applied sciences

MDPI

Article

SIV: Raise the Correlation of Second-Order Correlation Power Analysis to 1.00

Ju-Hwan Kim [1,†], Bo-Yeon Sim [1,†] and Dong-Guk Han [1,2,*,†]

[1] Department of Mathematics, Kookmin University, Seoul 02707, Korea; zzzz2605@kookmin.ac.kr (J.-H.K.); qjdusls@kookmin.ac.kr (B.-Y.S.)
[2] Department of Financial Information Security, Kookmin University, Seoul 02707, Korea
* Correspondence: christa@kookmin.ac.kr; Tel.: +82-02-910-4744
† These authors contributed equally to this work.

Received: 9 April 2020; Accepted: 10 May 2020; Published: 14 May 2020

Abstract: The major factors that determine the performance of the second-order correlation power analysis (SOCPA) include the accuracy of the power model and the correlation between the hypothetical intermediate value and preprocessed power consumption. Because of the tradeoff between the accuracy and correlation, the correlation coefficient of the general SOCPA using 8-bit SubBytes output is only up to 0.35. Therefore, based on the operational characteristic of the cryptographic algorithm, we propose to find a special intermediate value, called sparse intermediate value (SIV). The SIV significantly improves the performance of the SOCPA because it accurately models the power consumption while the correlation coefficient is 1.00. Further, the experimental results on OpenSSL advanced encryption standard (AES) show that the SIV-based SOCPA can disclose the entire secret key with only about a quarter of the power trace required by the general SOCPA.

Keywords: side-channel analysis; correlation power analysis; second-order correlation power analysis; OpenSSL; AES

1. Introduction

Cryptanalysis is the study of the analysis of the cryptographic algorithm's vulnerabilities to construct a secure system. Mathematical analysis, conventional cryptanalysis, reveals secret information based on the fact that an analyst knows plaintext or ciphertext. Meanwhile, Paul Kocher discovered that physical information of a cryptographic device is associated with the secret information [1]. Although a cryptographic algorithm could be secure against mathematical analysis, it is subjected to vulnerability using physical information. Consequently, cryptanalysis research is required under the assumption that an analyst is aware of physical information in addition to plaintext/ciphertext. Side-channel analysis discloses a secret key by using physical information such as power consumption, electromagnetic, acoustic, and photon [2–5].

Power analysis, which analyzes power consumption patterns of a cryptographic device, includes the simple power analysis (SPA) [1], differential power analysis (DPA) [2], and correlation power analysis (CPA) [6]. The DPA/CPA is based on the fact that power consumption when storing data in a register is related to the data. The important factor, which determines the DPA/CPA performance, is the accuracy of the power model that describes the relationship. To raise the model's accuracy, the analyst considers every single bit of the register: i.e., the length of intermediate value should be as long as the length of the register.

The first-order correlation power analysis (FOCPA) is a statistical method that utilizes the correlation between a single point of power consumption and the sensitive intermediate value. Thus, the countermeasure such as masking is generally used [7]. To counter the FOCPA, the first-order

masking, which splits an intermediate value into two random variables, is utilized. That is, it spreads the power consumption related to the intermediate value into two points. The first-order masked implementation is vulnerable to the second-order correlation power analysis (SOCPA) that utilizes the correlation between guessable intermediate value and two split points of power consumption. Two points of power consumption are preprocessed to related to the guessable intermediate value, and the preprocessing function determines the SOCPA performance.

For the SOCPA, the longer bit length of the intermediate value reduces the correlation between the hypothetical intermediate value and preprocessed power consumption. That is, 1-bit intermediate value has the highest correlation with the preprocessed power consumption when performing the SOCPA. However, 1-bit intermediate is generally not used because shorter bit length reduces the accuracy of the power model. In this tradeoff relation, analysts commonly use an 8-bit intermediate value to improve the performance of the SOCPA.

SOCPA requires more power traces than the FOCPA to determining whether the guessed key is the right key because the correlation of the SOCPA is much lower than the FOCPA. Accordingly, several preprocessing functions had been suggested to raise the correlation of the SOCPA. The first function is the product of two points [8]. If the power model is the Hamming weight model and the length of the intermediate value is 8, the absolute correlation coefficient is only about 0.09. In this case, theoretically, the general SOCPA requires at least $\frac{1}{0.09^2} \approx 123$ times more traces than the FOCPA because the correlation is only 0.09 [9]. Thus, the first function is generally not used. In 2000, the absolute-difference (AD) function was proposed by Messerges [10]. However, the correlation was still only 0.24, much lower than 1.00. To raise the correlation, Prouff suggested the product-combining (PC) function in 2009, and the correlation was enhanced to 0.35 [11]. Existing researches are focused on the preprocessing function to enhance SOCPA performance, and the correlation has not raised significantly.

As discussed above, a 1-bit intermediate value is typically not used because it decreases the accuracy of the power model. However, the correlation between the hypothetical intermediate value and preprocessed power consumption is 1.00, if the preprocessing function is AD or PC. Thus, if there exists the 8-bit intermediate value that has the same characteristic as 1-bit, it allows significantly raising the correlation. We focus on the characteristic that the number of cases is 2, for 1-bit intermediate value. Therefore, unlike existing researches, we aim to find the intermediate values that have a smaller cardinality, such as a 1-bit intermediate value. In this paper, we propose the special intermediate value, named sparse intermediate value (SIV), based on the operational characteristics of the cryptographic algorithm, and remarkably raise the correlation from 0.35 to 1.00. That is, we reduce the the minimum trace to disclose the secret key to the same as the FOCPA.

The rest of this paper is organized as follows. Section 2 briefly describes the overview of the CPA and MixColumns of the advanced encryption algorithm (AES). We analyze the operational characteristics to find the SIV and demonstrate the existence of the power consumption related to SIV in Section 3. Section 4 analyzes OpenSSL AES and compares the general SOCPA using 8-bit SubBytes output to SIV-based SOCPA. Section 5 recommends two countermeasures against the proposed method. Section 6 summarizes results obtained and the contribution of this paper. Finally, Section 7 concludes the paper.

2. Related Works

2.1. Symbols and Notations

Table 1 shows the notations used throughout this paper.

Table 1. Notations.

Notation	Description
XOR, $\oplus$	Exclusive-OR operation
$\ll$ ($\gg$)	Binary left (right) shift operation
P_D	Power consumption when manipulating data D
P_{noise}	Noise power
ϵ	Constant
$(u_7u_6\cdots u_0)_2$	The binary representation of u
K	The secret key, $K = (K_{n-1}K_{n-2}\cdots K_0)_{2^w}$, $nW = \lvert K\rvert$
W	Number of bits in a word
GK	Guessed key, $GK \in \{0,1\}^W$
N	Number of traces
$\mathbb{P}$	Plaintext set ($\mathbb{P} = \{p_1, \cdots, p_N\}$)
$\mathbb{T}$	Trace set ($\mathbb{T} = \{t_1, \cdots, t_N\}$)
pre	Preprocessing function
f	Arbitrary operation
$f(\mathbb{P}, K)$	Set of $f(p_i, K)$ $(1 \le i \le N)$
Corr (X, Y)	Pearson's correlation coefficient for variables X and Y
$E[X]$	The expectation for variable X
HW (x)	Hamming weight of x
MSB (x)	Most significant bit of x
PC	Product-combining
AD	Absolute-difference
$C(S)$	Collection of all possible outcomes of intermediate value S
$L(S)$	Bit length of the intermediate value S
$\lvert\mathbb{X}\rvert$	Cardinality of a set $\mathbb{X}$

2.2. Correlation Power Analysis

The CPA is a statistical method that analyzes a huge amount of power traces $\mathbb{T}$ of cryptographic device encrypting different plaintexts $\mathbb{P}$ to reveal the secret key K [6]. It is based on the fact that $\mathbb{T}$ is related to the intermediate values $f(\mathbb{P}, K)$ calculated when encrypting $\mathbb{P}$. The power consumption model is a method that describes the relationship between $\mathbb{T}$ and $f(\mathbb{P}, K)$. Typically, the Hamming weight model is utilized as the power consumption model in software implementation. It assumes that the power consumption is linearly related to the number of 1's in the binary representation of the intermediate value [12]. Therefore, when data $d = (d_7d_6d_5\cdots d_0)_2$ is stored in an 8-bit register, the power consumption P_d is linearly related to the Hamming weight of d HW $(d) = \sum_{i=0}^{7} d_i$.

$$P_d = \epsilon \times \text{HW}(d) + P_{noise}.$$

The brute-force attack must guess the entire secret key to determining whether the supposed key is the right key, whereas the CPA guesses a much shorter partial key K_i and decides that. Thus, the CPA applies the divide-and-conquer algorithm that recovers the K_i and combines that to disclose the whole secret key K. An analyst calculates the set of Hamming weights of an intermediate value HW $(f(\mathbb{P}, GK))$, where GK is the guessed key. If $GK = K_i$, the set of Hamming weight is linearly related to $\mathbb{T}$. Therefore, the analyst can confirm that GK is the secret key by Pearson's correlation coefficient Corr $(\mathbb{T}$, HW $(f(\mathbb{P}, GK)))$. The Pearson's correlation coefficient is a measure of linear correlation with a value between -1 and 1.

Masking is a countermeasure against the CPA performed by randomizing the power consumption unrelated to the guessable intermediate value. The Boolean masking conceals sensitive data x by XORing a random value (mask) m to the data $x \oplus m$. Thus, the intermediate value x is split into more than two random variables $r_1, r_2, \cdots, r_n, r_0 = x \oplus r_1 \oplus r_2 \oplus \cdots \oplus r_n$. The analyst cannot disclose the key via the CPA because the intermediate value, which is related to power consumption, is not guessable. Typically, the 8-bit masked AES is implemented as a schema proposed by Herbst [7]. As the schema can efficiently resist the first-order CPA (FOCPA) with only six masks. Figure 1 shows the

mask used to conceal the output of each transformation in the Herbst schema. Note that SubBytes and Shiftrows require only one mask each, and MixColumns requires four masks.

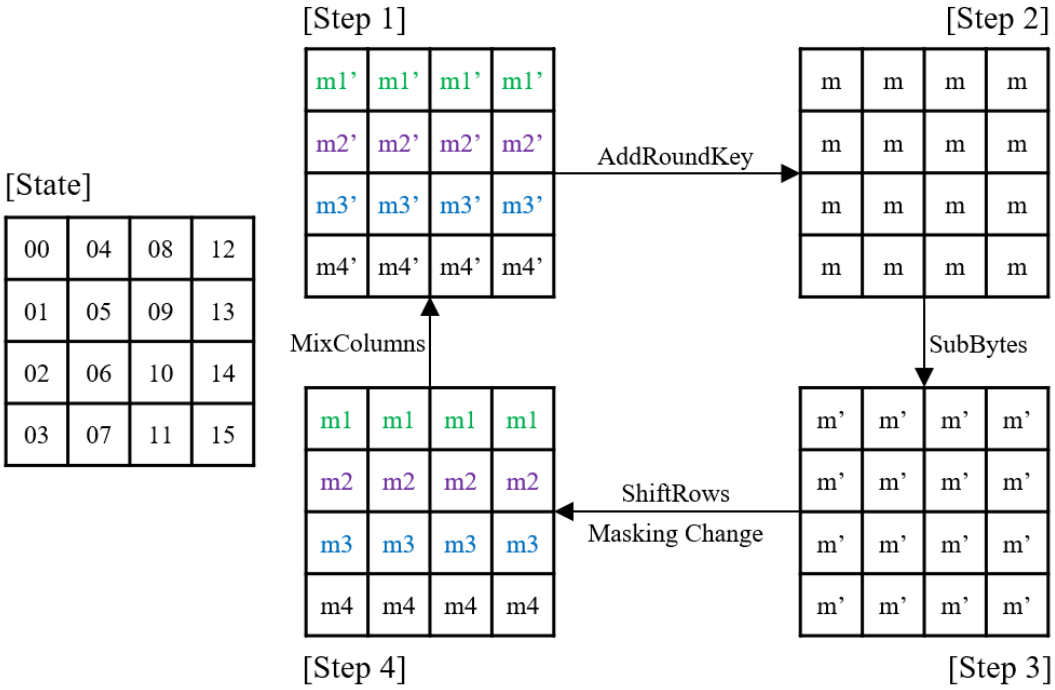


Figure 1. Masks to XOR to intermediate values for each transformation.

2.3. Second-Order CPA

If every mask is independent, the masked cryptography is secure against the CPA. However, in general implementation, some intermediate value shares the same mask because of the spatial and time complexity. In this case, the implementation might be vulnerable to the CPA.

The SOCPA is an analytical method that reveals the secret key by combining the power consumption of the two intermediate values x, y that share the same mask m. The analyst can disclose the key based on the fact that $\mathrm{HW}(x \oplus y)$ is linearly related to $pre(\mathrm{HW}(x \oplus m), \mathrm{HW}(y \oplus m))$ for some preprocessing function pre. Therefore, the SOCPA utilizes the fact that the correlation of Equation (1) is not zero, as shown in Table 2. l_I denotes the bit length of the intermediate value.

$$\begin{aligned} &\mathrm{Corr}\left(\mathrm{HW}(x \oplus y),\ pre\left(\mathrm{HW}(x \oplus m),\ \mathrm{HW}(y \oplus m)\right)\right) \\ &= \mathrm{Corr}\left(\sum_{i=0}^{l_I-1}(x_i \oplus y_i),\ pre\left(\sum_{i=0}^{l_I-1}(x_i \oplus m_i),\ \sum_{i=0}^{l_I-1}(y_i \oplus m_i)\right)\right). \end{aligned} \tag{1}$$

The commonly used preprocessing functions are product-combining (PC) and absolute-difference (AD). The definitions of PC and AD are expressed as follows:

$$\begin{aligned} pre_{\mathrm{PC}}(X, Y) &= (X - E[X]) \times (Y - E[Y]) \\ pre_{\mathrm{AD}}(X, Y) &= |X - Y|, \end{aligned}$$

where X and Y denote the random variables.

The correlation of Equation (1) depends on the preprocessing function and the bit length of the intermediate value l_I, as shown in Table 2. In Table 2, the correlation coefficients decrease as the bit length increases for all preprocessing functions. Note that the correlation is 1.00 when the bit length is 1; however, the correlation is only 0.35 at most when the bit length is 8.

Table 2. Correlation coefficients of Equation (1) for preprocessing functions and the bit length of intermediate values [11,13].

Preprocessing Function	Bit Length (l_I)			
	1	**2**	**4**	**8**
PC	1.00	0.71	0.50	0.35
AD	1.00	0.53	0.34	0.24

Because $x \oplus y$ is a guessable intermediate value, and two points of power consumption $P_{x\oplus m}$, $P_{y\oplus m}$ are linearly related to HW $(x \oplus m)$, HW $(y \oplus m)$, respectively, Equation (1) can be modified to Equation (2). Therefore, the analyst can perform the CPA by combining two intermediate values and two points of power consumption.

$$\begin{aligned} &\text{Corr}\left(\text{HW}\left(x \oplus y\right),\ pre\left(P_{x\oplus m},\ P_{y\oplus m}\right)\right) \\ &= \text{Corr}\left(\sum_{i=0}^{l_I-1}\left(x_i \oplus y_i\right),\ pre\left(P_{x\oplus m},\ P_{y\oplus m}\right)\right). \end{aligned} \tag{2}$$

Note that the Hamming weight is only modifiable to power consumption when l_I is equal to the length of the register l_R. If $l_I < l_R$, the remaining $l_R - l_I$ bits act as noise. For example, if data $d = (d_7 d_6 \cdots d_0)_2$ is stored in an 8-bit register and an analyst uses only a 1-bit intermediate value d_7, then the remaining 7 bits $(d_6 d_5 \cdots d_0)_2$ behave similar to noise:

$$P_d = \epsilon \times d_7 + \left(\sum_{i=0}^{6} d_i + P_{noise}\right). \tag{3}$$

The correlation of Equation (2) may be much lower compared to the correlation of Equation (1) because of the effect of noise. Consequently, even though the correlation of Equation (1) is 1.00 when bit length is 1, generally, the 1-bit intermediate is not an optimal choice. Table 3 shows the theoretical correlation coefficients of Equation (2) when the bit length of the register is 8. In Table 3, the correlation of the 1-bit intermediate value is only up to 0.13, which is much less than that of the 8-bit intermediate value. The theoretical correlation of the table is calculated as Equation (4). Unlike Equation (1), the upper limit of the sigma notation of the preprocessing function input is fixed at 7.

$$\text{Corr}\left(\sum_{i=0}^{l_I-1}\left(u_i \oplus v_i\right),\ pre\left(\sum_{i=0}^{7}\left(u_i \oplus m_i\right),\ \sum_{i=0}^{7}\left(v_i \oplus m_i\right)\right)\right). \tag{4}$$

Table 3. Theoretical correlation coefficients of Equation (2) for preprocessing functions and the bit length of intermediate values when the length of the register is 8.

Preprocessing Function	Bit Length (l_I)			
	1	**2**	**4**	**8**
PC	0.13	0.18	0.25	0.35
AD	0.08	0.12	0.17	0.24

2.4. MixColumns of AES

The MixColumns of AES is defined as the multiplication of the constant matrix. In the equation below, $s_{i,j}$ and $s'_{i,j}$ denote the input and output, respectively.

$$\begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}.$$

In AES, byte values are interpreted as Galois field GF (2^8) elements, i.e., if the binary representation of the value d is $(d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0)_2$, it is interpreted as the element of the Galois field as follows:

$$d_7 x^7 + d_6 x^6 + d_5 x^5 + d_4 x^4 + d_3 x^3 + d_2 x^2 + d_1 x + d_0.$$

Furthermore, AES operation is also defined in the Galois field GF (2^8). Multiplication is defined by multiplying two binary polynomials and reducing with an irreducible polynomial $x^8 + x^4 + x^3 + x + 1$. The general approach to implement multiplication in the Galois field is to repeat the `xtime` operation, i.e., multiplying the input with x. Multiplying the above polynomial with the polynomial x results in

$$d_7 x^8 + d_6 x^7 + d_5 x^6 + d_4 x^5 + d_3 x^4 + d_2 x^3 + d_1 x^2 + d_0 x.$$

As the degree of the irreducible polynomial is 8, if d_7 is zero, the above result does not require reduction. However, if d_7 is 1, subtraction of the irreducible polynomial is needed, i.e., byte-level implementation of the `xtime` performs different operation depending on d_7. The MSB of the input is determined as follows:

$$\texttt{xtime}\,(d) = \begin{cases} d \ll 1 & \text{if MSB}\,(d) = 0, \\ (d \ll 1) \oplus 0\text{x}1b & \text{if MSB}\,(d) = 1. \end{cases} \tag{5}$$

Note that the MSB extraction is required to implement the `xtime` operation.

Our Challenge

The major challenge we faced in this paper is finding a special intermediate value, shortly SIV, that maintains the correlation coefficient presented in Table 2 when $l_I = 1$ because it is less affected by noise, unlike Equation (3). Thus, because the SIV can accurately model power consumption and retain the theoretical correlation coefficient of the SOCPA is 1.00, using the SIV can significantly increase the performance of the SOCPA.

3. Sparse Intermediate Value in AES

In this section, we find the SIV based on the operational characteristic of the cryptographic algorithm. If there exists some intermediate value that behaves like a shorter length of the intermediate value, it can enhance the SOCPA performance because it raises the correlation between hypothetical intermediate value and preprocessed power consumption, as shown in Table 2 while accurately model power consumption. Typically, the number of all possible outcomes $C\,(S)$ of an $L\,(S)$ bit intermediate value S is $2^{L(S)}$.

$$|C\,(S)| = 2^{L(S)}.$$

However, a special intermediate value SIV may exist such that $|C\,(\text{SIV})|$ is much lower than $2^{L(\text{SIV})}$ because of operational characteristics.

$$|C\,(\text{SIV})| < 2^{L(\text{SIV})}.$$

Definition 1. *An intermediate value S is SIV if* $|C\,(S)| < 2^{L(S)}$.

3.1. Finding the SIV Based on Operational Characteristics

The `xtime` performs different operations depending on the input's MSB, as shown in Equation (5). To determine the instruction to be executed in software implementation, the MSB must be extracted

and stored in a register. When storing the MSB in an 8-bit register, only 1-bit changes depending on the MSB and the remaining 7 bits are always zero. Let the binary representation of the input d be $(d_7d_6d_5d_4d_3d_2d_1d_0)_2$. Then the intermediate value S_1, which is stored in the register when extracting MSB, is $(s_7s_6s_5s_4s_3s_2s_1s_0)_2 = (0000000d_7)_2$. The $|C(S_1)|$ is only 2, which is much smaller than $2^8 = 256$. Therefore, S_1 can be utilized as the SIV. Thus, we define the Property 1 as follows.

Property 1. *MSB extraction is essential to implement* `xtime`*. Therefore, the SIV exists as* $S_1 = (0000000d_7)_2$*, such that* $|C(S_1)|$ *is only 2.*

$$C(S_1) = \{(00000000)_2,\ (00000001)_2\} = \{0x00,\ 0x01\}.$$

Listing 1 is one of the `xtime` implementations in C language. It can be divided into two parts. One part is multiplication by x as the left-side of the XOR operator $(\text{input} \ll 1)$, and the other part is reduction by the irreducible polynomial as the right-side $((\text{input} \gg 7) * 0x1b)$.

Listing 1: 8-bit implementation of `xtime` using C language.

```
1 #define xtime(input)  ((input << 1) ^ ((input >> 7) * 0x1b))
```

In the reduction part, the MSB is not only extracted at $(\text{input} \gg 7)$ but also multiplied by 0x1b to decide whether to reduce the left-side result. The intermediate value S_2, which is stored when the MSB is multiplied by 0x1b, is $d_7 \times (00011011)_2 = (000d_7d_70d_7d_7)$. As the $|C(S_2)|$ is only 2, S_2 can be utilized as the SIV.

Note that the difference between the Hamming weights of elements in $C(S_2)$ is 4, whereas the difference of $C(S_1)$ is only 1. The effect of S_2 on power consumption is relatively greater compared to S_1.

$$\begin{aligned} P_{S_1} &= \epsilon \times \mathrm{HW}(S_1) + P_{noise} = \epsilon \times d_7 + P_{noise} \\ P_{S_2} &= \epsilon \times \mathrm{HW}(S_2) + P_{noise} = \epsilon \times (4d_7) + P_{noise}. \end{aligned} \tag{6}$$

Therefore, S_2 can relatively reduce the effect of noise. Thus, we define the Property 2 as follows.

Property 2. *If SIV is computed using a certain constant, the computation result is not only utilized as the SIV, it may also significantly affect power consumption. Thus, the computation result can relatively reduce the effect of noise* P_{noise}*. In the case of Listing 1, the multiplication result* $S_2 = (000d_7d_70d_7d_7)_2$ *can reduce the effect. The power consumption of* S_2 *and the collection of all possible outcomes of* S_2 *are expressed as follows:*

$$P_{S_2} = \epsilon \times (4d_7) + P_{noise}$$

$$C(S_2) = \{(00000000)_2,\ (00011011)_2\} = \{0x00,\ 0x1b\}.$$

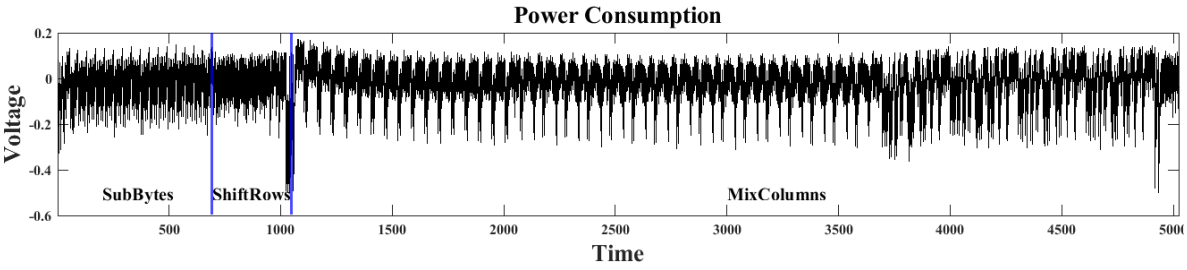
We demonstrate the existence of SIV via CPA and t-value. In this paper, we compare the two versions of the CPA, as follows.

- `General CPA`: CPA using SubBytes output as an intermediate value.
- `SIV-based CPA`: CPA using SIV as an intermediate value.

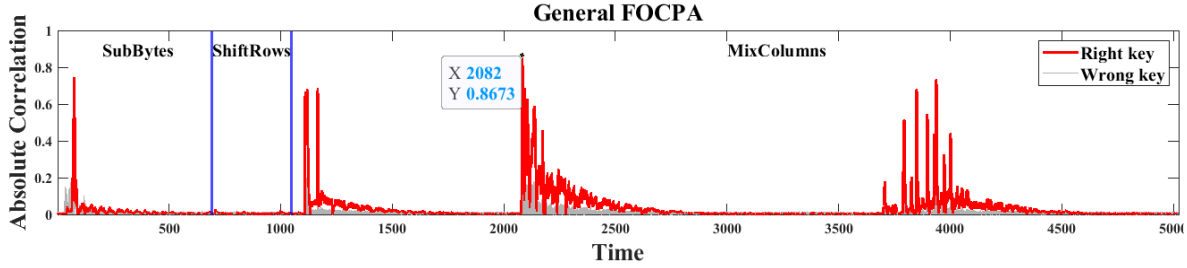
Demonstration of Existence of SIV

Our experiments demonstrated that the SIV-related power consumption occurs. We analyzed 100,000 power consumption traces at a 29.538 MS/s sampling rate when AES ran on a ChipWhisperer-Lite ATXMEGA128D4 (8-bit processor) [14]. We utilized S_2 as the SIV.

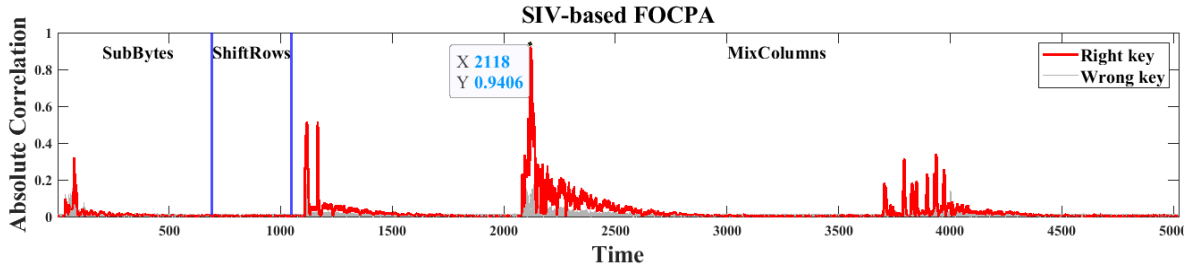
For the FOCPA of the MixColumns, the peak correlation coefficient of the `General FOCPA` is approximately 0.87, and the peak correlation of the `SIV-based FOCPA` is around 0.94, as shown in Figure 2. The peak correlation of the two versions of FOCPA is similar, demonstrating the existence of the SIV-related power consumption when performing `xtime`, as shown in Property 1.



(**a**) Power consumption when performing the first round of AES



(**b**) The absolute correlation coefficient of `General FOCPA`



(**c**) The absolute correlation coefficient of `SIV-based FOCPA`

Figure 2. Results of `General FOCPA` and `SIV-based FOCPA` (0th byte).

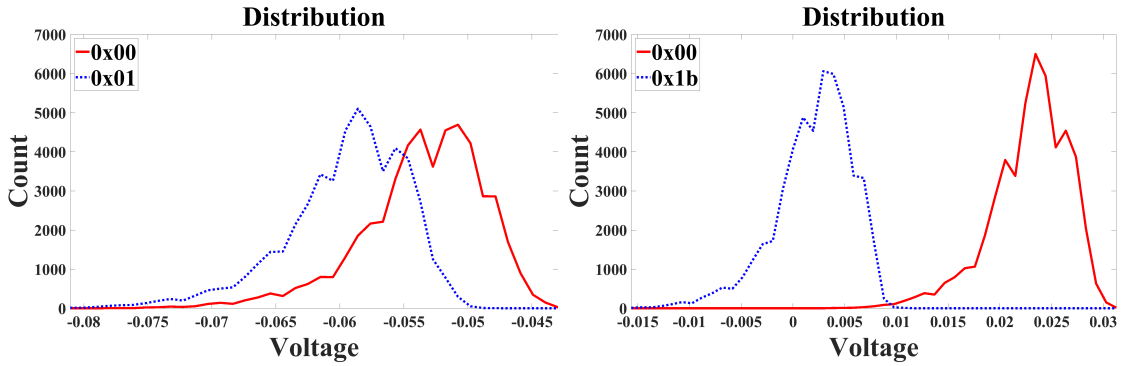
The power consumption of MixColumns is linearly related to the Hamming weight of SubBytes output because the SubBytes output is identical to the MixColumns input. In addition, the correlation of SubBytes is not zero because the Hamming weight of the SIV is linearly related to the MSB of the SubBytes output, as shown in Equation (6).

To demonstrate Property 2, we compared each power consumption of S_1 and S_2. The power consumption is divided into two groups based on the SIV, and we verified the distributions of each group. Based on Equation (6), the effect of d_7 on the power consumption P_{S_2} is four times greater than that on P_{S_1}. Therefore, theoretically, the difference between the mean of the two distributions of S_2 is four times larger than that of S_1. We utilized the t-value of Welch's t-test [15] to measure the difference between the two distributions as follows:

$$t = \frac{E[X] - E[Y]}{\sqrt{\frac{\sigma_X^2}{|X|} + \frac{\sigma_Y^2}{|Y|}}},$$

where σ_X and σ_Y denote the standard deviation of X and Y, respectively

Figure 3 shows the experimental proof of Property 2. The difference between the mean of the two distributions of S_2 is around 0.02084, which is approximately 3.5 times larger than that of S_1 (0.005981). This is very close to the theoretical ratio of 4. The t-value of S_2 is approximately 876.22 and that of S_1 is 198.84. Owing to the substantial difference in the means of the two distributions of S_2, the two distributions are completely separated despite the noise, as shown in Figure 3b, i.e., the effect of noise can be relatively moderated, as demonstrated in Property 2.



(**a**) Distributions of S_1 (**b**) Distributions of S_2

Figure 3. Distributions of power consumption that occurs when storing S_1 and S_2.

3.2. The Performance Improvement of SOCPA Using SIV

In Section 3.1, we demonstrated the existence of SIV using the operational characteristic. In the case of AES, the cardinality of all possible outcomes of SIV is only 2. Moreover, the Hamming weights of outcomes are different. Note that constant multiplication with random variables only affects the sign of the correlation coefficient:

$$\mathrm{Corr}\left(aX, bY\right) = \mathrm{sign}\left(ab\right) \cdot \mathrm{Corr}\left(X, Y\right),$$

where a and b are the arbitrary constants, and sign is the sign function. Thus, the theoretical correlation coefficient of Equation (1) is the same as the 1-bit correlation presented in Table 2, although the length of the SIV is 8. For example, let u, v be the S_2 intermediate value, then the theoretical correlation of the SIV is the same as that of some 1-bit intermediate value.

$$\begin{aligned}
&\mathrm{Corr}\left(\mathrm{HW}\left(u \oplus v\right),\ pre\left(\mathrm{HW}\left(u\right), \mathrm{HW}\left(v\right)\right)\right) \\
=\ &\mathrm{Corr}\left(\mathrm{HW}\left(4u_7 \oplus 4v_7\right),\ pre\left(\mathrm{HW}\left(4u_7\right), \mathrm{HW}\left(4v_7\right)\right)\right) \\
=\ &\mathrm{Corr}\left(4 \times \left(u_7 \oplus v_7\right),\ pre\left(4u_7, 4v_7\right)\right) \\
=\ &\mathrm{Corr}\left(u_7 \oplus v_7,\ pre\left(u_7, v_7\right)\right).
\end{aligned}$$

Therefore, when the preprocessing functions are AD and PC, the SIV can increase the correlation coefficient of SOCPA from 0.24 to 1.00 and from 0.35 to 1.00, respectively. That is, the SIV theoretically allows to reduce the minimum trace to disclose the secret key, $\frac{1}{0.24^2} \approx 17.3611$ times and $\frac{1}{0.35^2} \approx 8.1633$ times, respectively [9].

4. Application to OpenSSL AES

In this section, we analyzed the AES implementation of the OpenSSL, one of the most commonly used secure socket layer (SSL) toolkit, and demonstrate that SIV can significantly improve the performance of SOCPA.

4.1. Finding SIV Based on Operational Characteristics

Listing 2 is MixColumns implementation of the OpenSSL version 1.1.1c, which is the latest version. In this listing, `t` is the AES state, which is the 32-bit array of length 4; `r0`, `r1`, and `r2` are 32-bit variables for `xtime` operations and matrix multiplication.

Listing 2: The implementation of MixColumns in OpenSSL version 1.1.1c `aes_x86core.c`.

```
1  for (i = 0; i < 4; i++)
2  {
3    r0 = t[i];
4    // Bytewise MSB extraction
5    r1 = r0 & 0x80808080;
6    // xtime
7    r2 = ((r0 & 0x7f7f7f7f) << 1) ^ ((r1 - (r1 >> 7)) & 0x1b1b1b1b);
8    // matrix multiplication
9    t[i] = r2                                   ^
10          ((r2 ^ r0) << 24) ^ ((r2 ^ r0) >>  8) ^
11          (r0        << 16) ^ (r0        >> 16) ^
12          (r0        <<  8) ^ (r0        >> 24);
13 }
```

In this listing, line 7 is the implementation of `xtime`. Line 7 can be divided into two parts: performing bytewise 1-bit left shift operation (multiplication by x) as the left-side of the XOR operator $((\text{r0} \,\&\, 0\text{x7f7f7f7f}) \ll 1)$ and executing bytewise reduction as the right-side of the XOR operator $((\text{r1} - (\text{r1} \gg 7)) \,\&0\text{x1b1b1b1b})$.

Note that the bytewise MSB of the input is not only extracted and stored to r1 at line 5 but also calculated by itself at line 7. Table 4 shows the binary representation of the SIVs, wherein `X[31:24]` denotes the first byte of `X`. In particularly, because the MSB of each byte of the input determines 7 bits of the subtraction result S_5, it can reduce the effect of noise to the maximum, similar to Property 2. Therefore, the subtraction result can be utilized as the SIV with the best property. Thus, we define the Property 3 as follows.

Property 3. *Let the binary representation of input d be* $(d_{31}d_{30}\cdots d_0)_2$*, then the MSB of each byte is* d_{31}*,* d_{23}*,* d_{15}*,* d_7*. Assume, without loss of generality, that the analyst considers the first byte of the intermediate value. Thus, the power consumption of* S_i $(i \in \{3,4,5,6\})$ *is given as follows:*

$$P_{S_i} = \epsilon \times (n \times d_{31}) + \left(n \times \sum_{i=0}^{2} d_{8i+7} + P_{noise} \right),$$

where n is the Hamming weight of S_i*. The remaining 24 bits behave similar to noise. Note that if n is large, the effect of* P_{noise} *can be relatively reduced.*

Table 4. Binary representation of the first byte of the SIVs for each operation when performing `xtime` in OpenSSL `aes_x86core.c`.

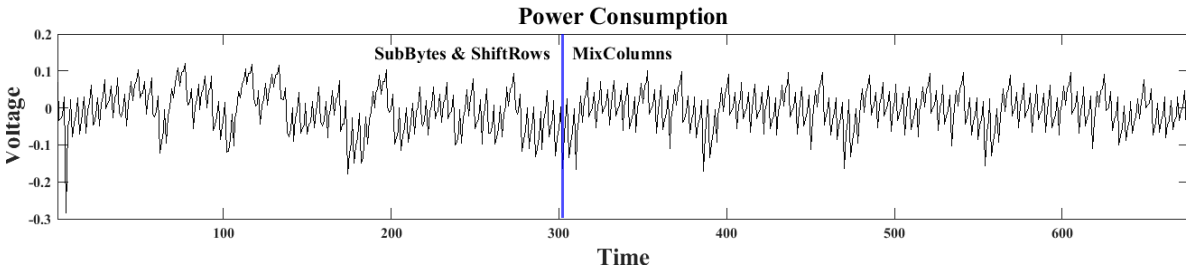
SIV	Operation	MSB 1	MSB 0
S_3	`r1[31:24]`	$(10000000)_2$	$(00000000)_2$
S_4	`(r1 >> 7)[31:24]`	$(00000001)_2$	$(00000000)_2$
S_5	`(r1 - (r1 >> 7))[31:24]`	$(01111111)_2$	$(00000000)_2$
S_6	`((r1 - (r1 >> 7)) & 0x1b1b1b1b)[31:24]`	$(00011011)_2$	$(00000000)_2$

Demonstration of Existence of SIV

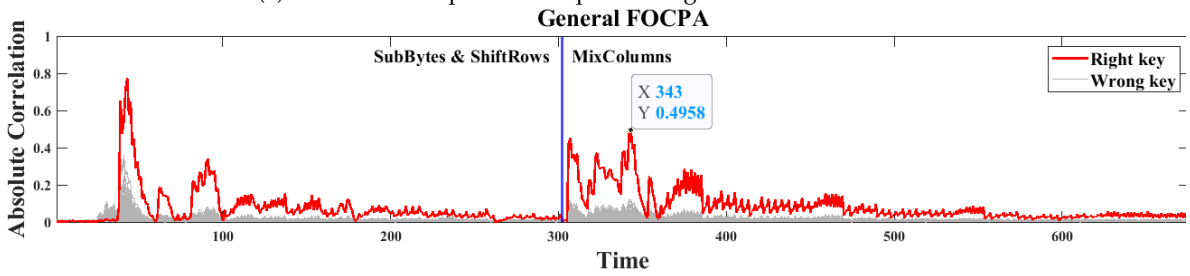
Our experiments demonstrate that the power consumption, which occurs when performing OpenSSL AES on ChipWhisperer UFO STM32F3 (32-bit processor) has the same features as the ATXMEGA128D4 in Section 3.1 [16].

Figure 4 shows the results of the two versions of FOCPA. For the FOCPA of the MixColumns, the peak correlation of the `General FOCPA` is about 0.50, whereas the peak correlation of the `SIV-based FOCPA` is about 0.37. The peak correlation coefficients of the two versions of FOCPA are the same. Therefore, the SIV-related power consumption exists.

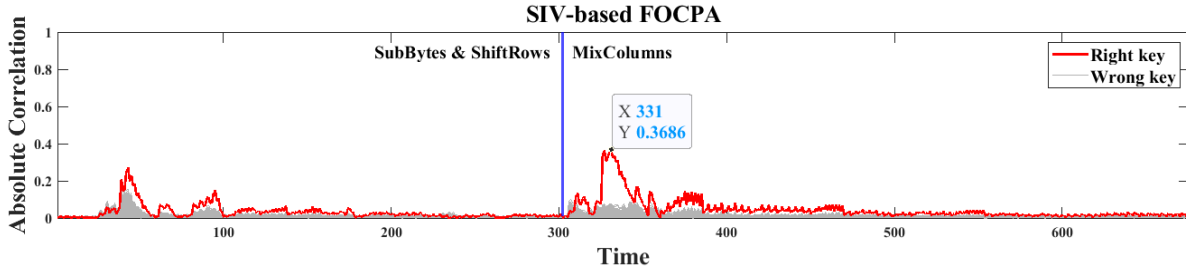
Figure 5 shows the distribution of power consumption of S_3 and S_5. The difference between the means of the two distributions of S_5 is approximately 0.003485, which is about 6.8 times larger than that of S_3 (0.000512). This is very close to the theoretical ratio of 7. The t-value of S_5 and S_3 is 107.73 and 24.81, respectively. Consequently, the effect of noise can be relatively reduced, as stated in Property 3.



(**a**) Power consumption when performing the first round of AES

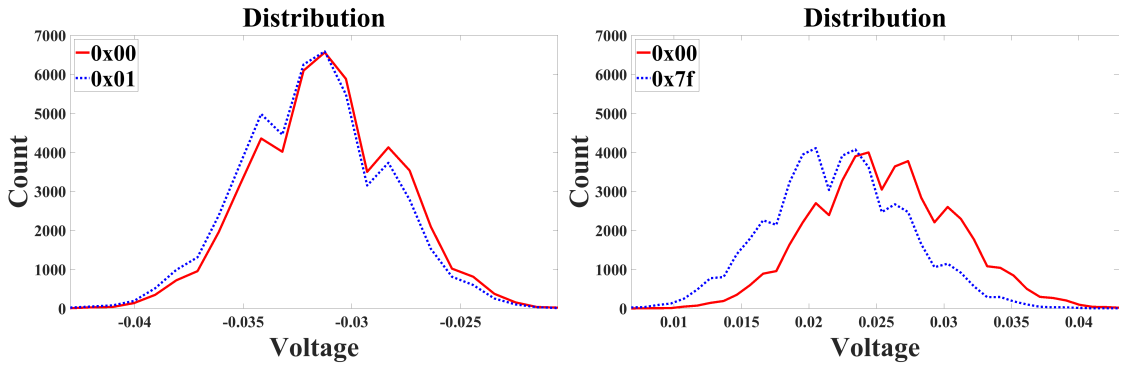


(**b**) The absolute correlation coefficient of the `General FOCPA`



(**c**) The absolute correlation coefficient of the `SIV-based FOCPA`

Figure 4. Results of `General FOCPA` and `SIV-based FOCPA`. (0th byte).



(**a**) Distribution of S_3 (**b**) Distribution of S_5

Figure 5. Distributions of power consumption that occurs when executing copy operation and `xtime` operation of OpenSSL.

4.2. Experimental Results of General SOCPA and SIV-Based SOCPA

In this section, we demonstrate that the correlation coefficient of the `SIV-based SOCPA` is considerably higher than that of `General SOCPA`. The PC and AD are utilized as preprocessing functions, and the results for AD are presented in Appendix A. The experimental environment is the same as that described in Section 4.1. We analyze the power consumption that occurs while performing OpenSSL AES on ChipWhisperer UFO STM32F (32-bit processor) [16].

To perform SOCPA, the analyst must combine two intermediate values concealed by the same mask. The inputs of the MixColumns share the same mask by row, as shown in Figure 1 [Step 4]. Our attack scenario performs minimum times of the attacks to recover the entire secret key. We analyze each row of the MixColumns by dividing it into two pairs. Thus, the combination of byte indexes of intermediate value for analysis is $(00, 04)$, $(01, 05)$, $\cdots$, $(11, 15)$. Recall that the state of AES is a column-major order array.

Figure 6 shows the bytewise peak correlation of two versions of the SOCPA. The correlation of the `SIV-based SOCPA` for every combination is higher than that of the `General SOCPA`, and the average correlation of the `SIV-based SOCPA` is approximately 1.7 times higher than the `General SOCPA`. The correlation of the `SIV-based SOCPA` is not 1.00 because the remaining 24 bits behave similar to noise.

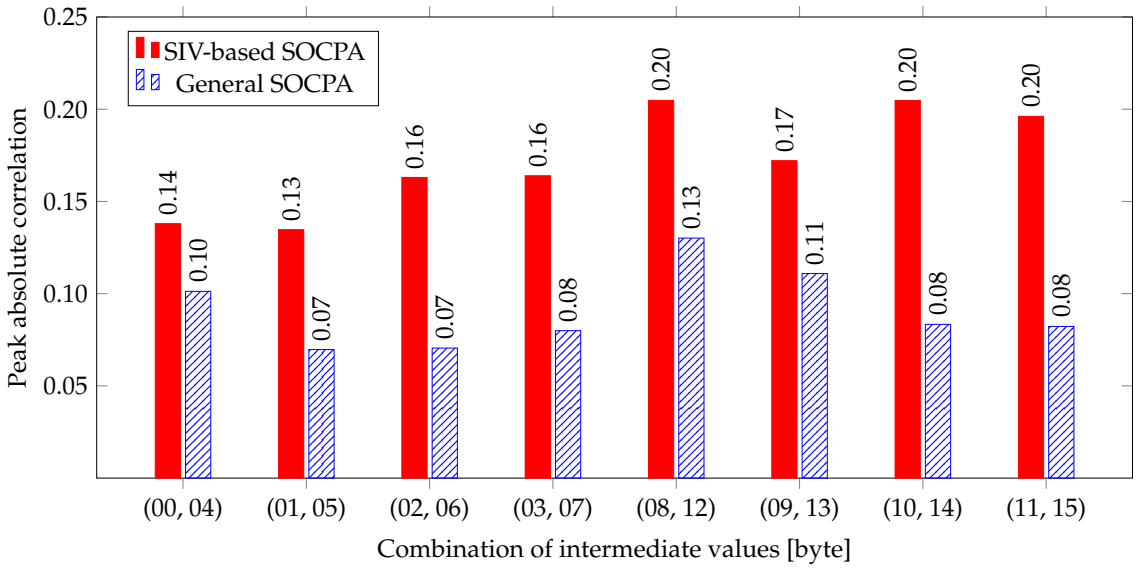
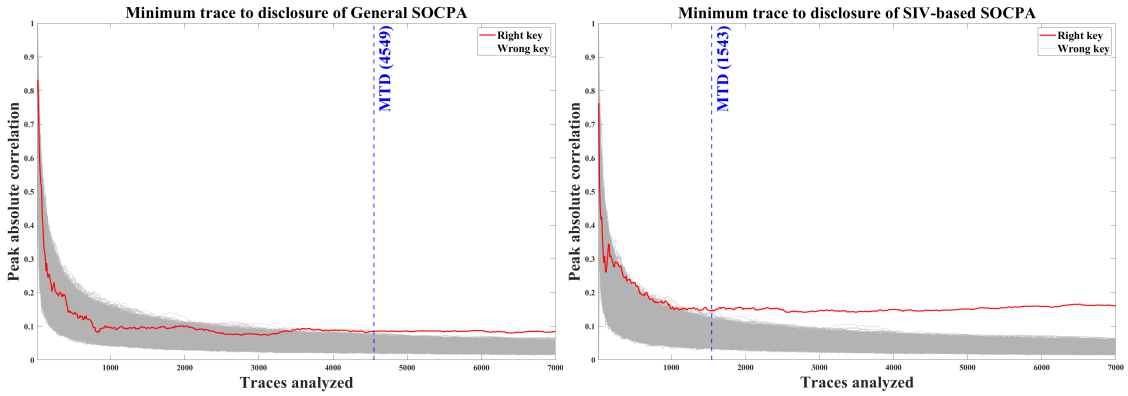


Figure 6. Peak correlation coefficient of the two versions of SOCPA on MixColumns (Product-Combining).

Furthermore, we find the minimum trace to disclose (MTD) to illustrate that the `SIV-based SOCPA` is more effective, i.e., it can reveal the secret key with less information than the `General SOCPA`. Figure 7 shows the MTD for a combination of 1st and 5th bytes. In this figure, the `SIV-based SOCPA` can disclose the secret key with only about 34% of the power trace than that required by the `General SOCPA`.

Figure 8 shows the MTD of every combination of intermediate values; the maximum MTD of the `SIV-based SOCPA` is 1717, and that of the `General SOCPA` is 6643. Therefore, SIV allows the disclosure of the entire secret key using only a quarter of the trace required by the `General SOCPA`.



(**a**) General SOCPA (**b**) SIV-based SOCPA

Figure 7. Minimum trace to disclosure for MixColumns for a combination of 1st and 5th bytes (Production-Combining).

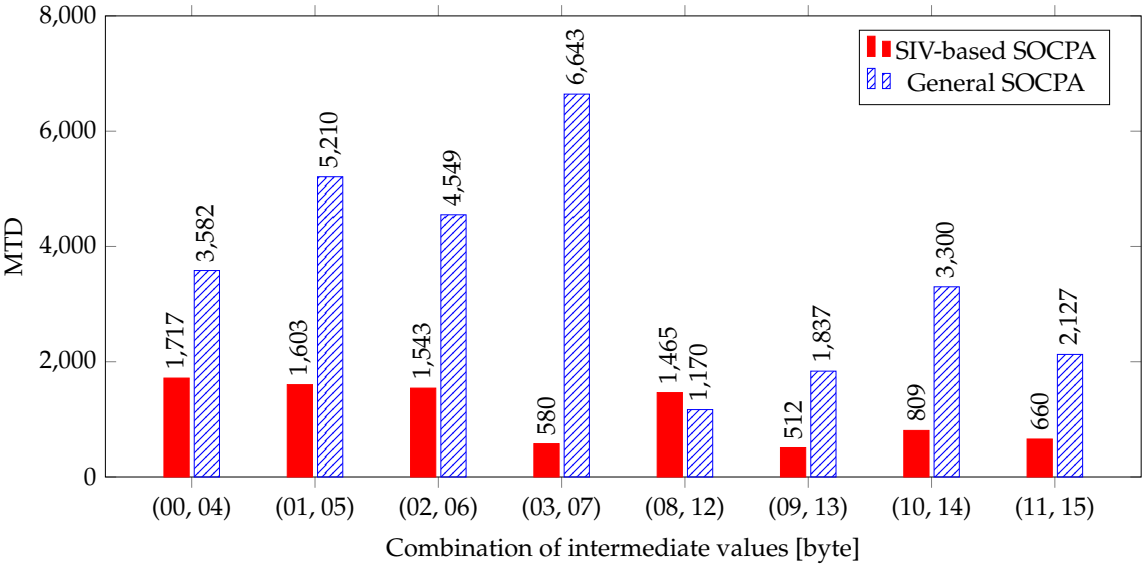


Figure 8. Minimum trace to disclosure for MixColumns (Product-Combining).

5. Countermeasures

We recommend two countermeasures against the `SIV-based SOCPA`. The first countermeasure changes the sequence of computations to increase the time complexity in calculating the SIV. The second countermeasure is to implement the SIV generating operation using a precomputed table to eliminate power consumption related to the SIV. Typically, Listing 3 is an implementation of the MixColumns to reduce the time complexity in the 8-bit device.

5.1. Increasing the Time Complexity of the SIV-Based SOCPA

The first countermeasure is to modify the MixColumns implementation, as shown in Listing 3. This listing utilizes the associative property to change the sequence of computations to complicate the calculation of the SIV.

Listing 3: Countermeasures to increase the time complexity of the intermediate value [17].

```
for (i = 0; i < 4; i++)
{
  r0 = t[i];
  rt = r0 ^ (r0 << 24) ^ (r0 >> 8); // added
  r1 = rt & 0x80808080;
  r2 = ((rt & 0x7f7f7f7f) << 1) ^ ((r1 - (r1 >> 7)) & 0x1b1b1b1b);
  t[i] = r2 ^ (rt ^ r0) ^ (r0 << 16) ^ (r0 >> 16) ^
         (r0 << 8) ^ (r0 >> 24);
}
```

The existing MixColumns implementation in Listing 2 performs `xtime` for each term and then adds two terms, as shown on the left-hand side of Equation (7). Contrarily, the proposed implementation adds two terms and then performs `xtime`, as shown on the right-hand side of Equation (7). To calculate the input of `xtime`, the analyst must guess two keys XORed with $s_{0,0}$ and $s_{0,1}$.

$$2 \cdot s_{0,0} + (2 \cdot s_{1,0} + s_{1,0}) + s_{2,0}, + s_{3,0} = 2 \cdot (s_{0,0} + s_{1,0}) + s_{1,0} + s_{2,0} + s_{3,0}. \quad (7)$$

Although the SIV occurs in this implementation, the size of keyspace to perform the `SIV-based SOCPA` increases from 2^{16} to 2^{32}. Therefore, performing the `SIV-based SOCPA` is impractical because the complexity of calculating the MSB of the `xtime` input increases to $2^{16} = 65{,}536$ times. As this countermeasure only changes the sequence of computations, there is no overhead for countermeasure.

5.2. Removing Bit Extraction Operation

The second countermeasure is to implement `xtime`, which is the SIV generating operation, by precomputation table. This implementation does not extract the input's MSB or generate the SIV; it only refers to the table. Thus, the power consumption related to the SIV does not occur. However, to implement this countermeasure, additional memory is required. Therefore, if the AES is implemented using a precomputed table of SubBytes and MixColumns known as T-table, it requires 4KB memory. Additionally, if a precomputation table achieves the only `xtime`, 256 bytes of memory are required. As these implementations substitute several operations to memory reference operations, the time complexity is generally lower than Listing 2.

6. Discussion

Herein, the special intermediate value, named SIV, was proposed based on the operational characteristics of the cryptographic algorithm. The SIV of the AES is determined by analyzing the reduction operation of the MixColumns. It remarkably raises the correlation of SOCPA to 1.00 compared to existing studies that have only increase the correlation to 0.35 [8,10,11]. That is, SIV theoretically allows reducing the MTD approximately 8.1633 times than `General SOCPA` [9].

We analyzed the AES of OpenSSL, which is one of the most commonly used secure socket layer toolkit. As shown in Table 5, correlation is increased from 0.0910 to 0.1722, and the `SIV-based SOCPA` can disclose the entire secret key with only about quarter trace required for the `General SOCPA`. The correlation is lower than the theoretical value 1.00 and MTD is not reduced as much as the theory, because the remaining 24 bits behave similarly to noise. However, we showed that the SIV-based SOCPA could improve the analysis performance by more than four times.

Table 5. Average peak correlation coefficient and maximum of minimum trace to disclosure of two versions of SOCPA on MixColumns (Product-Combining).

Experimental Result	`SIV-Based SOCPA`	`General SOCPA`
Average of peak correlation	0.1722	0.0910
Maximum of minimum trace to disclosure	1717	6643

Two countermeasures against the `SIV-based SOCPA` were recommended. The first is the ability to double the keyspace that must be guessed to calculate SIV, which increases the time complexity by 65,536 times. The countermeasure does not require any computation and memory overhead. Next is the implementation of the lookup table, which performs transformation without the subroutine that causes the SIV. This countermeasure has a 4 KB memory overhead. Typically, these two countermeasures provide guidelines for implementation that are resistant to the SIV-based SOCPA.

Our limitation is that, unlike improvements the preprocessing function is applicable to every cryptography, `SIV-based SOCPA` has to analyze the operational characteristic to determine the SIV for each cryptography, and we only discovered the SIV of the AES. And we applied only to the AES

7. Conclusions

In this paper, we proposed a special intermediate value, SIV, that has a unique characteristic.

The SIV improves the SOCPA performance significantly because it can accurately model power consumption and retain the theoretical correlation of 1-bit intermediate value. When the length of the intermediate value is 8, existing researches have only improved the correlation to 0.35, but the SIV has remarkably raised the correlation to 1.00. That is, the MTD for the `General SOCPA` requires at least 8.1633 times more traces than the FOCPA, whereas the `SIV-based SOCPA` is theoretically the same as the FOCPA.

We analyzed the OpenSSL, which is the most commonly used open-source secure socket layer, and confirmed that there exists the SIV. Consequently, the proposed `SIV-based SOCPA` can disclose the whole secret key using only a quarter of the trace required by the `General SOCPA`. Further, we recommended two countermeasures as a guideline for implementing a cryptographic algorithm that is resistant to the proposed method.

The limitation of the proposed method is that it has to discover the SIV for each cryptographic algorithms or implementation. In this paper, only the AES was analyzed. Finding the SIV from other cryptographic algorithms is an interesting further work. Thus in the future, we are going to apply to the proposed method for other cryptographic algorithms.

Author Contributions: Writing–original draft, J.-H.K., B.-Y.S. and D.-G.H.; Writing–review and editing, J.-H.K., B.-Y.S. and D.-G.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No. 2017-0-00520, Development of SCR-Friendly Symmetric Key Cryptosystem and Its Application Modes). Additionally, this work was supported as part of Military Crypto Research Center(UD170109ED) funded by Defense Acquisition Program Administration(DAPA) and Agency for Defense Development(ADD).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SIV	Sparse intermediate value
CPA	Correlation power analysis
FOCPA	First-order correlation power analysis
SOCPA	Second-order correlation power analysis
HW	Hamming weight
MSB	Most significant bit
PC	Product-combining
AD	Absolute-difference
AES	Advanced encryption standard

Appendix A. Second-Order CPA Results

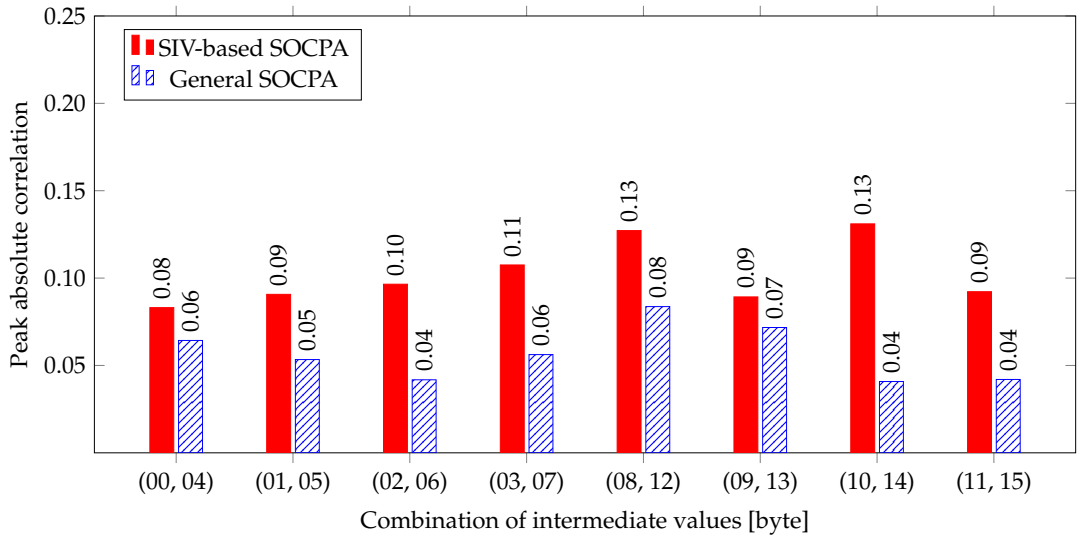
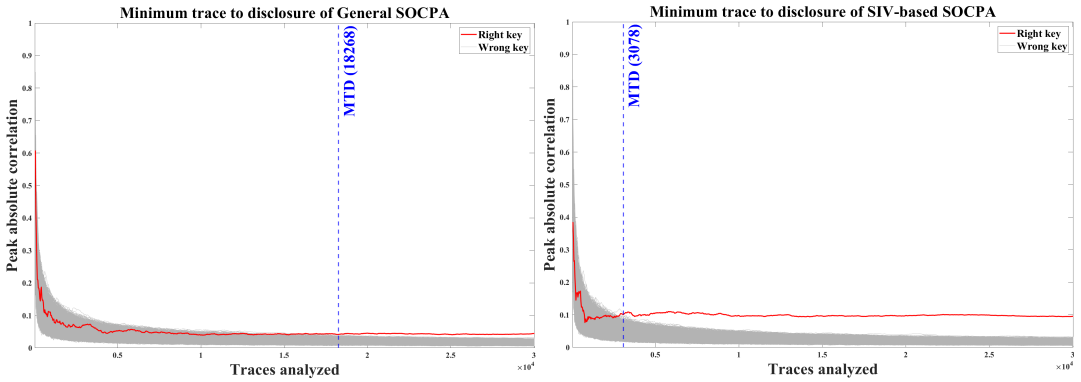


Figure A1. Correlation coefficient of the two versions of SOCPA on MixColumns (Absolute-Difference).



(**a**) General SOCPA (**b**) SIV-based SOCPA

Figure A2. Minimum trace to disclosure for MixColumns for a combination of 1st and 5th bytes (Absolute-Difference).

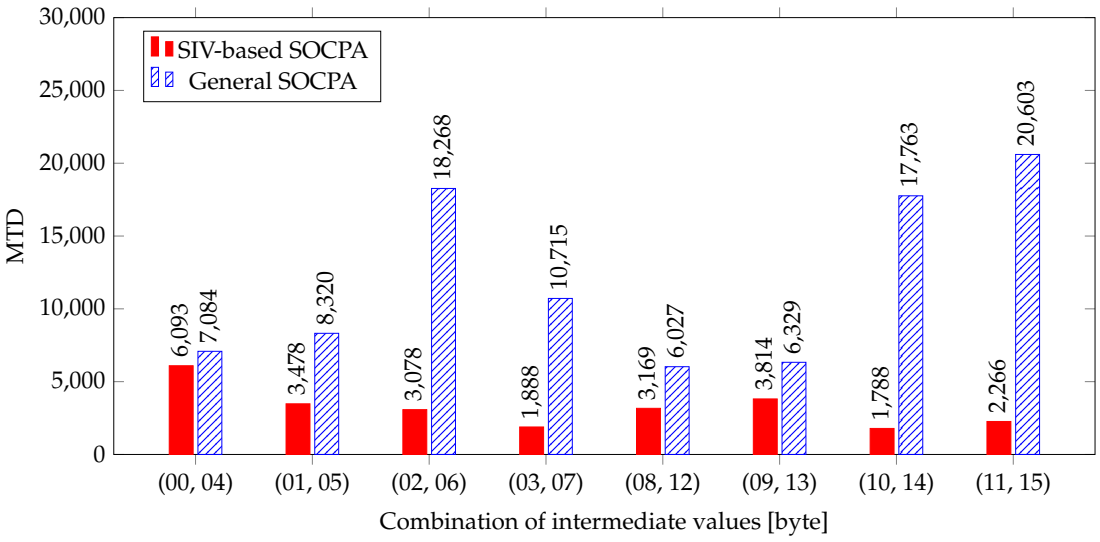


Figure A3. Minimum trace to disclosure (MTD) for MixColumns (Absolute-Difference).

References

1. Kocher, P.C. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Proceedings of the Advances in Cryptology—CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 1996; pp. 104–113. [CrossRef]
2. Kocher, P.C.; Jaffe, J.; Jun, B. Differential Power Analysis. In Proceedings of the Advances in Cryptology—CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999; pp. 388–397. [CrossRef]
3. Gandolfi, K.; Mourtel, C.; Olivier, F. Electromagnetic Analysis: Concrete Results. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2001, Third International Workshop, Paris, France, 14–16 May 2001; pp. 251–261. [CrossRef]
4. Genkin, D.; Shamir, A.; Tromer, E. RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis. In Proceedings of the Advances in Cryptology—CRYPTO 2014—34th Annual Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 2014; Part I, pp. 444–461. [CrossRef]
5. Ferrigno, J.; Hlavác, M. When AES blinks: introducing optical side channel. *IET Inf. Secur.* **2008**, *2*, 94–98. [CrossRef]
6. Brier, E.; Clavier, C.; Olivier, F. Correlation Power Analysis with a Leakage Model. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2004: 6th International Workshop, Cambridge, MA, USA, 11–13 August 2004; pp. 16–29. [CrossRef]
7. Herbst, C.; Oswald, E.; Mangard, S. An AES Smart Card Implementation Resistant to Power Analysis Attacks. In Proceedings of the Applied Cryptography and Network Security, 4th International Conference, ACNS 2006, Singapore, 6–9 June 2006; pp. 239–252. [CrossRef]
8. Chari, S.; Jutla, C.S.; Rao, J.R.; Rohatgi, P. Towards Sound Approaches to Counteract Power-Analysis Attacks. In Proceedings of the Advances in Cryptology—CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999; Wiener, M.J., Ed.; Lecture Notes in Computer Science; Springer: Berlin, Germany, 1999; Volume 1666, pp. 398–412. [CrossRef]
9. Tillich, S.; Herbst, C.; Mangard, S. Protecting AES Software Implementations on 32-Bit Processors Against Power Analysis. In Proceedings of the Applied Cryptography and Network Security, 5th International Conference, ACNS 2007, Zhuhai, China, 5–8 June 2007; pp. 141–157. [CrossRef]
10. Messerges, T.S. Using second-order power analysis to attack DPA resistant software. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Worcester, MA, USA, 17–18 August 2000; Springer: Berlin, Germany, 2000; pp. 238–251.
11. Prouff, E.; Rivain, M.; Bevan, R. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Comput.* **2009**, *58*, 799–811. [CrossRef]
12. Moradi, A. Side-Channel Leakage through Static Power—Should We Care about in Practice? In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2014—16th International Workshop, Busan, Korea, 23–26 September 2014; pp. 562–579. [CrossRef]
13. Joye, M.; Paillier, P.; Schoenmakers, B. On Second-Order Differential Power Analysis. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2005, 7th International Workshop, Edinburgh, UK, 29 August–1 September 2005; pp. 293–308. [CrossRef]
14. ChipWhisperer-Lite. Available online: https://wiki.newae.com/CW1173_ChipWhisperer-Lite (accessed on 24 April 2020).
15. Welch, B.L. The generalization ofstudent's' problem when several different population variances are involved. *Biometrika* **1947**, *34*, 28–35. [PubMed]
16. ChipWhisperer UFO. Available online: https://wiki.newae.com/CW308T-STM32F (accessed on 24 April 2020).
17. Ahmed, E.G.; Shaaban, E.; Hashem, M. Lightweight Mix Columns Implementation for AES. In Proceedings of the 11th WSEAS International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering, MMACTEE'09, Athens, Greece, 28–30 September 2009; World Scientific and Engineering Academy and Society (WSEAS): Stevens Point, WI, USA, 2009; pp. 48–53.