Power Analysis Attacks on SASEBO

January 6, 2010

Research Center for Information Security,
National Institute of Advanced Industrial Science and Technology

Table of Contents

1. OVERVIEW

A series of side-channel attack experiments were conducted on various cryptographic circuits using the standard evaluation boards SASEBO-R and SASEBO-G. The cryptographic LSI mounted on the SASEBO-R and the Xilinx FPGA Virtex-2 used by SASEBO-G are both manufactured with 1.2-V and 130-nm CMOS processes, while they are based on different semiconductor process technologies.

In this evaluation, 128-bit key AES was used for the targeted common-key block cipher. For the SASEBO-R, we selected the AES core on the cryptographic LSI, which implements the S-boxes with the single stage PPRM (Positive Polarity Reed Muler) logic, and tested using 6 different attack methods. The same Verilog-HDL codes of AES circuit were used in both the LSI and the FPGA on the SASEBO-G. Several countermeasures described in "Standard Cryptographic LSI Specification -with Side Channel Attack Counter Measures- Ver. 1.0" were implemented on the SASEBO-G. They are AES4 core with the composite field S-box, and AES8 (MAO), AES9 (MDPL), and AES11 (WDDL) based on AES4 with DPA countermeasures. We intended to test the AES10 (Threshold Implementation) core, but its evaluation was skipped for a lack of operation stability caused by a power system relevant problem. For the targeted public-key cipher, the 1,024-bit RSA implementations on the LSI of SASEBO-R and the FPGA of SASEBO-G were used and applied with SPA attacks using various input data patterns. The following sections expound these attack methods and the experimental results obtained.

2. POWER ANALYSIS ATTACK AGAINST AES

2.1 Attack Methods

Table 1 lists well-known power analysis attack methods applicable to AES circuits. The experiments targeted the AES circuits described in "ISO/IEC 18033-3 Standard Cryptographic LSI -with Side Channel Attack Countermeasures- Specification Version 1.0". Each analysis was performed on the power trace segment of the 10th (final) round or the segment that reflects register switching at the moment of data output, identifying 11 peaks on the power trace as shown in Figure 1.

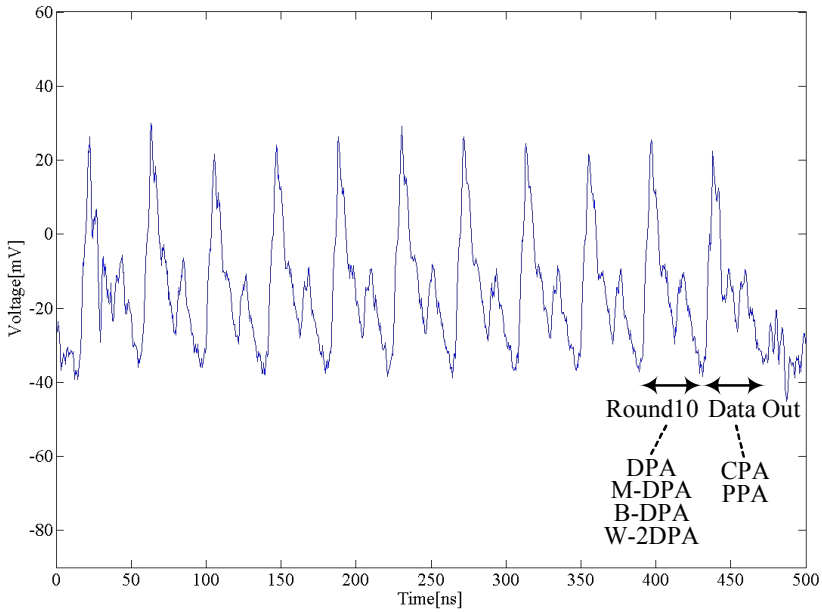


Figure 1 Power trace segments targeted by the attack methods for the AES circuit (PPRM1) of the SASEBO-R's LSI

Table 1 Attack methods against AES circuits

Attack Method	Description	Attacked Segment
DPA	The most basic and versatile attack, which analyzes a correlation between a set of power traces and a particular intermediate 1-bit value corresponding to a guessed partial key by computing a difference of the averages of two power trace groups distinguished by the bit value.	10^{th} round
M-DPA	An attack that examines the correlation between power traces and the Hamming weight of a particular intermediate multi-bit value corresponding to a guessed partial key by computing a difference of the averages of two power trace groups distinguished by whether the Hamming weight is equal to or larger than a threshold value or not. The accuracy of the attack highly depends on the circuit implementation.	10^{th} round
B-DPA	A versatile attack that combines the DPA results for each bit of a particular intermediate multi-bit value corresponding to a guessed partial key.	10^{th} round
CPA	An attack that analyzes a correlation between power traces and the Hamming distances of the transitioning of a register that stores a particular intermediate value corresponding to a guessed partial key. If a circuit lacks a countermeasure, this method would be successful with as few as or even less than 1/10 power traces of that B-DPA requires.	Data output
PPA	An attack extended from CPA by weighing to the Hamming distances. However, no efficient coefficient determining methods have been proposed.	Data output
M2-DPA	An attack that analyzes a correlation between two certain segments in the power traces. The attack accuracy depends on the implementation.	10^{th} round
W2-DPA	A versatile attack that computes a difference of the means of power trace squares, not of the means of power traces as in DPA.	10^{th} round

The following outline each of the attack methods. Equation (1) represents the meanings of symbols used in those explanations. $G_{condition}$, $N_{condition}$, and $\overline{W}_{condition}$ denote a set of the power traces that meet the *condition*, the number of the traces, and the mean of the traces, respectively.

$$\begin{cases} G_{condition} = \{W_i, i \in 1...N \mid condition\} \\ N_{condition} = card(G_{condition}) \\ \overline{W}_{condition} = \sum_{G_{condition}} W_i / N_{condition} \end{cases} \quad (1)$$

- **Differential Power Analysis (DPA)**

DPA, proposed by Kocher, *et al.*[1)], evaluates the differences between the means of power traces indicated by the following equation to estimate the secret key:

$$\Delta(b) = \overline{W}_{b=1} - \overline{W}_{b=0} \quad (2)$$

where b is a bit of an intermediate variable used in the cryptographic algorithm and $\Delta(b)$ is called

the DPA trace, representing the difference between the mean powers for b=1 and b=0. DPA computes the bit value b using known plaintext or ciphertext and a partial key hypothesis (namely guessed). Hereinafter, b will be called the *selection bit*. $\Delta(b)$ is also called a *selection function* for selecting the right key among key candidates. The most typical DPA attack against an AES implementation makes a guess for a partial key byte $\boldsymbol{k}$ at the final round and calculates the following from the known ciphertext byte $\boldsymbol{c}$:

$$\boldsymbol{b} = \{b_7, \cdots, b_1, b_0\} = S^{-1}(\boldsymbol{c} \oplus \boldsymbol{k}) \tag{3}$$

where S^{-1} is an 8-bit S-box used in the InvSubBytes function of AES. The attacker computes each $\Delta(b)$ for all the possible 256 patterns of the 8-bit partial key $\boldsymbol{k}$ and determines the correct key $\boldsymbol{k}$ that makes the maximum $\Delta(b)$. In this example of DPA, the selection bit may be chosen from ANY one of the 8 bits of $\boldsymbol{b} = \{b_7, \cdots, b_1, b_0\}$ to obtain the corresponding $\Delta(b)$ and thus to derive the same partial key.

DPA on a logic circuit succeeds only if the values of a circuit node designated as a selection bit cause a power consumption difference. For example, reference 2) shows that when an input port of a non-linear gate such as NAND or NOR is chosen as a selection bit, a deviation in transition probability arises on the subsequent logic circuit including that gate. References 2), 3), and 4) report that DPA may work effectively even if countermeasures such as random masking and complementary logic are deployed. Literature 5) shows an example of a successful DPA attack against an ASIC circuit that implements the random masking countermeasure method. The selection function of equation (3) is used to estimate the partial key given to the previously adjacent operation using the corresponding ciphertext being output and the power trace measured during the time segment of the final round. Literature 6) indicates that a selection function associated with the output of the first round SubBytes or its linear transform can be used if plaintext is selectively settable.

- **Messerges' multi-bit DPA**

As an extension to equation (7.1), several attack methods that exploit multiple selection bits have been proposed in reference 7) and 8) and can be classified as multi-bit DPA (M-DPA).

In reference 7), Messerges *et al*. present a power trace grouping technique that judges whether the Hamming weight H_W of d selection bits is equal to or higher than d/2, as shown in equation (4).

$$\begin{cases} G_0 = \{W_i, i \in 1...N \mid H_W(\boldsymbol{b}) < d/2\} \\ G_1 = \{W_i, i \in 1...N \mid H_W(\boldsymbol{b}) \geq d/2\} \\ \Delta(Hw(\boldsymbol{b})) = \overline{W}_{G_1} - \overline{W}_{G_0} \end{cases} \tag{4}$$

If more than one of the multiple bits chosen as a selection function contributes to the magnitude tendency of power consumption of the logic circuit in the same way, the attack accuracy of M-DPA would be superior to that of DPA. Unlike DPA, the presence of a difference does not necessarily improve the attack accuracy. For example, to take equation (3) as a selection function, every bit of $\boldsymbol{b} = \{b_7, \cdots, b_1, b_0\}$ has to show the same trend of whether the power consumption increases or decreases depending on its value of 0 or 1. Thus, if the power consumption is larger when b_0 is 1 than for 0, the power consumption should also be larger when each of the other bits is 1 as opposed to 0. In other words, it is important for achieving a high attack accuracy that the polarity of the DPA trace $\Delta(b)$ is invariant regardless of the bit position of the selection function. Since such a condition is met mainly when the SubBytes function consists of two-stage logic with AND-XOR combination, it is considered that there are narrower conditions and thus fewer targets in which an M-DPA attack is effective than those of DPA.

- **Bevan's multi-bit DPA**

Bevan *et al*. propose an attack method that makes use of a sum of the absolute values of DPA traces calculated with multiple selection bits as shown in equation (5) [8]. Hereinafter the attack method using this equation will be called B-DPA.

$$\sum_{b_i \in \boldsymbol{b}} |\Delta(b_i)| \tag{5}$$

In B-DPA, firstly multiple DPAs take place employing applicable multiple selection bits; secondly their results are combined by equation (5). If more than one of the multiple bits chosen as a selection function contributes to the difference in the power consumption of the logic circuit, the attack accuracy of B-DPA may be superior to that of DPA. Unlike M-DPA, the bit value does not have to determine the magnitude tendency of power consumption. However, if only a small number of bits in the selection function contribute to the power consumption difference, its accuracy turns out to be less than that of DPA.

- **Correlation Power Analysis**

Correlation Power Analysis (CPA) is a powerful attack method proposed by Brier *et al*[9]. CPA makes use of equation (6) to compute a correlation between a Hamming distance H_D at a register, which can be calculated from the estimated partial key $\boldsymbol{k}$, and the corresponding power consumption.

$$\begin{cases} G_j = \{W_i, i \in 1...N \mid H_D(\boldsymbol{b}) = j\} \\ \sigma_{W,H_D} = \sum_{j=0}^{d} j \cdot N_j \cdot \overline{W}_j - \overline{W} \cdot \overline{H_D(\boldsymbol{b})} \\ \rho(\boldsymbol{b}) = \dfrac{\sigma_{W,H_D}}{\sigma_W \sigma_{H_D}} \end{cases} \tag{6}$$

where d is the length of a register in which the stored value can be calculated from the estimated key; power traces are sorted into d+1 groups associated with the Hamming weights 0 to d of the register values. CPA determines the value of $\boldsymbol{k}$ as the right key that results in the largest value of $\rho(\boldsymbol{b})$. Let the register values before and after a transition be $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and the Hamming weight of $\boldsymbol{x}$ be $H_W(\boldsymbol{x})$. We obtain $H_D(\boldsymbol{b}) = H_W(\boldsymbol{x} \oplus \boldsymbol{y})$. Note that $\boldsymbol{x}$ and $\boldsymbol{y}$ can be derived from a known ciphertext output (or plaintext output) and the estimated partial key $\boldsymbol{k}$.

For a successful CPA attack, the power consumption of the logic circuit connecting with the register where a Hamming distance is computed must have a correlation with the number of transitioning bits of the register. This condition is normally met in a regular logic circuit. Due to a realistic computation limitation for analysis, the partial key length for Hamming distance calculation should be ranging from about 8 to 16 bits. The following equation is the selection function mostly widely used for a CPA attack against AES:

$$H_D(\boldsymbol{b}) = H_W(S^{-1}(\boldsymbol{c}_i \oplus \boldsymbol{k}_i) \oplus \boldsymbol{c}_j) \tag{7}$$

where $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$ are ciphertext (or plaintext) output bytes such that the jth position is moved to ith position by the function ShiftRows (or InvShiftRows). For this selection function to be effective, the intermediate value at the 9th round and the result of the 10th round (that is the ciphertext or plaintext output) have to be stored in the same register. Equation (7) involves power traces delayed by one cycle from ones in DPA shown in equation (3). This is because transitions of equation (7) take place when the register stores the ciphertext (or plaintext).

- **Partitioning Power Analysis**

Partitioning Power Analysis (hereinafter it will be called PPA), proposed by Le *et al.*[10], is an extension to CPA, in which the attacker adaptively sets the weight coefficient a_j for each Hamming distance depending on attack targets. This adaptive method enables a flexible association between Hamming distances and power traces. Literature 10) has left finding efficient coefficients as an open problem. The attack principle is the same as CPA except the normalization is not counted in equation (8) rather in equation (6).

$$\begin{cases} G_j = \{W_i, i \in 1...N \mid H_D(\boldsymbol{b}) = j\} \\ \Sigma_H(\boldsymbol{b}) = \sum_{j=0}^{d} a_j \cdot \overline{W}_{G_j} \end{cases} \tag{8}$$

- **Messerges' Second-Order DPA**

Messerges has proposed a second-order DPA (hereinafter we will refer to it as M2-DPA) that uses the selection function shown in equation (9) focusing on each power consumption segment of two separated cycles.

$$\begin{cases} \overline{S}_0 = \left| \overline{W}_{t,b=0} - \overline{W}_{t',b=0} \right| \\ \overline{S}_1 = \left| \overline{W}_{t,b=1} - \overline{W}_{t',b=1} \right| \\ \Delta_{2\text{nd}}(\boldsymbol{b}) = \overline{S}_1 - \overline{S}_0 \end{cases} \tag{9}$$

where $\overline{W}_{t,condition}$ is the average of the power traces at the *t*th cycle that meets the *condition*.

M2-DPA is an attack method targeting cryptographic implementations that employ the random masking countermeasure. For a software implementation of a countermeasure that XORs plaintext ***P*** with a random number ***R*** at the *t*th cycle and XORs the result of the *t*th cycle with a key ***K*** at the *t*'th cycle, it is considered that the effect of the random number ***R*** can be canceled by focusing on the power consumption difference between the *t*th cycle and the *t*'th cycle. For a logic circuit implementation, the attack is applicable as well, if the random number masking cycle is successfully separated from the key addition cycle. However, attacking AES with this method can not be generalized because it depends on the countermeasure adopted.

- **Waddle's Zero-Offset Second-Order DPA**

Waddle *et al*. have proposed a few second-order DPA attacks that are extensions to DPA in reference 12). The most basic attack method among them is Zero-Offset 2 DPA defined by equation (10). (Hereinafter it will be called W2-DPA.) Contrary to DPA, which computes a difference of the means of two power trace groups sorted by a selection bit, W2-DPA calculates a difference of the means of squares. If a countermeasure is adopted such that the average of power traces appear uniform regardless of a selection bit by means of random numbers or the like, and if, however, different power trace deviations are produced depending on the selection bit, it is possible to be attacked by W2-DPA.

$$\begin{cases} \overline{W}^{(2)}_{condition} = \Sigma_{G_{condition}} (W_i)^2 / N_{condition} \\ \Delta_{2\text{nd}}(\boldsymbol{b}) = \overline{W}^{(2)}_{b=1} - \overline{W}^{(2)}_{b=0} \end{cases} \tag{10}$$

2.2 Experimental Results

Table 2 lists the measurement conditions of the power analysis attack experiments conducted for AES. Power traces were measured as electric potential differences at both ends of each of the resistors of 3.3 Ω and 0.1 Ω inserted in the VDD lines of the cryptographic LSI and FPGA, respectively. Xilinx ISE 9.2i was used to implement the AES circuits on the FPGA. A series of analyses were performed for each S-box associated byte of the final round key of 16 bytes for different AES implementations and attack methods, with random plaintext inputs provided and 10,000 or 100,000 samples of power traces obtained. For DPA, every attack took place by making 8 power trace average differentials associated with each of the 8 bits of each S-box, and subsequently computing a sum of them. For PPA, every attack took place by obtaining correlation coefficients, setting -8, -6, -4, -2, 0, 2, 4, 6, 8 to the weight coefficients $a_0,\ldots,a_8$ in equation (8), respectively, associated with the Hamming distances 0 to 8 determined for an 8-bit intermediate value.

Table 2 Measurement Conditions

Measurement Factor		Condition
Digital oscilloscope		Agilent MSO8104A
Sampling frequency		2GSample/sec
Probe		Agilent 1130A
Probe head		Agilent E2695A SMA
Stabilized power supply		3.3 V
Operating clock frequency		24 MHz
Voltage measurement point	Cryptographic LSI	Both ends of the 3.3 Ω resistor inserted in the core VDD line
	FPGA (xc2vp30)	Both ends of the 0.1 Ω resistor inserted in the core VDD line
Secret key		2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C
Final round key of attack target		D0 14 F9 A8 C9 EE 25 89 E1 3F 0C C8 B6 63 0C A6

- **Attacking to the cryptographic LSI on the SASEBO-R**

The primary objective of the attack experiments executed on the cryptographic LSI is performance comparison between the attack methods. The AES2 (PPRM1) core, which has the S-boxes implemented with single-stage PPRM logic, which is of AND-XOR logic, was selected as the target because it consumes the most power and is thus the easiest to compromise. The attack methods used are DPA, CPA, W2-DPA, M-DPA, M2-DPA, and PPA.

Figure 2 shows the results of DPA with 10,000 traces. Differences of average power (DPA traces) for every 8-bit partial key hypotheses are shown for each of the 16 S-boxes (S0 is on the MSB side and S15 is on the LSB side). For every S-box, an obvious correlation peak emerges, indicating a proper key guess Figure 3 shows the variations of the rank of the correct key (the magnitude of difference) with the vertical axis in a log scale against the number of traces on the horizontal axis. While there are small variations among the S-boxes, it is observed that almost every partial key is estimated correctly in the very early stage (with a small number of traces).

Figure 4 and Figure 5 are the results of CPA with the same condition as in above-mentioned DPA. Like DPA, every partial key is estimated correctly in the early stage, even with rather small numbers of traces as a whole. Comparison between two or more attack accuracy graphs such as S0 and S10 in each of the attack methods indicates that the amount of information leakage is not uniquely determined by the S-box. It is important for improving the attack accuracy to match the implementation of the attack to the power model determined by the attack method. In general, the detailed implementation method of the cryptographic algorithm in a particular system is not disclosed. As shown in the experiment, even though the 16 S-boxes were synthesized from the same Verilog-HDL code, not only does the estimation accuracy vary, but also differences occur as key or data patterns change. Therefore, it is necessary to use attack methods based on various power models for security evaluations.

Figure 6 through Figure 9 represent the results of W2-DPA, M-DPA, M2-DPA, and PPA on S0. Since DPA and SPA, which these attack methods are based on, succeeded with high accuracy, these

graphs also show similar results. The experiment extracted all the correct partial keys estimated with from 1,000 to 4,000 traces, targeting the S-box implementation with the single-stage PPRM logic. In addition, we also obtained all the correct keys on each of the AES circuits with other S-box implementations (without a DPA countermeasure) while with different numbers of traces.

- **Attacking the FPGA implementations on the SASEBO-G**

Figure 10 and subsequent figures show the attack experiment results on the AES circuit implemented on the SASEBO-G's FPGA. Although the xc2vp7 on the SASEBO-G was originally the only FPGA for cryptographic circuit implementation, some AES implementations with a countermeasure became too large to fit within a single FPGA. Thus, we used the xc2vp30, which usually serves as the control FPGA, to implement each of the AES circuits that have S-boxes constructed over the composite field. Although we were able to implement the AES circuit with the DPA countermeasure of Threshold Implementation, which showed the largest size, on the FPGA, we omitted its evaluation result because the core voltage became unstable and it did not operate properly.

Figure 10 through Figure 15 show the experimental results of DPA, W2-DPA, and CPA with 100,000, 100,000, and 10,000 traces, respectively, on the AES circuit without a DPA countermeasure. The lower S/N ratio in the power traces mainly resulted in the lower accuracies compared to the results with the cryptographic LSI. Even with such a low quality of power traces, CPA effectively worked, yielding higher accuracies than DPA and W2-DPA by an order of magnitude or more.

Figure 16 through Figure 23 show the results of DPA, W2-DPA, and CPA using 100,000, 100,000, and 10,000 traces, respectively, on the AES circuit that uses the Masked-AND Operation (MAO) countermeasure. The DPA attempts on the S-boxes shown in Figure 16 and Figure 17 all failed to estimate the key. However, we performed further inspection on each of the 8 DPA traces before making a sum, and found that the bit-1 and bit-6 of the inputs of some S-boxes are particularly weak against DPA. Accordingly, we obtained the results of DPA experiments only for the 2 bits as shown in Figure 18 and Figure 19. It is observed that the 6 S-boxes S2, S3, S4, S9, S10, and S13 were compromised and exposed the key. This implies that, for the security conformance testing, knowledge of the correct key can be exploited to find the vulnerability of the module under test. In general, an attacker, who is a third person, does not have such knowledge about the correct key and it is impossible for him to construct the right selection function based on the knowledge. However, it should be noted that, in case an attacker possesses a target module, he may be able to put the key into it, perform an analysis on it, and exploit the result to attack another's same module. In addition, if the rank of the correct key is not highest but close, and if there exists some vulnerability in the implementation, the accuracy of key estimation may increase as the number of power traces increases.

In the Masked-AND countermeasure, since a single bit random number affects two or more signals, it is in theory possible for W2-DPA to attack it successfully. As shown in Figure 20 and Figure 21, however, all the attacks failed to compromise the circuit in the experiment. This may be due to large variations of signal delays and insufficient numbers of power traces. Furthermore, the CPA results shown in Figure 22 and Figure 23 indicate that data masking caused incorrect calculation of Hamming distances and successfully protected the key from every attack in the experiment.

Figure 24 and Figure 29 are the result graphs of attacks against the WDDL version of the AES circuit. Figure 24 and Figure 25 represent the DPA results for the idle phase (Precharge), and Figure 26 and Figure 27 show the DPA results for the active phase (Evaluation), each uses 100,000 power traces for analysis. Figure 28 and Figure 29 show the CPA results for the active phase with 10,000 traces. The results show that DPA successfully compromised some S-boxes for both the idle and active phases. Recall that WDDL is a countermeasure that attempts to yield no data-dependent power difference by causing switching on either signal of every signal pair for any input. In reality, however, there exists a difference among switching speeds of the primitive gates such as AND and OR that engage in activities of a signal pair. Also, there are variations in the parasitic capacitances and resistances including the effect of signal wires. Accordingly, a real circuit of WDDL still yields information leaks. Significant leaks observed particularly at S6 and S15 for both the idle and active

phases imply existence of exploitable imbalance between the involved signal pairs in the circuit. Like the Masked-AND case, a key estimation with even a higher accuracy would also be possible by identifying and summing the bits showing large leaks.

Recall that CPA is an attack method that leverages Hamming distances determined within the bit width of a basic operation (in this experiment, this is 8-bits of an S-box) by switching activities at a register or a set of signal wires. However, the experiment for WDDL's active phase used Hamming weights because the Hamming weight of the operation result will be the Hamming distance. Another characteristic of WDDL which leads us to use Hamming weight involves the signal pair being precharged during every idle phase, which causes the bits of the register to switch not by the difference between the result values of the previous and new operation, but by the change between the fixed precharged value (namely all the bits are zeros) and the new operation result value. As already mentioned, WDDL may cause significant differences in powers mainly due to differences in the output signal delays of the AND/OR gates that form signal pairs. On the contrary, however, it also may leave little differences in powers because the delays have irregular relationships in each signal pair and the correlation between the Hamming distance (also Hamming weight) of every 8 bits and the corresponding power consumption can be small enough such that the effects of the bits may almost cancel each other. Figure 28 and Figure 29 show that the attempts to derive the key all failed. However, it also appears that the rank of the correct key increases as the number of traces increases for S10 and S15. While this CPA experiment acquired only 10,000 power traces, another CPA experiment that obtained 100,000 traces in a different measurement environment resulted in successful attacks while with lower accuracy than DPA.

Figure 30 and Figure 35 represent the attack results with DPA and CPA against the MDPL version of the AES circuit. Similar to WDDL, DPA took place for both the idle phase (shown in Figures 30 and 31) and the active phase (shown in Figures 32 and 33) with 100,000 traces and CPA was performed for the active phase (shown in Figures 34 and 35) with 10,000 traces. Because MDPL consumes a lot of power and thus these experiments yielded very low S/N ratios, every attack failed and a subtle information leak could not be distinguished even by individual inspection of the analysis waveforms associated with the correct key. We performed another series of experiments by implementing the same Verilog-HDL code on the SASEBO-G under a different measurement condition with DPA and CPA each with 100,000 power traces. While some of the DPA experiments showed successful attack results for some bits of the selection function, all the CPA experiments failed. CPA with as many as 1 million power traces could successfully extract the correct key. However, for a guideline to security evaluation, further analysis is not necessary if vulnerability has been found against a single attack method.

- **Summary of the attack methods and countermeasures for AES circuits**

From the experimental results, we found that for AES circuits without countermeasures, attack accuracy of CPA, which exploits the correlation between Hamming distance based on data switching and power consumption, is obviously much higher than those of the other types of attacks. However, since the power model does not successfully match to the circuits with countermeasures, the accuracies are substantially reduced. On the other hand, for the most basic analysis method DPA, the model is simple enough and applicable to compromise various countermeasures effectively. Note that the above discussion is good when the power model of the attack target is unknown. In an evaluation test, however, the tester may be able to obtain the information on the implementation methods including countermeasures. Thus, taking advantage of the information to construct a correct power model based on the actual circuit's characteristics would make attacks (namely evaluation) with even higher accuracies possible.

It turned out that the countermeasures implemented on the FPGA effectively make power analyses difficult. The strength relationship among them is illustrated as follows:

MDPL > MAO > WDDL

Given that every countermeasure assumes a proper control of conditions such as timing delays and maintaining the balances of parasitic capacitances and resistances, the same requirement can hardly be met in an FPGA implementation. So, it should be understood that these experiments do not

necessarily indicate that each countermeasure has the strength in the order shown above or that DPA is always effective against the countermeasures with a sufficient number of power traces. Furthermore, not only the effect of the countermeasure algorithm employed but also the S/N ratio in the power traces significantly impact on the analysis result. Therefore, it is also important for security evaluation to include examination of the implementation forms and measurement environment.

Again, the two important things in which security evaluation differs from an attack are:

1. The detailed information on the implementation methods is available.
2. The correct key is known.

Utilizing this knowledge makes even power model construction possible as shown in the DPA against MAO for the bit-1 and bit-6. We consider a module to be adequately safe from attack, based on failure to find vulnerabilities in the more advantageous environment of security evaluation. It is also indicated that if vulnerability is found in the countermeasure with such an advantageous analysis, it does not necessarily mean a dangerous and useless countermeasure. Rather, the countermeasure is effective enough as long as it increases the required number of power traces for successful attacks over an implementation without the countermeasure. For example, an implementation that is known to the public to be able to be compromised with 1 million power traces would be very dangerous. On the contrary, even if the fact that information about a partial key in a cryptographic module leaks on a particular bit has been exposed by an evaluation experiment for a known key with 10,000 power traces, exploiting information on the implementation, if the module cannot be compromised with even as many as 1 million traces without such information, the module would be considered safe. This suggests that security evaluation should also take into account the costs of attacks for which attackers have no access to the crucial information to which the evaluator has access.

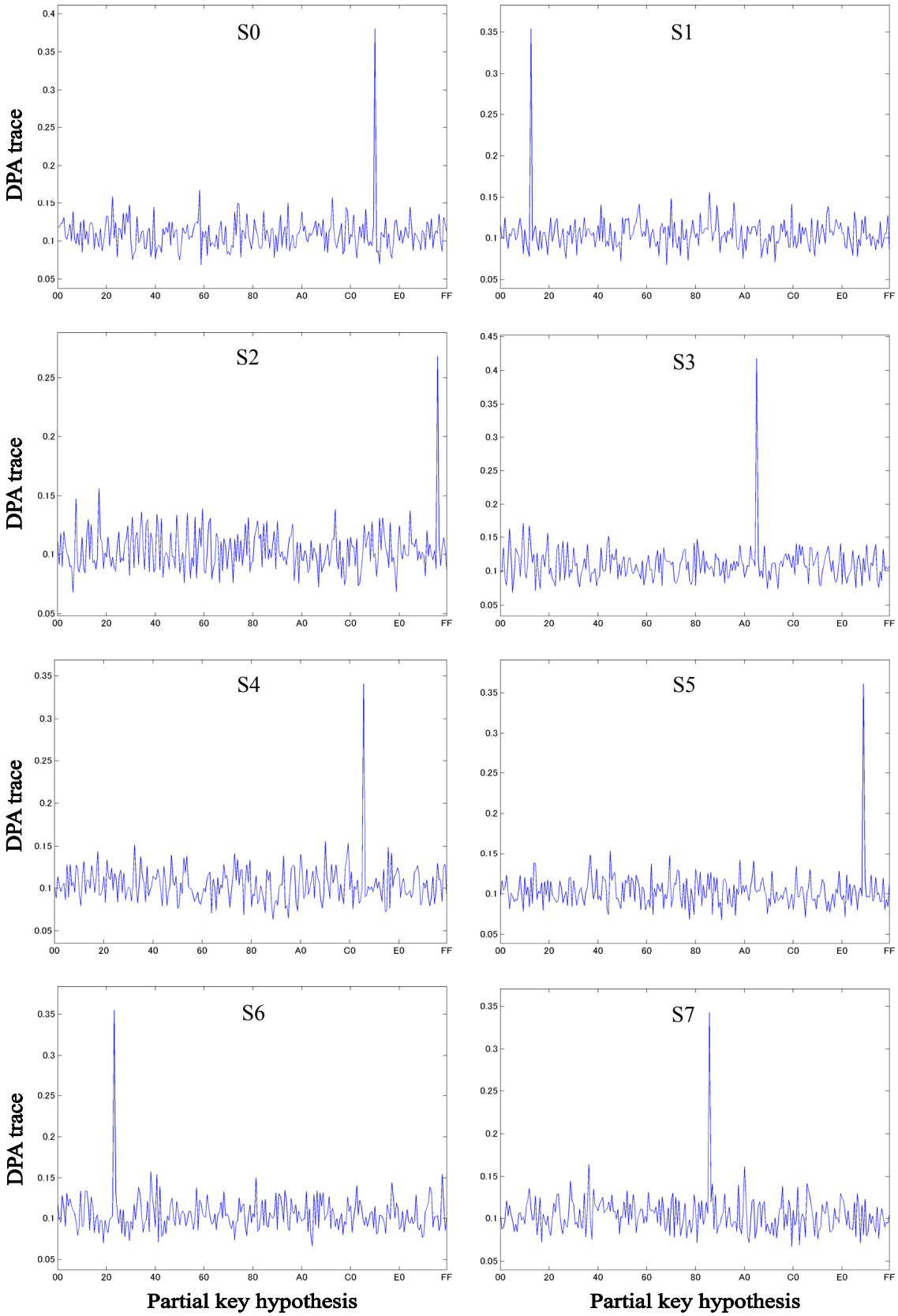
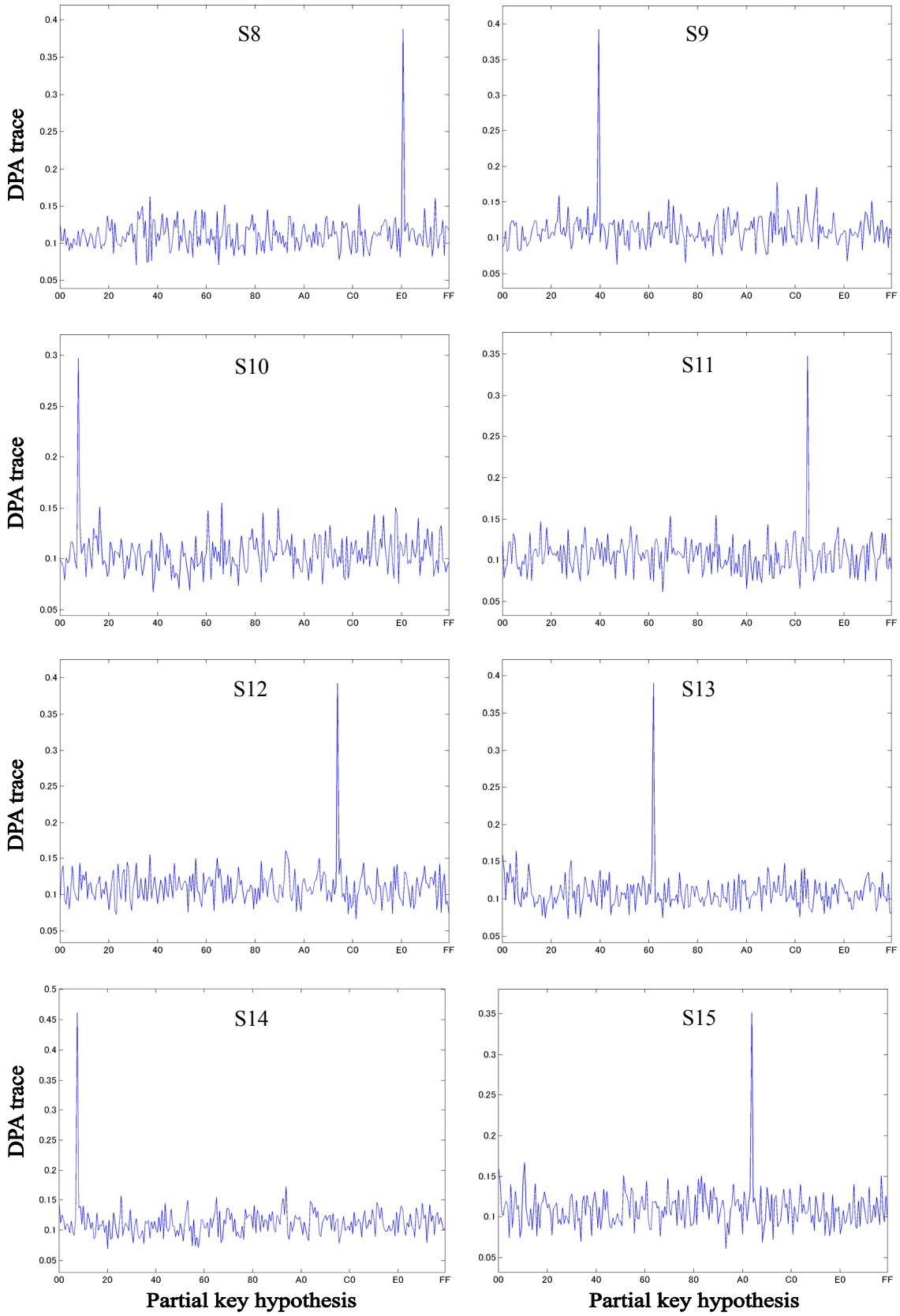


Figure 2-1 Average power differences (DPA traces) from DPA on the AES circuit (PPRM1) on the SASEBO-R



FigureFigure 2-2 Average power differences (DPA traces) from DPA on the AES circuit (PPRM1) on the SASEBO-R

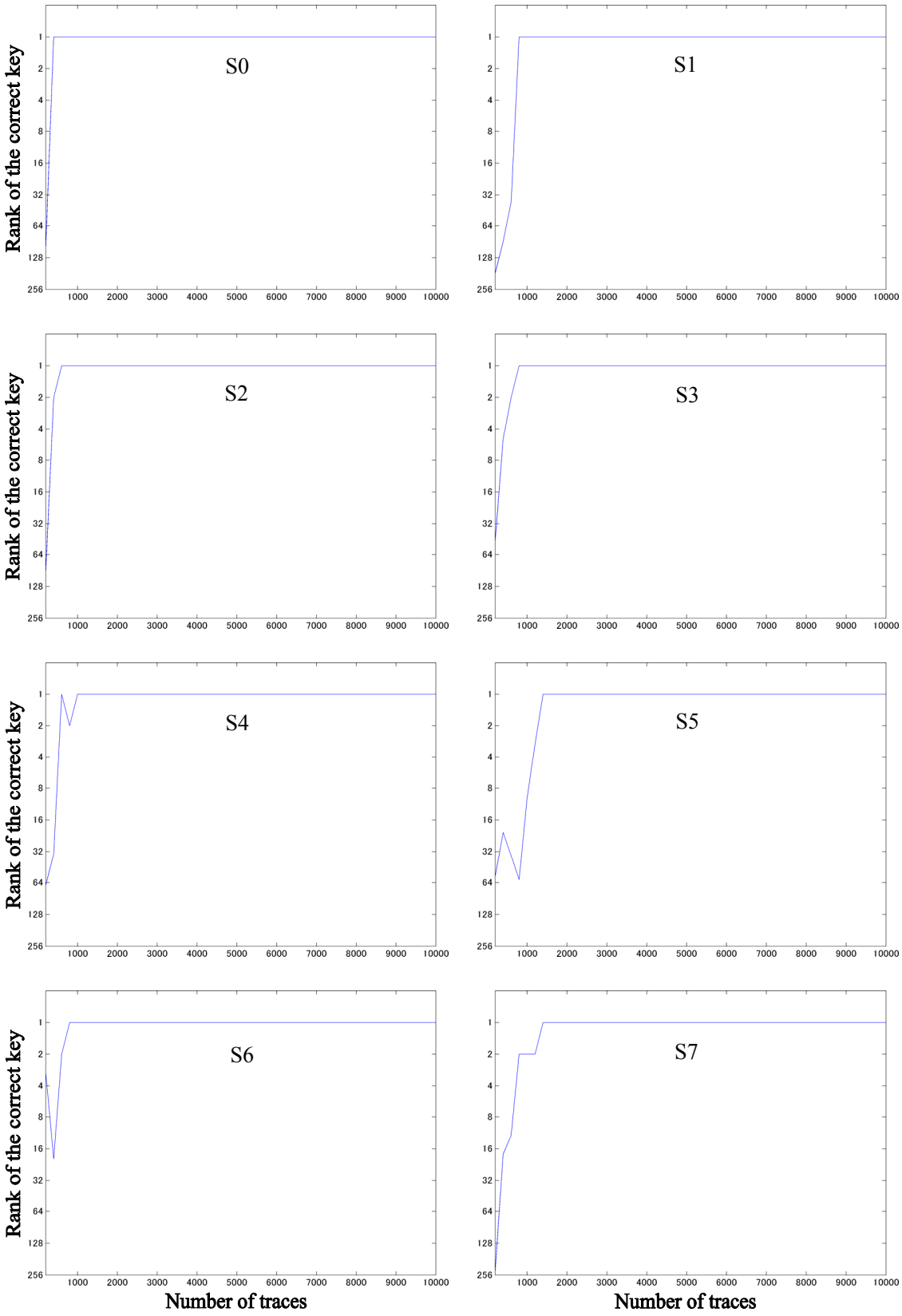


Figure 3-1 Number of power traces versus accuracy of DPA on the AES circuit (PPRM1) on the SASEBO-R

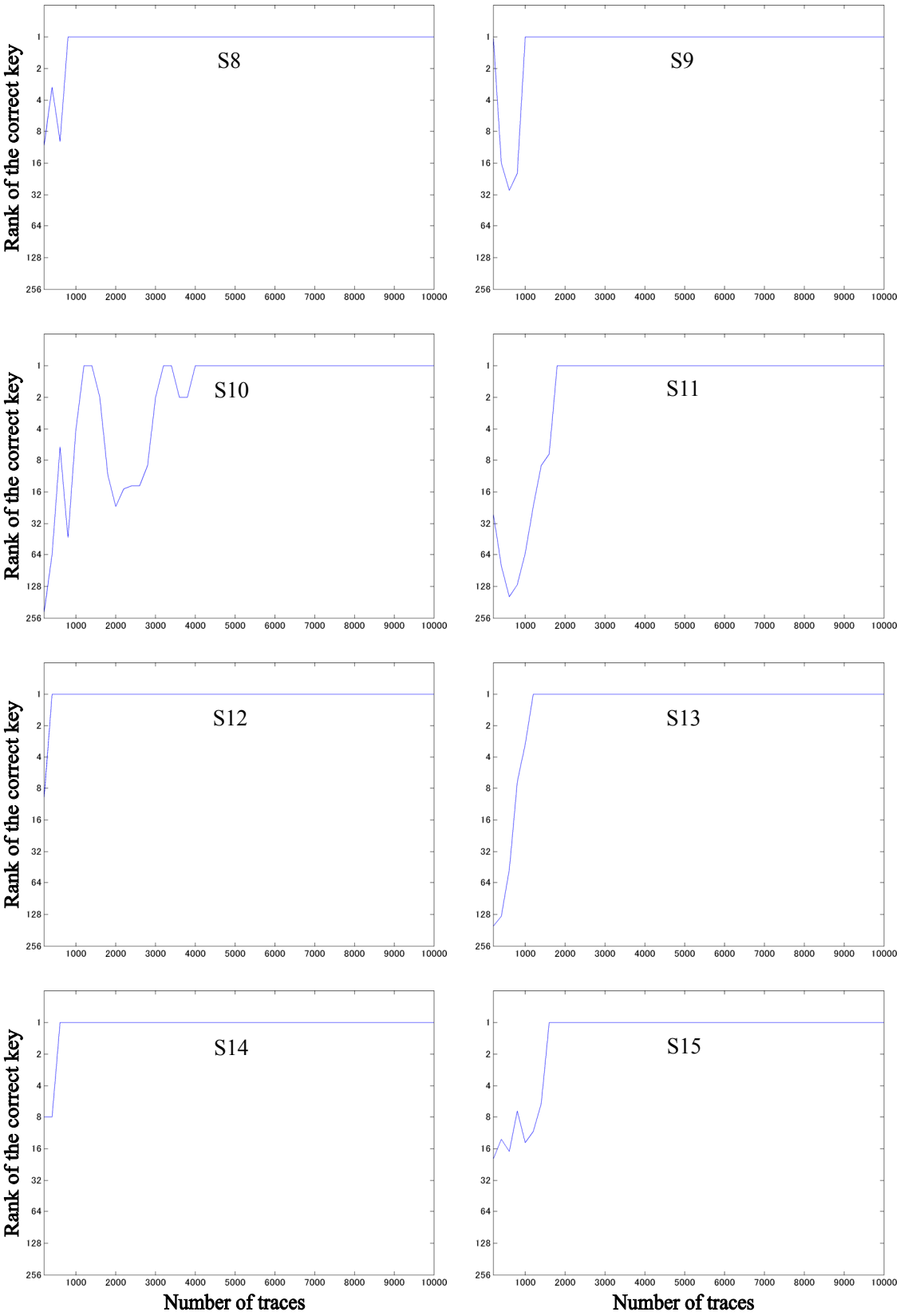


Figure 3-2 Number of power traces versus accuracy of DPA on the AES circuit (PPRM1) on the SASEBO-R

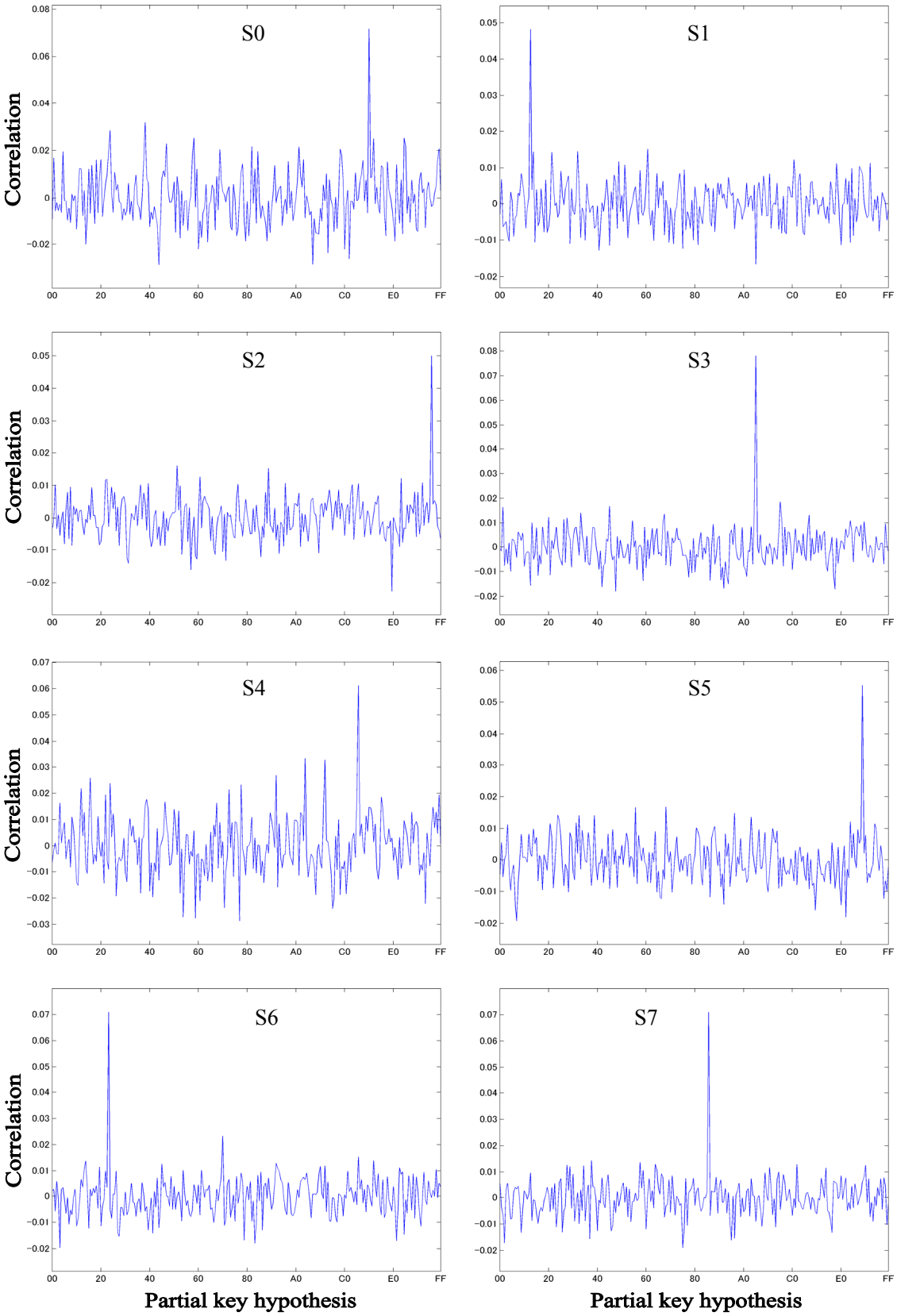


Figure 4-1 Correlation coefficients in CPA on the AES circuit (PPRM1) on the SASEBO-R

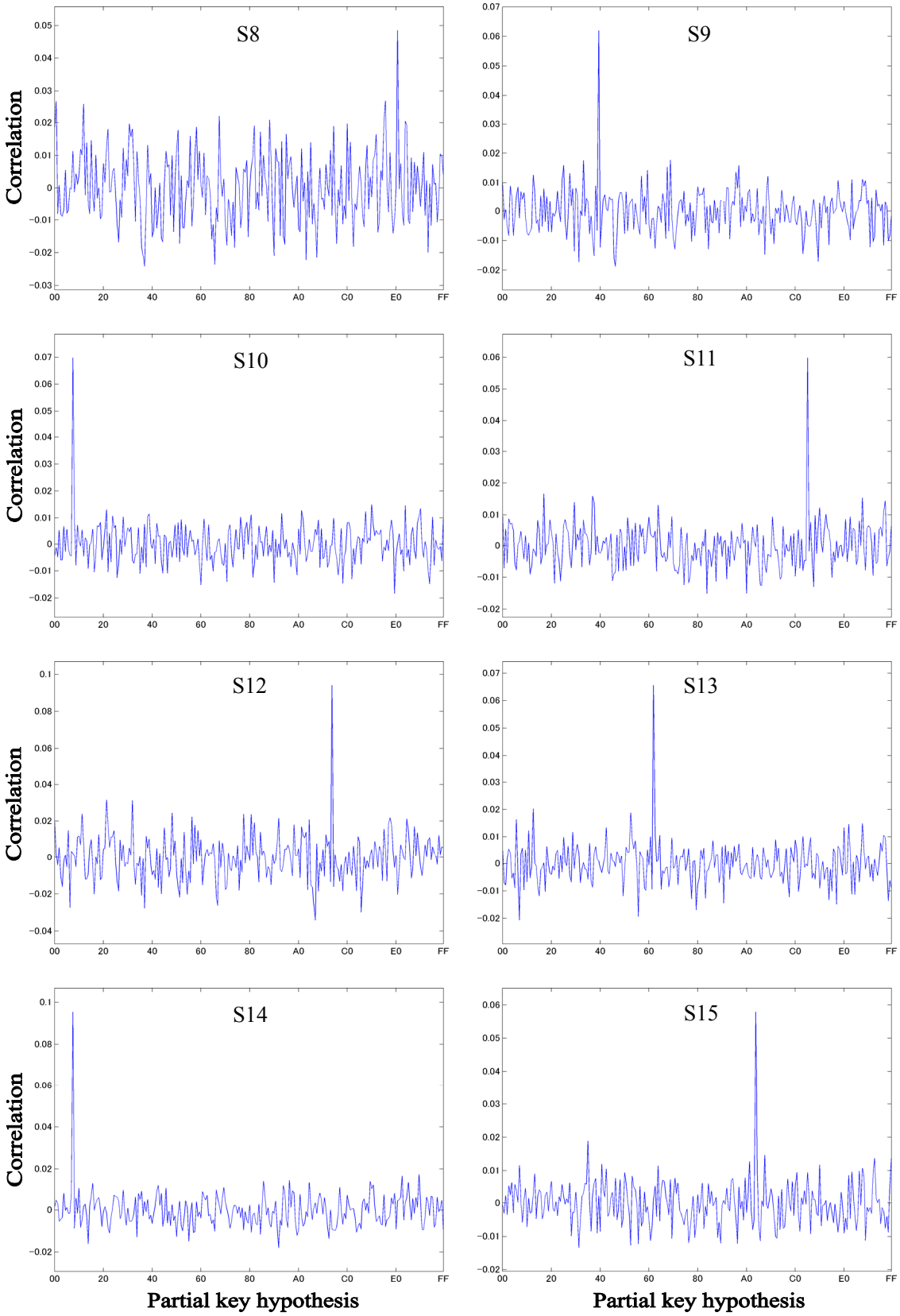


Figure 4-2 Correlation coefficients in CPA on the AES circuit (PPRM1) on the SASEBO-R

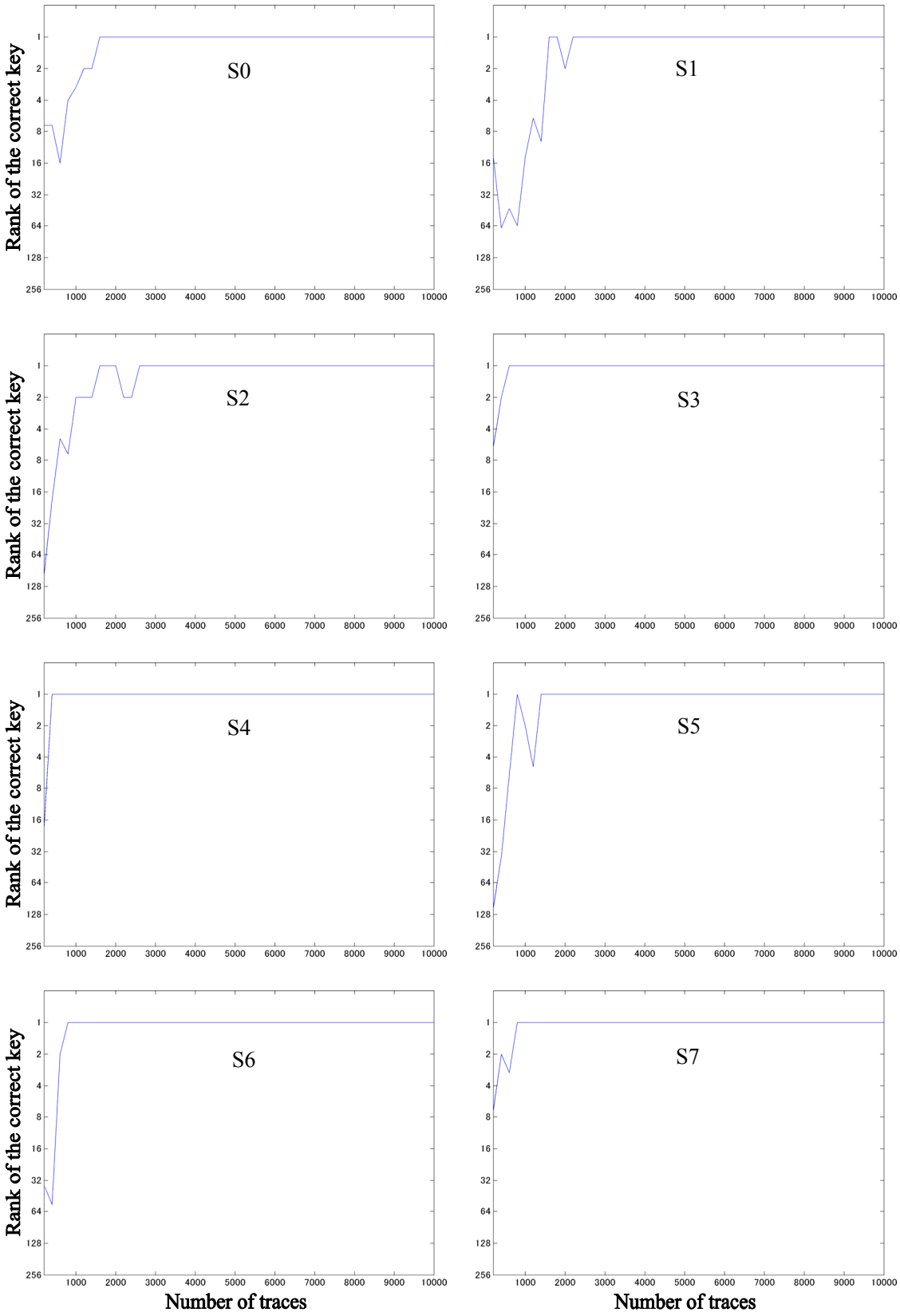


Figure 5-1 Number of power traces versus accuracy of CPA on the AES circuit (PPRM1) on the SASEBO-R

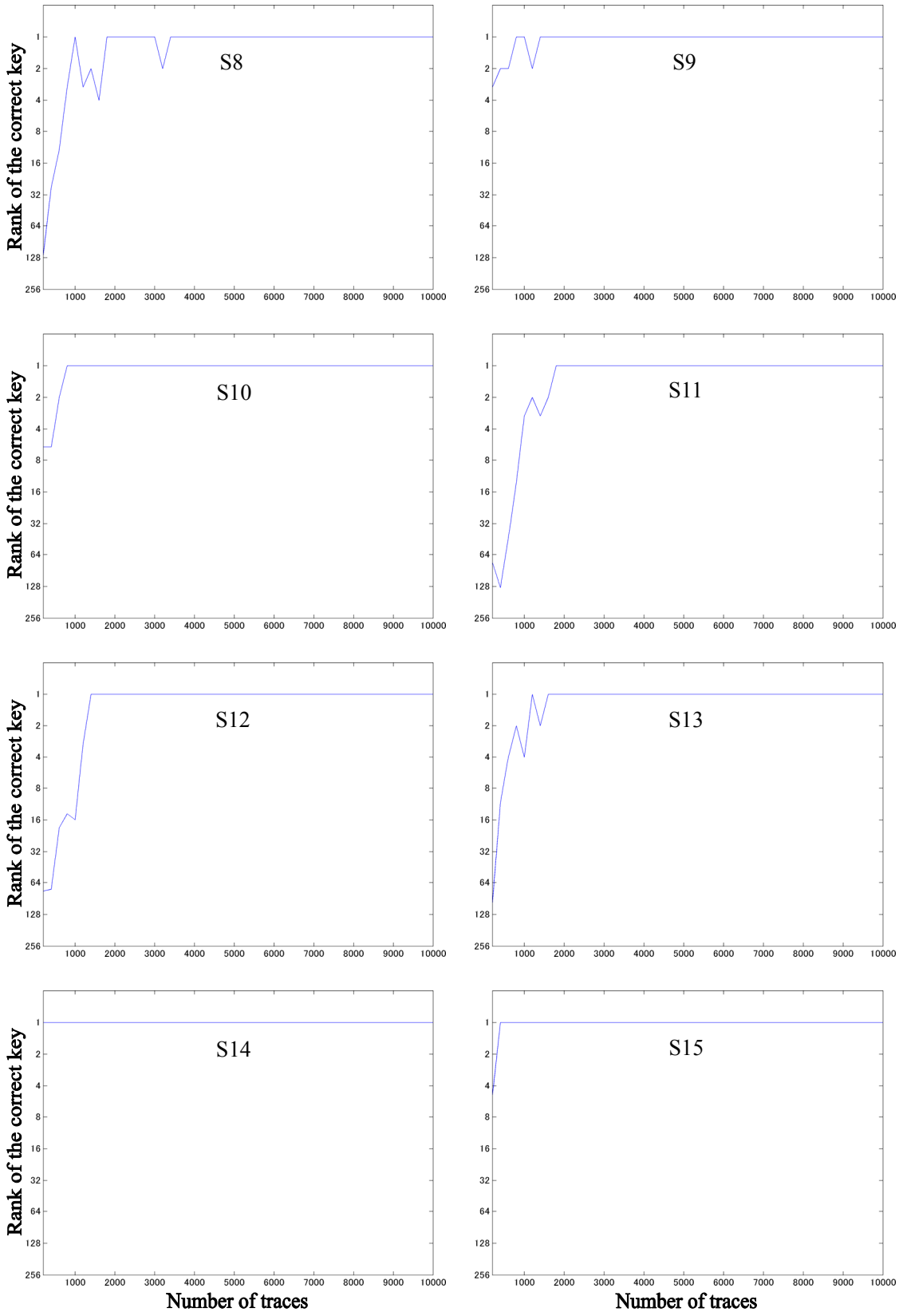


Figure 5-2 Number of power traces versus accuracy of CPA on the AES circuit (PPRM1) on the SASEBO-R

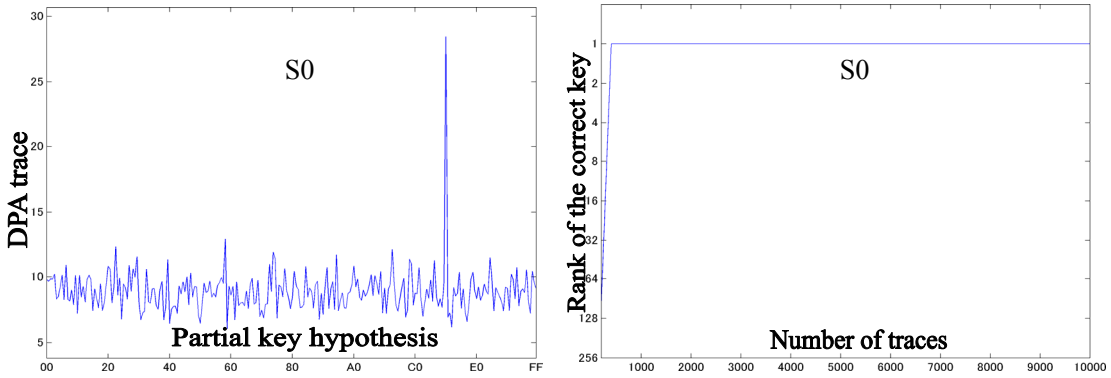


Figure 6 Result of W2-DPA on the AES circuit (PPRM1) on the SASEBO-R

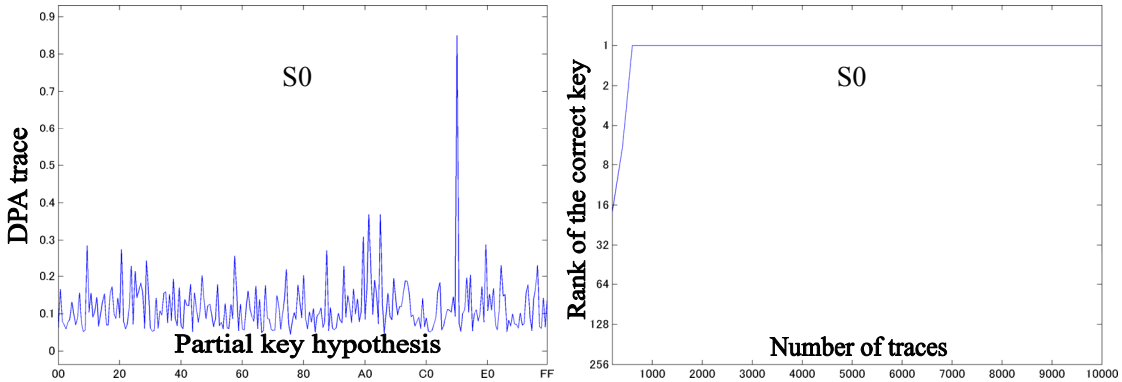


Figure 7 Result of M-DPA on the AES circuit (PPRM1) on the SASEBO-R

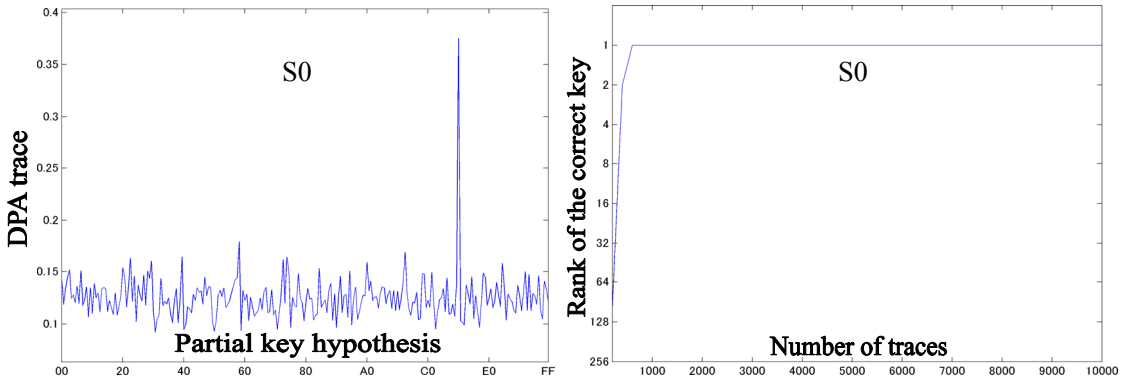


Figure 8 Result of M2-DPA on the AES circuit (PPRM1) on the SASEBO-R

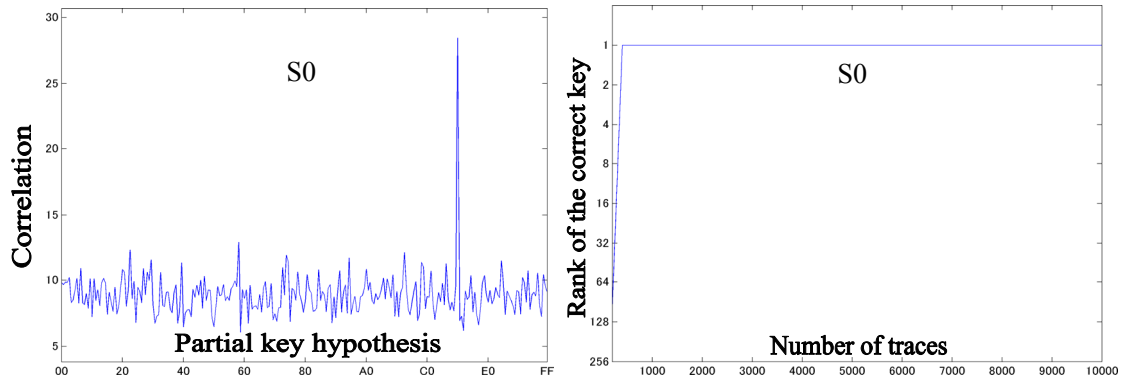


Figure 9 Result of PPA on the AES circuit (PPRM1) on the SASEBO-R

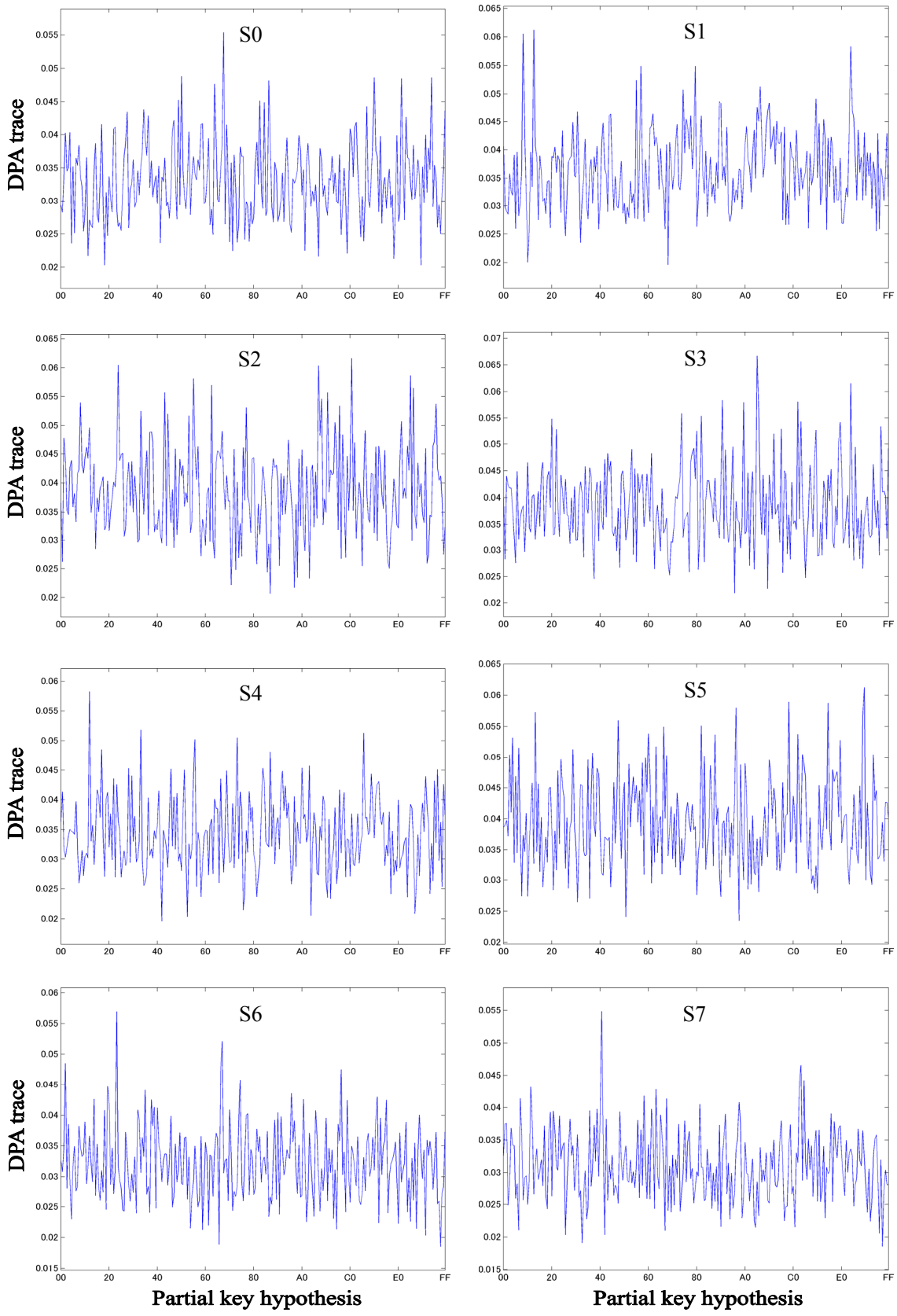


Figure 10-1 Average power differences (DPA traces) from DPA on the AES circuit (Comp) on the SASEBO-G

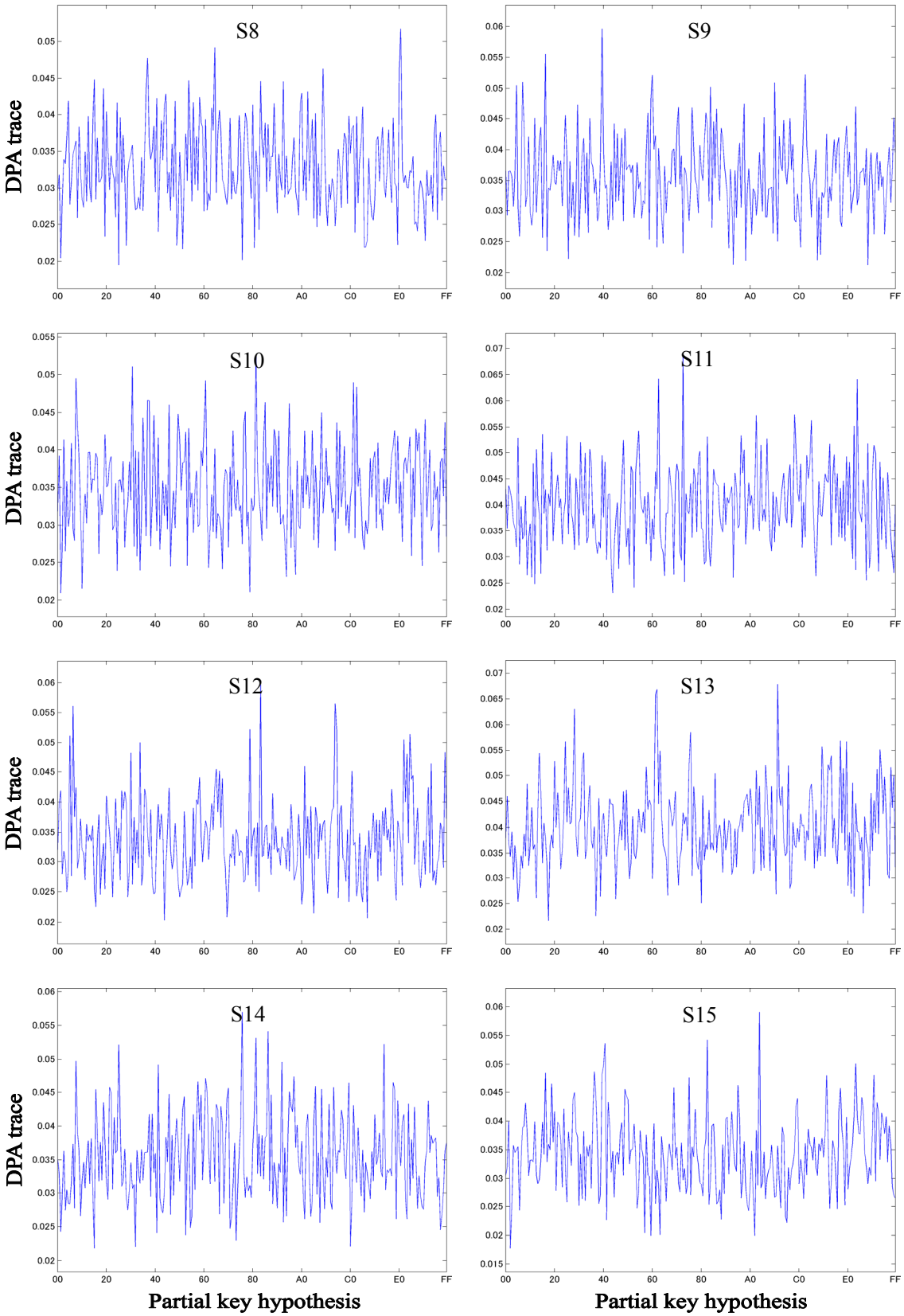


Figure 10-2 Average power differences (DPA traces) from DPA on the AES circuit (Comp) on the SASEBO-G

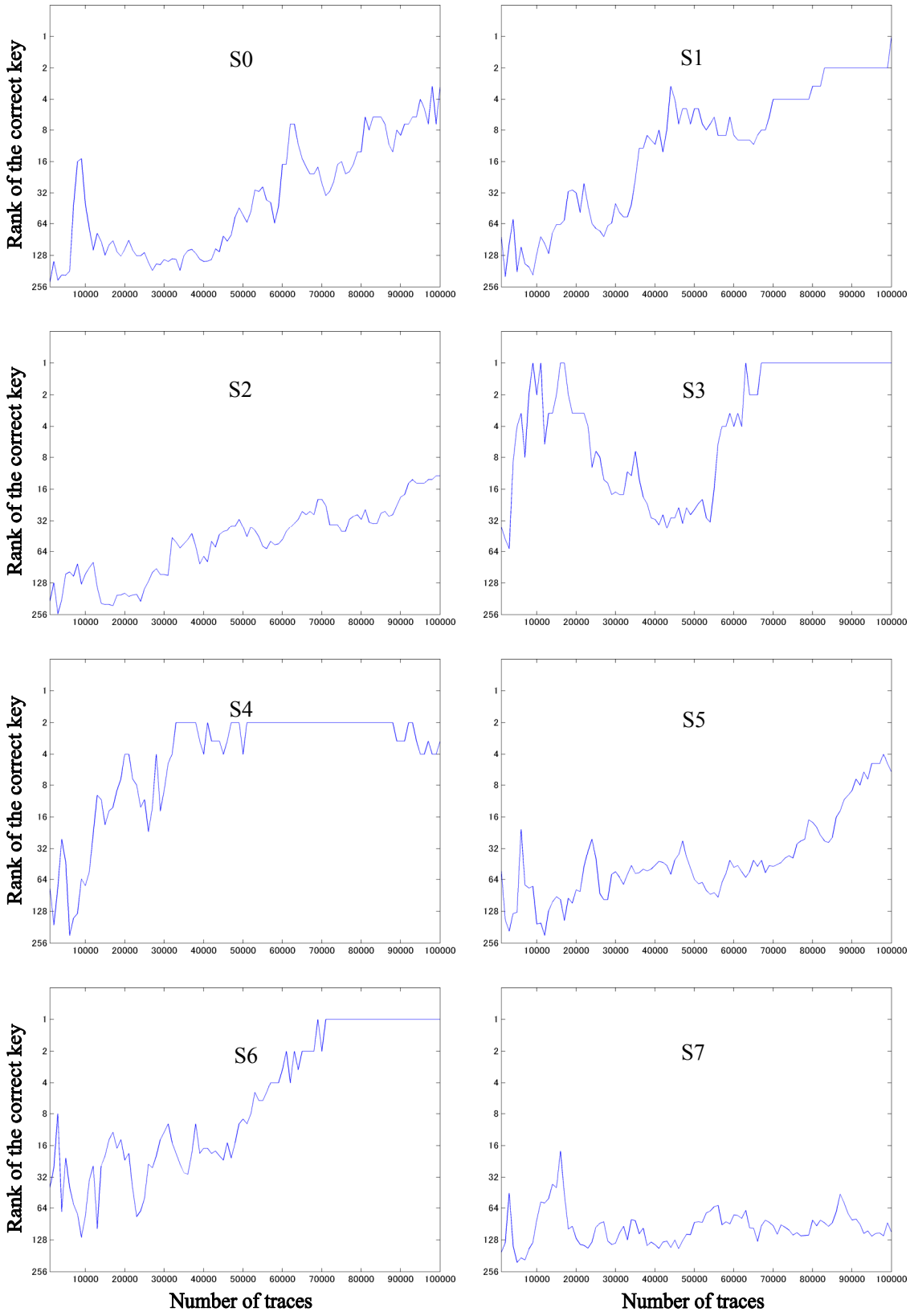


Figure 11-1 Number of power traces versus accuracy of DPA on the AES circuit (Comp) on the SASEBO-G

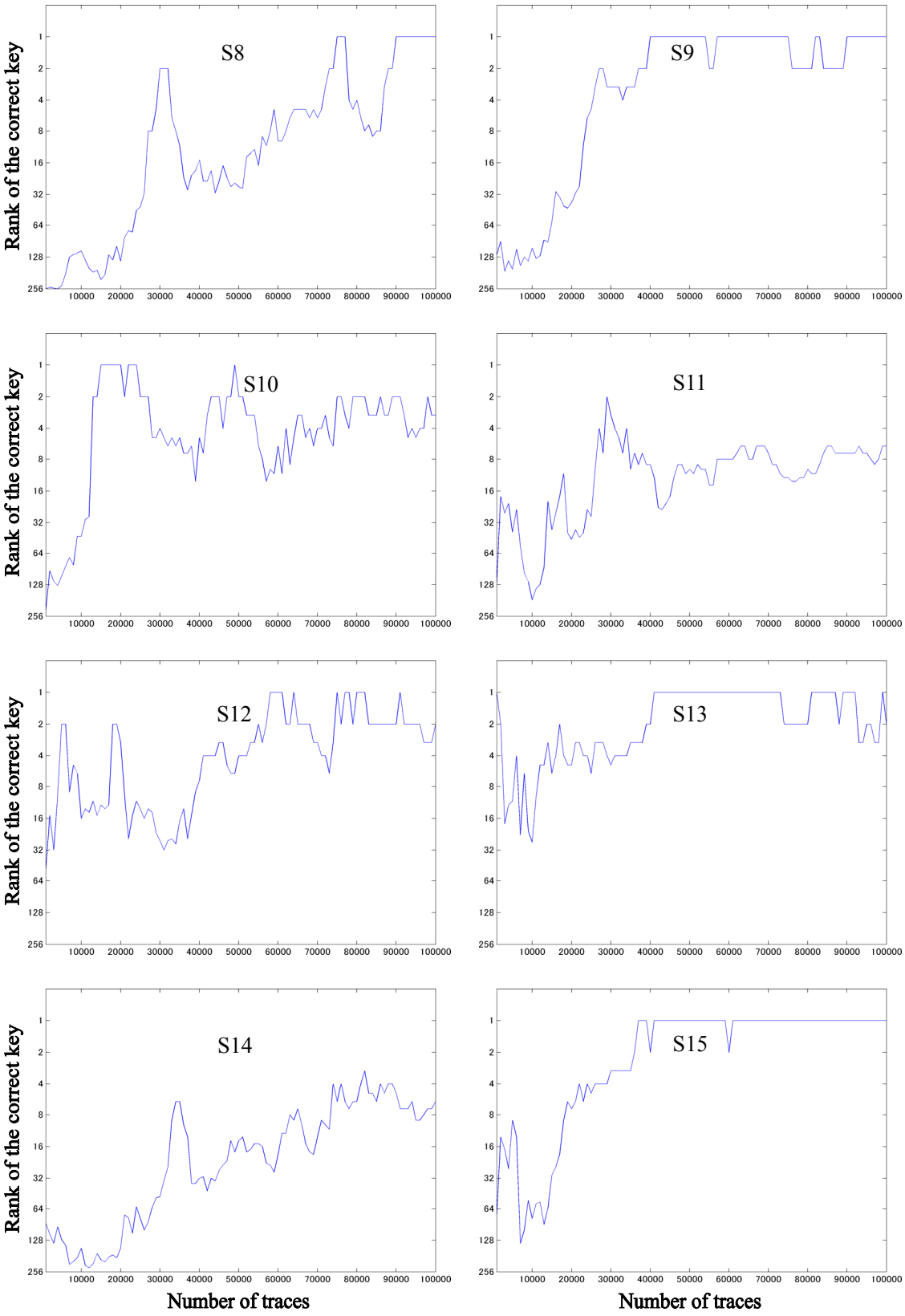


Figure 11-2 Number of power traces versus ahccuracy of DPA on the AES circuit (Comp) on the SASEBO-G

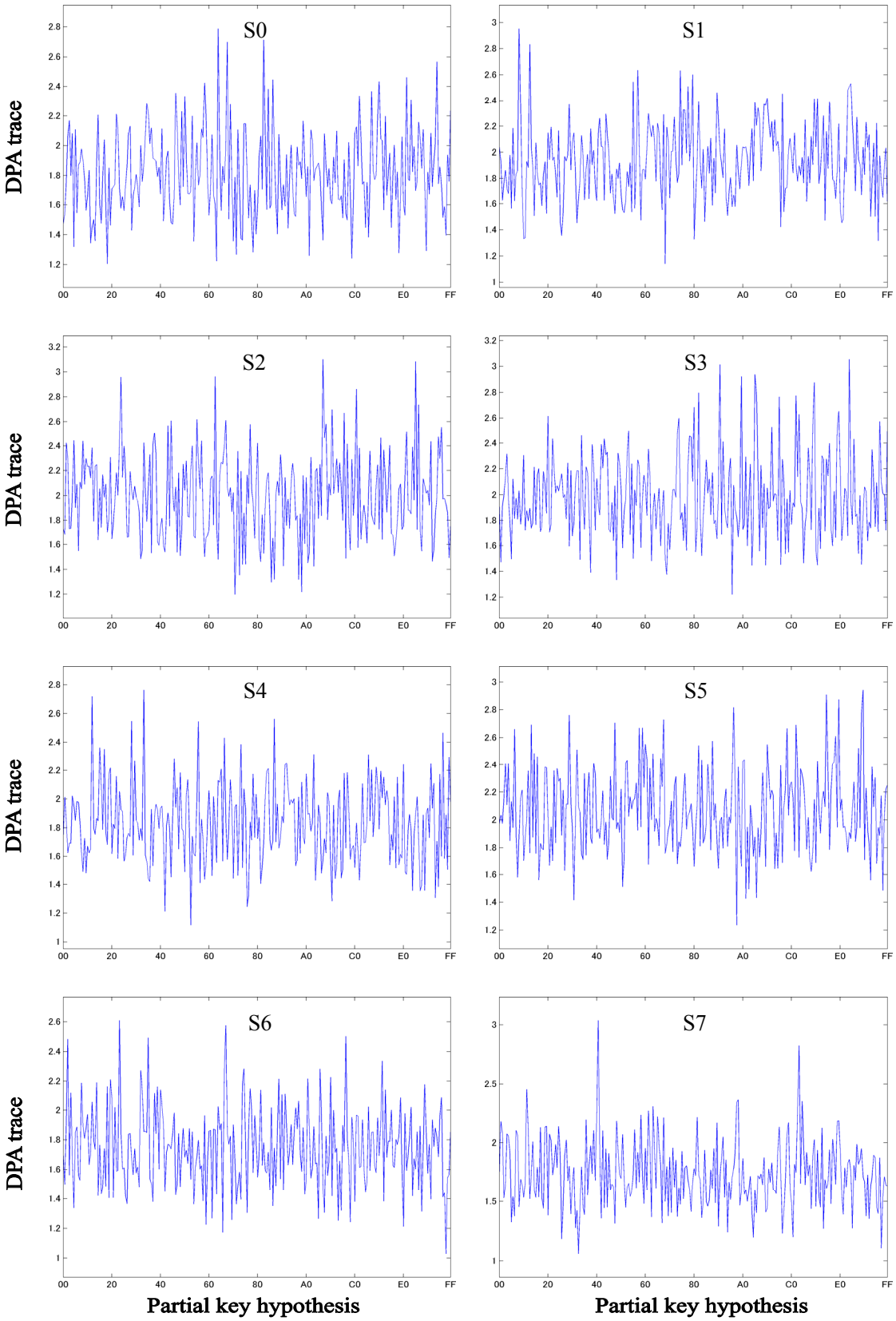


Figure 12-1 Correlation coefficients in W2-DPA on the AES circuit (Comp) on the SASEBO-G

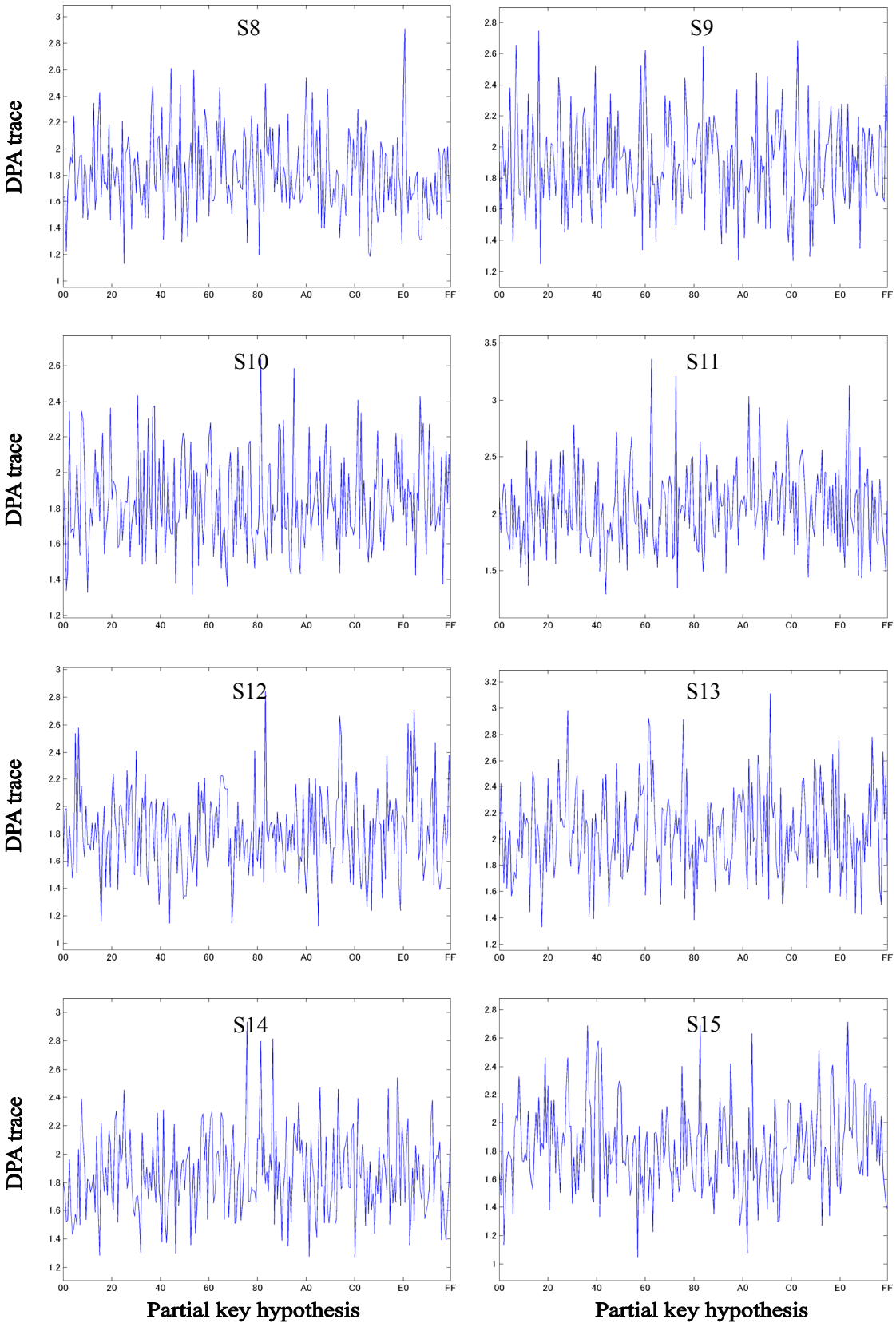


Figure 12-2 Correlation coefficients in W2-DPA on the AES circuit (Comp) on the SASEBO-G

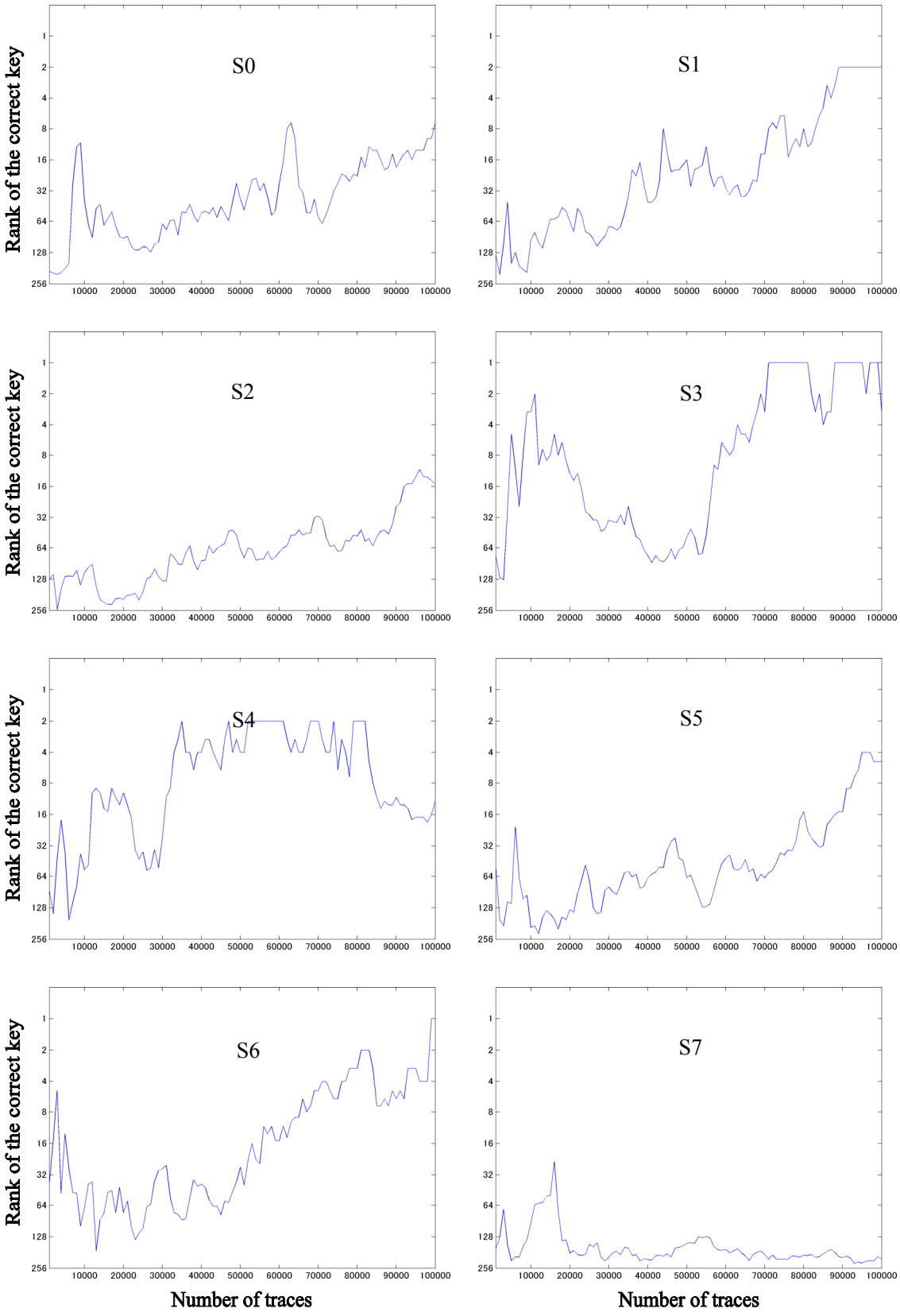


Figure 13-1 Number of power traces versus accuracy of W2-DPA on the AES circuit (Comp) on the SASEBO-G

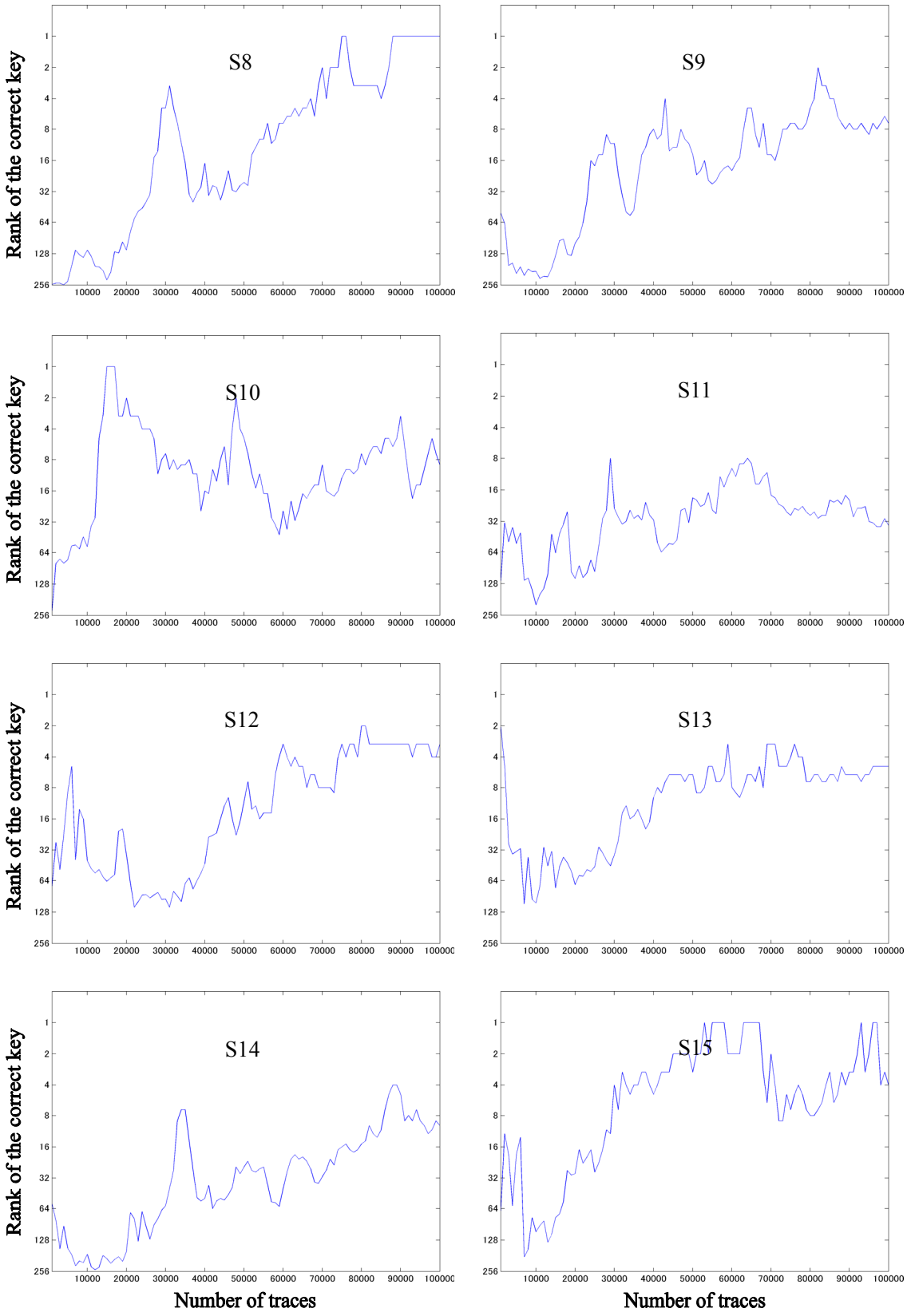


Figure 13-2 Number of power traces versus accuracy of W2-DPA on the AES circuit (Comp) on the SASEBO-G

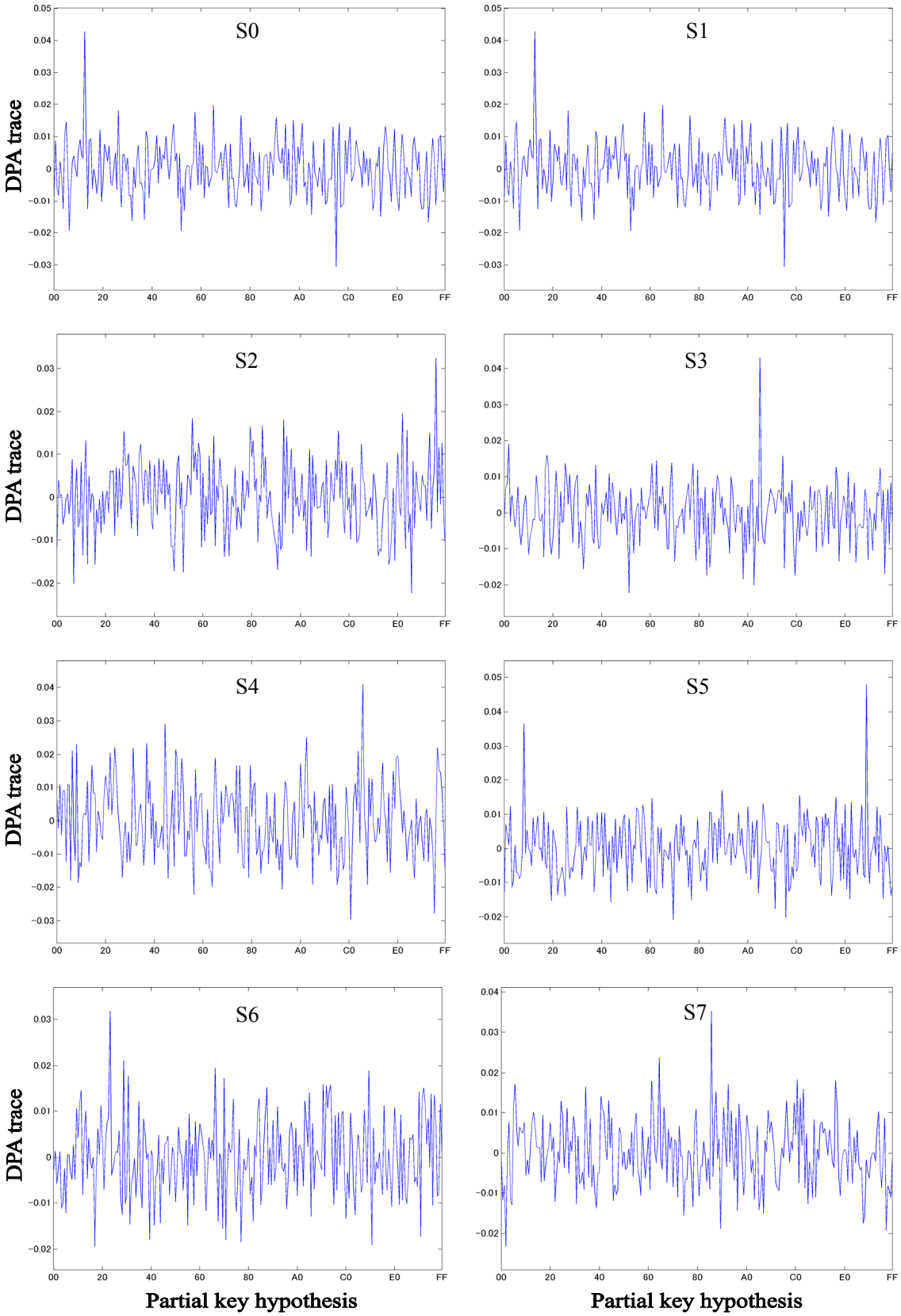


Figure 14-1 Correlation coefficients in CPA on the AES circuit (Comp) on the SASEBO-G

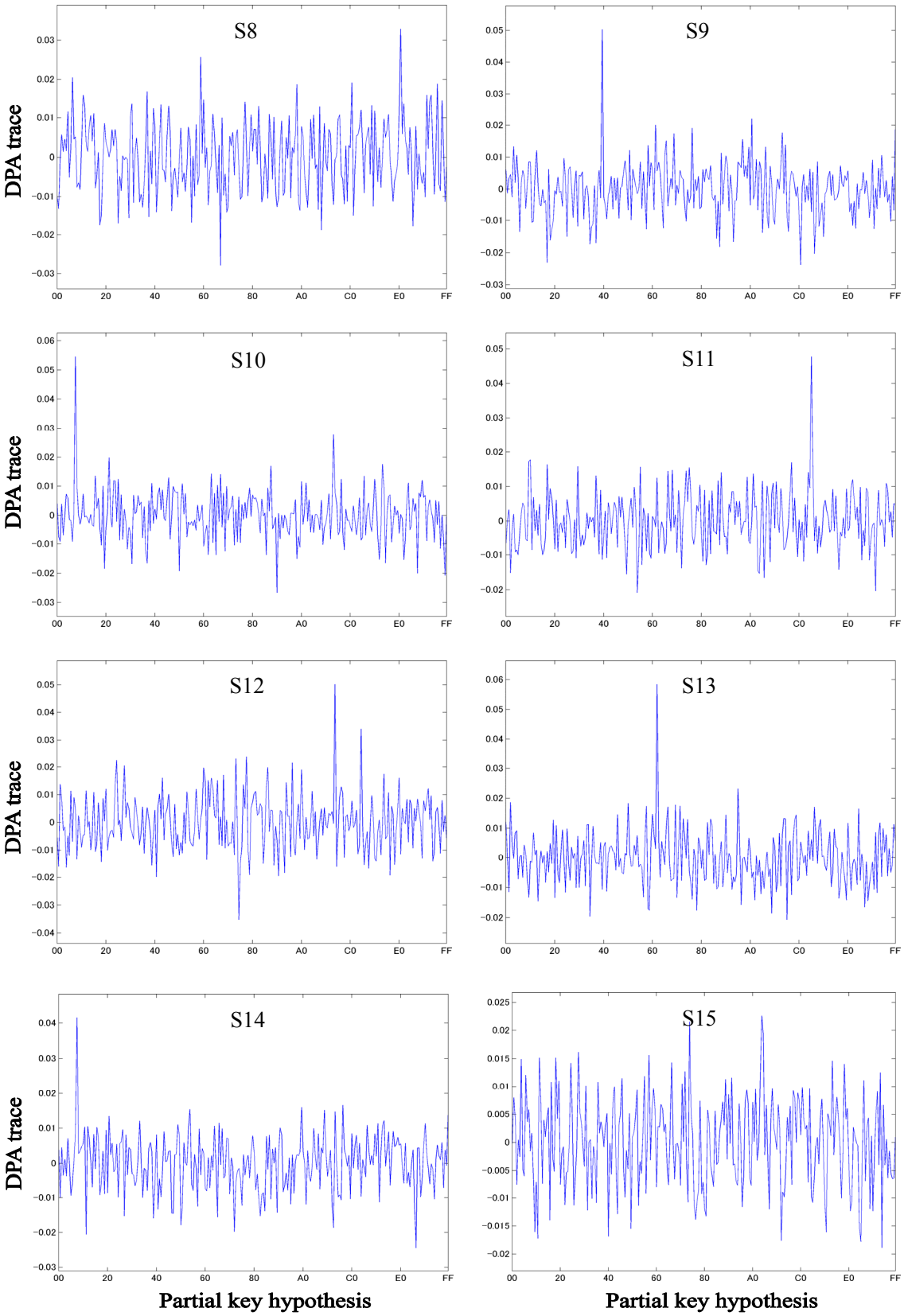


Figure 14-2 Correlation coefficients in CPA on the AES circuit (Comp) on the SASEBO-G

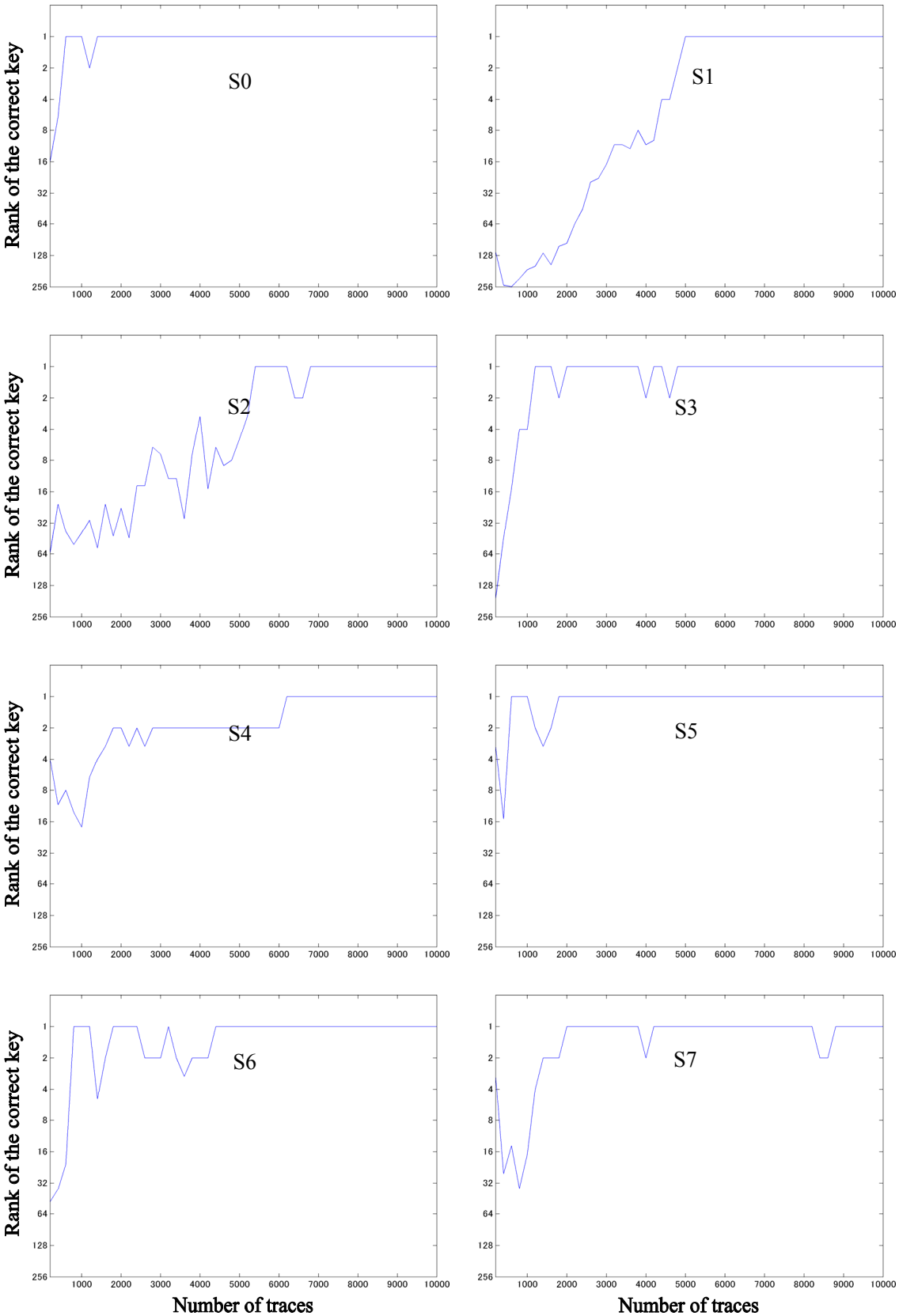


Figure 15-1 Number of power traces versus accuracy of CPA on the AES circuit (Comp) on the SASEBO-G

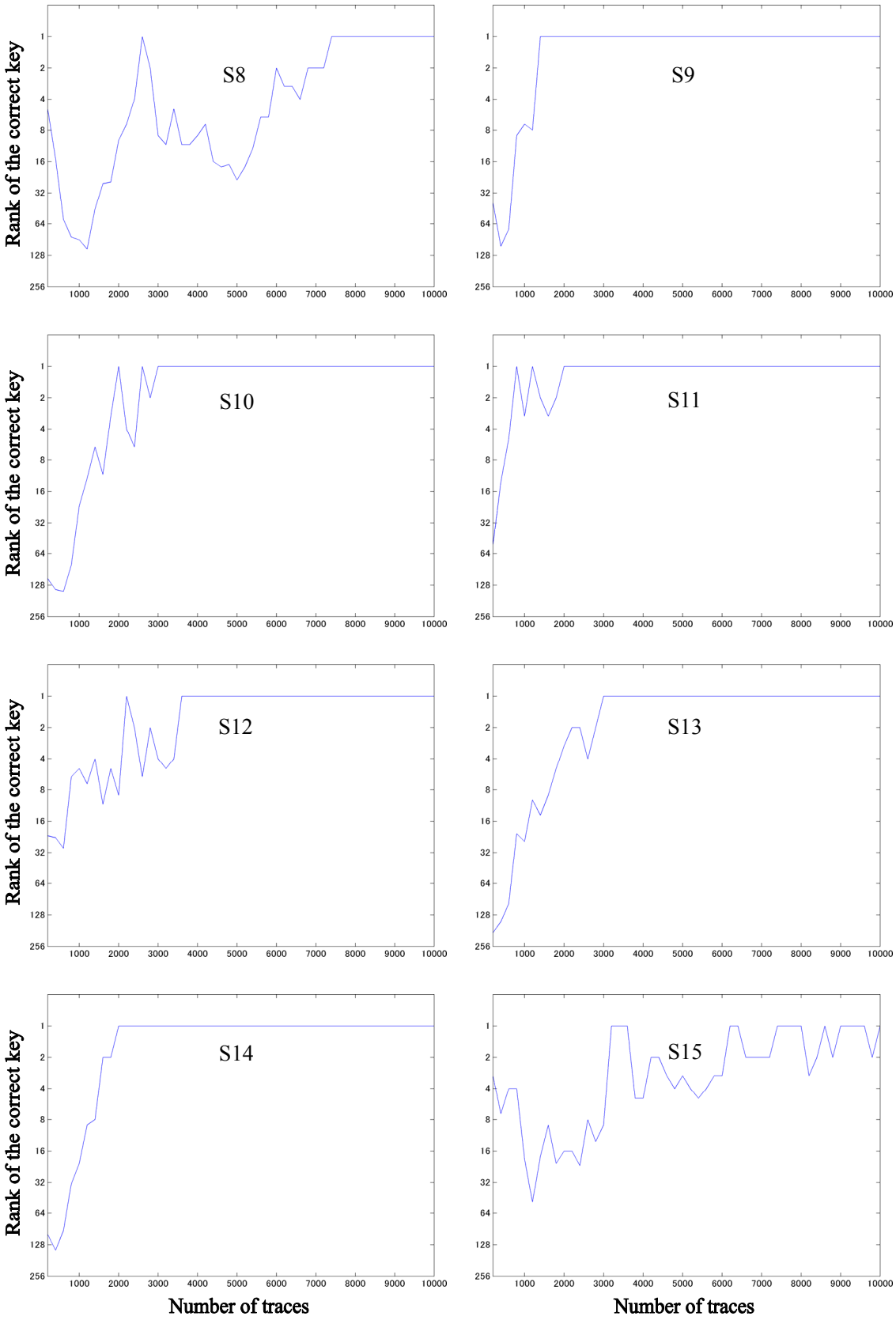


Figure 15-2 Number of power traces versus accuracy of CPA on the AES circuit (Comp) on the SASEBO-G

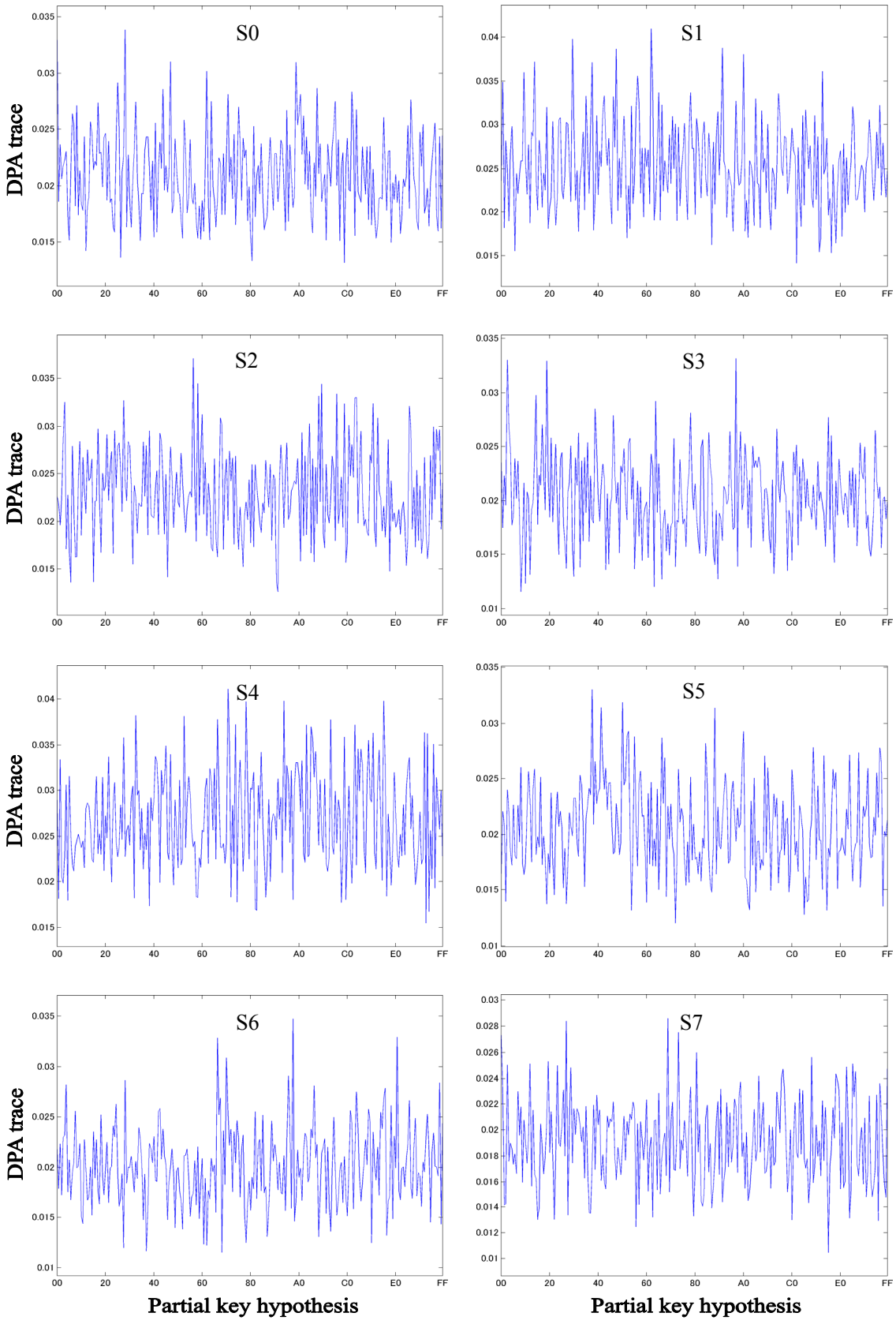


Figure 16-1 Average power differences (DPA traces) from DPA on the AES circuit (MAO) on the SASEBO-G

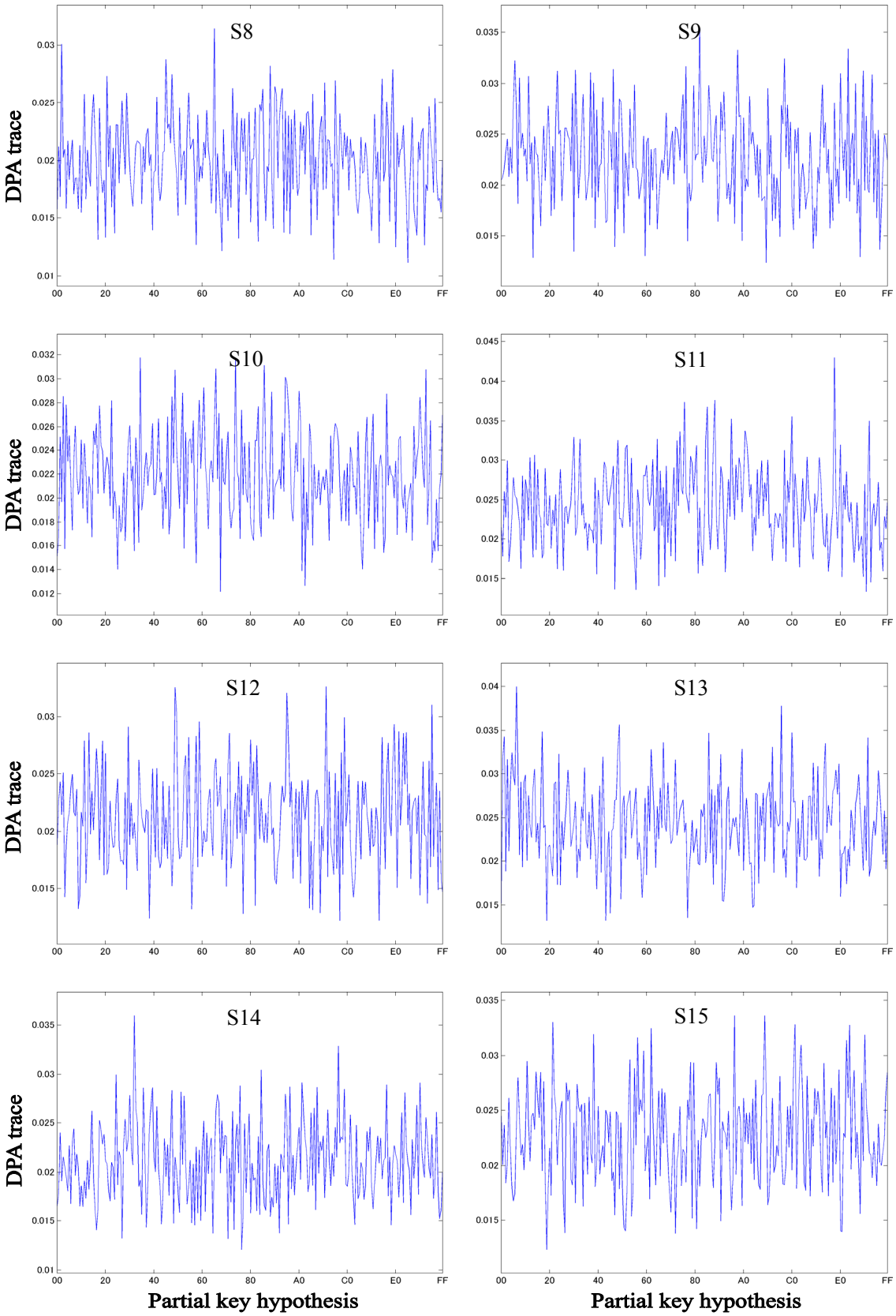


Figure 16-2 Average power differences (DPA traces) from DPA on the AES circuit (MAO) on the SASEBO-G

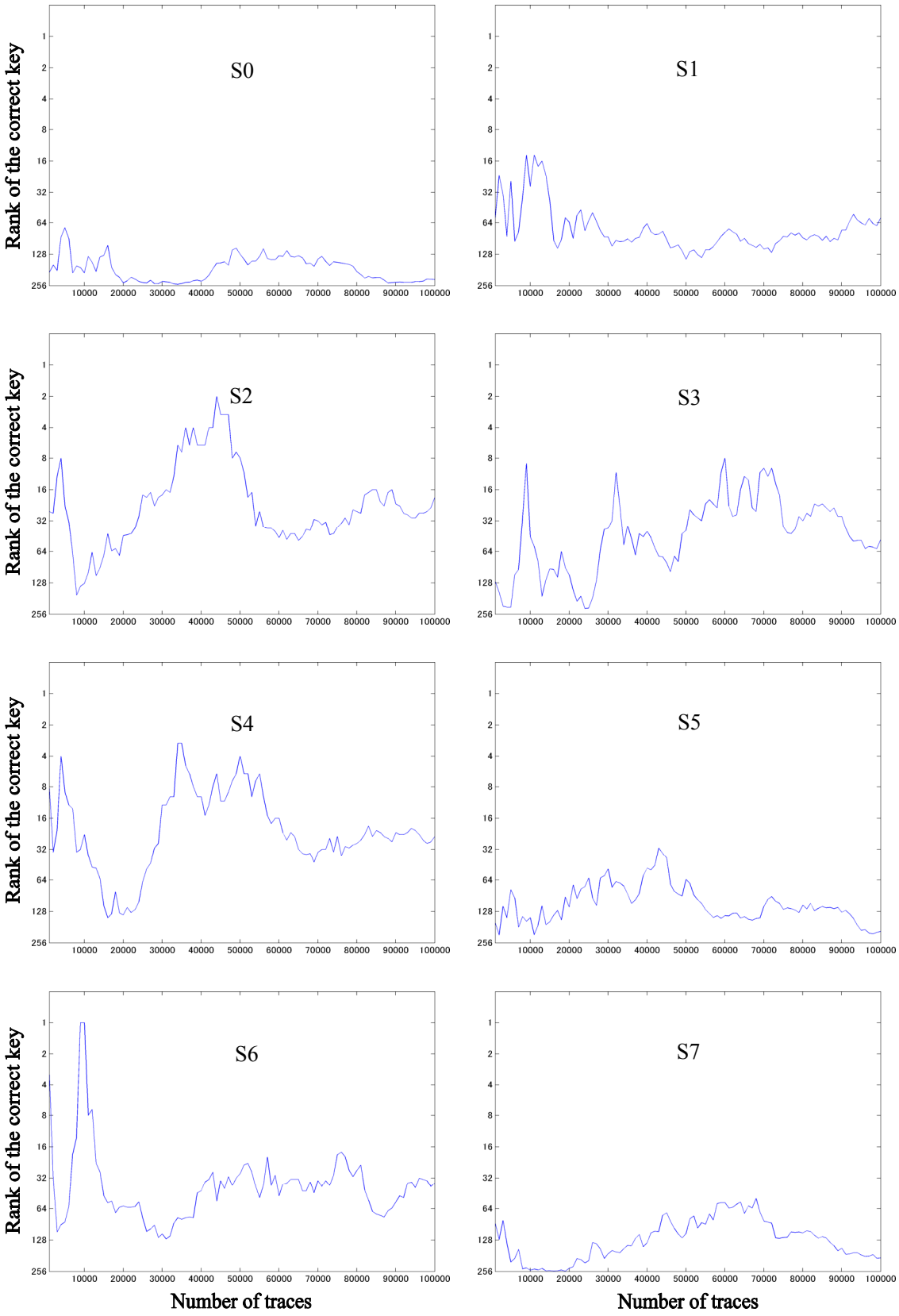


Figure 17-1 Number of power traces versus accuracy of DPA on the AES circuit (MAO) on the SASEBO-G

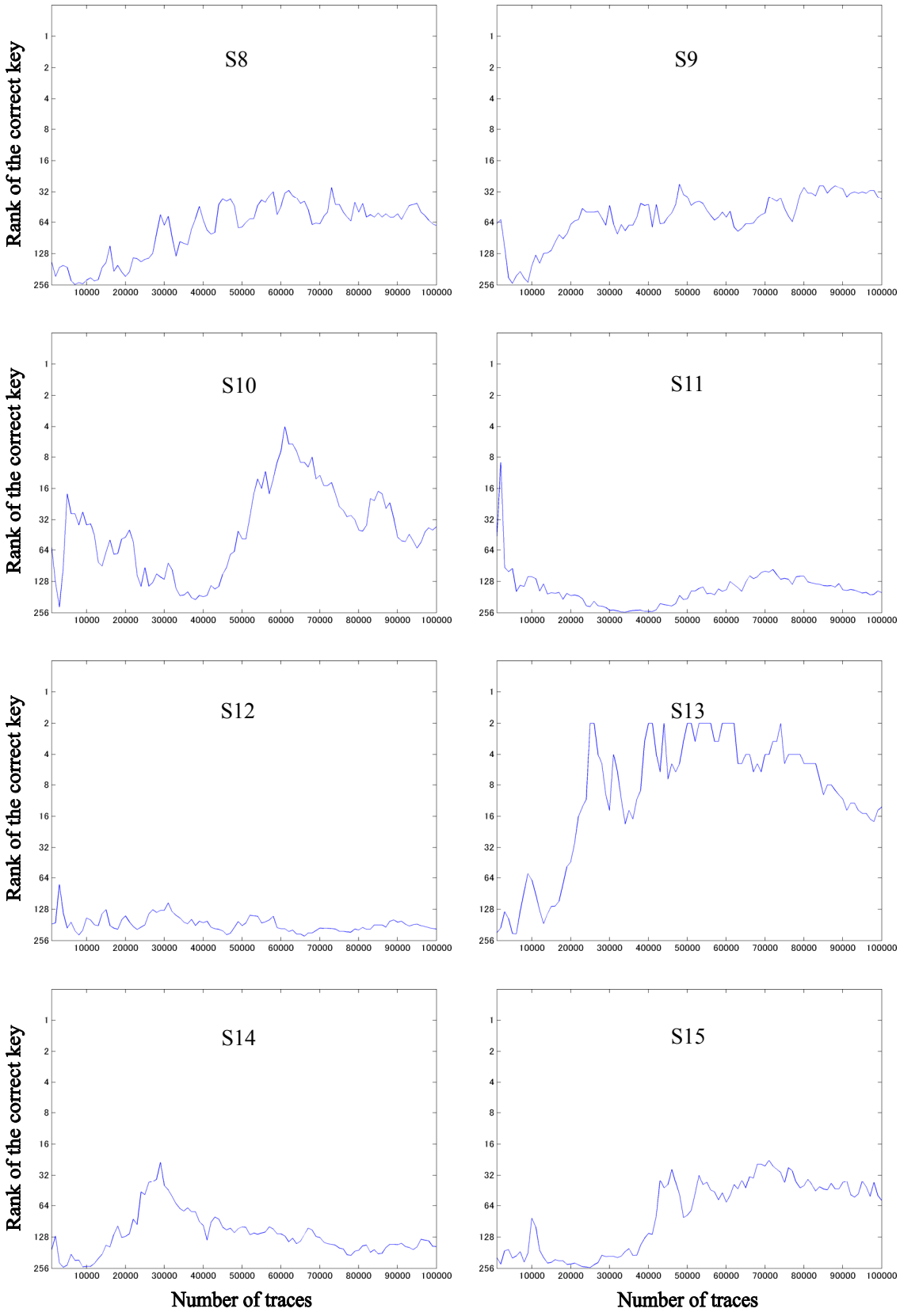


Figure 17-2 Number of power traces versus accuracy of DPA on the AES circuit (MAO) on the SASEBO-G

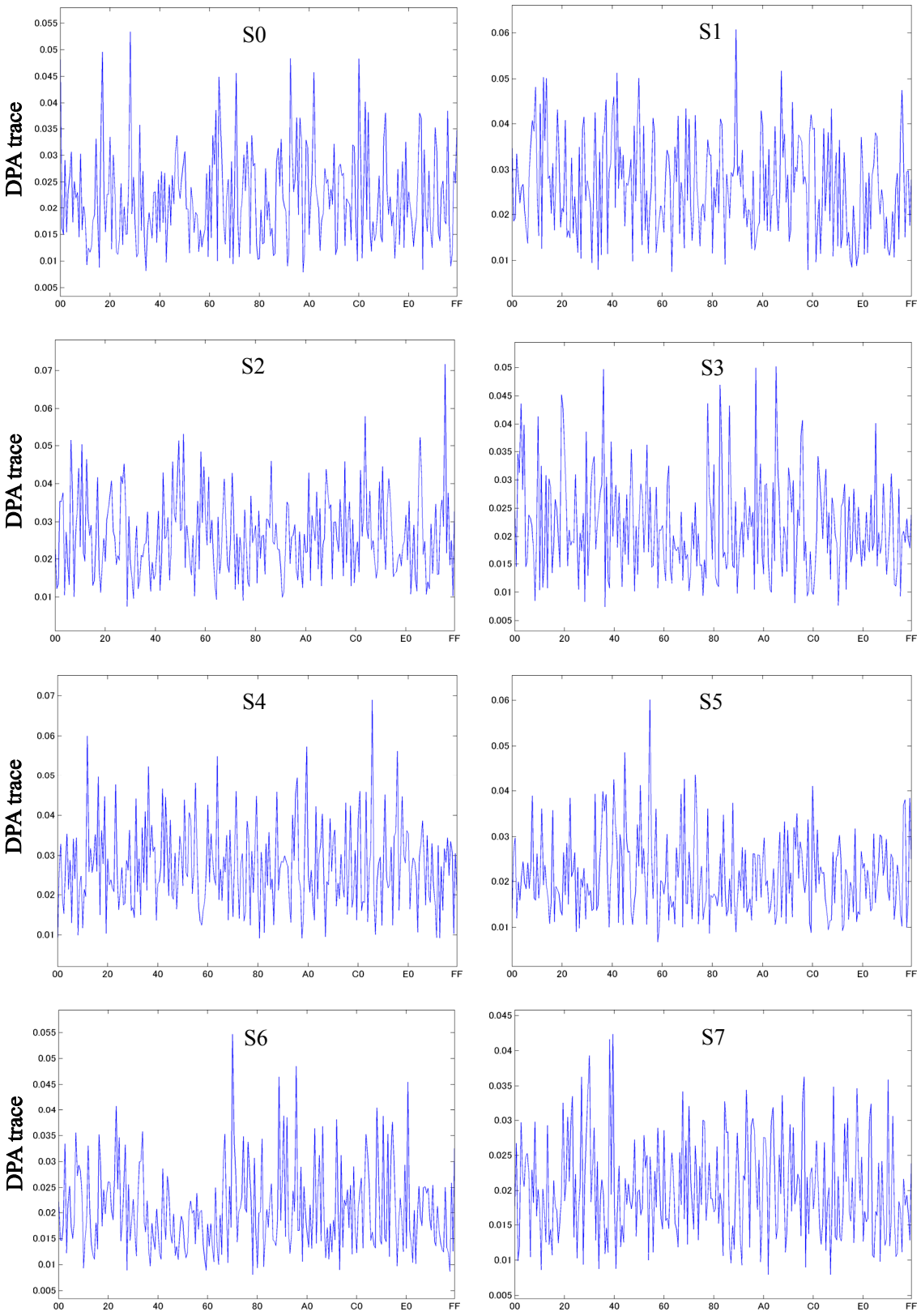


Figure 18-1 Average power differences (DPA traces) from DPA on the AES circuit (MAO) on the SASEBO-G (bit1 & bit6)

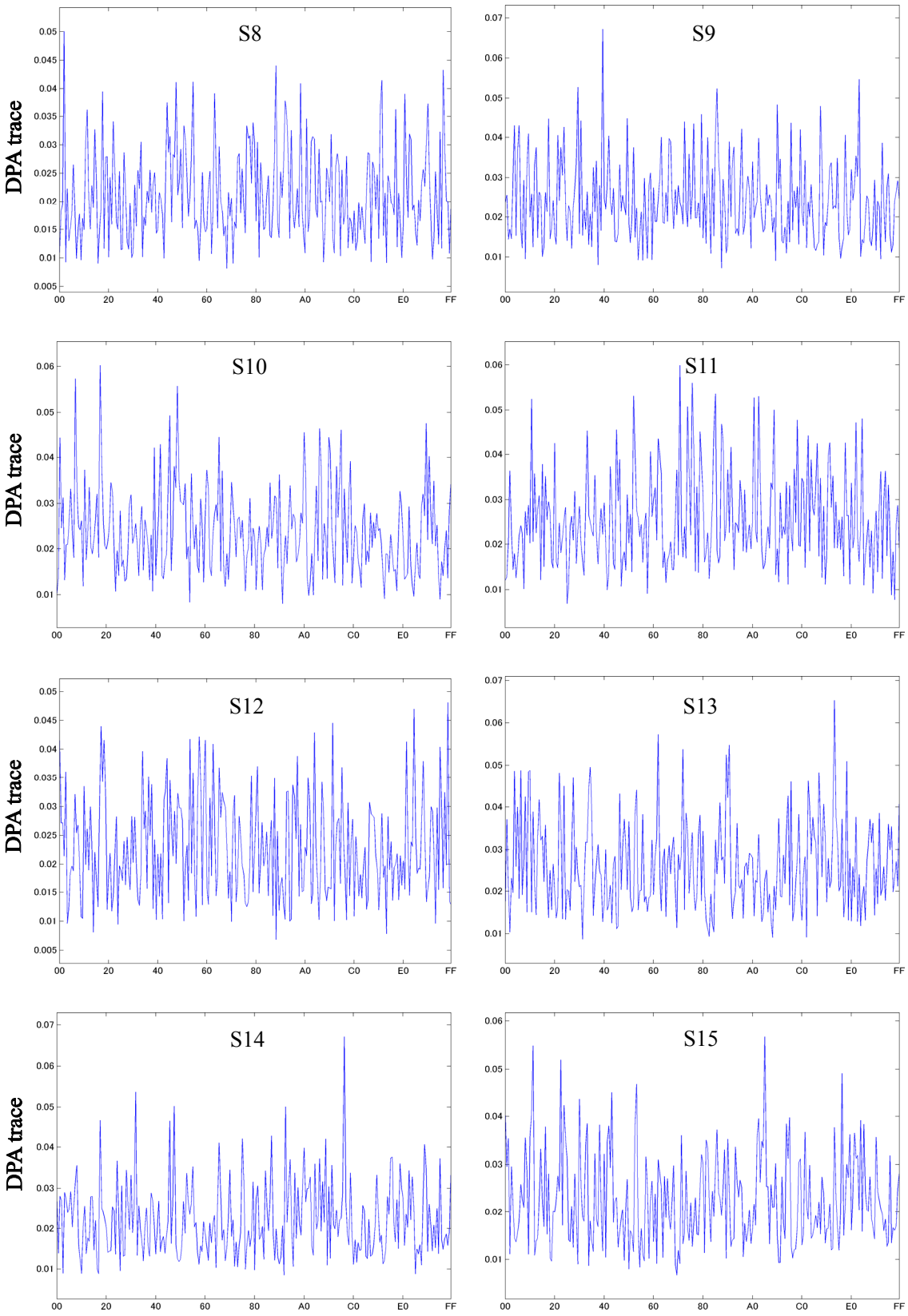


Figure 18-2 Average power differences (DPA traces) from DPA on the AES circuit (MAO) on the SASEBO-G (bit1 & bit6)

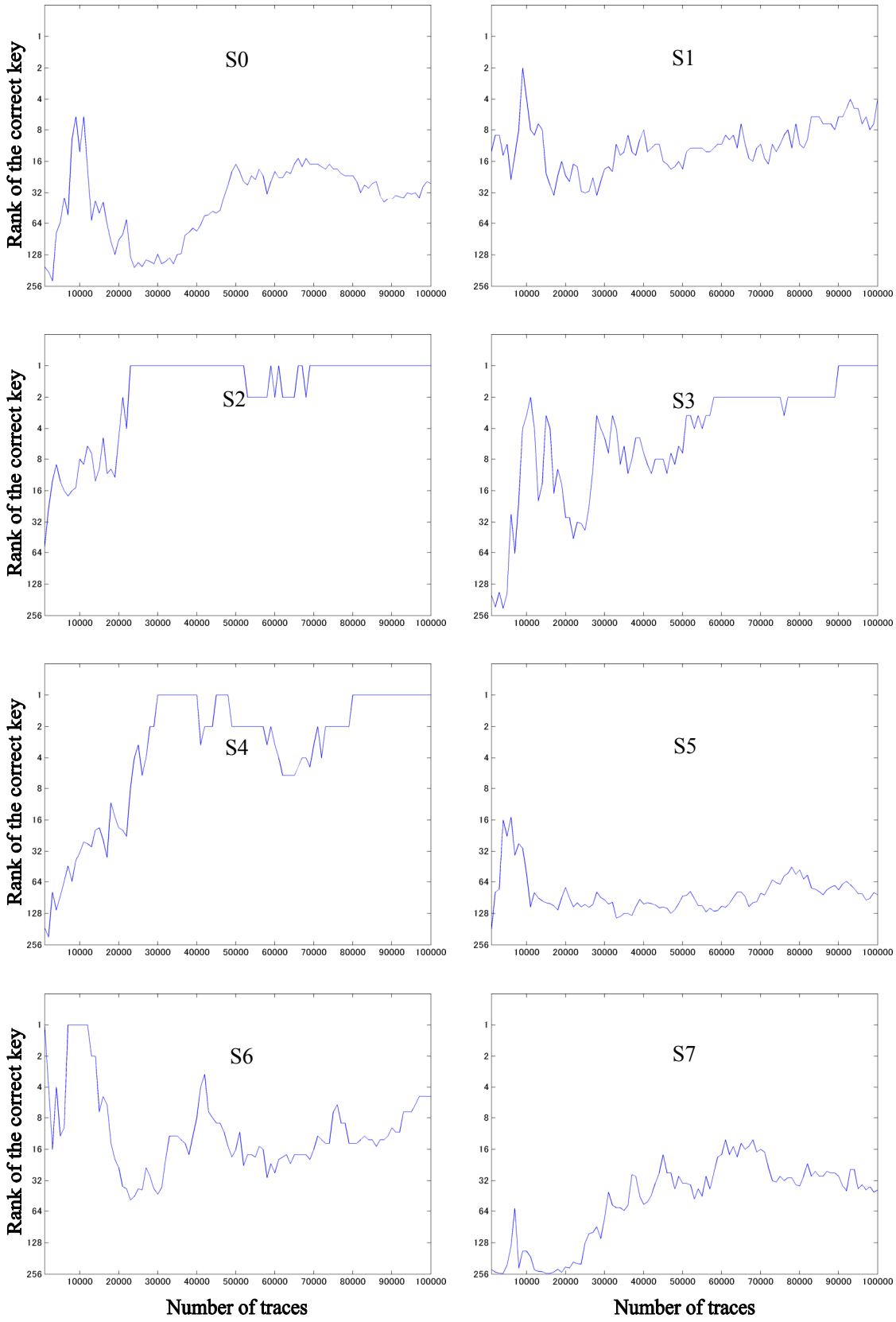


Figure 19-1 Number of power traces versus accuracy of DPA on the AES circuit (MAO) on the SASEBO-G (bit1 and bit6)

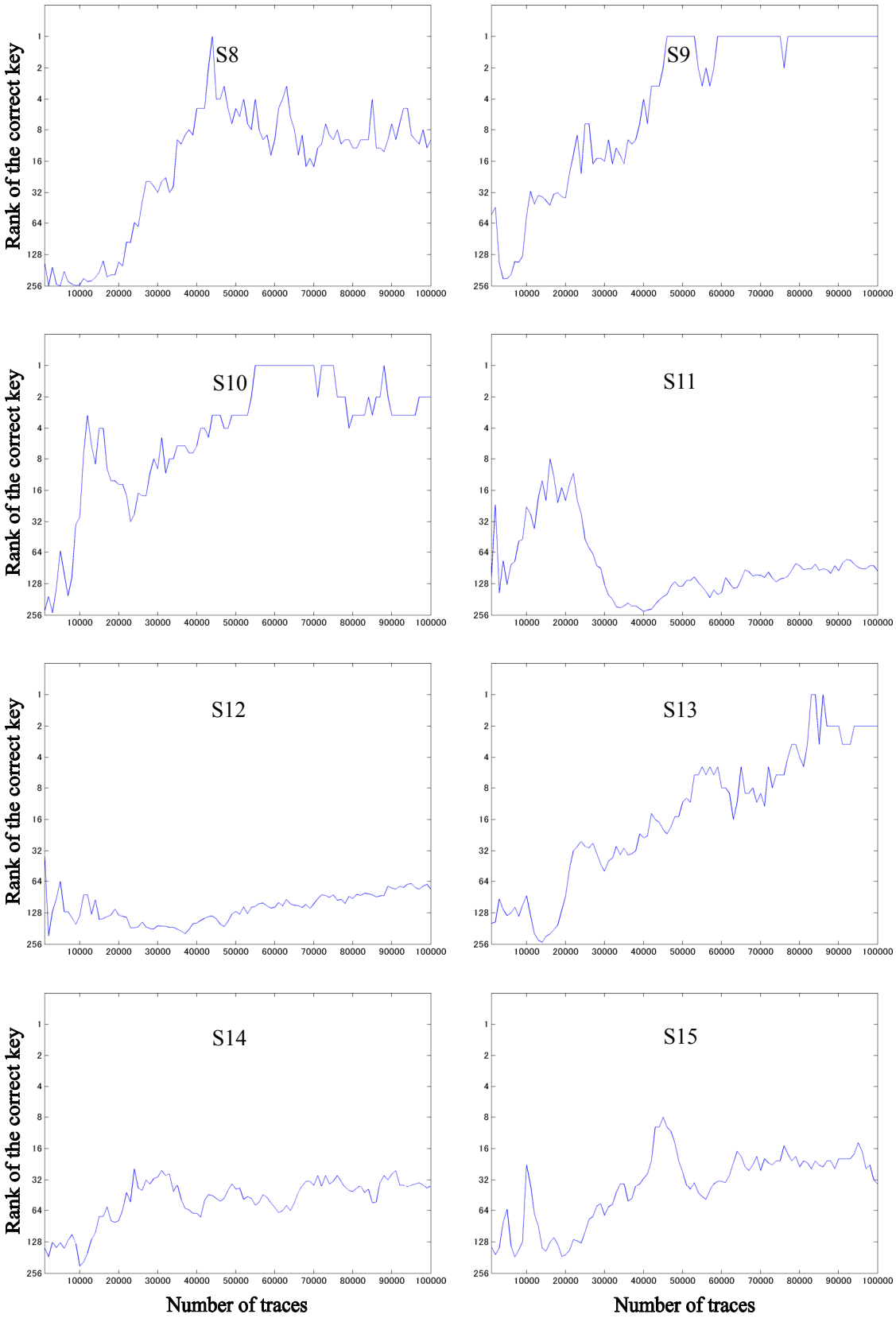


Figure 19-2 Number of power traces versus accuracy of DPA on the AES circuit (MAO) on the SASEBO-G (bit1 and bit6)

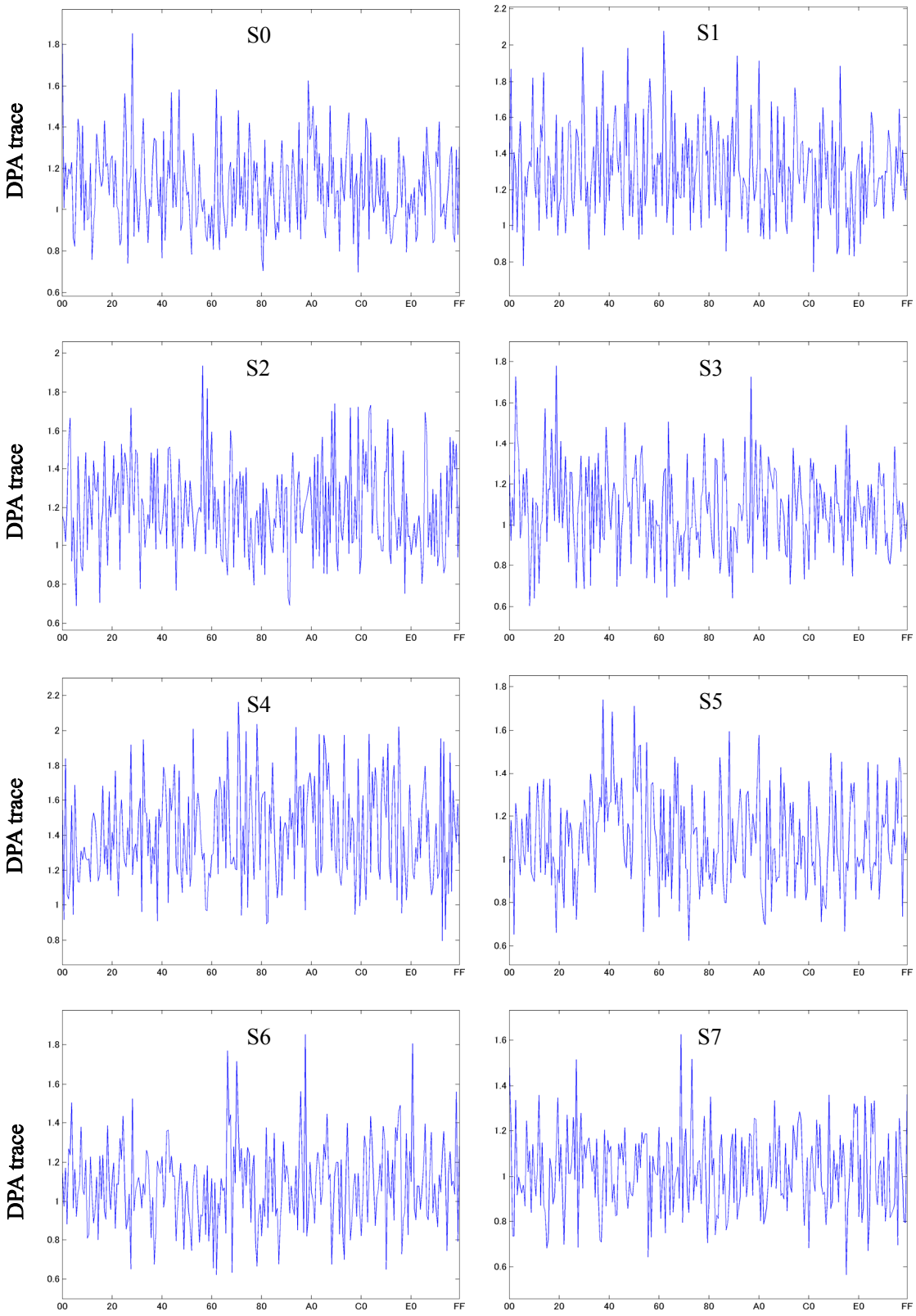


Figure 20-1 Average power differences (DPA traces) from W2-DPA on the AES circuit (MAO) on the SASEBO-G

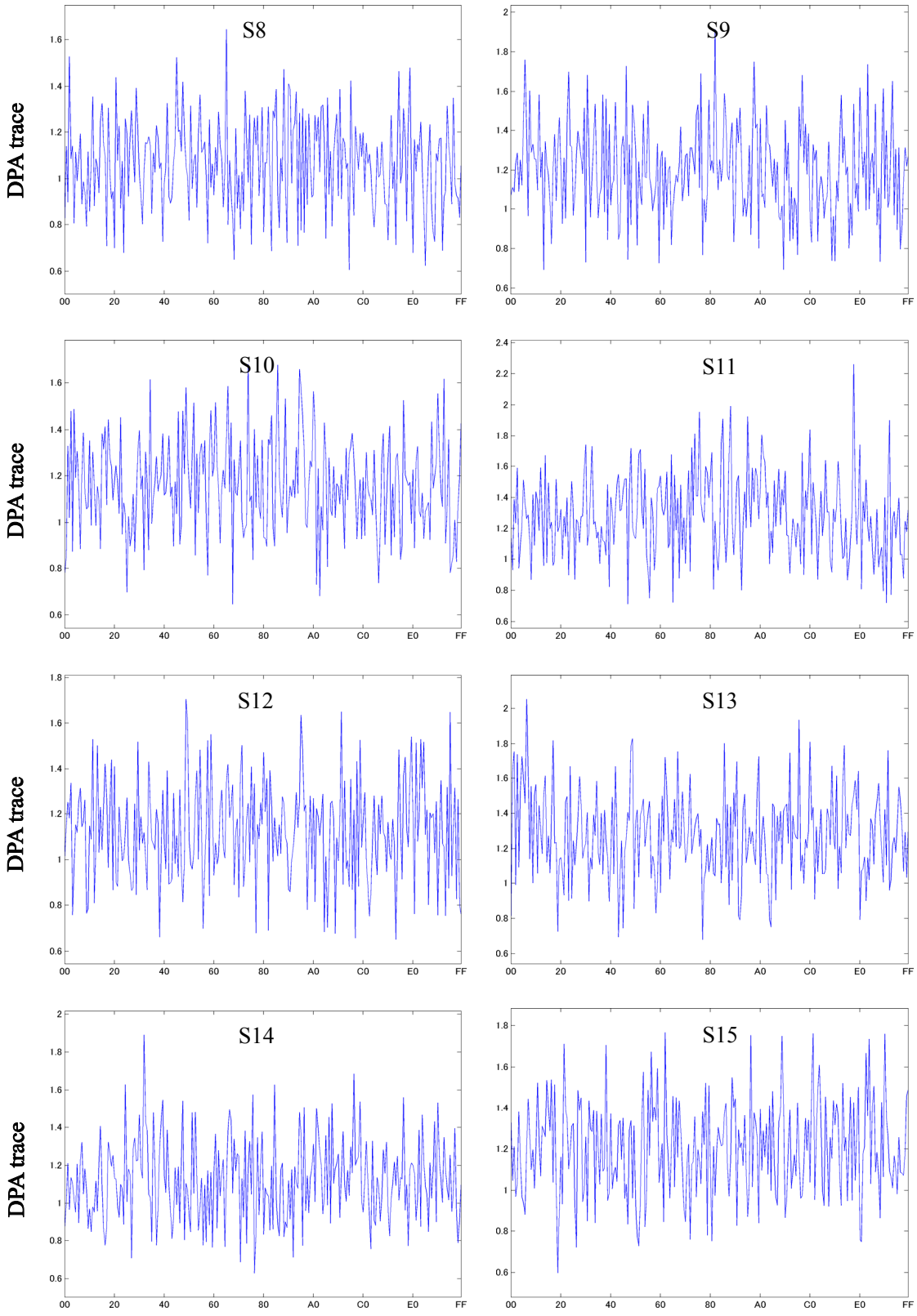


Figure 20-2 Average power differences (DPA traces) from W2-DPA on the AES circuit (MAO) on the SASEBO-G

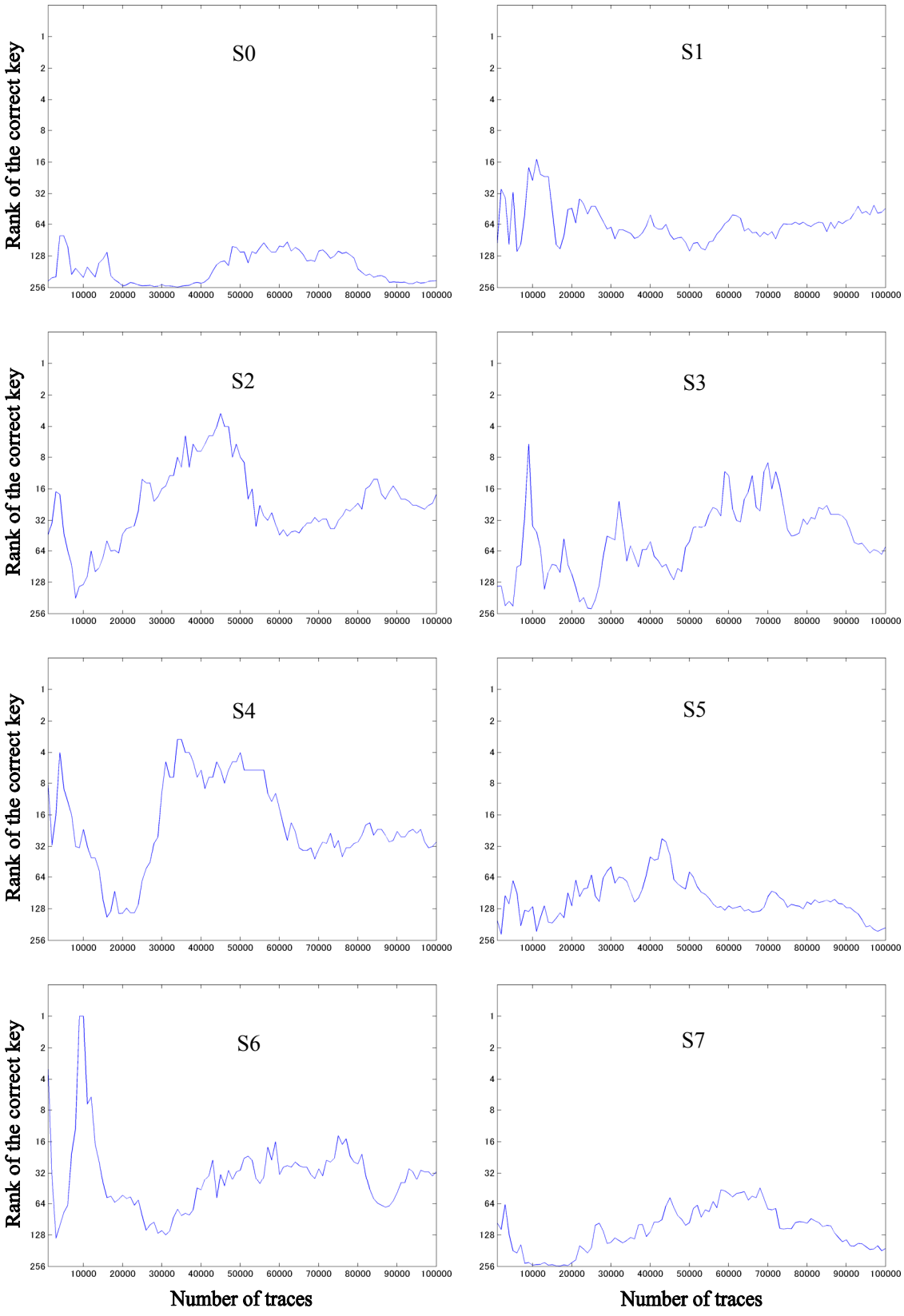


Figure 21-1 Number of power traces versus accuracy of W2-DPA on the AES circuit (MAO) on the SASEBO-G

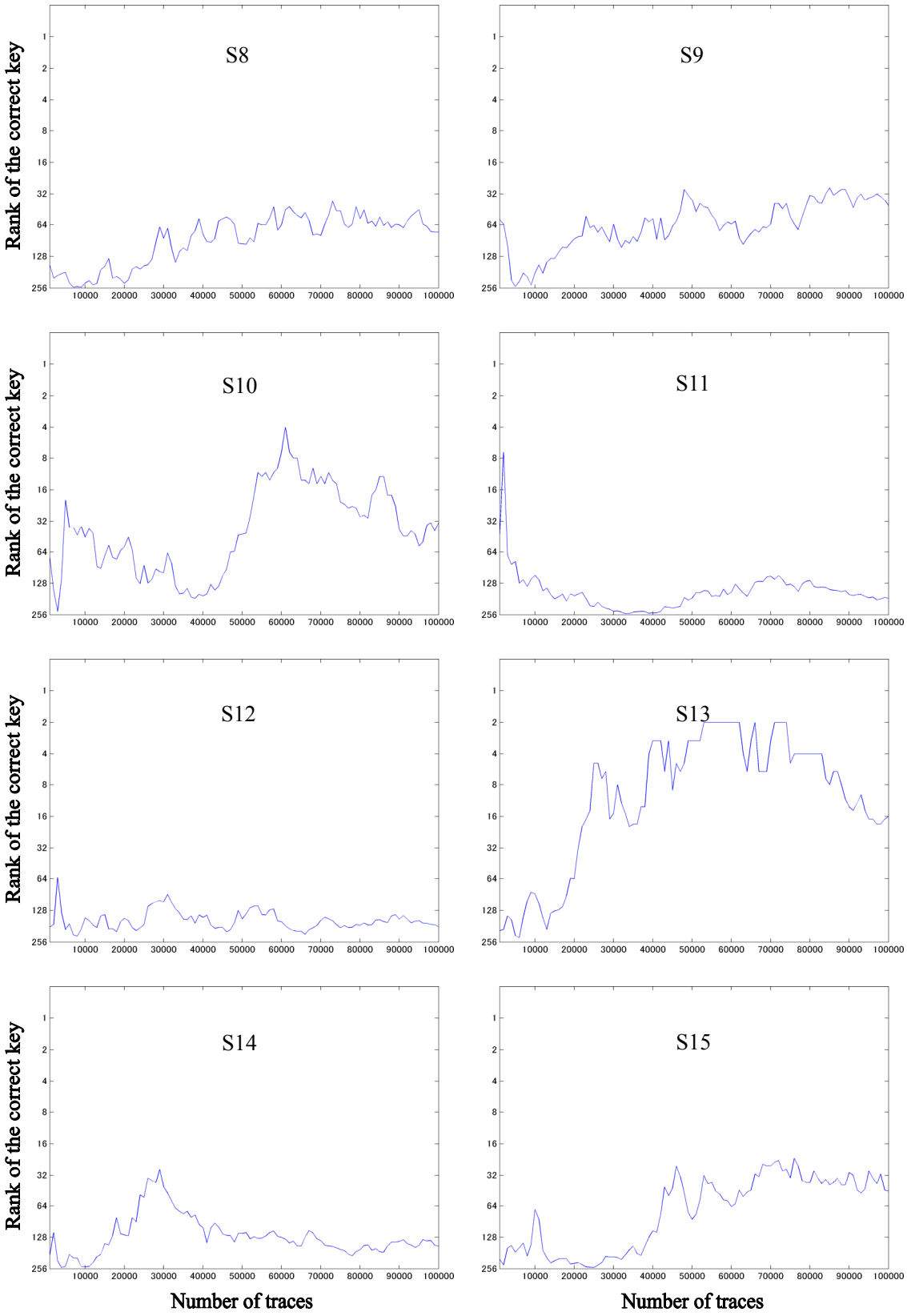


Figure 21-2 Number of power traces versus accuracy of W2-DPA on the AES circuit (MAO) on the SASEBO-G

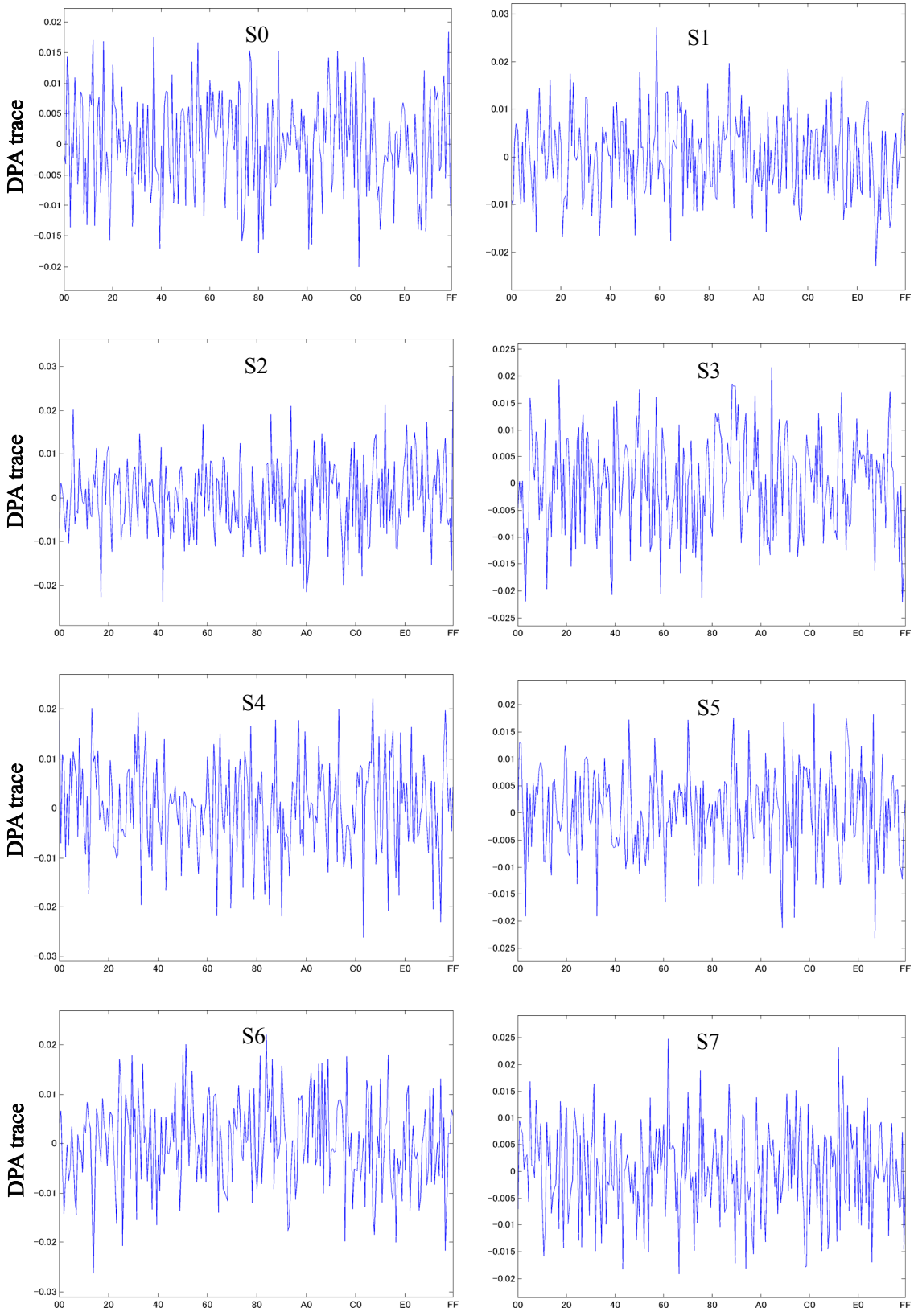


Figure 22-1 Average power differences (DPA traces) from CPA on the AES circuit (MAO) on the SASEBO-G

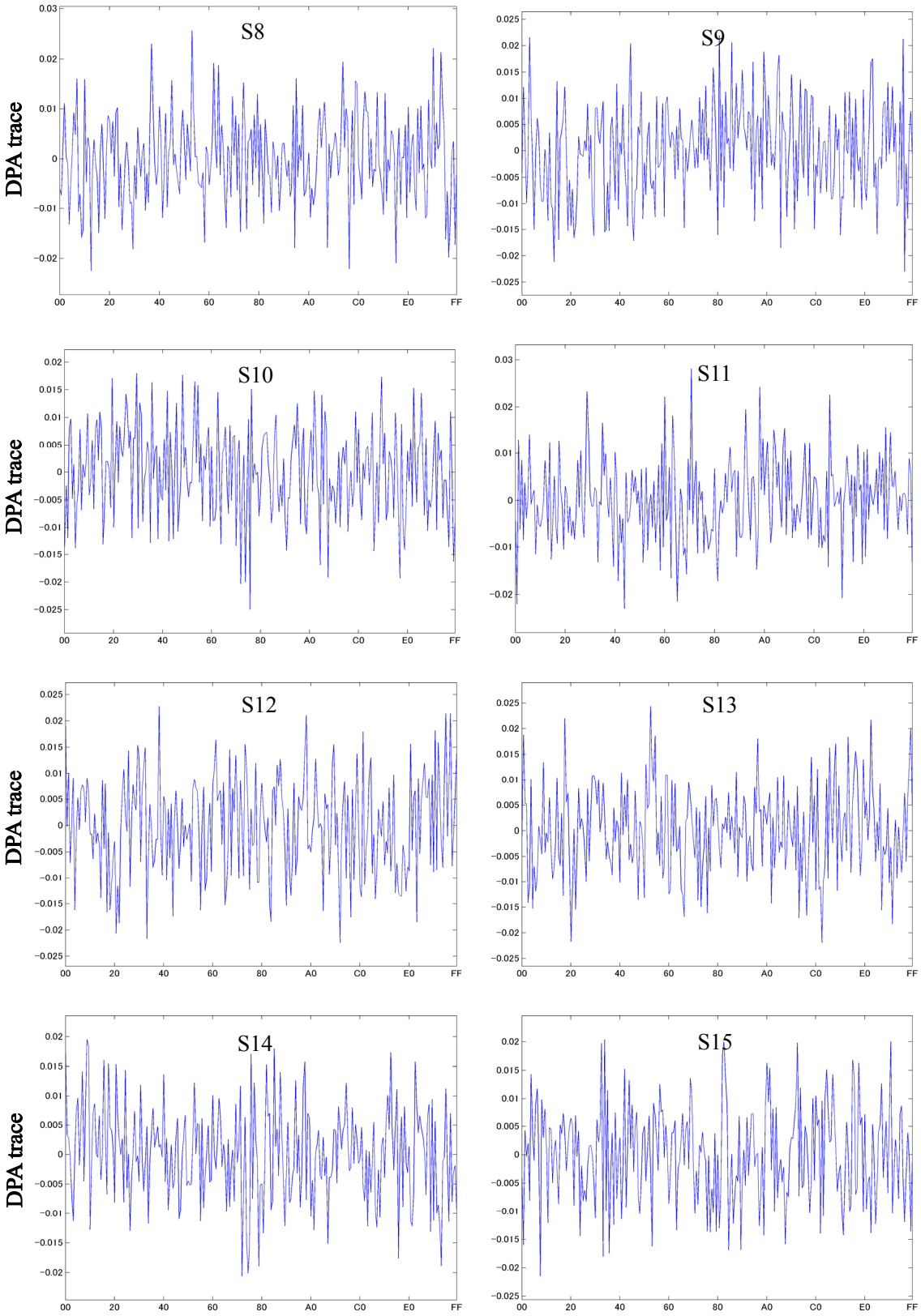


Figure 22-2 Average power differences (DPA traces) from CPA on the AES circuit (MAO) on the SASEBO-G

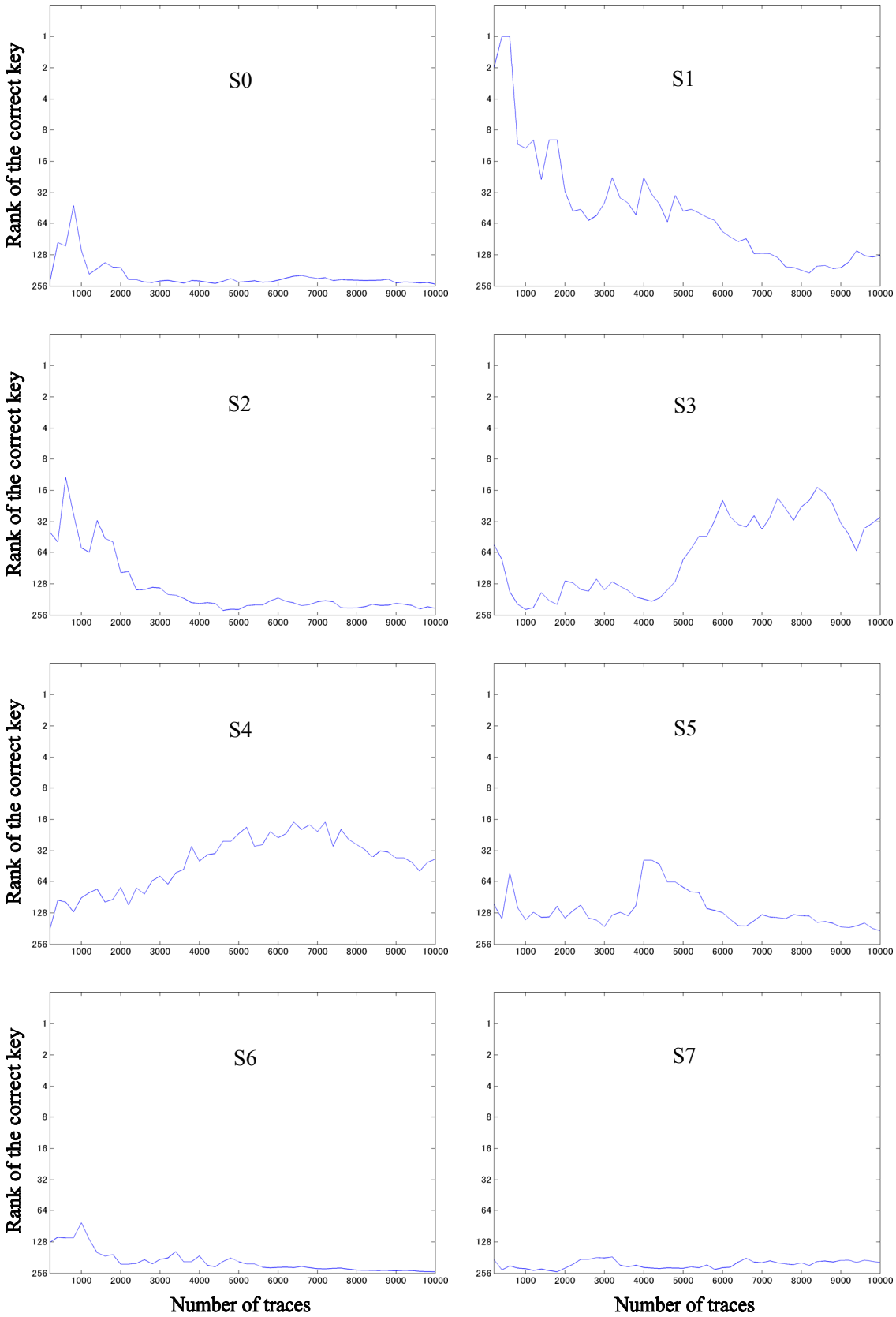


Figure 23-1 Number of power traces versus accuracy of CPA on the AES circuit (MAO) on the SASEBO-G

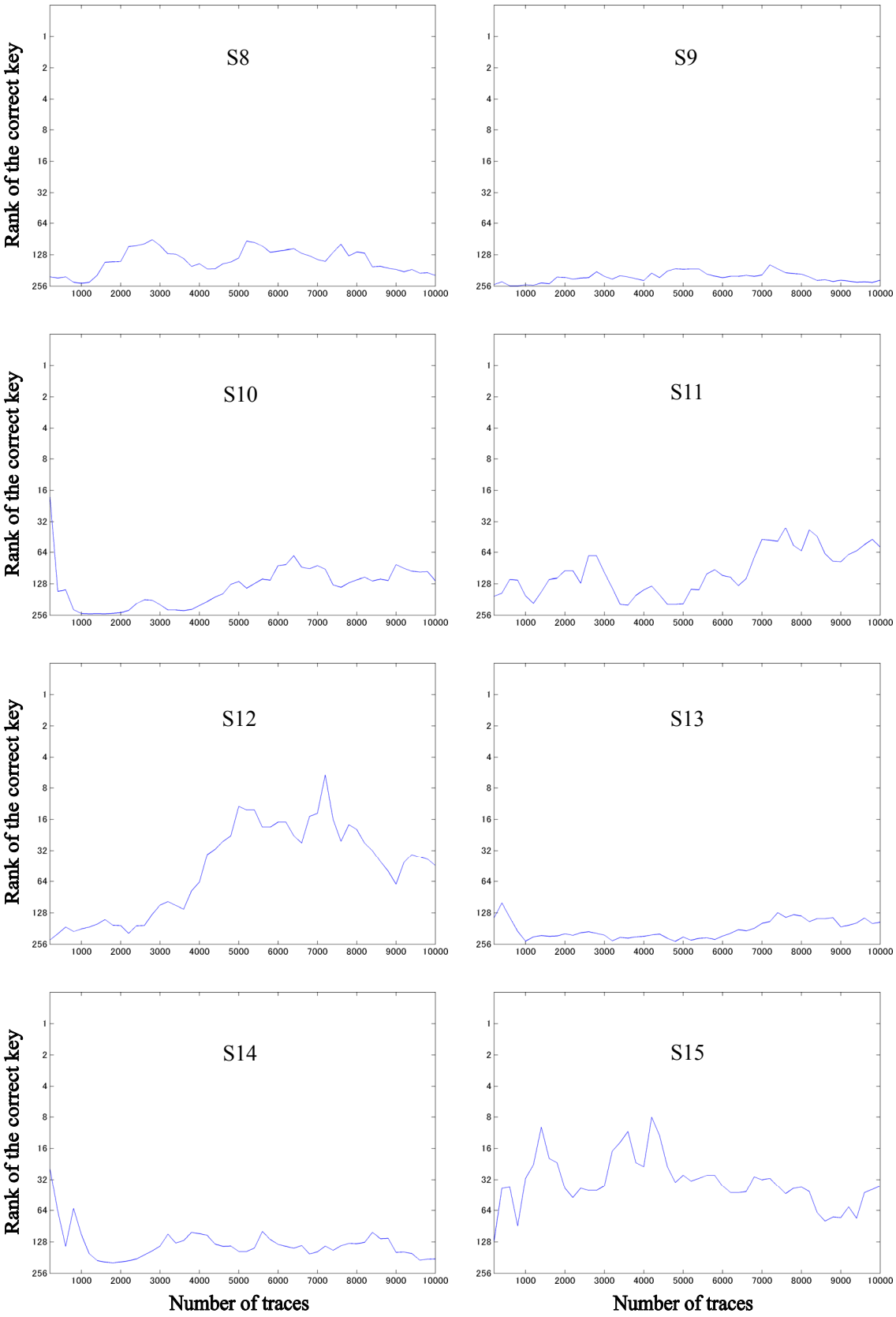


Figure 23-2 Number of power traces versus accuracy of CPA on the AES circuit (MAO) on the SASEBO-G

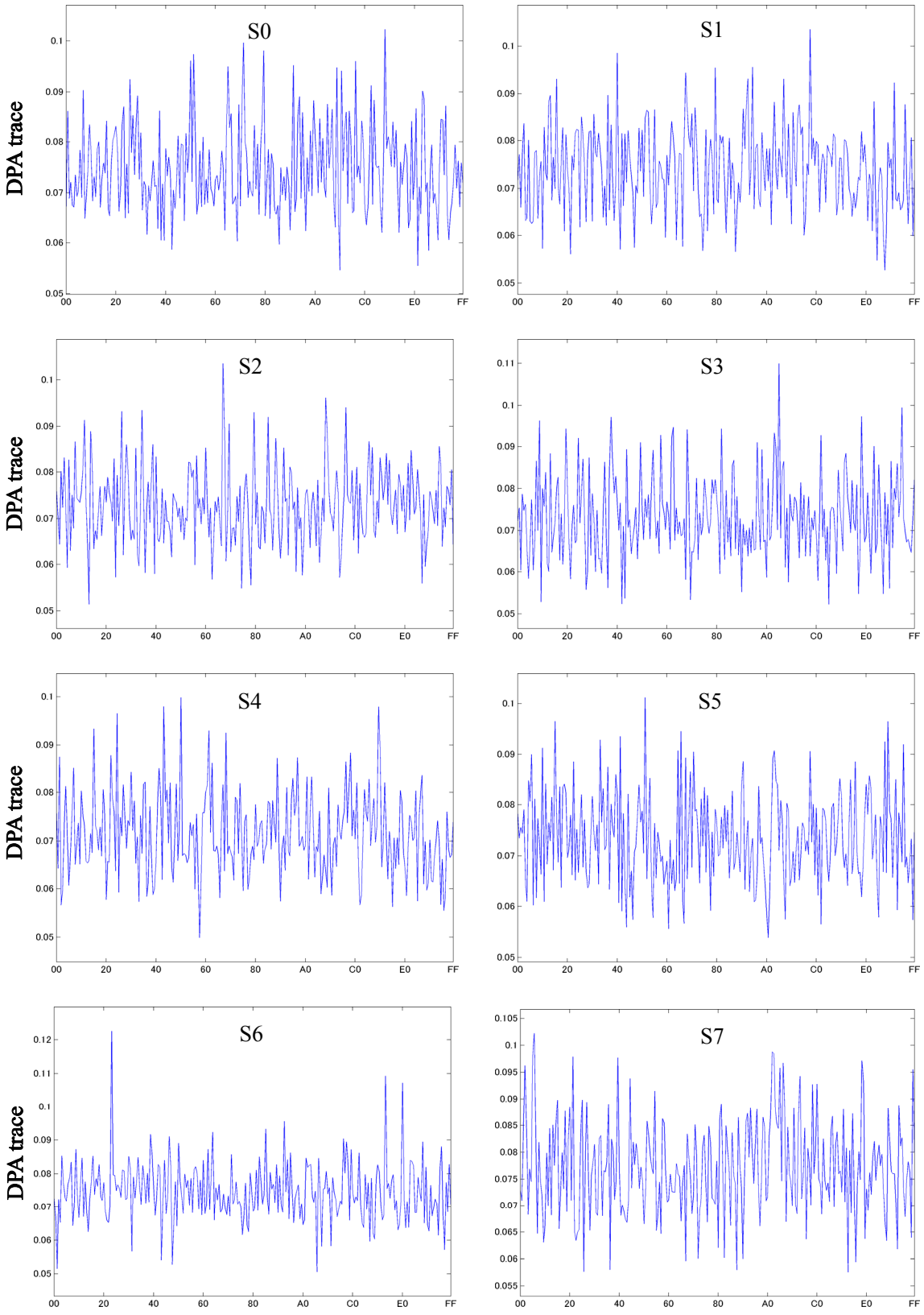


Figure 24-1 Average power differences (DPA traces) from DPA on the AES circuit (WDDL) on the SASEBO-G (Precharge phase)

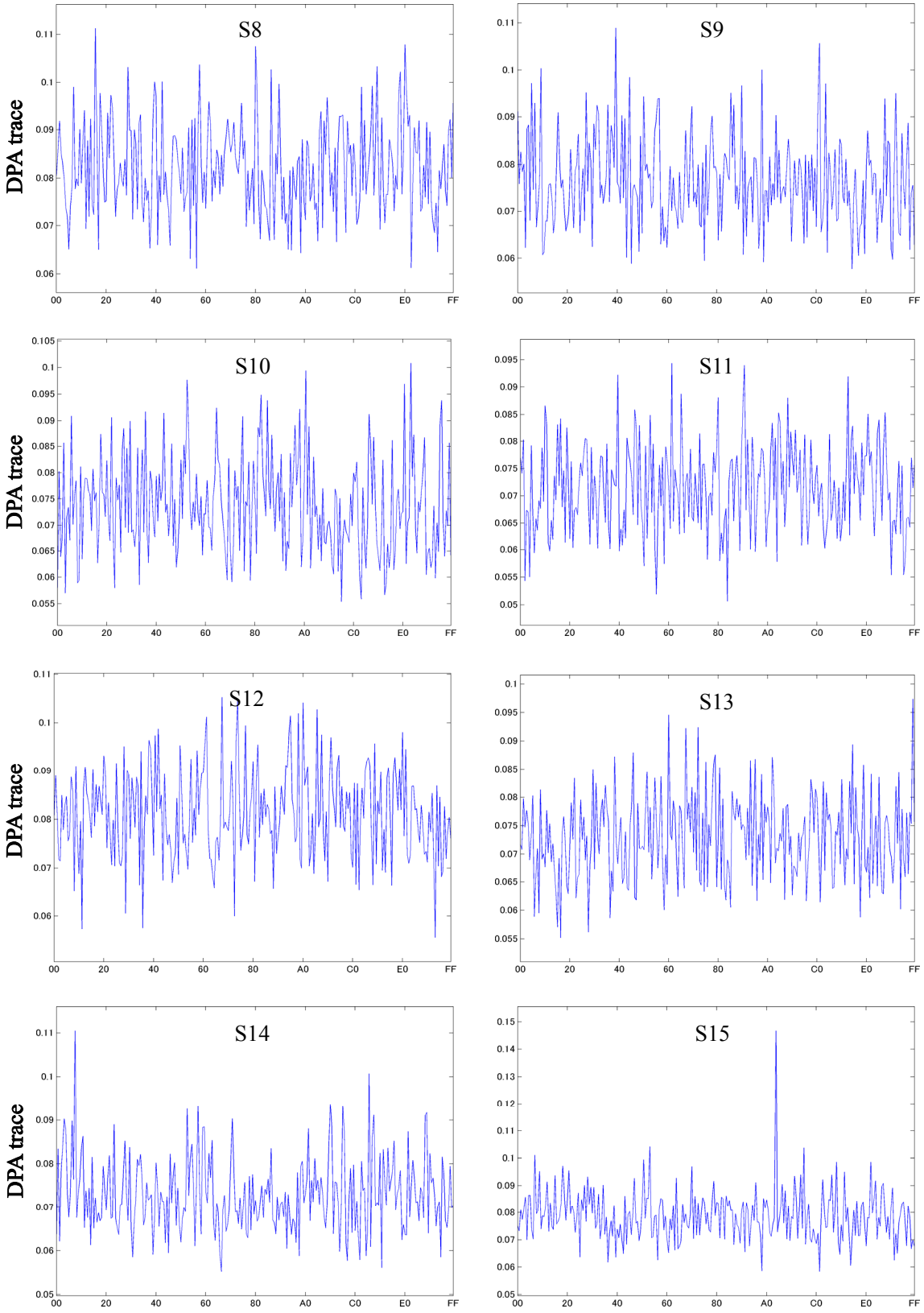


Figure 24-2 Average power differences (DPA traces) from DPA on the AES circuit (WDDL) on the SASEBO-G (Precharge phase)

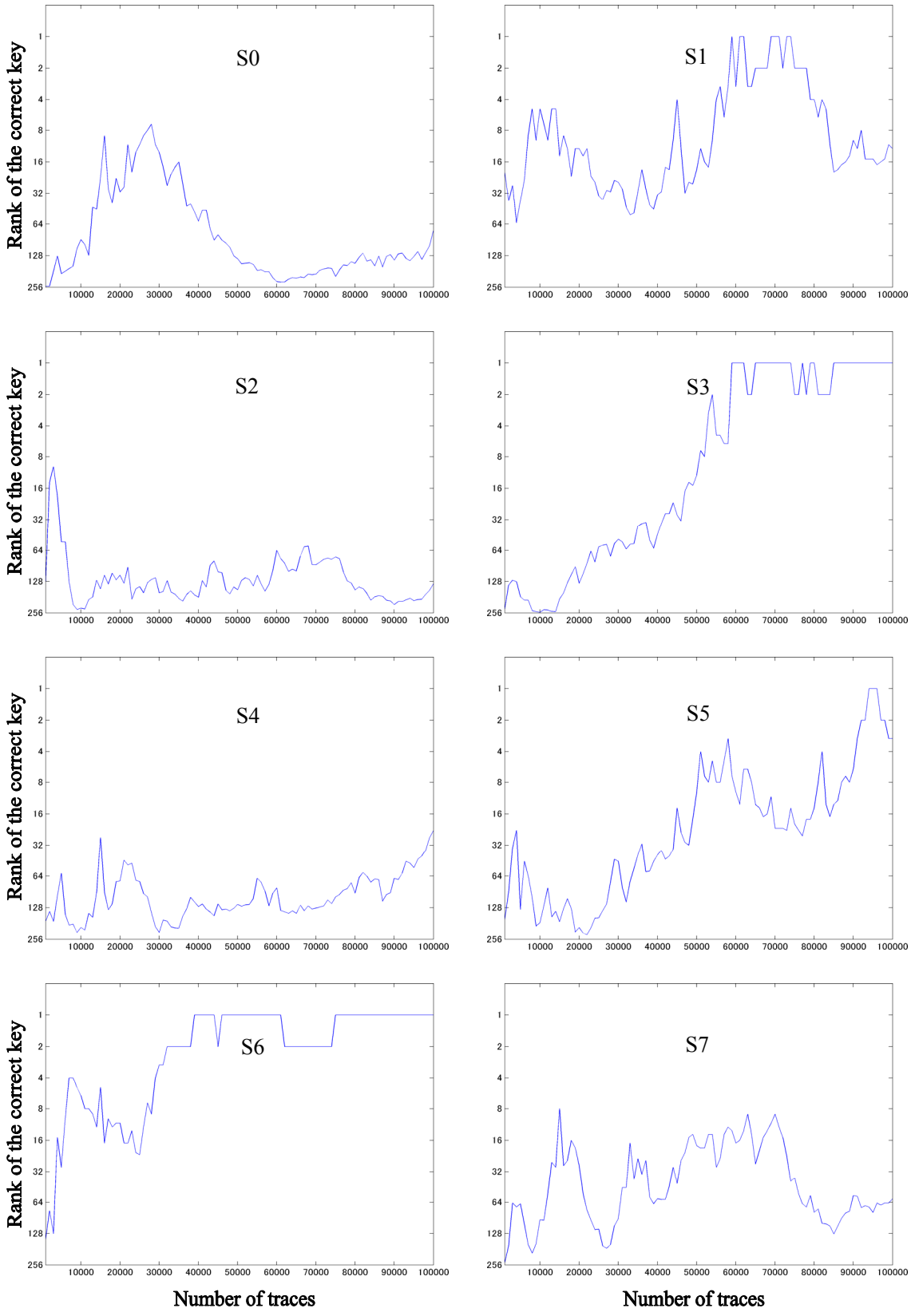


Figure 25-1 Number of power traces versus accuracy of DPA on the AES circuit (WDDL) on the SASEBO-G (Precharge phase)

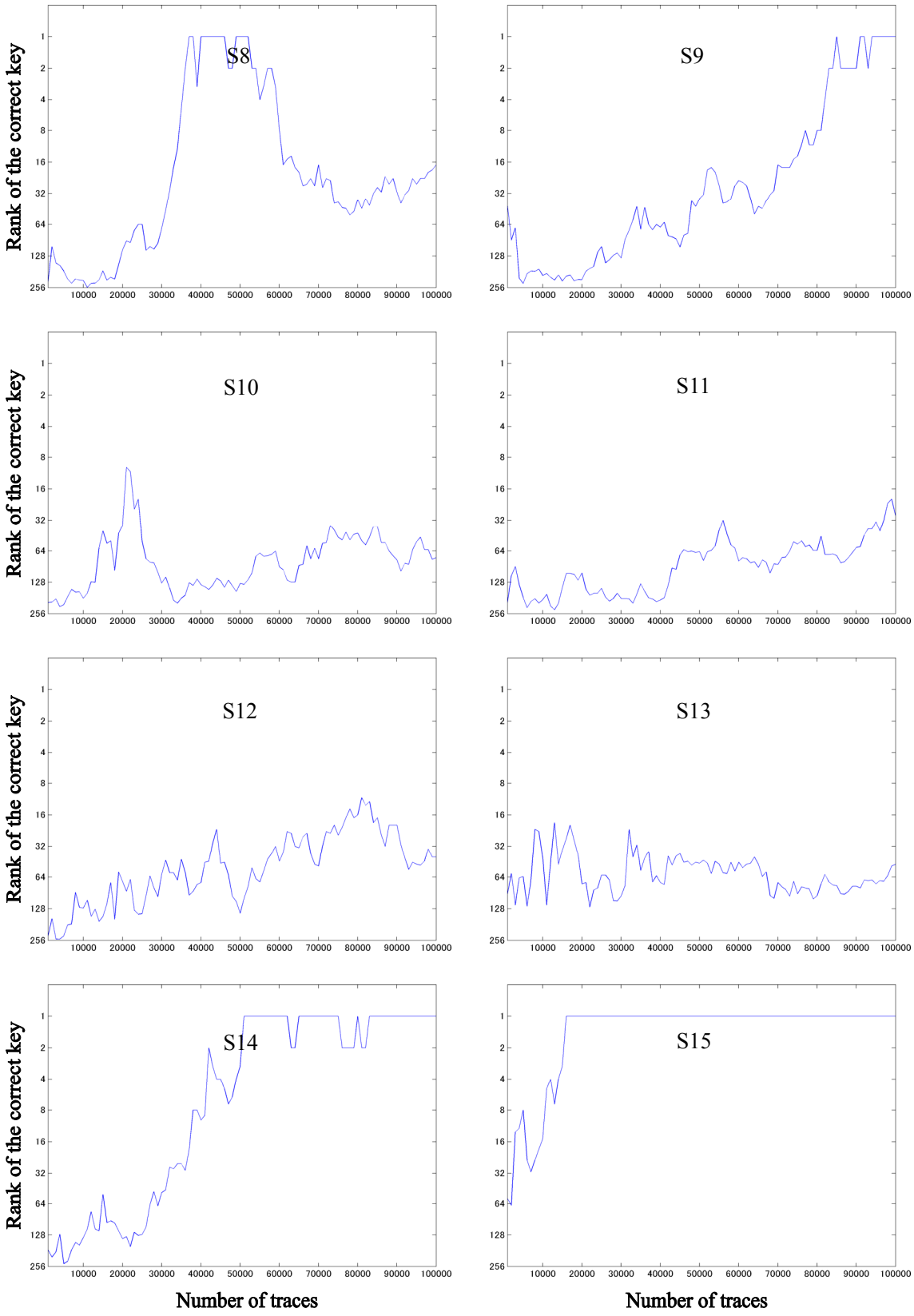


Figure 25-2 Number of power traces versus accuracy of DPA on the AES circuit (WDDL) on the SASEBO-G (Precharge phase)

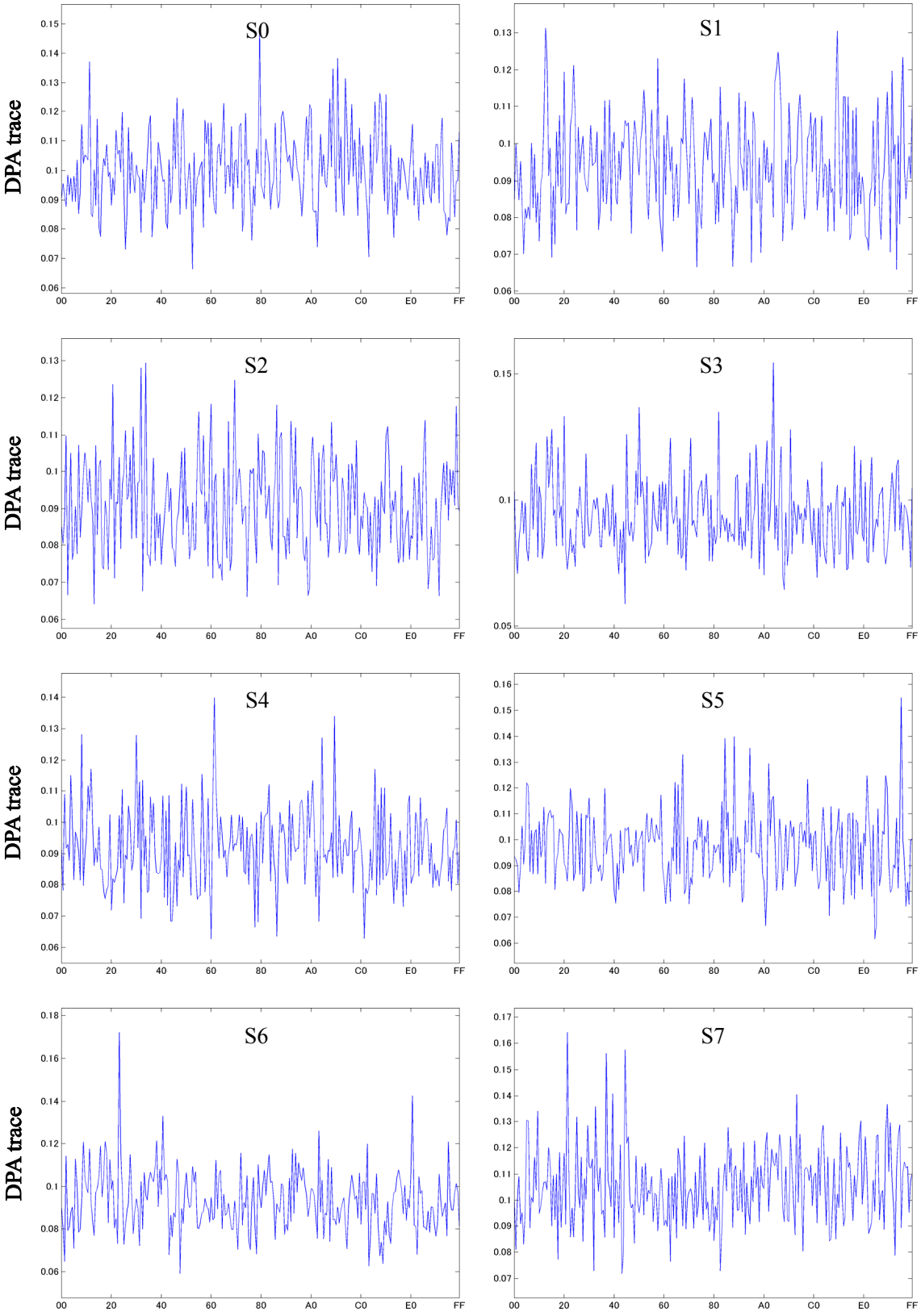


Figure 26-1 Average power differences (DPA traces) from DPA on the AES circuit (WDDL) on the SASEBO-G (Evaluation phase)

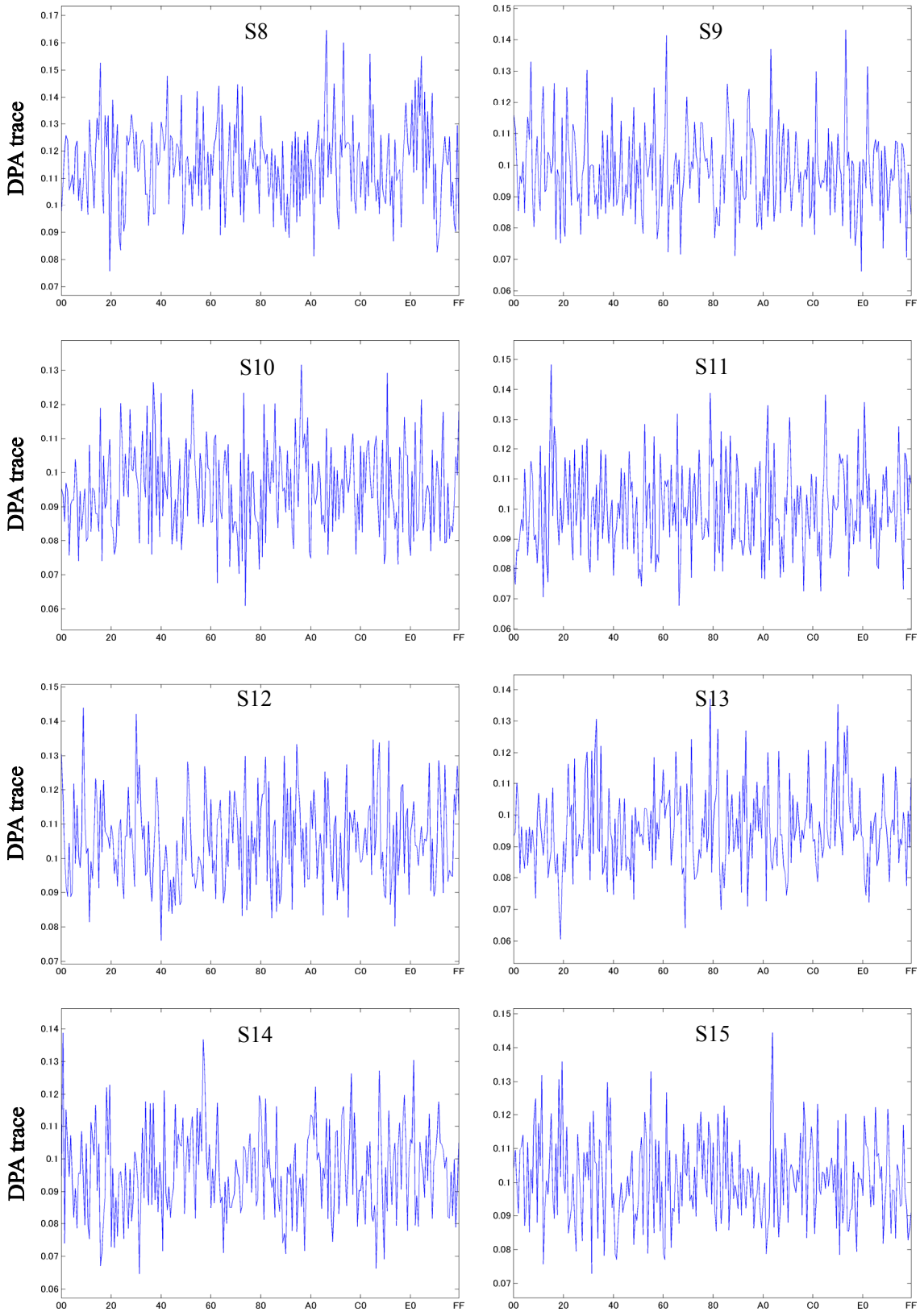


Figure 26-2 Average power differences (DPA traces) from DPA on the AES circuit (WDDL) on the SASEBO-G (Evaluation phase)

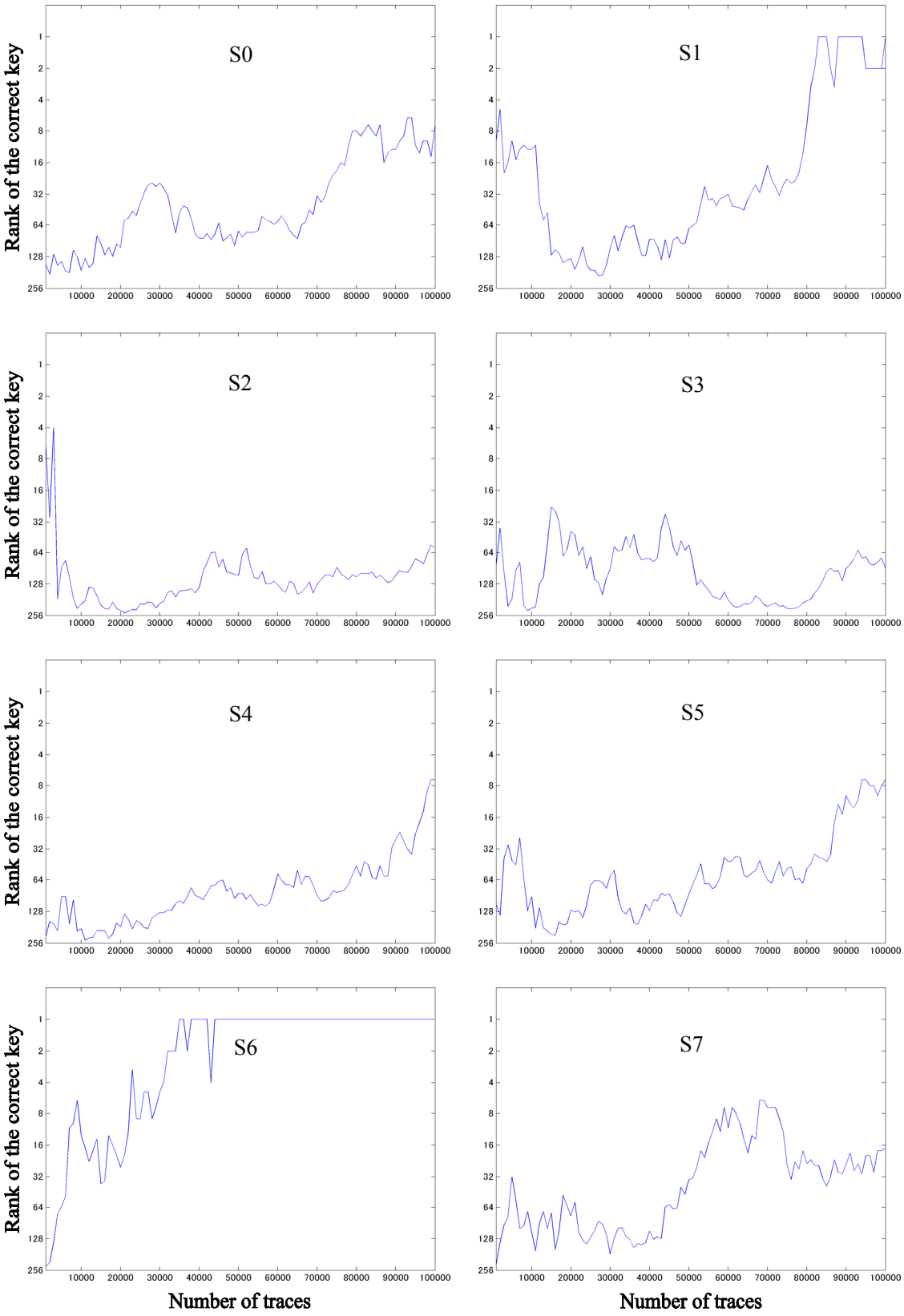


Figure 27-1 Number of power traces versus accuracy of DPA on the AES circuit (WDDL) on the SASEBO-G (Evaluation phase)

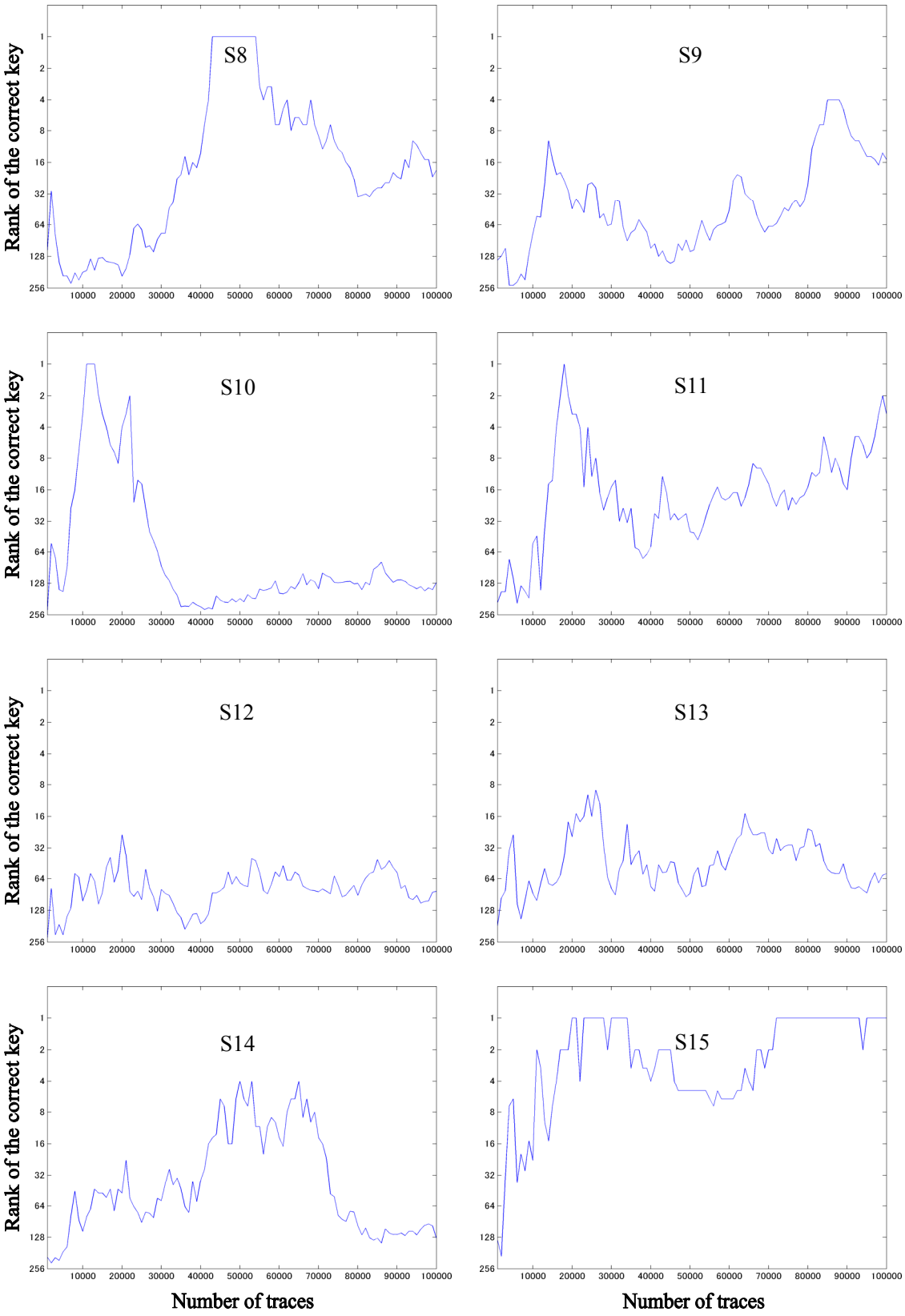


Figure 27-2 Number of power traces versus accuracy of DPA on the AES circuit (WDDL) on the SASEBO-G (Evaluation phase)

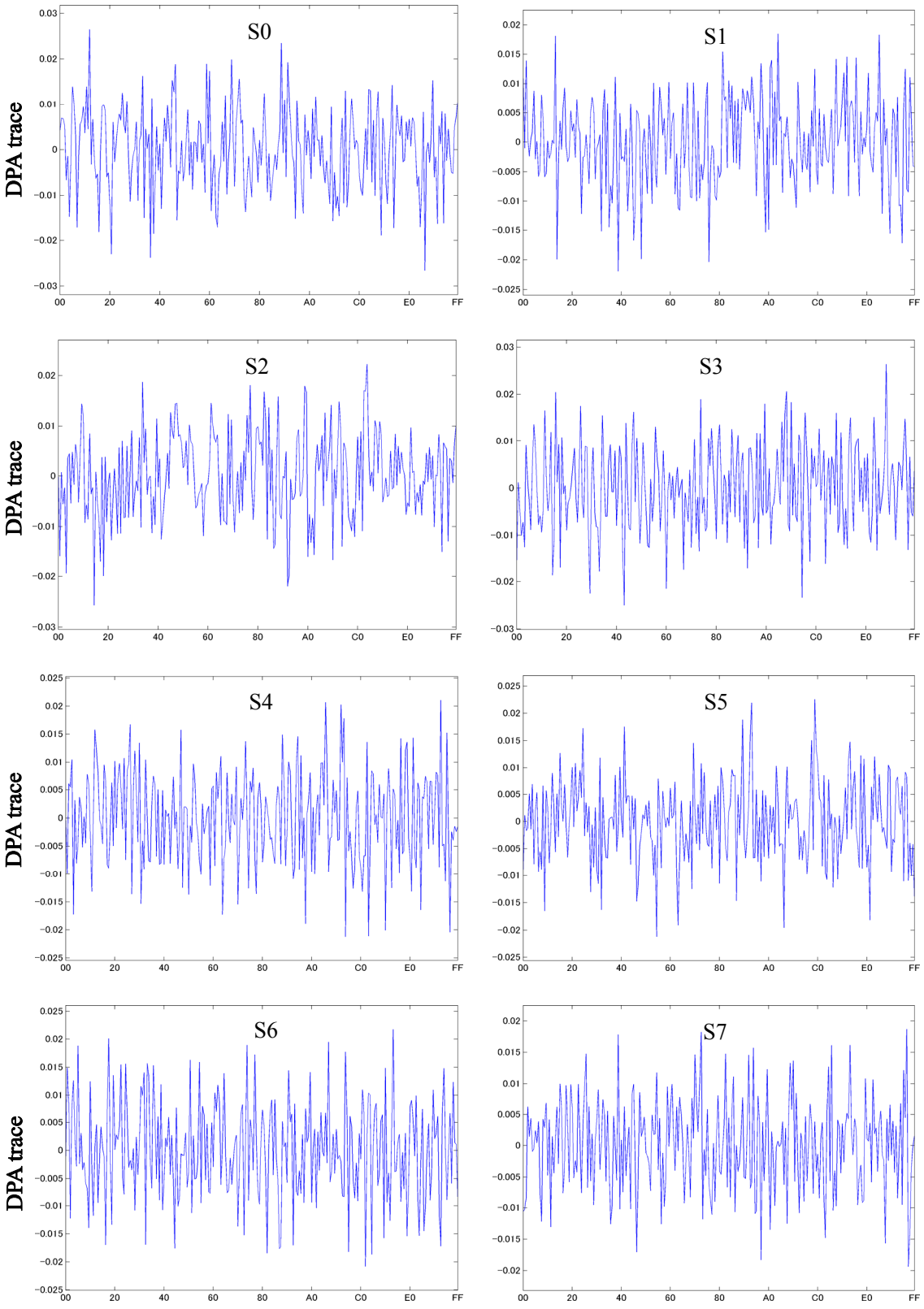


Figure 28-1 Correlation coefficients in CPA on the AES circuit (WDDL) on the SASEBO-G (Evaluation phase)

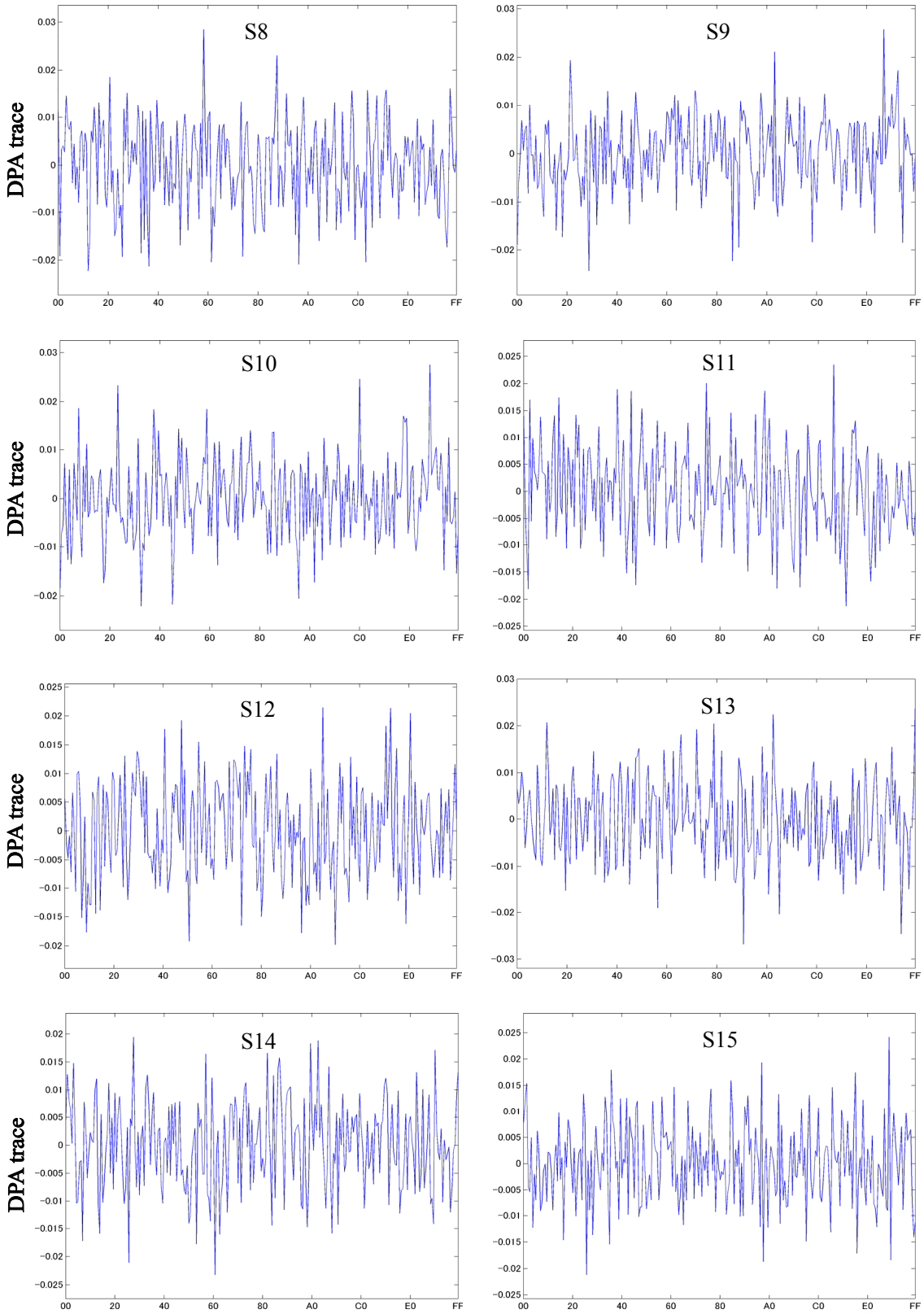


Figure 28-2 Correlation coefficients in CPA on the AES circuit (WDDL) on the SASEBO-G (Evaluation phase)

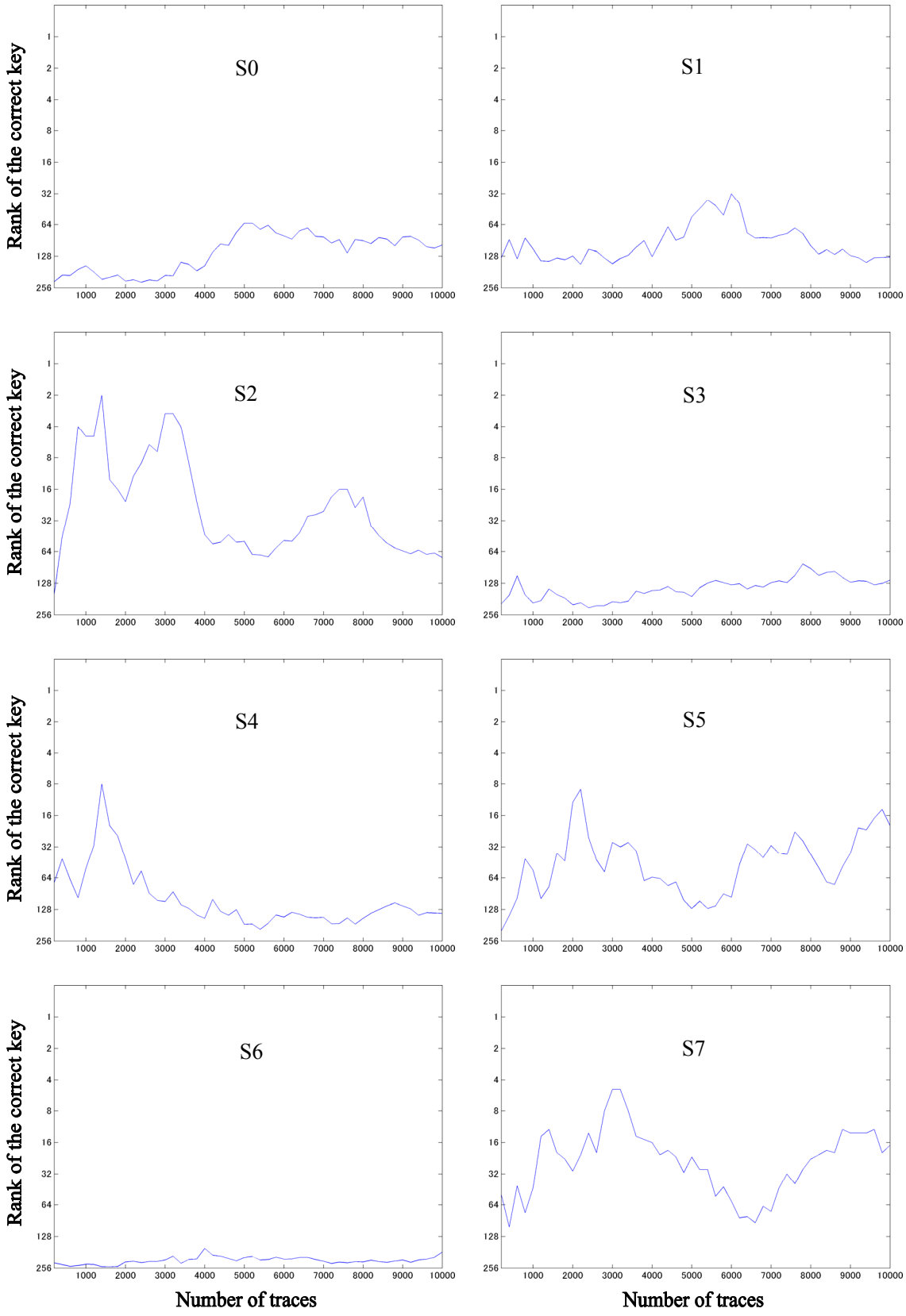


Figure 29-1 Number of power traces versus accuracy of CPA on the AES circuit (WDDL) on the SASEBO-G (Evaluation phase)

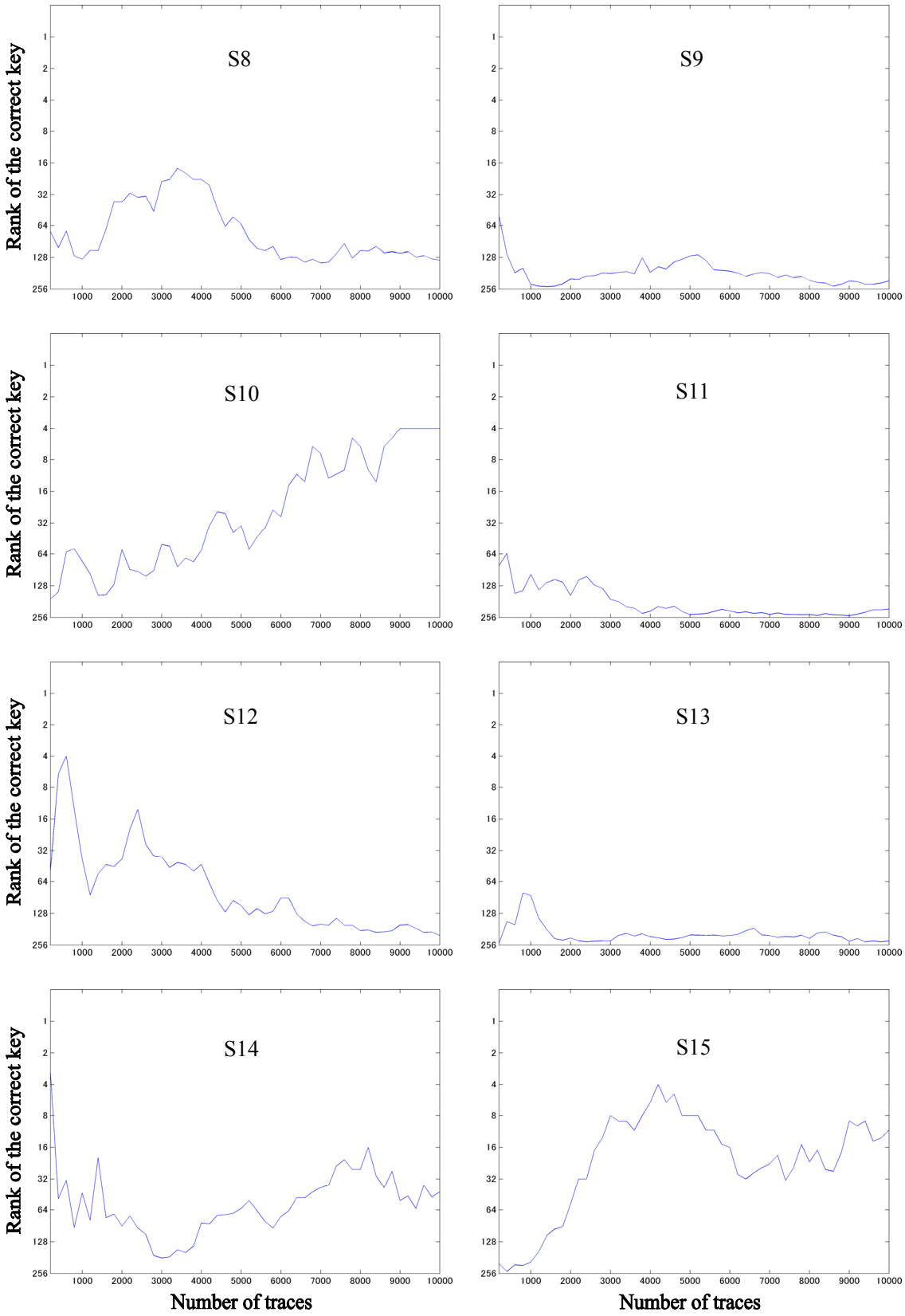


Figure 29-2 Number of power traces versus accuracy of CPA on the AES circuit (WDDL) on the SASEBO-G (Evaluation phase)

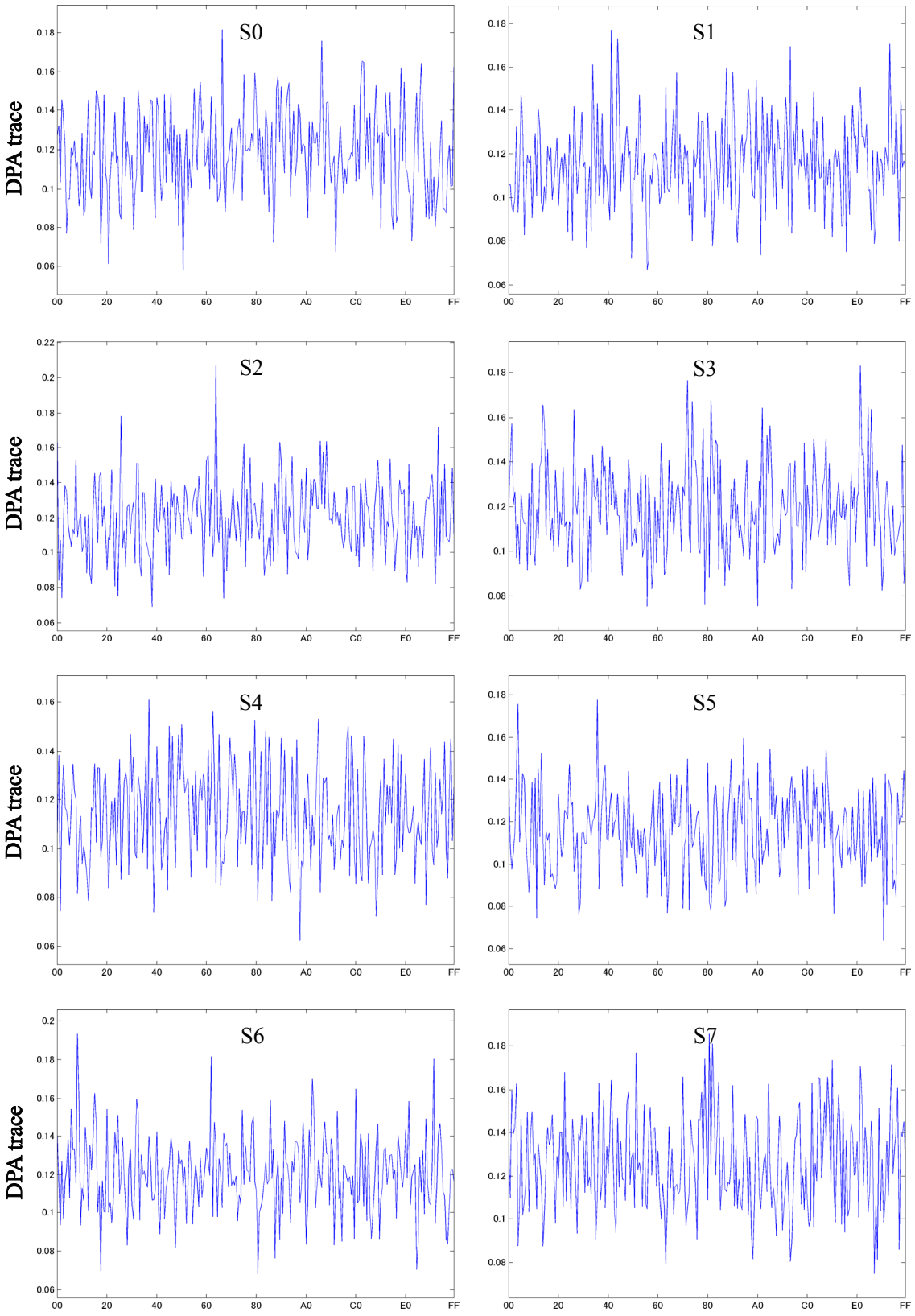


Figure 30-1 Average power differences (DPA traces) from DPA on the AES circuit (MDPL) on the SASEBO-G

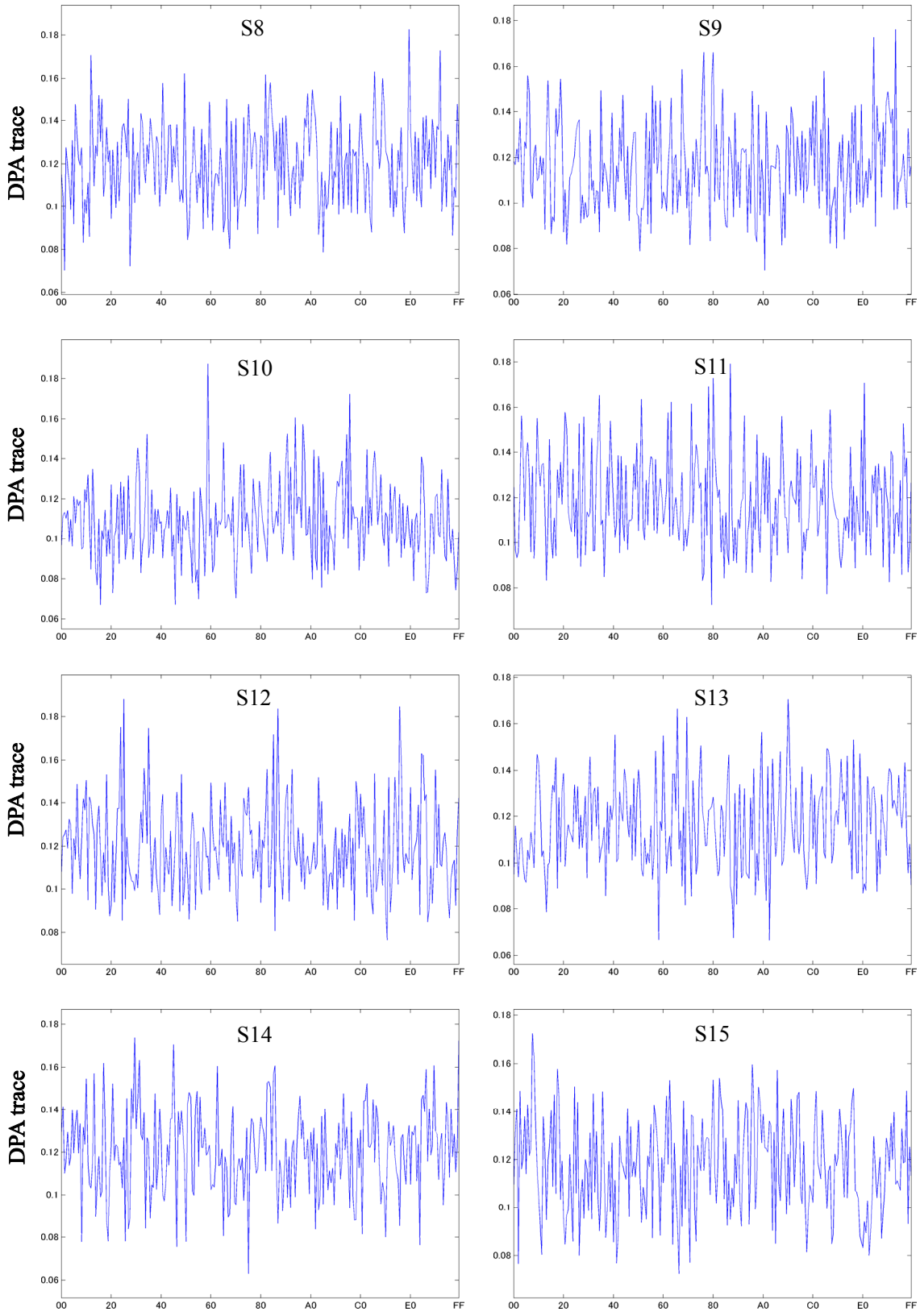


Figure 30 エラー! 参照元が見つかりません。-2　Average power differences (DPA traces) from DPA on the AES circuit (MDPL) on the SASEBO-G

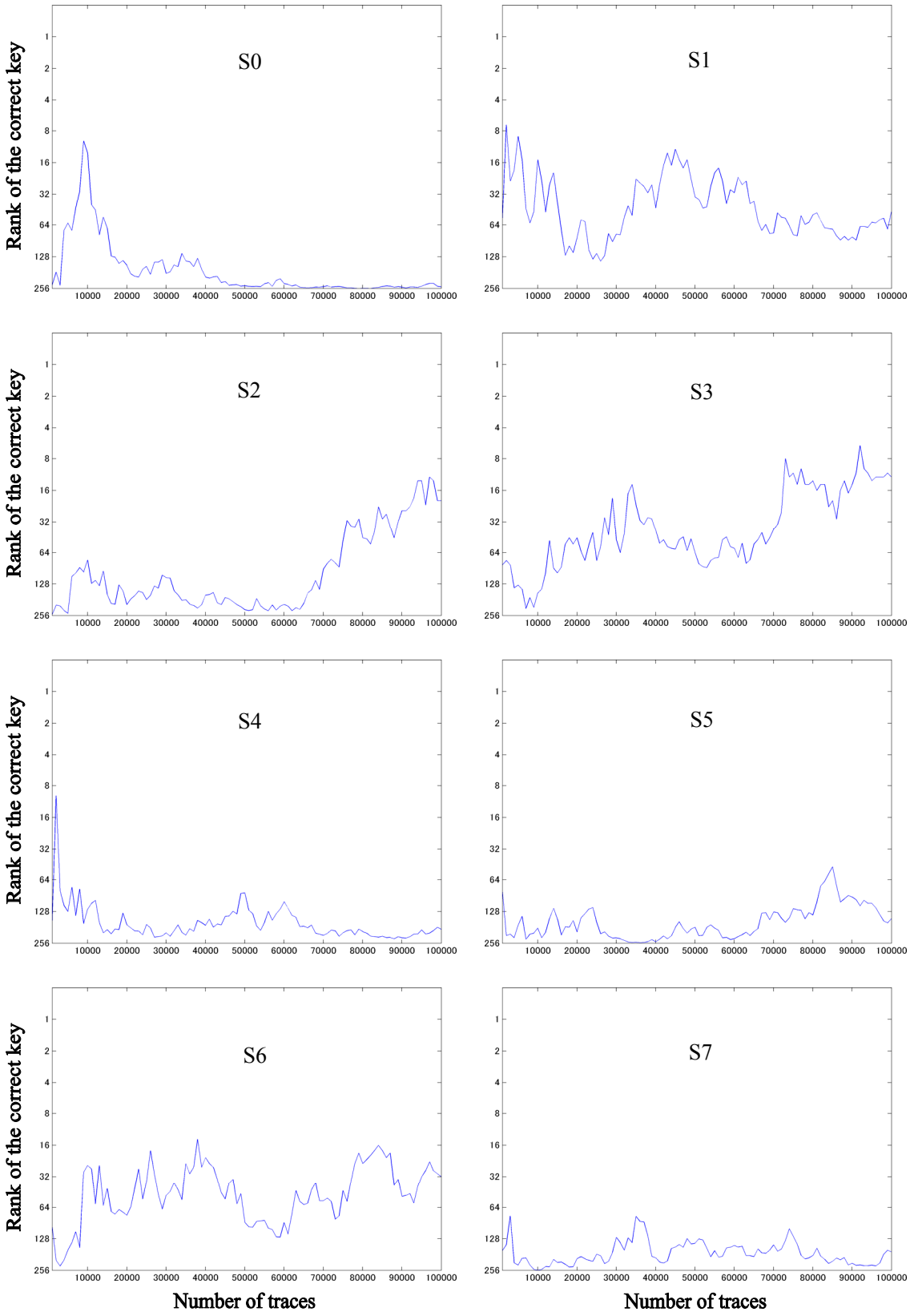


Figure 31-1 Number of power traces versus accuracy of DPA on the AES circuit (MDPL) on the SASEBO-G

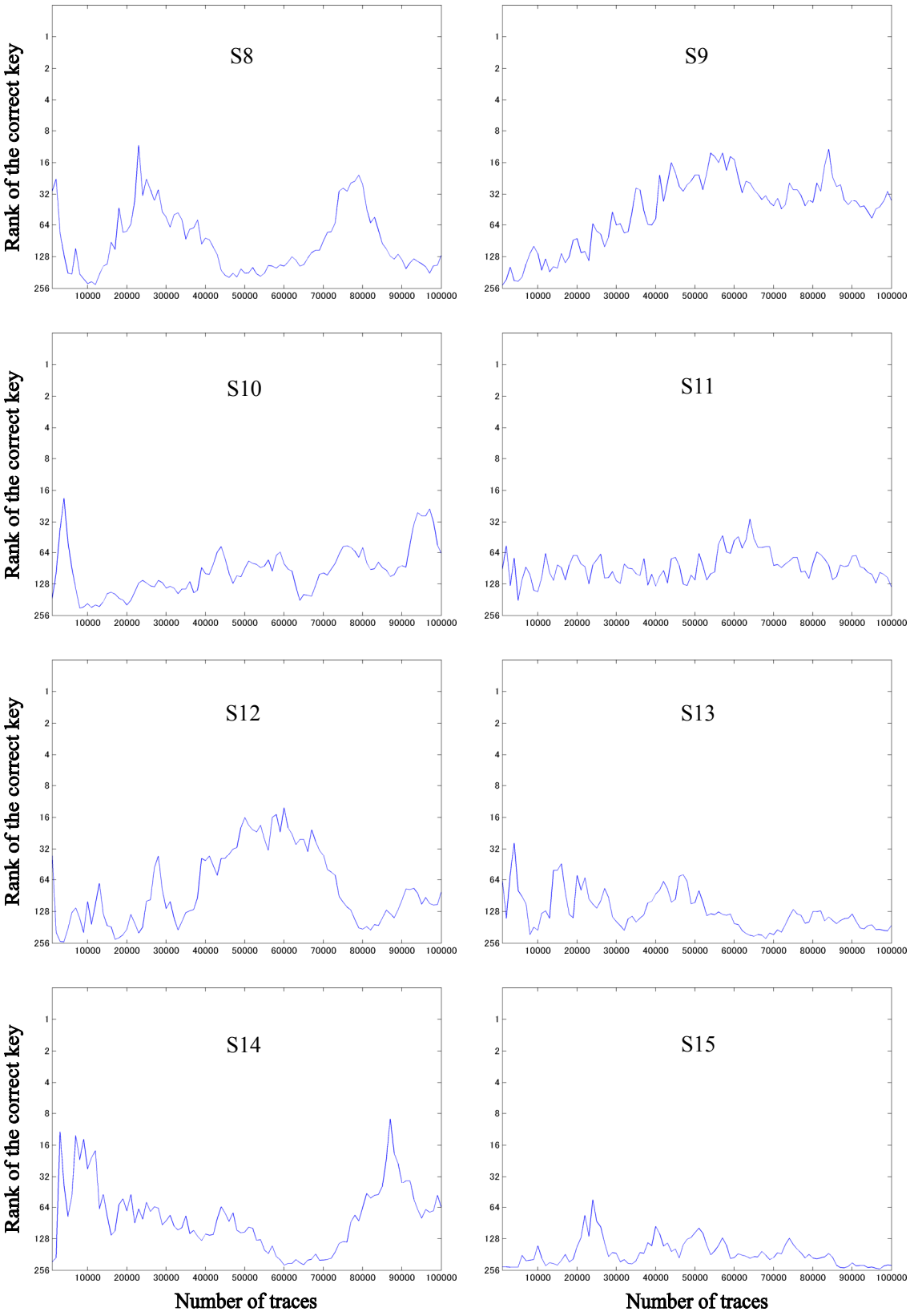


Figure 31-2 Number of power traces versus accuracy of DPA on the AES circuit (MDPL) on the SASEBO-G

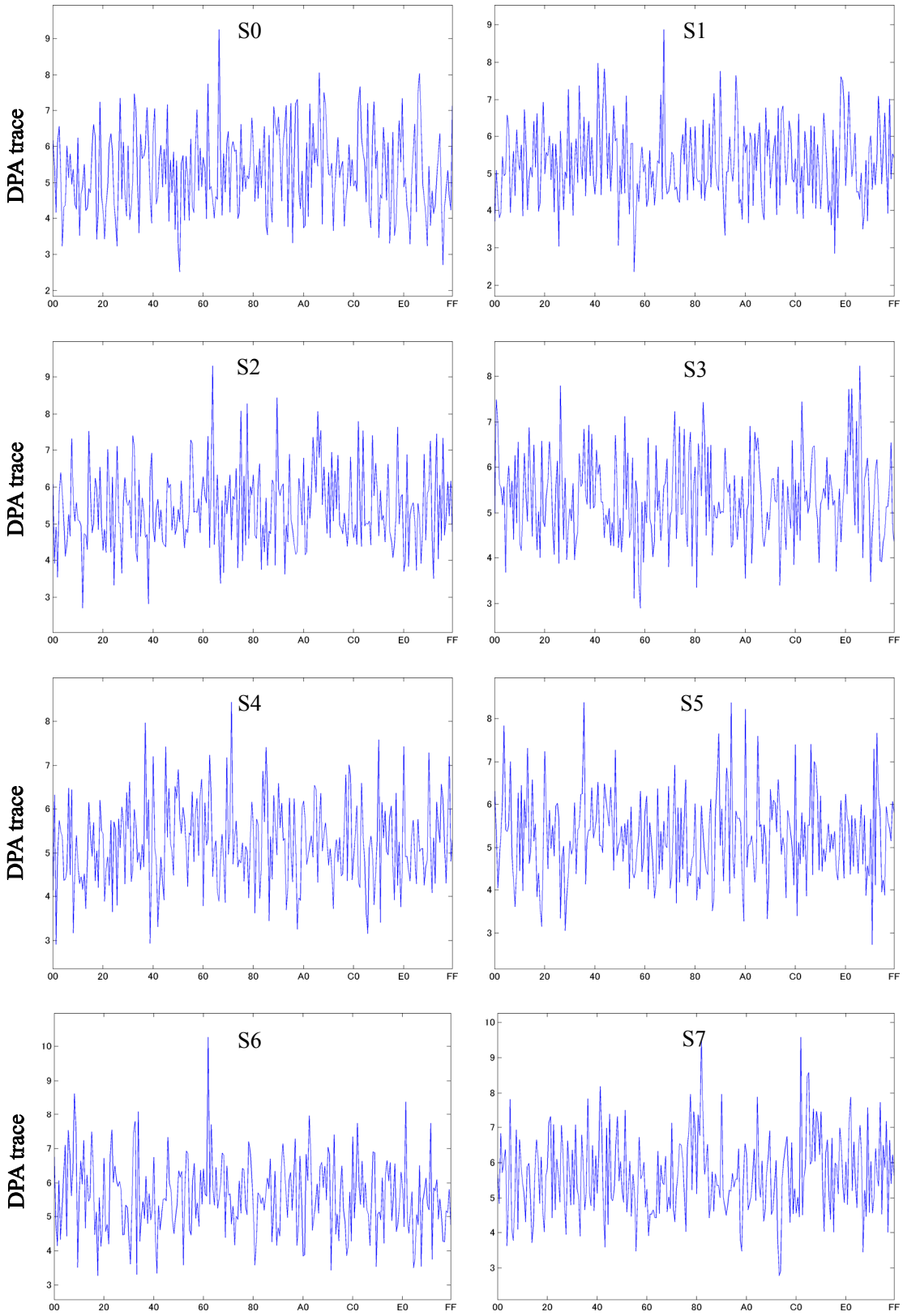


Figure 32-1 Average power differences (DPA traces) from W2-DPA on the AES circuit (MDPL) on the SASEBO-G

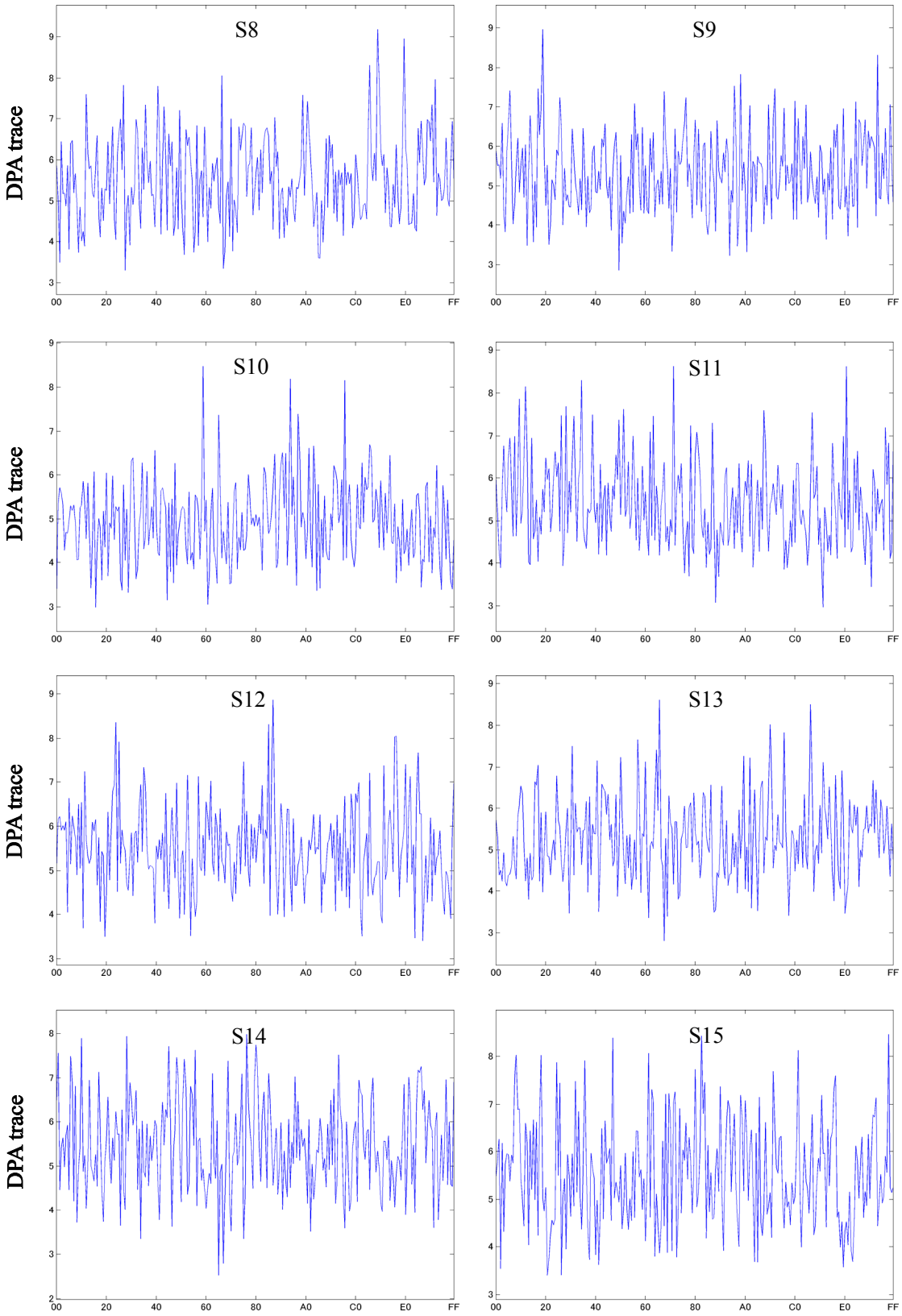


Figure 32-2 Average power differences (DPA traces) from W2-DPA on the AES circuit (MDPL) on the SASEBO-G

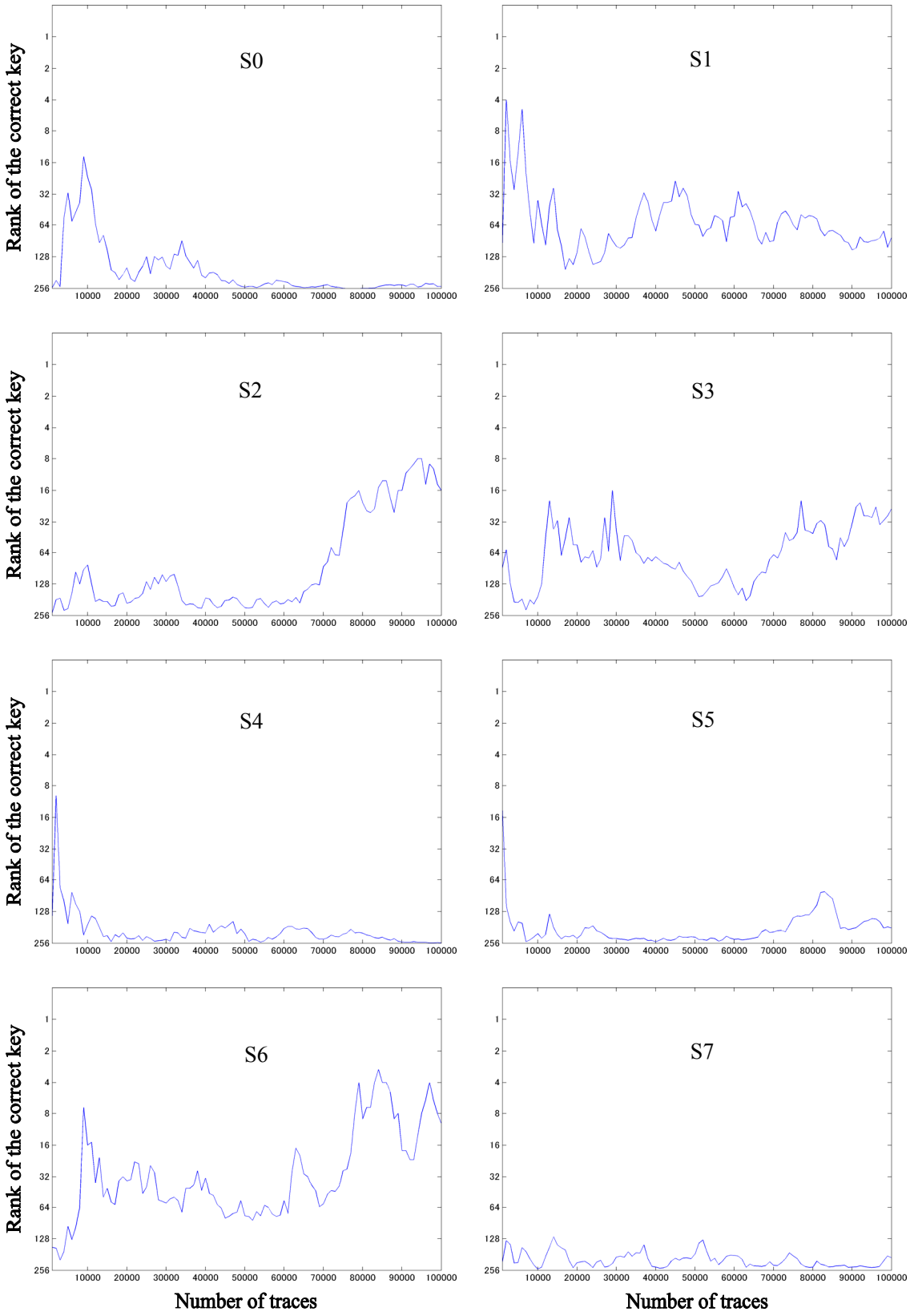


Figure 33-1 Number of power traces versus accuracy of W2-DPA on the AES circuit (MDPL) on the SASEBO-G

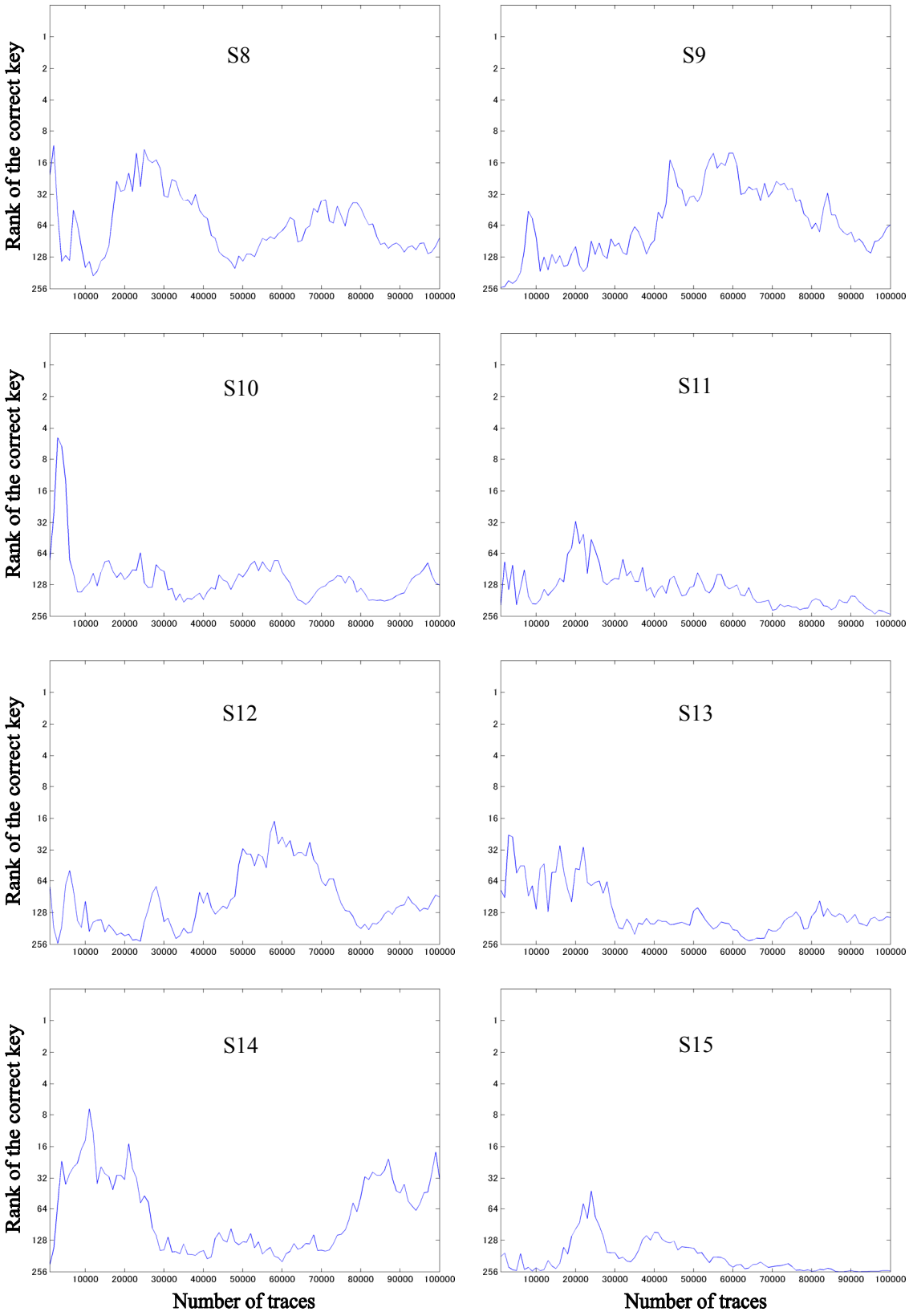


Figure 33-2 Number of power traces versus accuracy of W2-DPA on the AES circuit (MDPL) on the SASEBO-G

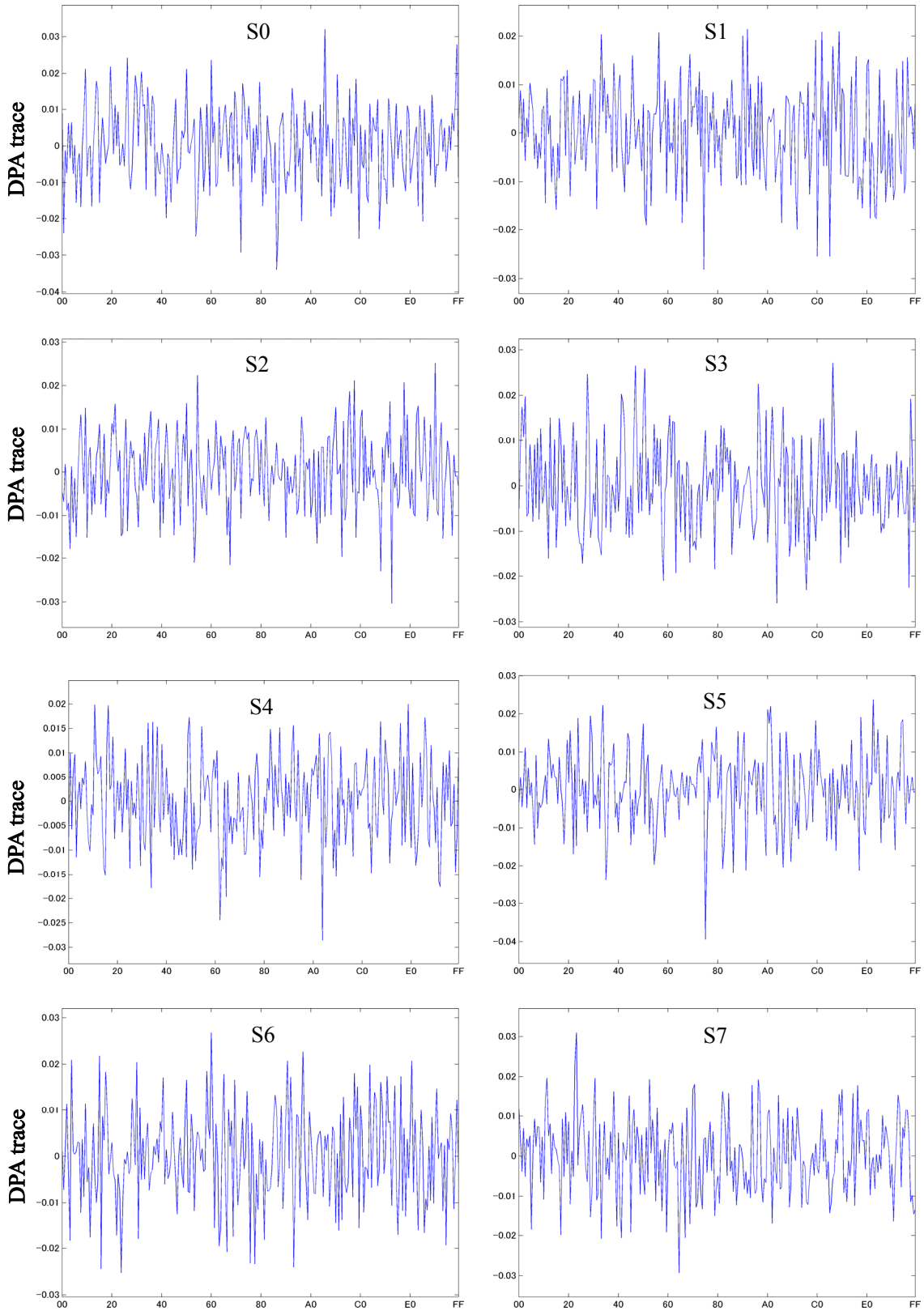


Figure 34-1 Correlation coefficients in CPA on the AES circuit (MDPL) on the SASEBO-G

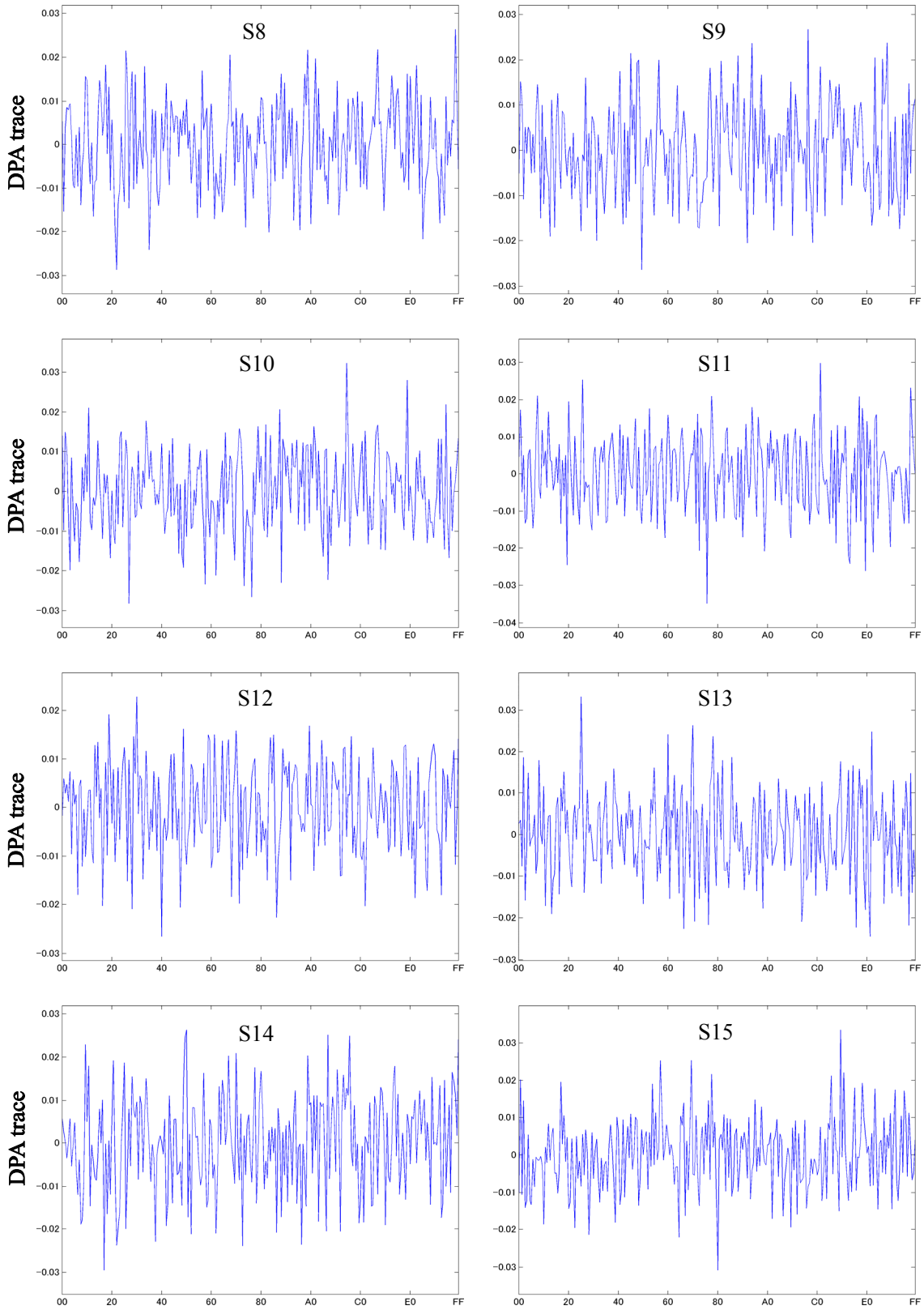


Figure 34-2 Correlation coefficients in CPA on the AES circuit (MDPL) on the SASEBO-G

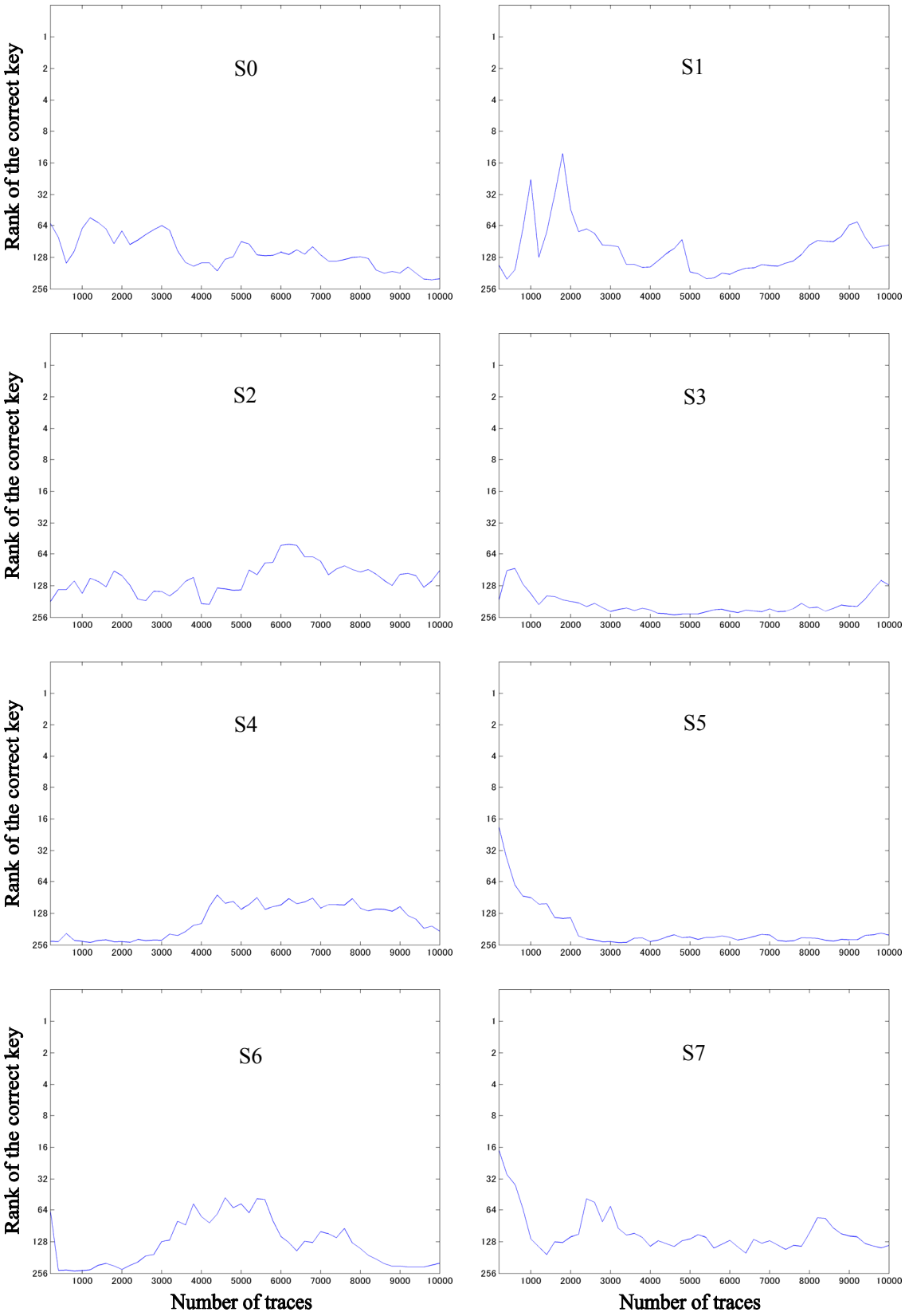


Figure 35-1 Number of power traces versus accuracy of CPA on the AES circuit (MDPL) on the SASEBO-G

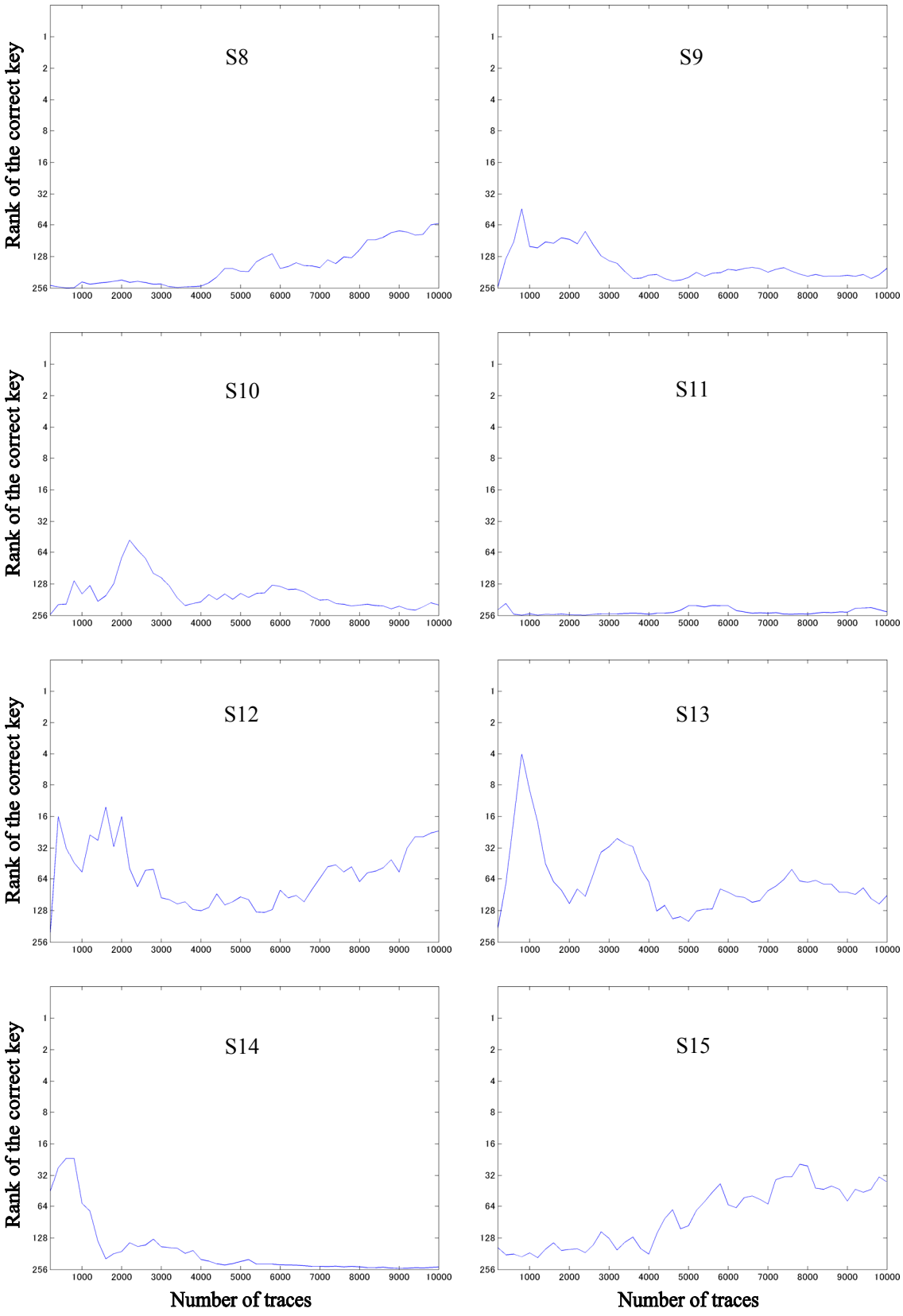


Figure 35-2 Number of power traces versus accuracy of CPA on the AES circuit (MDPL) on the SASEBO-G

3. POWER ANALYSIS ATTACK AGAINST RSA

3.1 Overview

This section deals with attacks against public key ciphers, particularly explaining Simple Power Analysis (SPA) and its variations against the RSA scheme. Through a series of experiments performed for the RSA scheme implemented on each of the cryptographic LSI and FPGA under the measurement conditions shown in Table 3, we will examine the attack methods' effectiveness.

The RSA scheme is a public key cipher that performs encryption and decryption with modular exponentiation operations. Let P be original data (plaintext), let C be ciphertext, let E and N be public key, and let D be secret key, we obtain expressions of encryption and decryption as in the following equations:

Encryption: $C = P^E \bmod N$
Decryption: $P = C^D \bmod N$

From a security perspective, 1,024-bit or longer multiple-precision integers are typically used as the modulus N, namely the public key, and the secret key D. The same word length as the modulus N is used for the plaintext P and ciphertext C. The modular exponentiation operation in the RSA scheme is realized by iterating modular square and modular multiply operations (we will refer to these as squaring and multiplication, respectively, for simplicity), reflecting the bit pattern of the exponent E or D. The most basic algorithm for its computation is the *binary method*. The method is further classified into the *left binary method* and the *right binary method* depending where the operation begins. For the left binary method, it begins from the left end (most significant) bit of the exponent bits, while from the right end (least significant) bit for the right binary method. For the both methods, a bit '0' involves a squaring cycle, whereas a bit '1' invokes both squaring and multiplication cycles, and the entire modular exponentiation operation completes by repeating the cycle operations as many times as the number of bits of the key. While the left binary method uses a single intermediate variable, the right binary method requires two of them. Accordingly, the left binary method is commonly used because of its higher implementation efficiency. Thus, the following sections describe attack methods targeting the left binary method.

Table 3 Measurement conditions

Measurement Factor	Condition
Digital oscilloscope	Agilent MSO6104A
Sampling frequency	800MSample/sec
Probe	Coaxial cable (50 Ω)
Stabilized power supply	3.3 V
Operating clock frequency	24 MHz
Measurement point	Both ends of the 1 Ω resister inserted in the GND line of the targeted cryptographic LSI or FPGA

3.2 Simple Power Analysis (SPA)

SPA[1)] is one of the most fundamental power analysis attacks. It is used to estimate the secret information directly from a power trace measured during a cipher operation. Because the RSA scheme deals with long words in its computation, it takes hundreds or thousands of cycles for a single modular exponentiation operation even with dedicated hardware. As shown in Figure 36, SPA on RSA derives the secret information by identifying the power trace segments of each of square and multiply operations. The factors of the difference in their power consumption include deviations in switching characteristics of transistors, operation times, and difference in the control logic.

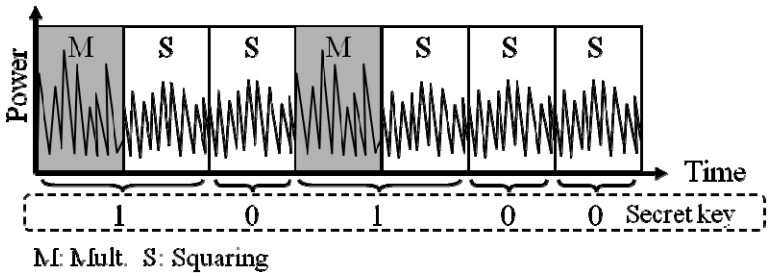


Figure 36 SPA on the RSA scheme

Figure 37(a) and (b) represent the power traces for the RSA hardware implemented with the same Verilog-HDL code on the ASIC (In this section, we refer to the cryptographic LSI as ASIC.) of the SASEBO-R and on the FPGA of the SASEBO-G, respectively. Both used random numbers as inputs. The power trace for the FPGA is about 5 times larger in amplitude than for the ASIC. A significant difference in shape between their power traces can also be seen. Thus, even though they have the similar circuit structures, the power traces significantly differ from each other depending on the implementation conditions and the device. It is difficult to distinguish between square operations and multiply operations on the power traces in Figure 37 (a) and (b). However, applying a low pass filter (cut-off frequency of 80 MHz) to the oscilloscope input to eliminate noise made a clear difference for the FPGA as shown in Figure 38.

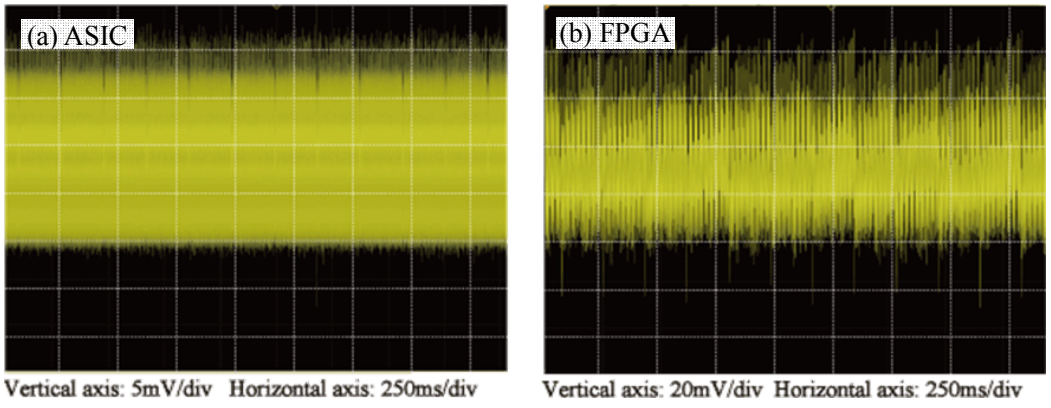


Figure 37 Power traces with random number inputs (without filtering)

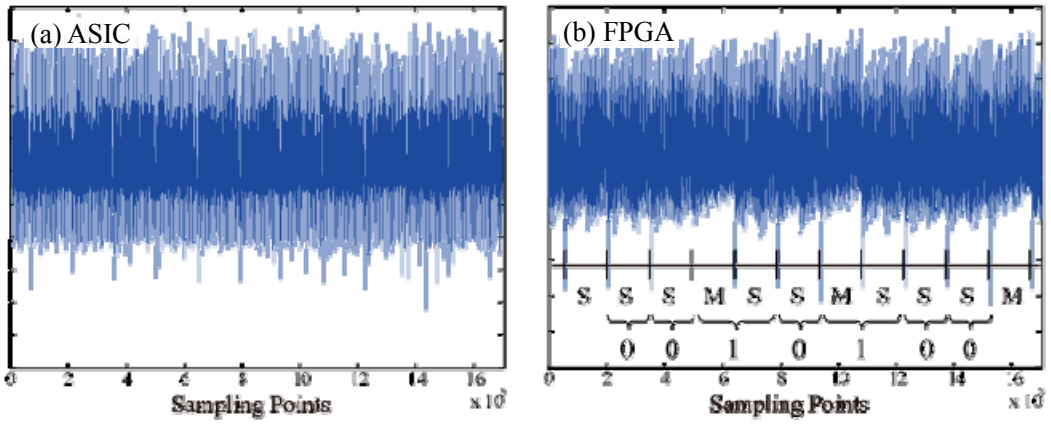


Figure 38 Power traces with random number inputs (with filtering)

3.3 Chosen-Plaintext SPA

For SPA, it is necessary to distinguish between the power trace shapes of square operations and multiply operations executed in a modular exponentiation operation. However, as shown above, the difference is not necessarily observable for a random number input because the operation data is different every time. Furthermore, if a single computing unit executes squaring and multiplication in the same sequence, distinguishing these operations will become harder. To address this problem, some attack methods that combine SPA with chosen plaintext to enhance the difference of the operations depending on a key bit have been proposed.

- **SPA with the input *N*-1**

Let the input be *N*-1, the left binary method's operations are classified into the following three types depending on the key bit pattern: (M) multiplication after squaring, (S1) squaring after multiplication, and (S2) squaring after squaring[13].

(M) $1 \times (-1) \bmod N = -1 \bmod N$

(S1) $(-1) \times (-1) \bmod N = 1 \bmod N$

(S2) $1 \times 1 \bmod N = 1 \bmod N$

These relationships are invariant through the entire left binary method's sequence. In addition, Montgomery multiplication, which is a fast computation technique of modular multiplication, can be applied. In Montgomery multiplication, because the radix of the operation is transformed into the Montgomery domain $(\times 2^k \bmod N)$, the above equations are given as follows:

(M) $2^k \times (-2^k) \times 2^{-k} \bmod N = -2^k \bmod N$

(S1) $(-2^k) \times (-2^k) \times 2^{-k} \bmod N = 2^k \bmod N$

(S2) $2^k \times 2^k \times 2^{-k} \bmod N = 2^k \bmod N$

Because this attack method estimates the key bit pattern from the difference of the power consumption of M, S1, and S2, detailed knowledge of the implemented modular multiplication algorithm or circuit architecture is not required. Further, since there are only three power trace patterns, it is easy to determine which of M, S1, and S2 is associated with the pattern. This determination is also applicable using a known public key.

Figure 39 shows the concept of the SPA attack with the input data *N*-1. With the left binary method, M and S1 always make a pair, leaving S2 a single entity to appear. Therefore, it is not necessary to identify all the three shapes in the power trace, but possible to estimate the key at a high probability as long as only one of the shapes is distinguished. Taking advantage of the emergence order of the shape patterns is one of the remarkable features of the attack using *N*-1 as the input.

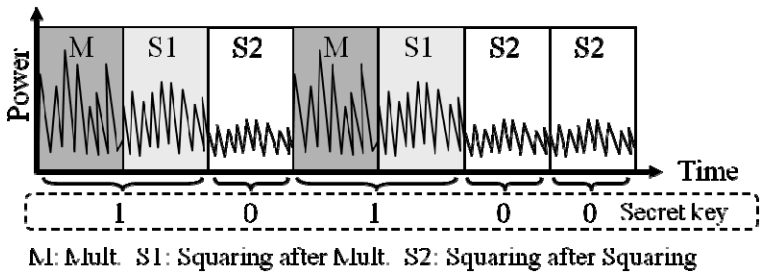


Figure 39 Chosen plaintext SPA with the input *N*-1

Figure 40 shows the power traces measured on both the ASIC and FPGA with the same input *N*-1. Both traces appear distinguishable between squaring and multiplication. Similar to Figure 38, Figure 41 illustrates that a low pass filter effectively cut the noise component so that the differences between the operations became clearer.

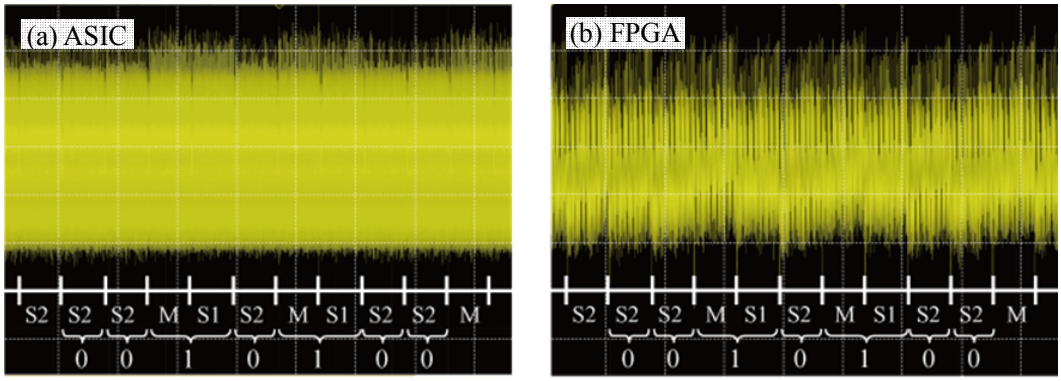


Figure 40 Power traces with the input *N*-1 (without filtering)

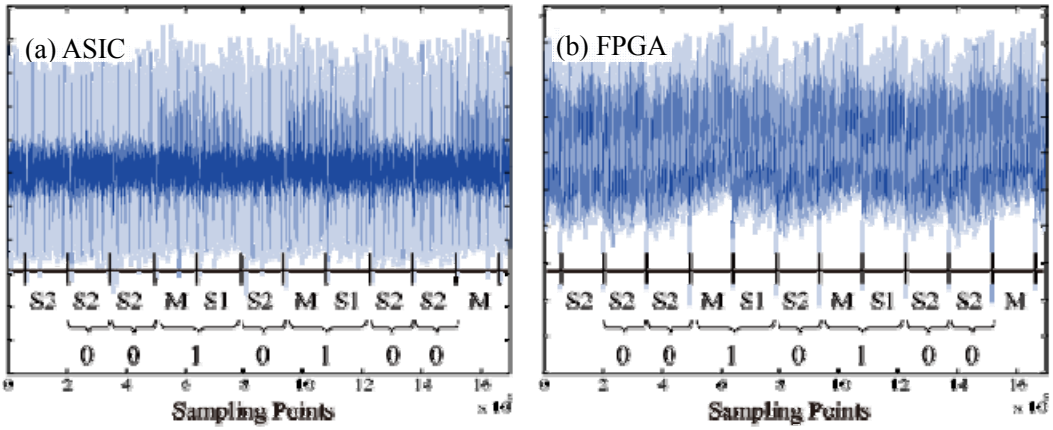


Figure 41 Power traces with the input *N*-1 (with filtering)

The attack using the input *N*-1 also works on the typical SPA countermeasure that inserts dummy multiply operations. The countermeasure strategy is to make a pair of squaring and multiplication for every bit of the key by performing dummy multiplication when the bit is '0'. As a result, the SPA shown in Figure 36 cannot extract the exponent, namely the key. However, if *N*-1 comes to the input data, every squaring S2 follows dummy multiplication DM, while every S1 comes after the original multiplication M as shown in Figure 42. Therefore, if the sequence of M→S2 is observed, it can be determined as dummy multiplication. Figure 43 indicates a power trace of the FPGA implementation employing the dummy operation countermeasure supplied with the input *N*-1.

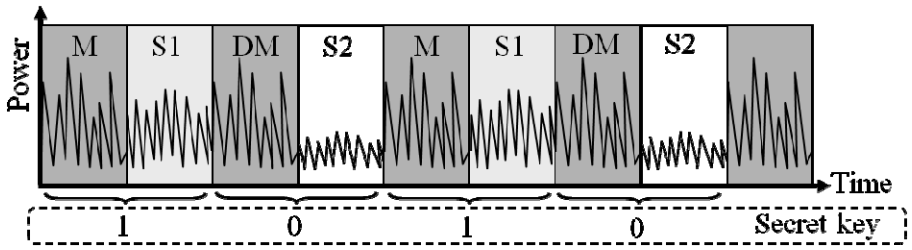


Figure 42 Chosen-plaintext SPA with the input *N*-1 on an RSA implementation with the dummy multiplication countermeasure

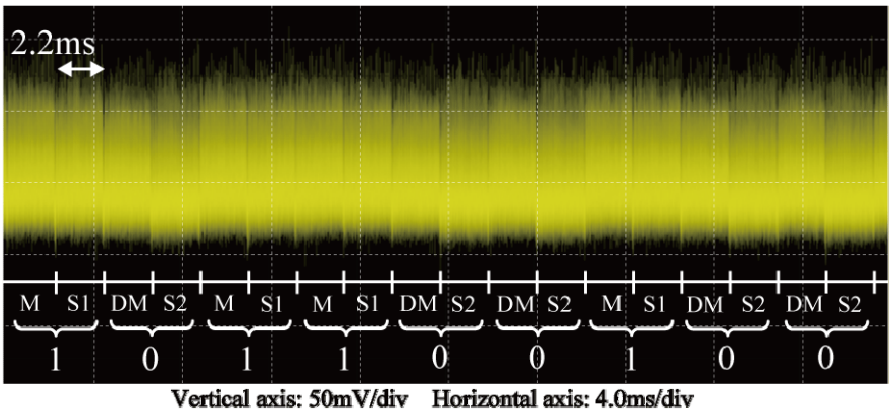


Figure 43 Power trace with the input N-1 on an FPGA implementation of RSA with the dummy multiplication countermeasure (without filtering)

- **SPA with the input 1 (the input 2^{-k})**

For the left binary method, every multiply operation is performed for an intermediate value and the input data. Therefore, by supplying the input with a particular bit pattern, the power consumption of the multiply operations derived by the input data can be relatively lowered. For instance, the input 1 mod N (the input 2^{-k} when using Montgomery multiplication) is applicable such that all the input bits are '0's but the first bit[14].

Figure 44 shows a power trace for the Montgomery multiplication with the input 2^{-k}. Figure 45 represents the same but with a low pass filter applied Both the ASIC and FPGA cases show a clearer distinction between squaring and multiplications than for the input N-1.

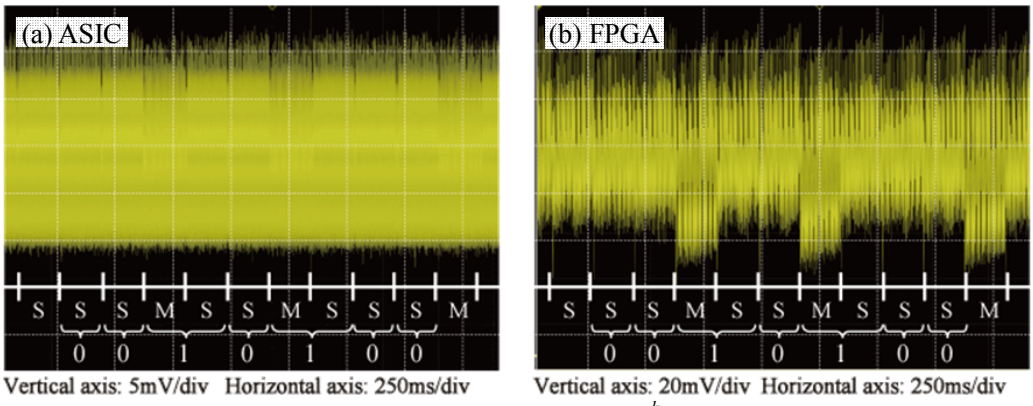


Figure 44 Power traces with the input 2^{-k} (before filtering)

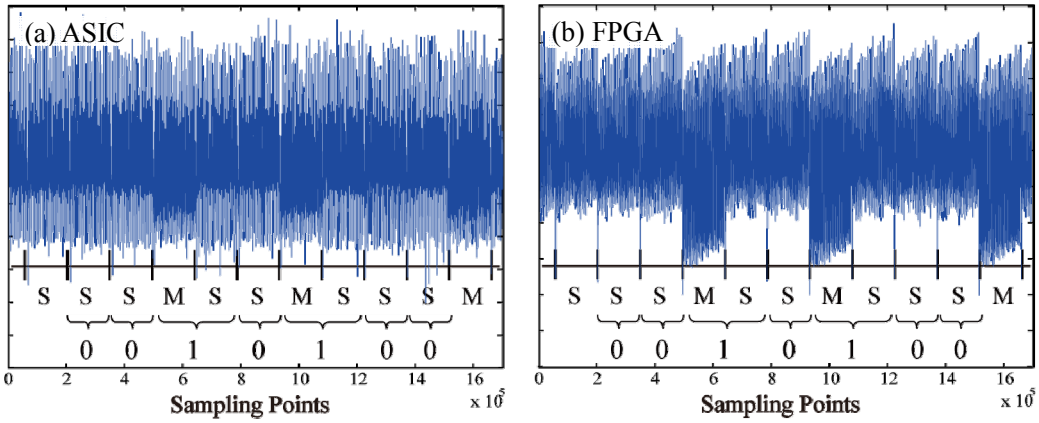


Figure 45 Power traces with the input 2^{-k} (after filtering)

The result suggests that there may be a difference in the power consumption between squaring and multiplication when the ratio of the 0s and 1s in the input is biased. This would even introduce threatening input patterns other than 2^{-k}. Through experiments on the SASEBO-G, we also observed a difference in power consumption between multiplication and squaring with an input biased with about 800 bits of '0's or '1's out of 1,024 bits. This suggests that even if particular input patterns such as N-1 and 2^{-k} are excluded, an RSA implementation can still be threatened for as many as $2^{224=1024-800}\times 2$ input patterns.

3.4 Chosen-Plaintext SPA

One of the powerful chosen-plaintext power analysis methods involves a key estimation method comparing power traces associated with a special input pair. The Doubling attack (or the Squaring attack for modular exponentiation)[15] estimates the key by using two power traces obtained from an input pair of X and X^2. Figure 46 shows an example of the Doubling attack on the left binary method. M and S in the diagram denote multiplication and squaring, respectively. Across the power traces P_X and $P_X{}^2$ for inputs X and X^2, the inputs circled and outputs of the two square operations on the positions shifted by 1 exponent bit to each other will match. The similarity between the square operation Ss is to be detected to determine the type of the operations that reflect the key bit sequence.

On the other hand, Yen *et al*. have proposed a variation of the attack above that uses an input pair of X and $-X$. The attack estimates the key by exploiting the matches of the inputs and outputs between the Ss performed at the same operation cycle as shown in Figure 47.

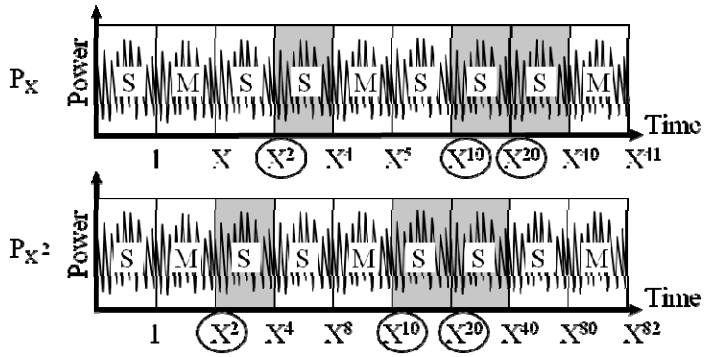


Figure 46 SPA with chosen-plaintext-pair (X , X^2) (Doubling attack)

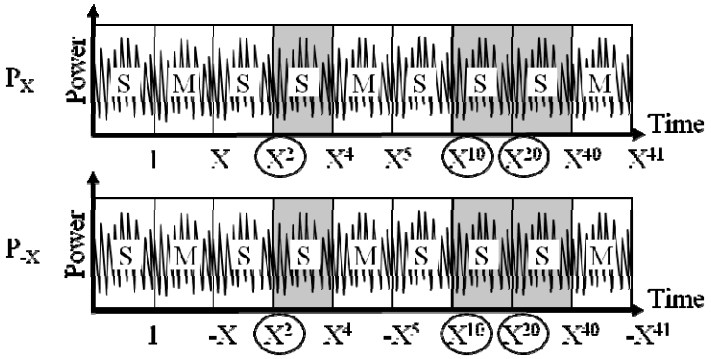


Figure 47 SPA with chosen-plaintext-pair (X , $-X$)

Figure 48 shows the differential power traces obtained from the input pair of X and $-X$. Figure 49 represents the results after applying a low pass filter to them. The differential power trace segments involving pairs of the same operations show smaller magnitudes than other parts. The effect of the low pass filter makes it easier to distinguish multiplication and squaring.

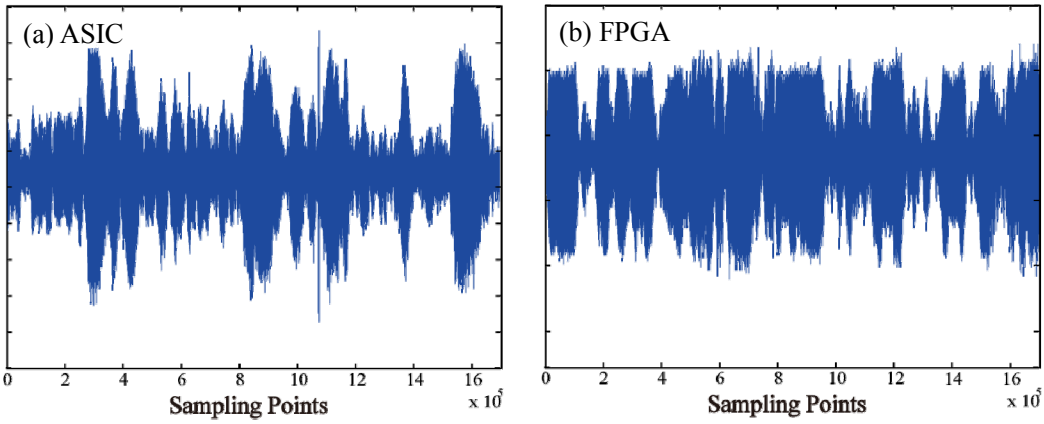


Figure 48 Differential power traces for a chosen-plaintext-pair (X , $-X$) (before filtering)

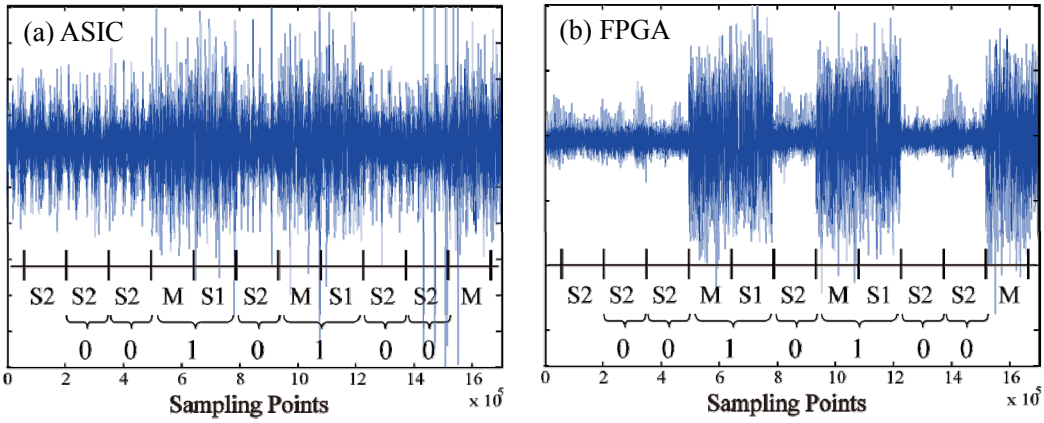


Figure 49 Differential power traces for a chosen-plaintext-pair (X , $-X$) (after filtering)

The methods proposed by Fouque *et al*. and Yen *et al*. take into account the emergence order of the operations so that these methods also work on the dummy operation countermeasure. However, they are applicable only to the left binary method. In contrast, Homma *et al*. have proposed a key estimation method that relates to square operations appearing in arbitrary cycles in two power traces[16]. This method is not only applicable to specific input pairs such as X, X^2 and X, $-X$, but also deals with wider input data settings, resulting in more flexible key estimation. Besides, the method is applicable to not only the left binary method, but also to the right binary method and to other algorithms such as the window method and the sliding window method.

Figure 50 illustrates the Homma's scheme using an example on the left binary method. The attacker provides the input with Y and Z such that $Y^{\alpha}=Z^{\beta}$ ($Y\neq Z$) and will be building up the key $E=\{e_{k-1}\ e_{k-2}...\ e_1e_0\}_2$ from the most significant bit in sequence. When the partial key $E^{(j)} = \{e_{k-1}e_{k-2}...e_{k-j}\}_2$ becomes known, he updates the input of Y and Z such that Y^{α} is the input of the operation for the unknown key bit $e_{k-(j+1)}$ (target operation) while Z^{β} is the input of a known square operation (reference operation). If $e_{k-(j+1)}=0$, the target operation is squaring, corresponding to the squaring that takes the input Z^{β}. On the other hand, if $e_{k-(j+1)}=1$, the target operation is multiplication, which is incongruent with the reference operation. By judging the similarity of the waveform patterns, the attacker can determine $e_{k-(j+1)}$. The value of α will be updated based on the value of $e_{k-(j+1)}$. By repeating the judging, the attacker determines the entire key sequence. It is easy to obtain an input combination such that $Y^{\alpha}=Z^{\beta}$ by computing $Y = r^{\beta} \bmod N$ and $Z = r^{\alpha} \bmod N$ (r is an arbitrary integer). The exponents α and β are given by $\alpha= 2E^{(j)}$, $\beta = E^{(t)}$ ($1 \le t \le j$) when $E_{(j)}$ is known.

Figure 50 shows an example of SPA using a plaintext pair such that $Y^{\alpha}=Z^{\beta}$. In this picture, the partial key $E^{(4)}=\{1100\}_2$ has been known, and the next e_{k-5} is about to be estimated. Given $\alpha=2E^{(4)}=24$ and $\beta= E^{(2)}=3$, we obtain $Y^{24}=Z^3$ so that the two operations that each input Y^{24} and Z^3 will be compared. The experimental results obtained from the SASEBO-G are shown in Figures 51 and

52. Figure 51 represents the case when the waveform of the estimation target and the reference waveform are the same, while Figure 52 shows the case when they are distinct.

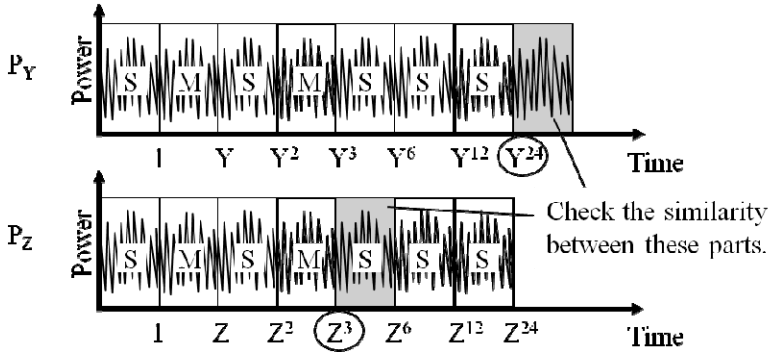


Figure 50 SPA with a plaintext pair such that $Y^{\alpha}=Z^{\beta}$

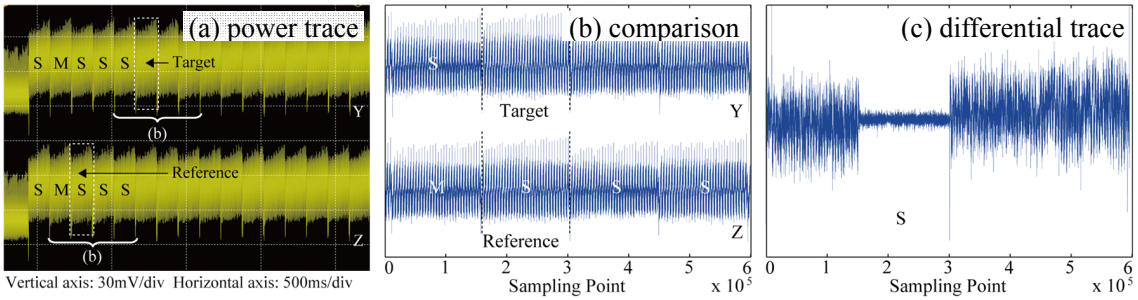


Figure 51 Power traces for inputs Y, Z such that $Y^{24} = Z^3$ (target waveform = reference waveform)

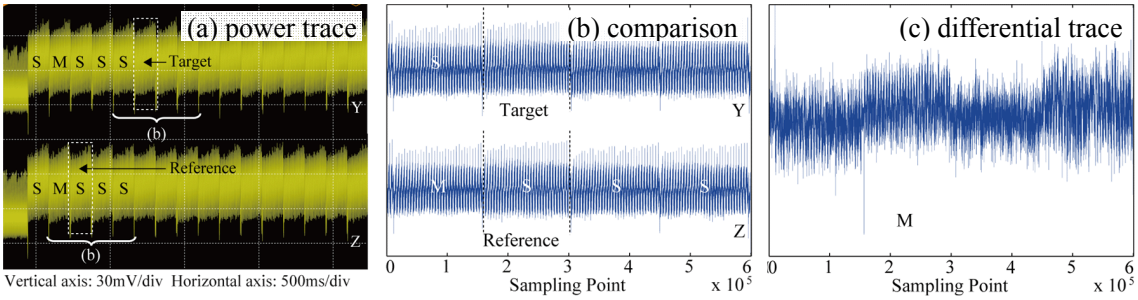


Figure 52 Power traces for inputs Y, Z such that $Y^{24} = Z^3$ (target waveform ≠ reference waveform)

3.5 SPA against Other Implementations

Although the SPA results shown above were obtained on a hardware implementation of RSA, the method is also applicable to a software implementation of the same kind. Figure 53 and Figure 54 show the power traces of SPA with a plaintext pair such that $Y^{\alpha}=Z^{\beta}$ performed on the RSA scheme implemented in C language on the PowerPC processor embedded in the SASEBO-G's FPGA. The program was made to have the same instruction sequence and the same memory access pattern regardless of whether the operation is multiplication or squaring so that SPA is difficult to perform on it. As a result, this implementation yields less differences in operation time and power consumption, compared to common software implementations. Nevertheless, the differential traces in the diagrams indicate that a proper key estimation was made. We also verified that the other above-mentioned SPA methods work on the same software implementation. In a software implementation, in general, a difference among operation segments may be observed because conditional branches make differences in their execution times. Instruction sequences or memory access patterns may be distinct depending on each operation. As a result, in a software implementation, it is easier to observe a difference in power traces than for a hardware

implementation.

Since chosen-plaintext SPA methods focus on input and output data of multiply and square operations, knowledge of the internal circuit structure is not needed. In these experiments, although we attacked on a multiplier based simple implementation of RSA, they are also applicable to other implementations such as adder based, Montgomery multiplication algorithm based, and Chinese Remainder Theorem (CRT) based implementations. For CRT implementations in particular, some specific chosen-plaintext SPA methods have been proposed in addition to the above-mentioned SPA[17), 18), 19), 20)].

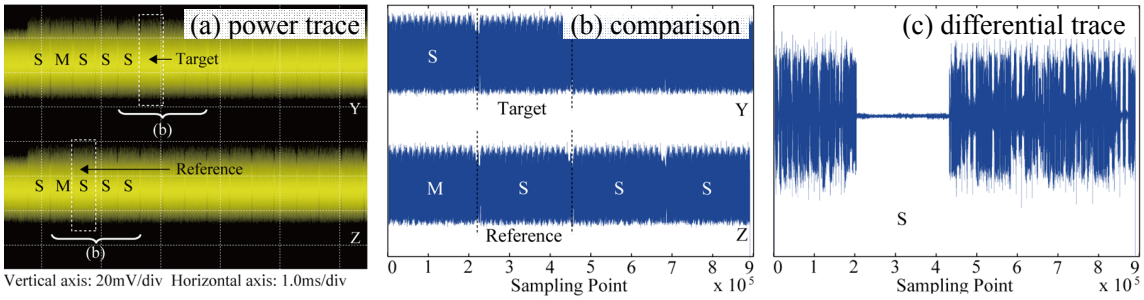


Figure 53 Power traces on a software implementation (target waveform = reference waveform)

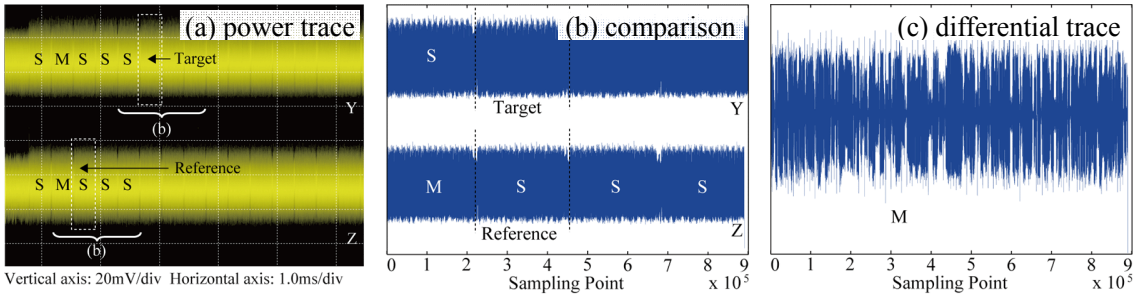


Figure 54 Power traces on a software implementation (target waveform ≠ reference waveform)

3.6 SPA Countermeasures and Their Evaluation

SPA Countermeasures are roughly classified as circuit level and algorithm level. Circuit level countermeasures eliminate the dependencies between power consumption and the secret key by using special logic for circuit implementation. Known methods include the Wave Dynamic Differential Logic (WDDL), which we apply to the AES cores implemented with DPA countermeasures, the Sense Amplifier Based Logic (SABL), and Simple Dynamic Based Logic (SDBL)[21)]. On the other hand, algorithm level countermeasures dissolve the dependencies by manipulating or changing operation sequences or data. Countermeasures that resolve the instruction sequence distinction include the above-mentioned square-and-multiply-always method[22)], which inserts dummy operations, and its extension Montgomery Powering Ladder[23)]. For countermeasures against attacks that exploit data characteristics, methods of masking on messages or keys have been proposed[24)].

For general SPA methods that use random inputs, countermeasures on operation sequences (Square-and-multiply-always method or Montgomery Powering Ladder) are effective. However, other SPA methods that use the input of N-1 or a chosen-plaintext pair may defeat such countermeasures. This is because the input values of the square operations in exponentiation directly reflect the secret information (secret key) values. To prevent such chosen-plaintext SPA methods, it is necessary to detach the positions and operations of squaring from the secret information in the sequence level. Taking into account the relationship between data and the secret information would also be effective against chosen-plaintext SPA attacks. In particular, plaintext masking would make plaintext choice impossible. Furthermore, combining these countermeasures with exponentiation masking improves their effects. However, care should be taken because the effectiveness of these

countermeasures depends on the size, generating method, and updating method of the mask (random numbers). Some countermeasure implementation methods are still vulnerable to a chosen-plaintext SPA[15].

To effectively evaluate SPA resistance properties, not only are choice of attack method and input data important, but also the observed power traces must sufficiently reflect the characteristics of each operation. Conversely, even without an explicit countermeasure, a cryptographic module can be considered attack resistant when the power traces do not provide adequate observed information or characteristics of each operation. Thus, as an index of sufficiency of observation data, we suggest to evaluate the *quality* of power traces by computing $S \times C \times 1/F$, where S is the sampling frequency of the measurement instrument (a digital oscilloscope) (Samples/s), C is the number of clock cycles for a single operation (multiplication or squaring) to complete (cycles), and F is the operating frequency of the module (Hz). This value is a sampling number per a single operation, which indicates that the information or characteristics used for attack resistance evaluation increases as its value gets larger. The larger index value also introduces the larger attack costs such as the required measurement time and computing time. The quality of the power traces collected during the experiments discussed here is 1.5×10^5 point for hardware implementations (ASIC and FPGA) and is 2.2×10^5 point for software implementations (PowerPC).

REFERENCES

1) P. Kocher, J. Jaffe, and B. Jun, “Differential Power Analysis,” Advances in Cryptology (*CRYPTO 1999*), LNCS 1666, pp. 388-397, Springer-Verlag, Aug. 1999.
2) D. Suzuki, M. Saeki, and T. Ichikawa “DPA Leakage Models for CMOS Logic Circuits,” Cryptographic Hardware and Embedded Systems (*CHES 2005*), LNCS 3659, pp. 366-382, Springer-Verlag, Sep. 2005.
3) S. Mangard, T. Popp, and B. M. Gammel “Side-Channel Leakage of Masked CMOS Gates,” Topics in Cryptology (*CT-RSA 2005*), LNCS 3376, pp. 361-365, Springer-Verlag, Feb. 2005
4) D. Suzuki and M. Saeki: Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style. Cryptographic Hardware and Embedded Systems (*CHES 2006*), LNCS 4249, pp. 255-269, Springer-Verlag, Oct. 2006.
5) S. Mangard, N. Pramstaller, and E. Oswald, “Successfully Attacking Masked AES Hardware Implementations,” Cryptographic Hardware and Embedded Systems (*CHES 2005*), LNCS 3376, pp. 157-171, Springer-Verlag, Sep. 2005.
6) D. Suzuki, M. Saeki, and K. Shimizu, “Side-Channel Resistance Evaluation for Block Cipher Circuit Architecture (1)(2),” Symposium on Cryptography and Information Security (*SCIS 2009*), Jan. 2009 (Japanese).
7) T. S. Messerges, E. A. Dabbish, and R. H. Sloan, “Investigations of Power Analysis Attacks on Smartcards,” USENIX1999, Jun. 1999. http://www.usenix.org/
8) R. Bevan and E. Knudsen, “Ways to Enhance DPA,” International Conference on Information Security and Cryptology (*ICISC 2002*), LNCS 2587, pp.327-342, Springer-Verlag, Dec. 2003.
9) E. Brier, C. Clavier, and F. Olivier, “Correlation Power Analysis with a Leakage Model,” Cryptographic Hardware and Embedded Systems (*CHES 2004*), LNCS 3156, pp. 16-29, Springer-Verlag, Aug. 2004.
10) T. Le, J. Clediere, C. Canovas, B. Robisson, C. Serviere, and J. Lacoume, “A Proposition for Correlation Power Analysis Enhancement,” Cryptographic Hardware and Embedded Systems (*CHES2006*), LNCS 4249, pp. 174-186, Oct. 2006.
11) T. S. Messerges, “Using Second-Order Power Analysis to Attack DPA Resistant Software,” Cryptographic Hardware and Embedded Systems (*CHES 2000*), LNCS 1965, pp.238-251, Aug. 2000.
12) J. Waddle and D. Wagner, “Towards Efficient Second-Order Power Analysis,” Cryptographic Hardware and Embedded Systems (*CHES 2004*), LNCS 3156, pp. 1-15, Springer-Verlag,

Aug. 2004.

13) S. M. Yen, W. C. Lien, S. J.Moon, and J. C. Ha, “Power analysis by exploiting chosen message and internal collisions - vulnerability of checking mechanism for RSA decryption,” Progress in Cryptology (*Mycrypt 2005*), LNCS 3715, pp. 183-195, Springer-Verlag, Sep. 2005.
14) A. Miyamoto, N. Homma, T. Aoki, and A. Satoh, “Chosen-message SPA attacks against FPGA-based RSA hardware implementations,” Proc. International Conference on Field Programmable Logic and Applications (*FPL 2008*), pp. 35-40, Sep. 2008.
15) A. P. Fouque and F. Valette, “The doubling attack - why upwards is better than downwards,” Cryptographic Hardware and Embedded Systems (*CHES 2003*), LNCS 2779, Springer-Verlag, pp. 269-280, Set. 2003.
16) N. Homma, A. Miyamoto, T. Aoki, A. Satoh, and A. Shamir, “Collision-based power analysis of modular exponentiation using chosen-message pairs,” Cryptographic Hardware and Embedded Systems (*CHES 2008*), LNCS 5154, pp. 15-29, Aug. 2008.
17) W. Schindler, “A timing attack against RSA with the Chinese remainder theorem,” Cryptographic Hardware and Embedded Systems (*CHES 2000*), LNCS 1965, pp. 109-124, Springer-Verlag, Aug. 2000.
18) C. D. Walter and S. Thompson, “Distinguishing exponent digits by observing modular subtractions,” Topics in Cryptology (*CT-RSA 2001*), LNCS 2020, pp. 192-207, Springer-Verlag, Apr. 2001.
19) R. Novak, “SPA-based adaptive chosen-ciphertext attack on RSA implementation,” Public Key Conference (*PKC 2002*), LNCS 2274, pp. 252-262, Springer-Verlag, Feb. 2002.
20) B. D. Boer, K. Lemke, and G. Wicke, “A DPA attack against the modular reduction within a CRT implementation of RSA,” Cryptographic Hardware and Embedded Systems (*CHES 2002*), LNCS 2523, pp. 228-243, Springer-Verlag, Aug. 2002.
21) T. Popp, S. Mangard and E. Oswald, “Power Analysis Attacks: Revealing the Secrets of Smart Cards,” Springer-Verlag, 2007.
22) J. S. Coron, “Resistance against differential power analysis for elliptic curve cryptosystems,” Cryptographic Hardware and Embedded Systems (*CHES 1999*), LNCS 1717, pp. 192–302, Springer-Verlag, 1999.
23) M. Joye and S. M. Yen, “The montgomery powering ladder,” Cryptographic Hardware and Embedded Systems (*CHES 2002*), LNCS 2523, pp. 291-302, Springer-Verlag, Aug. 2002.
24) P. Kocher, “Timing attacks on implementations of diffiehellman, RSA, DSS, and other systems,” Advances in Cryptology (*CRYPTO 1996*), LNCS 1109, Springer-Verlag, pp. 104-113, Aug. 1996.