International Journal of Cyber-Security and Digital Forensics (IJCSDF) 1(2): 99-114
The Society of Digital Information and Wireless Communications, 2012 (ISSN: 2305-0012)

# Power Amount Analysis: An efficient Means to Reveal the Secrets in Cryptosystems

Qizhi Tian and Sorin A. Huss
Integrated Circuits and Systems Lab (ICS)
TU Darmstadt, Germany
Email: tian, huss@iss.tu-darmstadt.de

## ABSTRACT

In this paper we propose a novel approach to reveal the information leakage of cryptosystems by means of a side-channel analysis of their power consumption. We therefore introduce first a novel power trace model based on communication theory to better understand and to efficiently exploit power traces in side-channel attacks. Then, we discuss a dedicated attack method denoted as Power Amount Analysis, which takes more time points into consideration compared to many other attack methods. We use the well-known Correlation Power Analysis method as the reference in order to demonstrate the figures of merit of the advocated analysis method. Then we perform a comparison of these analysis methods at identical attack conditions in terms of run time, traces usage, misalignment tolerance, and internal clock frequency effects. The resulting advantages of the novel analysis method are demonstrated by mounting both mentioned attack methods for an FPGA-based AES-128 encryption module.

## KEYWORDS

AES-128 Block Cipher; Power Model; Trace Model; Correlation Power Analysis; Power Amount Analysis.

## 1 Introduction

In 1999, Kocher et al. introduced the Differential Power Analysis (DPA) [1], as a novel analysis method for revealing the secret key of a cryptosystem. DPA then became the premier approach for exploiting the temporal power consumption for practical side-channel attacks of a cryptosystem. In the past decade, many researchers addressed the side-channel properties of cryptosystems and contributed their efforts to this area resulting in both new and powerful side-channel analysis methods next to DPA and in related countermeasures. Thus, the research in side-channel properties of cryptosystem implementations may be classified in two opposite domains: The one is aimed to the development of efficient analysis methods to eventually attack the system, whereas the other is dedicated to the invention or creation of countermeasures to harden the system and thus to reduce or even to avoid the success of such attacks. In other words, a still open competition between attack and defense of cryptosystems has been established meanwhile.

With regard to the attack methods, Chari et al. published in 2002 a paper on the so-called template attack [2]. In 2004, Brier et al. proposed the Correlation Power Analysis (CPA) method [3]. Later, in 2005, the stochastic analysis approach was introduced by Schindler et al. [4]. In 2012, Tian et al. [17] proposed an attack method called Power Amount Analysis (PAA) aimed to attack the cryptosystem by exploiting a large set of time points, which may contribute to information

leakage. Compared to the CPA attack, the PAA attack as outlined in [17] shows clear advantages in terms of run time, traces usage, misalignment tolerance, and internal Clock Frequency Effects (CFE). In the area of the defense of cryptosystems, on the other hand, a large number of countermeasures aimed at reducing the exploitable information leakage, i.e., the hardening of a cryptosystem, have been suggested as detailed in, e.g., [7], [8].

The fundamental idea of the attacking methods mentioned above is that adversaries mimic the variation of the power consumption behavior of the cryptosystem at hand in time domain by constructing a key dependent power model and by exploiting some mathematical functions. Then, various statistical methods are applied to analyze the relation between power model and measured power traces such as correlation coefficients, least squares, or maximum likelihood, aimed to help to eventually unveil the secret of the cryptosystem.

As discussed in [17], usually, the key dependent power model is based on some states produced by the cryptographic operations and then stored in registers of the cryptosystem hardware. Although these states seem to change instantaneously, it takes time in reality to calculate and to store them. For instance, if this process requires $0.1\text{ms}$ and is being monitored by means of an oscilloscope operated at a sample rate of 1MHz, i.e., the sample interval is $10^{-6}$s, the resulting 100 discrete points, which carry part of the information leakage, will be used to depict the results of this process in the monitored power curve in time domain. In other words, all these points should be used to reveal the secret key of the cryptographic system for the sake of efficiency.

In the CPA attack, the secret of the cryptographic system is indicated by the highest correlation peak, i.e., the maximum similarity between the power traces and the key dependent power model. Compared to the other information carrying time points, the highest correlation coefficient value comes from one certain fixed time point out of the captured traces. This means that the other time points of the recorded power traces do not explicitly contribute to the information leakage of this analysis method, they are only used for reference purposes. In other words, in the CPA attack just one time point is being exploited, while the other time points, which clearly do contain parts of the total information leakage, are discarded.

Compared to the CPA attack, in both the template attack and the stochastic approach several time points are in fact being used to identify the information leakage in order to reveal the secret key [4]. But their calculation complexity is considerably larger than in, e.g., CPA: The more time points are being used, the more computational time and memory space are needed. Note that in the profiling phase of the template attack and stochastic approach a certain amount of traces has to be captured using an identical training device [5], which takes even more execution time.

However, the PAA exploits a set of time points contributing to the information leakage in the attack without significantly increasing the computation effort. This property stems from a new power trace model. Such a method is able to exploit hundreds or even thousands of time points for revealing the secret key. In [17] the authors show that the related computational time is considerably less than needed for the CPA attack.

Therefore, in this paper we discuss more in-depth the PAA attack's properties and the resulting advantages when mounting practical attacks. The paper is structured as follows. In Section 2 we first detail how CPA works and how its trace model looks like. In Section 3, we define a new trace model based on communication theory and introduce the information leakage extraction from this model as well as a related attack procedure and highlight the resulting advantages. Section 4 presents a comparison of measured results by executing both CPA and the new analysis method PAA under identical conditions on raw, artificially misaligned, and clock frequency distorted traces produced from an FPGA-based cryptosystem running AES-128 encryption. Finally, we conclude with a summary of the advantages and benefits of this new analysis method.

## 2 Correlation Power Analysis

In this section some basic definitions will be given first, which are used in this and in the upcoming sections.

### 2.1 Basic Definitions

Input: A set of plaintext **d** with size D, where $d_i$ represents the $i^{th}$ plaintext and $i \in [1, D]$.

Output: A set of ciphertext **c** with size D, where $c_i$ represents the $i^{th}$ ciphertext, which corresponds to the plaintext $d_i$.

Subkey: All the possible subkey values form a set **k** with size K. For instance, a subkey byte has $2^8$ possible key values, i.e., $K = 256$, where $k_i$ denotes the $i^{th}$ subkey value.

Power Trace Matrix **T**: It is constructed from D power traces, captured by a sampling oscilloscope, while the cryptosystem is processing all inputs **d**. Each

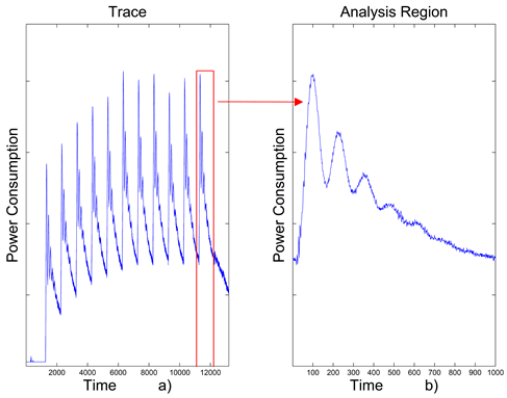


**Figure 1: Visualization of Analysis Region**

trace has M sample points. $T_{i,1:M}$ holds the $i^{th}$ measured power trace related to input $d_i$.

Analysis Region: Because there are a large number of time points in the captured traces, we do not need to analyze each and every of these points: A small portion of time points containing the information leakages is our analysis target. Therefore, we introduce an area of interest called *analysis region* in the captured traces, which contains the information leakage both of the selected power model and the part of the consumption the adversary focuses on. For instance, for an AES-128 power trace, if the power model is constructed on the basis of the last round operation, then the analysis region should be the area, where the last round peak exists, as depicted in Figure 1.

Following abbreviations are applied where appropriate: Expectation (E), Variance (Var), Standard Deviation (Dev), and Correlation Coefficient (CorrCoef).

### 2.2 Model of Power Traces

The power traces are captured and recorded by a sampling oscilloscope while either encryption or decryption is running. As a matter of fact, the existence of noise in the recorded power

$$\begin{bmatrix} R_{1,1} & \dots & R_{1,M} \\ \vdots & \ddots & \vdots \\ R_{k,1} & \dots & R_{K,M} \end{bmatrix} = StatAnlysis \begin{bmatrix} T_{1,1} & \dots & T_{1,M} \\ \vdots & \ddots & \vdots \\ T_{D,1} & \dots & T_{D,M} \end{bmatrix}, \begin{bmatrix} H_{1,1} & \dots & H_{1,K} \\ \vdots & \ddots & \vdots \\ H_{D,1} & \dots & H_{D,K} \end{bmatrix} \quad (1)$$

traces is inevitable in practice. The total consumption of the cryptosystem may then be determined as follows according to [7, p. 62]:

$$P_{total} = P_{op} + P_{data} + P_{el.noise} + P_{const} \quad (2)$$

At each point in time of the recorded trace the total power may thus be modeled by (2), where $P_{op}$ is the operation dependent power consumption, $P_{data}$ defines the data dependent power consumption, $P_{el.noise}$ denotes power resulting from the electronic noise in the hardware, which features a normal distribution, i.e., $P_{el.noise} \in \mathcal{N}(0, \sigma^2)$ holds, and $P_{const}$ represents, depending on the technical implementation, some constant power consumption. All these parameters are additive, independent, and functions of time. But the power model as exploited in CPA is restricted to analyze just a single point in time rather than the complete power function in time domain. CPA aims at a traversal of all the captured traces at a certain point in time to find the biggest information leakage point, i.e., the same time point, but in different traces. Therefore, the precondition of CPA to mount an attack successfully is that the power consumption values at each time point are yielded by the same operation in the cryptographic algorithm. In other words, the power traces must be correctly aligned in time as pointed out in, e.g., [7, p.120].

### 2.3 Power Model

Power models are in general based on both the algorithm running in the hardware and its architecture. Considering, e.g., the last round of the AES-128 algorithm, the Hamming Distance (HD) model of the output register before and after the S-Box, respectively, as discussed in, e.g., [7, p. 132], is given by (3):

$$HD = HammingWeight(c_i \oplus \tilde{d}_i) \quad (3)$$

where $\tilde{d}_i$ denotes a certain byte, e.g., the second byte of the register stored in the last round before the S-Box, which is the counterpart of $c_i$. In contrast, the Hamming Weight (HW) model of the output register is given by:

$$HW = HammingWeight(c_i \oplus k_i) \quad (4)$$

Another possible classification of the power model is proposed in the following.

Instantaneous Model: A power model based on the state at some time point of a certain register, e.g., HW power model.

Process Model: A power model based on the two states changing within a time interval, e.g., HD power model.

### 2.4 CPA Attack Phase

The attack procedure may be summarized as follows:

**Step1**: Plaintext **d** or ciphertext **c** and the subkey **k** are mapped by the power model, for example exploiting (3) or (4), to form a matrix, which is named *hypothesis matrix* **H** of size D × K.

**Step2**: Analysis of power trace matrix **T** and hypothesis matrix **H** is performed by calculating the correlation coefficient during StatAnalysis as shown in (1), which yields the result matrix **R** with

size K × M. The elements of **R** are calculated from:

$$R_{i,j} = CorrCoef(T_{1:D,j}, H_{1:D,i}) \qquad (5)$$

where $i \in [1, \mathrm{K}]$ and $j \in [1, \mathrm{M}]$ hold. Then, the unique time point featuring the maximum value of **R** is determined next, which indicates the correct key value.

## 3 Power Amount Analysis

In this section, we introduce a trace model to address the power consumption in a quite different way, which relies on principles adopted from communication theory. Then, based on this model, a new attacking method, i.e., PAA is proposed, which is characterized by an exploitation of a larger set of time points compared to CPA. In PAA we exploit in general more than one hundred points to efficiently extract the information leakages and to attack the cryptosystem successfully as detailed in the sequel.

### 3.1 Hardware Model

Communication theory has been developed for more than one hundred years. Many models were proposed and are currently used to evaluate and simulate the communication channel. Among these models, there exists a simple and easy one, which is named Additive White Gaussian Noise (AWGN) channel, as detailed in, e.g., [10, p. 167], [11]. A discrete time AWGN channel is given as follows:

$$O[i] = S[i] + N[i] \qquad (6)$$

where $S[\mathrm{i}]$ is the input signal of the channel at the discrete time point i, $O[\mathrm{i}]$ denotes the output of the channel, and $N[\mathrm{i}]$ represents the additive white Gaussian noise while the input signal passes through the channel. As generally assumed in the communications field, for the noise $N \in \mathcal{N}(0, \sigma^2)$ holds, see [10, pp. 29-30].

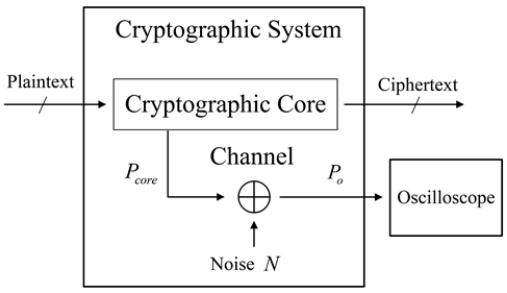


**Figure 2: Abstract Signal and Noise Model**

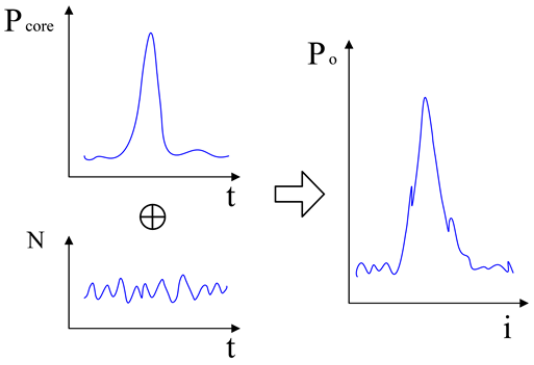


**Figure 3: Visualization of the Power Traces**

Consequently, we model the power trace of a cryptosystem based on the communication model in (6). As shown in Figure 2, the power consumption from the core chip is taken as the input to the channel and noise is being added while it propagates. The time discrete trace of the power consumption function, captured by the oscilloscope, now consists of two parts, as visualized in Figure 3: The first one is the power consumption function of the cryptographic chip while encryption or decryption runs and contains the information leakage of the cryptosystem; the second part contains the noise produced by the hardware, which can be modeled as in the AWGN

channel. Assume that the power consumption of the core chip is pure, i.e., without noise, and its temporal value is being transferred via the electric circuit network to the oscilloscope. Meanwhile, the AWGN adds the noise to it. Consequently, for each measurement time point, the power traces are modeled as follows:

$$P_o[i] = P_{core}[i] + N[i] \quad (7)$$

where $P_o[i]$ represents the output power consumption, which is captured by a sampling oscilloscope at time index $i$, $P_{core}[i]$ is the power consumption generated by the cryptographic core chip while running encryption or decryption, and $N[i]$ is taken from the AWGN, i.e., for any measurement in time domain, the noise features $N \in \mathcal{N}(0, \sigma^2)$. Please note as an important property that $P_{core}$ and $N$ are independent and uncorrelated in time domain.

Let us assume that an attack using a process model, e.g., the HD model, takes place from time points $m_1$ to $m_2$. Now, we intend to calculate the power variation of the cryptographic chip from $m_1$ to $m_2$, which contains the information leakage the adversary is looking for, i.e., the power variation of the analyzed register's state changing, which matches the HD model very well. Consequently, by calculating $Var(P_{core})$ in the time interval $[m_1, m_2]$ for each trace and then comparing the similarity between the variation of the power consumption of the core chip and the key dependent hypothesis matrix, one can eventually retrieve the secret key of the cryptographic system.

However, we cannot measure the $P_{core}$ values directly. Each time point of the measurement contains a mixture of power consumption $P_{core}$ and noise N as defined by (7), so that one cannot separate them easily. Therefore, a straight forward calculation of $Var(P_{core})$ is difficult, but we will show in the sequel how to derive it indirectly.

### 3.2 Power Consumption of the Hardware Module

In reality, when a hardware device is working, its power consumption is a continuous function in time domain. But when a sampling oscilloscope is being used to monitor this power function, only discrete points will be captured. These discrete points constitute the curve $P_o$, where $P_o[i]$ is the instantaneous power at time index $i$.

The average power consumption $\bar{P}_o$ in the time index interval $[m_1, m_2]$ can be approximately calculated by

$$\bar{P}_o = \frac{1}{m_2 - m_1 + 1}[P_o[m_1] + \cdots + P_o[m_2]] \quad (8)$$

Equation (8) denotes that the average power consumption is just the mean of the power values within $[m_1, m_2]$. If one increases the sample rate, i.e., more sample points in the interval $[m_1, m_2]$, then $\bar{P}_o$ becomes an increasingly precise estimator of $E(P_o)$. Because $E(N) = 0$ holds, (8) can be rewritten as follows:

$$E(P_o) = E(P_{core}) + E(N) = E(P_{core}) \quad (9)$$

Here $E(P_o) = E(P_{core})$, i.e., the mean value for the captured power traces denotes the average power consumption of the core chip in the time index interval $[m_1, m_2]$. We can assume that such an average value contains the constant average power from the hardware circuits its self and the average power variation

from the state changing in the time index interval $[m_1, m_2]$. We prefer to identify the power variation for the states changing in the register to the constant average power from hardware itself. However, the constant power of hardware circuits is difficult to determine. Therefore, filtering out the variation information form $E(P_o)$ seems to be impossible.

Now let us look at the $Var(P_{core})$ calculation as follows:

$$Var(P_{core}) = \mathrm{E}[P_{core} - \mathrm{E}(P_{core})]^2 \quad (10)$$

which denotes the average power variation around the average power $E(P_{core})$ for the register states changing in the time interval $[m_1, m_2]$ the adversary looks for. It contains two steps: in the first step, the information is compressed by calculation of the mean power consumption $E(P_{core})$. However, such a compression is not sufficient to being used for key revealing. Therefore each sample is compared to $E(P_{core})$ resulting in some differences. After that, the average differences power value is calculated to form $\mathrm{Var}(P_{core})$, which is the information carrier the adversary is looking for. This is called the information extraction step. Indeed, $\mathrm{Var}(P_{core})$ is very important for the key revealing in the cryptosystem, but it cannot be measured or calculated directly.

Nevertheless, $Var(N) = \sigma^2$ holds and $P_{core}$ and N are independent as well as uncorrelated in time domain. From (9) we can easily get $Var(P_o)$ as:

$$\begin{aligned} Var(P_o) &= Var(P_{core}) + Var(N) \\ &= E(P_{core}) + \sigma^2 \end{aligned} \quad (11)$$

Equation (11) consists of two parts: The first part is $Var(P_{core})$, the second part is $\sigma^2$, i.e., the noise in a trace matrix has the same variance $\sigma^2$ for each single trace, which is the fundamental property of the new trace model as mentioned before. The value $\sigma^2$ is a constant, thus items $\mathrm{Var}(P_o)$ and $\mathrm{Var}(P_{core})$ are in a linear relation. Now, instead of the calculation of $Var(P_{core})$, one just compares the similarity between $Var(P_o)$ and **H** yielding the same results.

$$Dev(P_o) = \sqrt{Var(P_{core}) + \sigma^2} \quad (12)$$

$$\sqrt{x} = 1 + \frac{1}{2}(x-1) + \frac{1}{8}(x-1)^2 + \cdots \quad (13)$$

$Dev(P_o)$ is a non-linear function because of the relation by a square root as given in (12). Therefore, we cannot use it as the substitution of $Var(P_{core})$ to attack the system. Nevertheless, the square root can be expanded to a Taylor series as given in (13), in which the first two terms of the series result in a linear relation. Thus, $Var(P_{core})$ and $Dev(P_o)$ can approximately be taken as being in a linear relation. So, we can exploit this approximation to further analyze the power consumption of the system.

### 3.2 Attack Phase

An important property of PAA is that an usage of more time points in the analysis region involved results in less traces needed for a successful attack. There are two ways to achieve this goal:

1) Use more time points in the analysis region for the attack.
2) Increase the sample rate of the monitor devices, i.e., use an oscilloscope of a good quality, e.g., with a higher sampling rate.

Theoretically, the proposed PAA takes the time interval factors into consideration, thus the process model fits such an attack very well. On the contrary, the

$$Var(T) = \begin{bmatrix} Var(T_{1,1} & \dots & T_{1,M}) \\ & \vdots & \\ Var(T_{D,1} & \dots & T_{D,M}) \end{bmatrix} = \begin{bmatrix} V_{1,1} \\ \vdots \\ V_{D,1} \end{bmatrix} \quad (14)$$

$$[R_{1,1} \quad \dots \quad R_{1,K}] = StatAnlysis \begin{bmatrix} T_{1,1} & \dots & T_{1,M} & V_{1,1} \\ \vdots & \ddots & \vdots & , \ \vdots \\ T_{D,1} & \dots & T_{D,M} & V_{D,1} \end{bmatrix} \quad (15)$$

instantaneous model, e.g., HW model, focuses on some time point only. By using such a model the attack results of PAA will be not as good as we would expect.

### 3.2.1 Attacking Procedure

The attacking procedure of the PAA is given as follows:

**Step1**: Plaintext **d** or ciphertext **c** and the subkey **k** are mapped by the power model, e.g., by equation (3), thus generating the hypothesis matrix **H** of size $D \times K$.

**Step2**: Calculate the variance or standard deviation of each row of the trace matrix $\boldsymbol{T}$, i.e., $V_{i,1} = Var(T_{i,1:M})$, where $i \in [1, D]$ holds, as given in (14).

**Step3**: Calculate the results matrix **R** with size 1 × K by analyzing statistically the vector **V** derived from (14) and the hypothesis matrix **H** according to (15), whereas the correlation coefficient is used as the distinguisher:

$$R_{1,i} = CorrCoef(V_{1:D,1}, H_{1:D,i}) \quad (16)$$

where $i \in [1, \mathrm{K}]$ holds. Subsequently, the maximum correlation value will be determined to find the correct key value.

### 3.3 Advantages

Usually, in the view of an attacker, some factors must be taken into consideration in a practical attack. For example, the key should be revealed within limited time and traces usage. If the targeted algorithm is hardened by a countermeasure, e.g., the power traces captured from oscilloscope are not aligned because of a related countermeasure, such as random clock or dummy wait state insertion, then before mounting a CPA attack some preprocessing should be done. Compared to the CPA attack, the proposed PAA method can deal with the mentioned requirements easily, which will be discussed in the upcoming sections.

### 3.3.1 Execution Time

The required execution or run time is a very important metric, which indicates the efficiency of the algorithm in a real attack. In PAA a large number of time points is taken into the variance or deviation calculation. Therefore, the trace matrix **T** is mapped to **V**, see (14), which will be used to calculate the correlation coefficient using the hypothesis matrix **H** as shown in (15). On the contrary, in the CPA attack, the correlation coefficient values matrix is directly calculated from the trace matrix **T** and the hypothesis matrix **H**, see (1). Therefore, under the same calculation condition, i.e., the number of time points CPA needs to traverse, the variance in PAA attack is being calculated, i.e., the PAA attack is faster than CPA attack. One can also find that in CPA the result is a matrix **R** with size K × M. In contrast, PAA yields

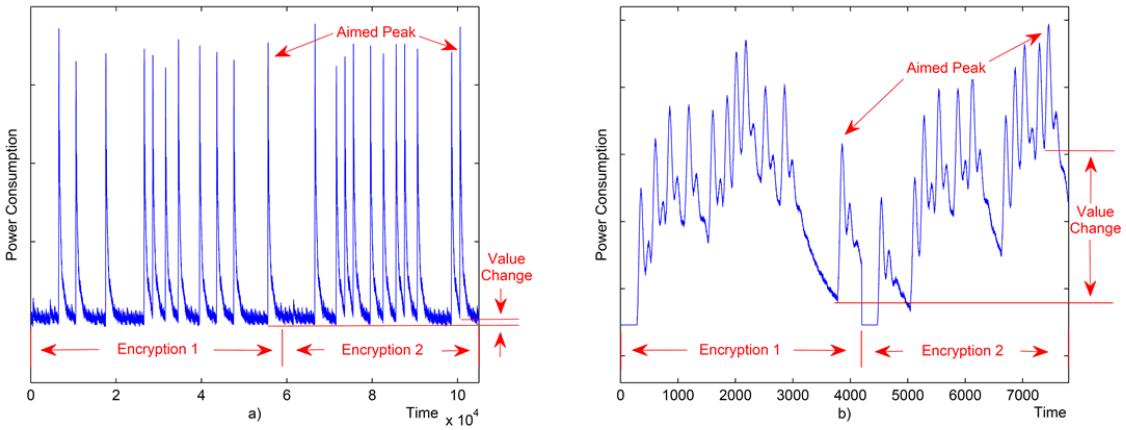


**Figure 4: Power Traces at different Base Clock Frequencies: a) 2 MHz, b) 24 MHz**

a vector **R**, with size 1 × K. Therefore, the calculation complexity is decreased, because it is easier to identify the correct key by means of a result vector than of a matrix. We will focus on the run time consumed in both attack methods by means of experimental results in the sequel.

### 3.3.1 Traces Usage

Traces usage is an important parameter in the evaluation of cryptosystem security. From an attacker's point of view, the less trace usage, the more efficient the attack method will be. In contrast, for a system designer, to some extent it denotes the degree of safety of the cryptographic algorithms. Therefore, traces usage reduction is crucial to come quickly to an assessment of the related SCA resistance level of a cryptosystem.

In general, the PAA attack calculates the variance and standard deviation for the captured power traces, where the information leakages at each single time point is compressed and extracted, i.e., more information leakages sources are considered. By means of such a method, one can achieve higher distinguisher values when the number of power traces is limited. In contrast, CPA just exploits one time point which contributes the maximum information leakage. Therefore, such an attack method requires a relatively large number of power traces to achieve the same attack results as in PAA. In the last section, we will demonstrate this important feature.

### 3.3.3 Misalignment Tolerance

$$\boldsymbol{T} = \begin{bmatrix} T_{1,3} & T_{1,4} & \boldsymbol{T_{1,5}} & T_{1,6} & T_{1,7} & T_{1,8} \\ T_{2,3} & T_{2,4} & \boldsymbol{T_{2,5}} & T_{2,6} & T_{2,7} & T_{2,8} \\ T_{3,3} & T_{3,4} & \boldsymbol{T_{3,5}} & T_{3,6} & T_{3,7} & T_{3,8} \end{bmatrix} \quad (17)$$

$$\tilde{\boldsymbol{T}} = \begin{bmatrix} T_{1,3} & T_{1,4} & \boldsymbol{T_{1,5}} & T_{1,6} & T_{1,7} & T_{1,8} \\ T_{2,4} & \boldsymbol{T_{2,5}} & T_{2,6} & T_{2,7} & T_{2,8} & T_{2,9} \\ T_{3,2} & T_{3,3} & T_{3,4} & \boldsymbol{T_{3,5}} & T_{3,6} & T_{3,7} \end{bmatrix} \quad (18)$$

As mentioned before, the model of the power traces in CPA attack concentrates on a common time point in different power traces. For example, (17) denotes a matrix constructed from aligned power traces. The third column contains the maximum information leakage point, i.e., the elements $T_{1,5}$, $T_{2,5}$, $T_{3,5}$, are the best combination for the information contribution. If there are some misalignments in the constituting such power traces as indicated in (18), the third column's

combination is broken. Then the new combination $T_{1,5}$, $T_{2,6}$, $T_{3,4}$ in the third column cannot provide the maximum leakage for the CPA attack. Therefore, the attack results become worse. In other words, the prerequisite for mounting a CPA attack successfully is that the power traces must be aligned. However, in the PAA attack a large number of time points are taken to the variance calculation. Therefore, for each misaligned power trace compared to the original power trace, only a few time points are missing. Thus, the difference between the second rows of $\mathbf{T}$ and $\widetilde{\mathbf{T}}$, respectively, is that $T_{2,3}$ is substituted by $T_{2,9}$. If there are enough time points in the interval, then such a substitution cannot greatly impact the variance values and hence the overall attack results. Consequently, PAA features a considerably stronger misalignment tolerance during a real attack. In other words, a small misalignment does not affect the final results to a large extent. Therefore, such a property of the analysis method can be exploited to improve attacks of some power traces featuring a misalignment injection as a hardening countermeasure.

### 3.3.4 Clock Frequency Effects

Misalignment may be a countermeasure to impede CPA attacks in practice, see [13]. In presence of misaligned power traces, a preprocessing is required for improving the CPA attack results in order to cope with traces manipulated by changes in the clock frequency of the cryptosystem as shown in Figure 4 a), where the aimed peaks are shifted in time domain. The authors of [18] proposed a method to align misaligned power traces by exploiting dynamic time warping. The authors of [15] presented a *horizontal alignment* method to align the power traces in time domain both partially and dynamically. Later, in [16], these authors reported on a phenomenon called *clock frequency effect*, which occurs in random clock featured cryptosystem, when the base clock runs at a higher clock frequency. Then the power peaks in the captured traces not only shift in time domain, but also change their power values in the amplitude domain. One finds easily that the power value change in Figure 4 b) is considerably larger than that in Figure 4 a), where the base clock frequencies are 24MHz and 2MHz, respectively. In order to cope with such effects, these authors proposed a *vertical matching* after the horizontal alignment, where each horizontally aligned power trace is moved up and down in the amplitude domain in order to find the minimal distance between the moved trace and an arbitrarily chosen template. Finally, these vertically matched power traces are attacked. The experimental results in [16] show that by exploiting vertical matching as a preprocessing step, the efficiency of the CPA attack can greatly be improved. This is because in the CPA attack the focus is on finding a certain time point in different power traces only. Let us take $\mathbf{T}_{2,3:9}$ as an example and shift its element values in the amplitude domain, i.e., to each element the same positive or negative value $o$ is added, an operation, which does not change its variance. In other words, regardless of a possible shifting in the amplitude domain, the attack results will always be the same as visible from (20).

$$\mathbf{T}_{2,3:9} + \boldsymbol{o} = \left[T_{2,3} + o, \ldots, T_{2,9} + o\right]^{\mathrm{T}} \quad (19)$$

$$Var(\mathbf{T}_{2,3:9} + \boldsymbol{o}) = Var(\mathbf{T}_{2,3:9}) \quad (20)$$

Therefore, for attacking misaligned power traces, only the horizontal alignment is required. The vertical matching step in PAA attack can be completely omitted in contrast to CPA. This property saves a lot of traces processing time without affecting the quality in the attack results.

## 4 Application Examples

In this section several attacks on an AES-128 cryptosystem featuring the TBL S-Box [12] are presented and discussed. The HD model from (3) is being taken as the power model. In order to evaluate and to compare the mentioned four main properties to be taken as a metric, the results are produced by mounting the attack exploiting both CPA and PAA, respectively. The experiments were ordered into three sections as follows:

1) Attack of the captured power traces directly with CPA and PAA, respectively.
2) The captured power traces will be misaligned artificially to some extent and then be attacked by mounting CPA and PAA, respectively, in order to assess the misalignment tolerance and thus the robustness of both attack methods.
3) In order to verify the internal clock frequency effects for the PAA attack, each captured power trace will be shifted in the amplitude domain by some random offset, i.e., a high clock frequency effect injection takes place. Finally, the attack results for both CPA and PAA are compared.

Here, the run time and success rate for each byte and global key will be exploited as the metric to evaluate both the CPA and PAA attack results. The run time is a relative value, which depends on the calculation computer's processor, memory, configurations, etc. The success rate is detailed in [14], which defines the possible rate such that all the key bytes are to be successfully recovered under the constraint of a limited amount of experiments. Therefore, we ran 30 times different attack experiments, each experiment with 1000 power traces. Then the success rate is calculated accordingly.

### 4.1 Platform

The side channel attack standard evaluation board version G (SASEBO) [6] is exploited as the target platform, which embodies two Xilinx Virtex-II pro series FPGAs: One for board control and one for cryptographic algorithms implementation. Both FPGAs are running at 2MHz clock frequency.

### 4.2 Run Time and Traces Usage

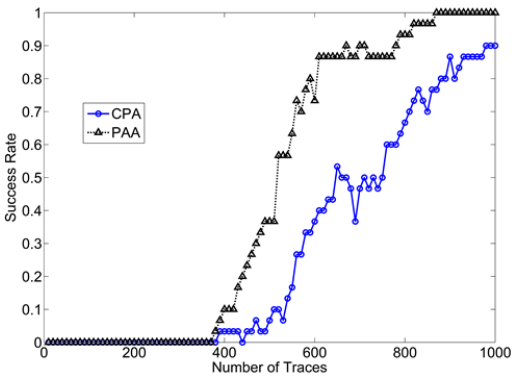


**Figure 5: Global Success Rate for CPA and PAA**

Table 1: Run Time Comparison

| | CPA | PAA | | Ratio |
|---|---|---|---|---|
| Run Time | 54.44s | Var | 33.29s | 60.0% |
| | | Std | 33.54s | 60.5% |

In order to compare the run time for both CPA and PAA attacks, we executed these two methods under the same con-

ditions. For the CPA attack, we traverse 600 time points in the analysis region to find the maximum information leakage. In PAA attack, these 600 time points will be taken to the variance or standard deviation calculation, respectively. After that, the total run time values for all 16 key bytes are compared.

Table I shows the run times for all 16 attackable bytes when mounting CPA and PAA attacks. When applying the methods Var and Dev, the PAA attack results to run times of 33.29s and 33.54s, respectively, which in both cases result in 60.0% and 60.5% of the run time consumed in the CPA attack. Therefore, under the same attack condition, the PAA is faster than CPA, which can thus shorten the breaking time of a cryptosystem in practical attacks.

With regard to the traces usage Figure 5 illustrates that, when we consider the power traces range of from 0 to 1000, all 16 bytes are revealed after an usage of 850 traces only by PAA, i.e., the global success rate is 1. However, under the same condition, CPA can reveal only 90% correct keys when consuming all available 1000 power traces, i.e., such an attack needs more power traces to reveal all correct key bytes. At the same time, the success rate curve for PAA attack raises faster after 400 power traces usage compared to its counterpart in CPA attack. This is because, PAA exploits more time points, which contribute to the information leakage, thus resulting in a lower traces usage in comparison to CPA. In the Appendix, we visualize in Figure 10 the success rate individually for each key byte. One easily finds out from this figure that for each byte to be revealed, PAA always consumes less traces than the CPA method.

### 4.3 Misalignment Tolerance

As mentioned above, we expect that the PAA attack shows a good robustness in presence of a reasonably misalignment of traces. Sometimes, the power trace misalignment is intended as a hardening countermeasure against power consumption attacks [13]. In order to demonstrate this additional robustness feature, the misaligned traces are first generated by applying Algorithm 1 and then attacked by means of both CPA and PAA, respectively.

In order to generate comparable results for CPA and PAA attacks, we set the range B of the random number in Algorithm 1 from 0 to 20, 50, and 100, respectively. The global success rate curves for CPA and PAA attacks before misalignment (MA) are depicted for comparison purposes as illustrated in Figure 6 to 8.

**Algorithm 1** Misaligned Traces Generation

**Require:** Aligned Traces $\mathbf{T}_{i,1:M}$

**Ensure:** Misaligned Traces $\widetilde{\mathbf{T}}_{i,1:W}$

1: Find a start point $a$ in $\mathbf{T}_{i,1:M}$

2: Generate an integer random number $r$, $r \in [0, B]$

3: Cut the traces from point $a + r$, with width $W$.

4: Save the cut trace into set $\widetilde{\mathbf{T}}_{i,1:W}$

**Return:** $\widetilde{\mathbf{T}}_{i,1:W}$

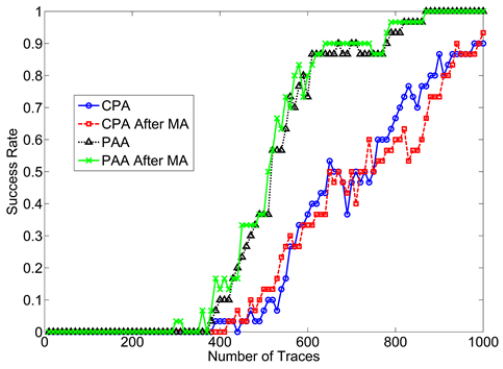


**Figure 6: Global Success Rate, B=20**

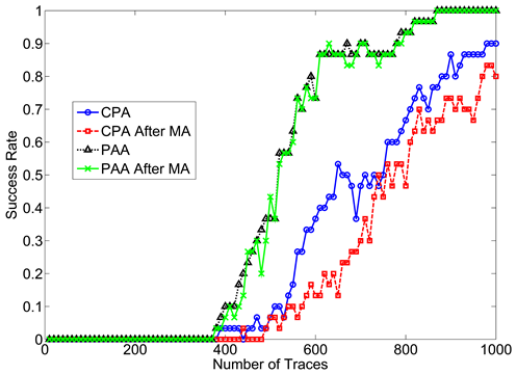


**Figure 7: Global Success Rate, B=50**

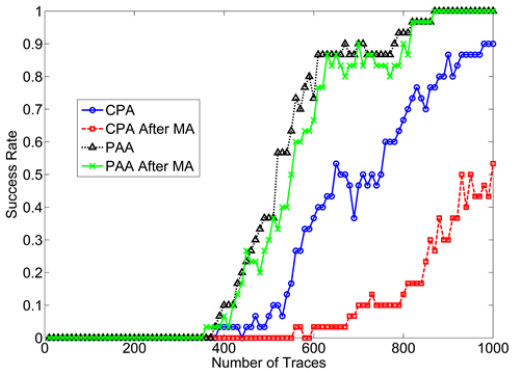


**Figure 8: Global Success Rate, B=100**

In Figure 6, the maximum shifting of the power traces is 20 time points. One finds that the global success rates for PAA attack before and after misalignment are overlapped, i.e., the misalignment cannot affect the PAA attack results greatly. For CPA, the global success rate curve deviates a bit from its counterpart before misalignment, i.e., a small misalignment affects the CPA attack not that much.

Then we increase the maximum shifting to 50 points, as shown in Figure 7. For PAA, the success rate curves still overlap. In contrast, in CPA, because of the stronger misalignment, the attack becomes harder, and then the deviation of the success rate curves before and after misalignment is enlarged. Thus, when increasing the maximum number of shifting time points, the deviation of PAA attack results is smaller than that in CPA, i.e., PAA is more robust.

In order to show this characteristic more clearly, the parameter B is now set to 100, i.e., the maximum shifting of the power trace is 100 time points. Now for PAA, the success rate curves show a small deviation. However, in CPA, the deviation between the corresponding curves unveils a big gap as shown in Figure 8. Therefore, we can state that the PAA features a considerably stronger misalignment tolerance compared to CPA.

### 4.4 Internal Clock Frequency Effects

In order to simulate the clock frequency effects environment condition, mentioned in [16], Algorithm 2 is used to inject such effects into power traces by moving each power trace in the amplitude domain by a random offset vector $\boldsymbol{r}$.

**Algorithm 2** Clock Frequency Effects Injection (CFEI)

**Require:** Aligned Traces $\mathbf{T}_{i,1:M}$
**Ensure:** CFE Injected Traces $\widehat{\mathbf{T}}_{i,1:M}$
1: Generate an integer random number $r$, $r \in [-F, F]$ holds
2: Generate an $M$ elements constant vector $\boldsymbol{r}$, where $\boldsymbol{r} = [r_1, \ldots r_M]$
3: Do $\widehat{\mathbf{T}}_{i,1:M} = \mathbf{T}_{i,1:M} + \boldsymbol{r}$
**Return:** $\widehat{\mathbf{T}}_{i,1:M}$

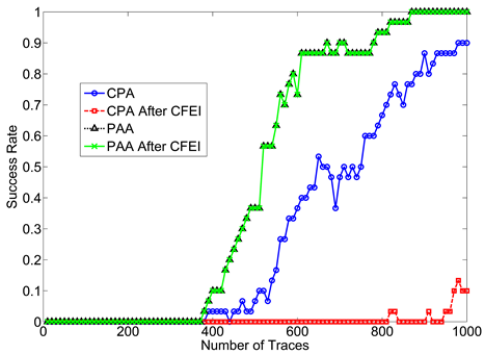


**Figure 9: Global Success Rate F=10**

The chosen parameter value is $F = 10$, thus each trace will be moved by an offset from the interval $[-10,10]$ unit in the amplitude domain. Then, CPA and PAA are mounted.

As already mentioned, a power trace shifts in amplitude domain results in unchanged variance and standard deviation values. Therefore, for PAA, the attack results are always the same, as illustrated in Figure 9. One finds that the success rate curves for PAA attack are completely overlapped, i.e., the numerical results are almost the same. In contrast, in CPA after the CFEI, the success rate is decreased from 90% to just 10% at 1000 traces usage. This means that the clock frequency effects decline the success rate in CPA attack considerably for random clock hardened cryptosystems. In contrast, the PAA method can counteract such shifts in the amplitude domain automatically and yields the same attack efficiency as for the original data set.

## 5 Summary

In this paper we discussed in detail the Power Amount Analysis method, which is based on a new trace model originating from communication theory. This novel SCA analysis exploits many time points within the power traces that contribute to the information leakage in the captured traces and thus helps significantly in revealing the secret key. Starting from the original PAA paper, we first performed a comparison to the well-known CPA attack and we then elaborated four advantages of the proposed methods in terms of run time, traces usage, misalignment tolerance, and internal clock frequency effects. These advantages were demonstrated by mounting both CPA and PAA attacks on the power traces captured from an FPGA-based AES-128 cryptosystem. We have shown that the advocated analysis method takes advantage in presence of both aligned and misaligned power traces. We see PAA as a new means, which provides a different way to view and to understand power traces. Its specific properties help to reveal the secret key in cryptosystems more easily and thus to qualify the security of cryptographic algorithm implementations.

**Acknowledgement**

This work was supported by CASED (www.cased.de).

## 6 REFERENCES

1. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun, Differential Power Analysis, International Cryptology Conference (CRYPTO), 1999, pp.388-397.
2. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi, Template Attacks, Cryptographic Hardware and Embedded Systems (CHES), 2002, pp. 13-28, Springer-Verlag.
3. Eric Brier, Christophe Clavier, and Francis Olivier, Correlation Power Analysis with a Leakage Model, Cryptographic Hardware and Embedded Systems (CHES), 2004, pp. 16-29, Springer-Verlag.
4. Werner Schindler, Kerstin Lemke, and Christof Paar, A Stochastic Model for Differential Side Channel Cryptanalysis, Cryptographic Hardware and Embedded Systems (CHES), 2005, pp. 30-46, Springer-Verlag.
5. Werner Schindler, Advanced Stochastic Methods in Side Channel Analysis on Block Ciphers in the Presence of Masking, J. of Mathematical Cryptology 2(3), 2008, pp. 291-310.
6. N. N., Research Center for Information Security National Institute, Side Channel Attack Standard Evaluation Board Version G Specification, 2008, http://www.morita-tech.co.jp/SASEBO/en/board/index.html.
7. Stefan Mangard, Elisabeth Oswald, and Thomas Popp, Power Analysis Attacks: Revealing the Secrets of Smart Cards

(Advances in Information Security), 2007, Springer-Verlag, New York, USA.
8. Kai Schramm and Christof Paar, Higher Order Masking of the AES, Cryptographers' Track of RSA Conference (CT-RSA), 2006, pp. 208-225, Springer-Verlag.
9. Danil Sokolov, Julian P. Murphy, Alexandre V. Bystrov, and Alexandre Yakovlev, Design and Analysis of Dual-Rail Circuits for Security Applications, IEEE Trans. On Computers, 2005, vol. 54, pp. 449-460.
10. David Tse and Pramod Viswanath, Fundamentals of Wireless Communication, 2005, Cambridge University Press.
11. Andrea Goldsmith, Wireless Communications, 2005, Cambridge University Press.
12. Atri Rudra, Pradeep K. Dubey, Charanjit S. Jutla, Vijay Kumar, Josyula R. Rao, and Pankaj Rohatgi, Efficient Rijndael Encryption Implementation with Composite Field Arithmetic, Cryptographic Hardware and Embedded Systems (CHES), 2001, pp. 175-188, Springer-Verlag.
13. Shengqi Yang, Wayne Wolf, Narayanan Vijaykrishnan, Dimitrios N. Serpanos, and Yuan Xie, Power Attack Resistant Cryptosystem Design: A Dynamic Voltage and Frequency Switching Approach, ACM/IEEE DATE, 2005, pp. 64-69.
14. Francois-Xavier Standaert, Tal Malkin and Moti Yung, A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks, EUROCRYPT, 2009, pp. 443-461, Springer-Verlag.
15. Qizhi Tian, Abdulhadi Shoufan, Marc Stoettinger, and Sorin A. Huss, Power Trace Alignment for Cryptosystems featuring Random Frequency Countermeasures, IEEE Int. Conf. on Digital Information Processing and Communications, 2012.
16. Qizhi Tian and Sorin A. Huss, On Clock Frequency Effects in Side Channel Attacks of Symmetric Block Ciphers, IEEE Int. Conf. on New Technologies, Mobility, and Security, 2012.
17. Qizhi Tian and Sorin A. Huss, Power Amount Analysis: Another Way to Understand Power Traces in Side Channel Attacks, IEEE Int. Conf. on Digital Information Processing and Communications, 2012.
18. Jasper G. J. van Woudenberg, Marc F. Witteman, and Bram Bakker, Improving Differential Power Analysis by Elastic Alignment, Cryptographers' Track of RSA Conference (CT-RSA), 2011, pp. 104-119, Springer-Verlag.

## 7 Appendix

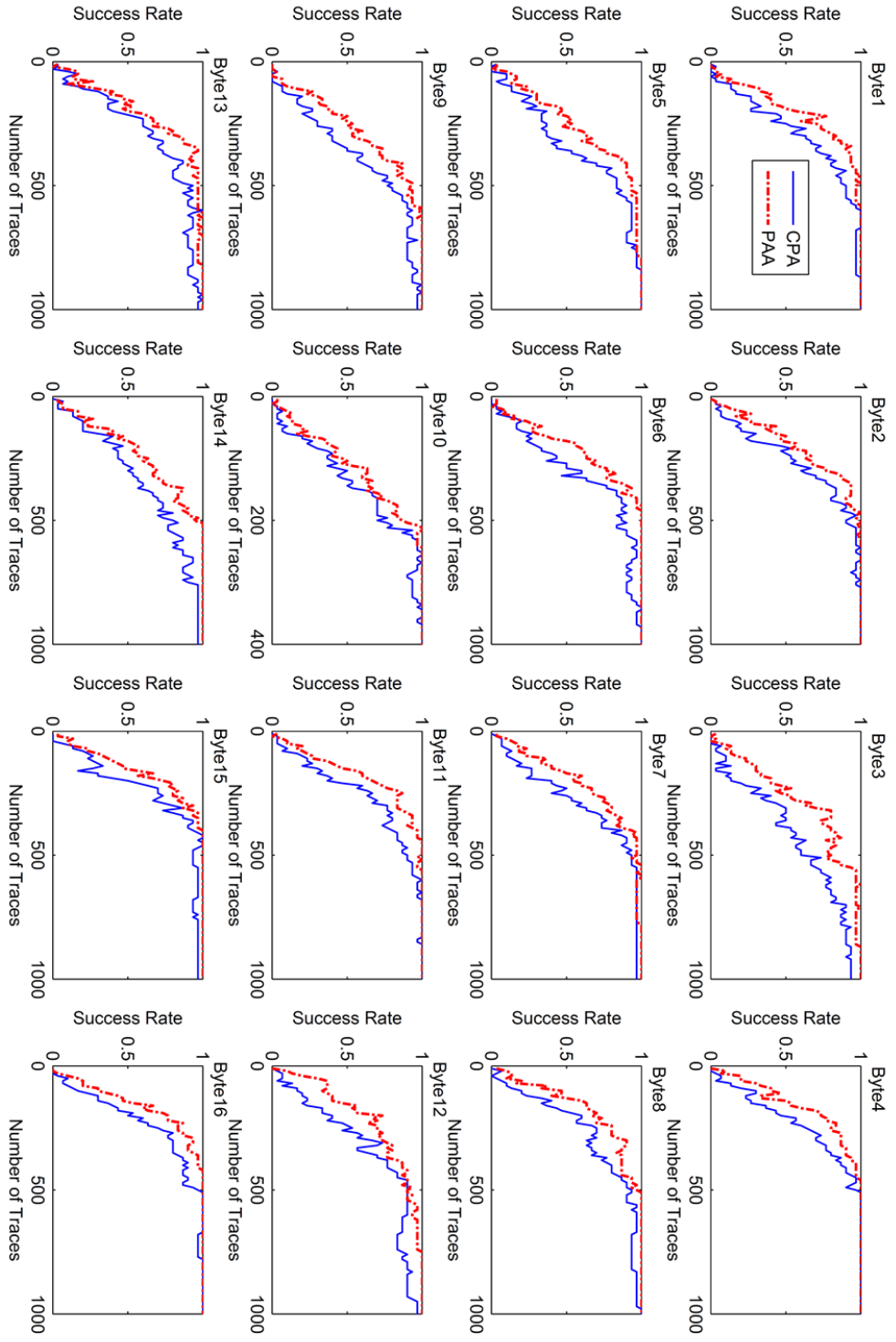


**Figure 10: Success Rate for each Byte in CPA and PAA attacks**