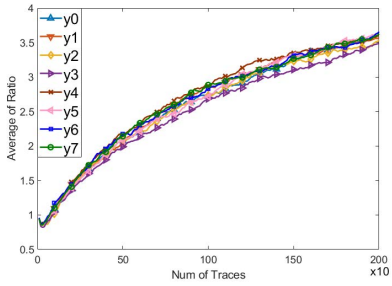
In this section, we present experimental CPA results for other ciphers implemented by bitslice structures.

AES is the most representative block cipher that uses an 8-bit S-Box. We chose the target by the bitslice implementation of AES. Fantomas is a Robin family bitslice block cipher that uses an 8-bit S-Box. RECTANGLE, RoadRunneR, PRIDE, and CRAFT are block ciphers that use a 4-bit S-Box. We performed the CPA for the above block ciphers.

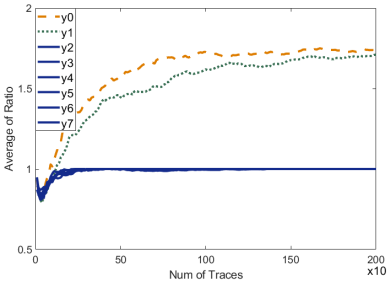
Table 5 shows the R_i values of each block cipher's S-Box. Figure 15 shows the average *Ratio* based on the number of traces for each block cipher. The experimental environment is the same as in Section IV.

The predictive value for AES in Table 5 shows that the R_i values are all equally high values of 8.0 because the AES's S-Box is designed as a mathematical structure to perfectly satisfy nonlinearity. As shown in Figure 15 (a), on AES CPA, every single-bit analyzed has an almost equal *Ratio*. This result is the same as predicted in Table 5, and AES has a higher *Ratio* than other ciphers because of higher R_i values.

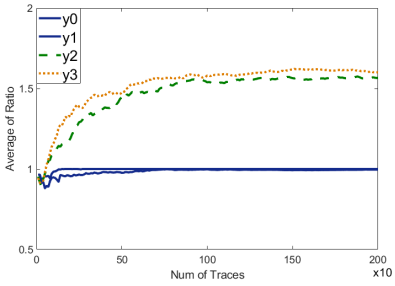
Table 5 shows that the R_i values for Fantomas, RECTANGLE, RoadRunneR, PRIDE, and CRAFT are 2.0 or 1.0. According to the prediction result of Fantomas, R_i has



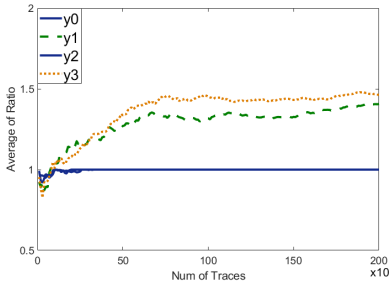
(a) The CPA results for bitslice implementation AES



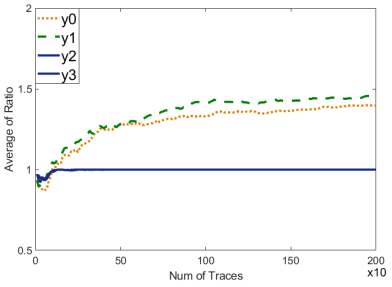
(b) The CPA results for Fantomas



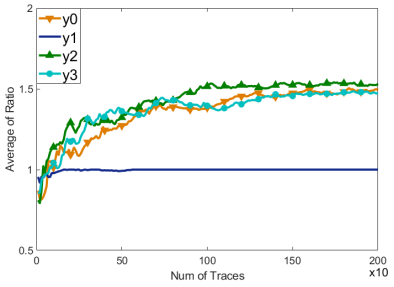
(c) The CPA results for RECTANGLE



(d) The CPA results for RoadRunneR



(e) The CPA results for PRIDE



(f) The CPA results for CRAFT

FIGURE 15. ***Ratio*** **by bit according to the number of traces, the CPA results against bitslice implementation ciphers.**

TABLE 6. ***Ratio*** **of other ciphers.**

Cipher	i	0	1	2	3	4	5	6	7
AES	*Ratio* of y_i	3.62	3.57	3.53	3.48	3.59	3.62	3.65	3.62
Fantomas	*Ratio* of y_i	1.74	1.71	1.0	1.0	1.0	1.0	1.0	1.0
RECTANGLE	*Ratio* of y_i	0.99	1.0	1.57	1.60	-	-	-	-
RoadRunneR	*Ratio* of y_i	1.0	1.41	1.0	1.46	-	-	-	-
PRIDE	*Ratio* of y_i	1.39	1.46	1.0	1.0	-	-	-	-
CRAFT	*Ratio* of y_i	1.49	1.0	1.53	1.47	-	-	-	-

the maximum value of 2.0 when i is 0 or 1; thus, these bits are expected to have the best performance in CPA. Figure 15 (b) shows that at 2,000 traces, y_0 and y_1 have a *Ratio* value of approximately 1.6, whereas other bits have a *Ratio* value of 1.0. Similarly, for RECTANGLE, R_i has the maximum value of 2.0 when i is 2 or 3. In the actual CPA results (Figure 15 (c)) at 2,000 traces, y_2 and y_3 have a *Ratio* value of approximately 1.5, and the other bits have a *Ratio* value of 1.0. For RoadRunneR, PRIDE, and CRAFT, the prediction results in Table 5 match the actual CPA results in Figure 15. Notably, in RoadRunneR, y_1 and y_3 have higher *Ratio* values than other bits. In PRIDE, y_0 and y_1 have higher *Ratio* values than other bits. Finally, in CRAFT, y_0, y_2, and y_3 have higher *Ratio* values than y_1.

Table 6 summarizes the *Ratio* results of each single-bit CPA using 2,000 traces for block ciphers. These results demonstrated that the proposed Algorithm 2 correctly predicted the CPA results for each block cipher, *i.e.*, AES, Fantomas, RECTANGLE, RoadRunneR, PRIDE, and CRAFT.

VI. CONCLUSION

In this paper, we propose a method to predict the CPA performance. Additionally, using the method, we propose an algorithm to predict the CPA performance of each single-bit using a target bitslice block cipher's S-Box. Applying the proposed algorithm to Robin, Fantomas, AES, RECTANGLE, RoadRunneR, PRIDE, and CRAFT provided experimental confirmation of its efficacy. The proposed algorithm can assist systematic and improved CPA for bitslice block ciphers, and through this, it also can be utilized in the CPA security verification model for bitslice block ciphers.

Additionally, our findings can be extended to higher-order CPA for bitslice block ciphers.

REFERENCES

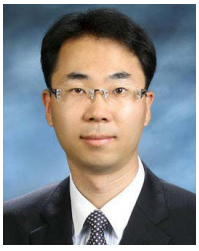
[1] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Advances in Cryptology*. Santa Barbara, CA, USA, Aug. 1996, pp. 104–113.
[2] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1666, M. J. Wiener, Ed. Santa Barbara, CA, USA: Springer, 1999, pp. 388–397.
[3] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems*. Cambridge, MA, USA, Aug. 2004, pp. 16–29.
[4] V. Grosso, G. Leurent, F. Standaert, and K. Varici, "LS-designs: Bitslice encryption for efficient masked software implementations," in *Fast Software Encryption* (Lecture Notes in Computer Science), vol. 8540, C. Cid and C. Rechberger, Eds. London, U.K.: Springer, Mar. 2014, pp. 18–37.
[5] A. Baysal and S. Sahin, "Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors," in *Lightweight Cryptography for Security and Privacy* (Lecture Notes in Computer Science), vol. 9542, T. Güneysu, G. Leander, and A. Moradi, Eds. Bochum, Germany: Springer, Sep. 2015, pp. 58–76.
[6] C. Beierle, G. Leander, A. Moradi, and S. Rasoolzadeh, "CRAFT: Lightweight tweakable block cipher with efficient protection against DFA Attacks," *IACR Trans. Symmetric Cryptol.*, vol. 2019, no. 1, pp. 5–45, 2019.
[7] J. Balasch, B. Gierlichs, O. Reparaz, and I. Verbauwhede, "DPA, bitslicing and masking at 1 GHz," in *Cryptographic Hardware and Embedded Systems* (Lecture Notes in Computer Science), vol. 9293, T. Güneysu and H. Handschuh, Eds. Saint-Malo, France: Springer, Sep. 2015, pp. 599–619. 2015.
[8] *Advanced Encryption Standard*, NIST FIPS PUB 197, NIST, Gaithersburg, MD, USA, 2001.
[9] W. T. Zhang, Z. Z. Bao, D. D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, "RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms," *Sci. China Inf. Sci.*, vol. 58, no. 12, pp. 1–15, Dec. 2015.
[10] M. R. Albrecht, B. Driessen, E. B. Kavun, G. Leander, C. Paar, and T. Yalçin, "Block ciphers—Focus on the linear layer (feat. PRIDE)," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 8616, J. A. Garay and R. Gennaro, Eds. Santa Barbara, CA, USA: Springer, Aug. 2014, pp. 57–76.
[11] A. Adomnicai, B. Lac, A. Canteaut, J. Fournier, L. Masson, R. Sirdey, and A. Tria, "On the importance of considering physical attacks when implementing lightweight cryptography," Tech. Rep., 2016.
[12] M. Akkar, R. Bevan, P. Dischamp, and D. Moyart, "Power analysis, what is now possible," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1976, T. Okamoto, Ed. Kyoto, Japan: Springer, Dec. 2000, pp. 489–502.
[13] J. Doget, E. Prouff, M. Rivain, and F. Standaert, "Univariate side channel attacks and leakage modeling," *IACR Cryptol. ePrint Arch.*, vol. 2011, p. 302, 2011.
[14] NewAE Techonology. *ChipWhisperer-Lite (CW1173) Two-Part Version*. [Online]. Available: https://rtfm.newae.com/Capture/ChipWhisperer-Lite/

JAESEUNG HAN received the M.S. degree in financial information security from Kookmin University, Seoul, Republic of Korea, in 2022, where he is currently pursuing the Ph.D. degree in financial information security. His research interests include side-channel attacks, symmetric key cryptography, and lattice-based cryptography.

YEON-JAE KIM received the B.S. degree in information security, cryptology, and mathematics from Kookmin University, Seoul, Republic of Korea, in 2021, where she is currently pursuing the master's degree in financial information security. Her research interests include side-channel attacks and public key cryptography.

SOO-JIN KIM received the B.S. degree in information security, cryptology, and mathematics from Kookmin University, Seoul, Republic of Korea, in 2021, where she is currently pursuing the master's degree in financial information security. Her research interests include side-channel attacks, symmetric key cryptography, and post quantum cryptography.

BO-YEON SIM received the Ph.D. degree in information security from Kookmin University, Seoul, Republic of Korea, in 2020. She worked as a Research Professor at Kookmin University, in 2020. She is currently working as a Researcher at the Electronics and Telecommunications Research Institute (ETRI). Her research interests include side-channel attacks, cryptography, reverse engineering, and implementation of information protection technology for embedded systems.



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