Higher Order Masking of the AES

Kai Schramm and Christof Paar

Horst Görtz Institute for IT Security (HGI),
Universitätsstr. 150, Ruhr University Bochum, Germany, 44780 Bochum, Germany
{schramm, cpaar}@crypto.ruhr-uni-bochum.de

Abstract. The development of masking schemes to secure AES implementations against side channel attacks is a topic of ongoing research. Many different approaches focus on the AES S-box and have been discussed in the previous years. Unfortunately, to our knowledge most of these countermeasures only address first-order DPA. In this article, we discuss the theoretical background of higher order DPA. We give the expected measurement costs an adversary has to deal with for different hardware models. Moreover, we present a masking scheme which protects an AES implementation against higher order DPA. We have implemented this masking scheme for various orders and present the corresponding performance details implementors will have to expect.

Keywords: AES, Higher Order DPA, Masking Countermeasure.

1 Introduction

The *Advanced Encryption Standard* (AES) is the worldwide de-facto standard for symmetric encryption [10]. Therefore, it is very likely that it will be used for many different purposes ranging from high-performance applications such as video stream encryption to low-cost (low memory, low power consumption) implementations on smart cards. Especially in the case of software implementations for smart cards limited memory (ROM, RAM, XRAM) poses a challenging constraint for implementors. Even worse, side channel attacks based on differential power analysis (DPA) [13, 17] and its various branches such as higher order differential power analysis (HODPA) [16, 2] require considerable effort to come up with efficient yet secure implementations which do not succumb to these attacks.

A lot of effort has been devoted in the past years to the development of efficient countermeasures for AES implementations against first-order DPA [7]. Especially the so-called random masking technique has been suggested many times [15, 8, 1, 11] but also algebraic techniques to protect AES implementations against side channel attacks have been proposed in various publications [4, 18, 9, 19]. Unfortunately, HODPA attacks which have been discussed in several articles [16, 21, 12] are often capable to break these first-order countermeasures.

In this article, we would like to present a masking scheme which protects AES implementations against HODPA attacks. In Section 2, we start with a theoretical discussion of HODPA attacks and derive the correlation coefficients of HODPA attacks for various orders and hardware architectures. In Section 3, we propose an AES masking scheme which uses multiple masks and is secure against HODPA attacks. In Section 4, we

D. Pointcheval (Ed.): CT-RSA 2006, LNCS 3860, pp. 208–225, 2006.

present performance-related details of various HODPA-resistant AES implementations which we have programmed on a test device. We conclude this article in Section 5.

2 HODPA: Theoretical Issues

We assume an adversary encrypts N plaintexts X_j and measures the corresponding power traces $P_j(t)$. As discussed in [7, 16, 2, 21], we define a DPA of order d as the correlation of the product of d power signals $P_j(t_1), ..., P_j(t_d)$ with a selected function f of the known plaintext X_j and a key hypothesis K_h.

$$\rho(\prod_{i=1}^{d} P(t_i), f(X_j, K_h)) = \frac{COV[\prod_{i=1}^{d} P(t_i), f(X_j, K_h))]}{\sqrt{V[\prod_{i=1}^{d} P(t_i)]}\sqrt{V[f(X_j, K_h))]}} \tag{1}$$

Since the adversary is generally only able to measure a finite number N of power traces $P_j(t)$, the estimated correlation coefficient $\hat{\rho}(N)$ is computed using the approximated covariance and variances.

$$\begin{aligned} COV[\prod_{i=1}^{d} P(t_i), f(X_j, K_h))] &= \frac{1}{N}\sum_{j=0}^{N-1}\prod_{i=1}^{d} P_j(t_i) f(X_j, K_h) \\ &\quad - \Big(\frac{1}{N}\sum_{j=0}^{N-1}\prod_{i=1}^{d} P_j(t_i)\Big)\Big(\frac{1}{N}\sum_{j=0}^{N-1} f(X_j, K_h)\Big) \qquad (2) \\ V[\prod_{i=1}^{d} P(t_i)] &= \frac{1}{N}\sum_{j=0}^{N-1}\Big(\prod_{i=1}^{d} P_j(t_i) - \frac{1}{N}\sum_{j=0}^{N-1}\prod_{i=1}^{d} P_j(t_i)\Big)^2 \qquad (3) \\ V[f(X, K_h))] &= \frac{1}{N}\sum_{j=0}^{N-1}\Big(f(X_j, K_h) - \frac{1}{N}\sum_{j=0}^{N-1} f(X_j, K_h)\Big)^2. \qquad (4) \end{aligned}$$

2.1 Second Order DPA Against the AES S-Box Input (HW-Model)

We suppose an adversary performs a second-order DPA based on an l-bit key hypothesis K_h against the masked input of an AES S-box in round one with $1 \leq l \leq n = 8$. Furthermore, we assume that a random mask M leaks at time t_1, the masked S-box input $X \oplus K \oplus M$ leaks at time t_2 and that the adversary is able to measure the corresponding power signals $P(t_1)$ and $P(t_2)$. If the power contribution ϵ of all bits is equal[1] and coupling effects among the bits are neglected, the power signals $P(t_1)$ and $P(t_2)$ can be modelled as

$$P(t_1) = \epsilon\sum_{i=0}^{n-1} M[i] + N_1 \quad \text{and} \quad P(t_2) = \epsilon\sum_{i=0}^{n-1}(X \oplus K \oplus M)[i] + N_2$$

[1] Hamming weight model.

where the additive noise terms N_1 and N_2 are assumed to be independent Gaussian random variables with (0,σ^2). As shown in Appendix A, the correlation coefficient is

$$\rho(P(t_1)P(t_2), W(X \oplus K_h)) = \frac{\epsilon^2 \frac{1}{4}(\frac{l}{2} - u)}{\sqrt{(\epsilon^4 \frac{n^2}{16}(3 + 2(n-1)) + \epsilon^2 \frac{n}{2}(\sigma^2 + \sigma^2 n) + \sigma^4)}\sqrt{\frac{l}{4}}} \tag{5}$$

where $W(X \oplus K_h) \leq l$ denotes the Hamming weight of the guessed lower l bits of the unmasked S-box input and u denotes the number of correctly guessed key bits, $0 \leq u \leq l$. The expression can be simplified, if we assume that the power signals only depend on the Hamming weights (ϵ =1) and that the uncorrelated noise terms N_1 and N_2 are neglected ($\sigma = 0$).

$$\rho(P(t_1)P(t_2), W(X \oplus K_h)) = \frac{\frac{1}{4}(\frac{l}{2} - u)}{\sqrt{(\frac{n^2}{16}(3 + 2(n-1))}\sqrt{\frac{l}{4}}} \tag{6}$$

The AES S-box input in the first round is a linear function of a plaintext byte X and a key byte K. As a result, the resulting correlation coefficient shows a linear characteristic. It is proportional to the number of correctly guessed key bits and reaches its minimum (maximum)[2], if all key bits are guessed correctly (incorrectly). Moreover, wrong key guesses which are "close" to the correct key guess (e.g. $l-1$ bits are guessed correctly and only one bit incorrectly) will result in a correlation coefficient which is "close" to its minimum. Therefore, DPA attacks usually focus on the output of non-linear functions, such as the AES S-boxes, because wrong key guesses will result in correlation coefficients which are clearly distinguishable from the correct key guess. Please note, however, that in [5] it was shown that S-boxes which are not perfectly non-linear (e.g. the DES S-boxes) may result in "ghost peaks".

2.2 Multi-bit HODPA Against the AES S-Box Output (HW-Model)

We suppose an adversary performs a DPA of order d against the S-box output, i.e., the adversary correlates the product of d power signals with a selected function of the unmasked S-box output based on the key hypothesis K_h. Moreover, we presume that the leakage of a variable is equal to its Hamming weight (i.e. $\epsilon = 1$ and $\sigma = 0$) and that the adversary knows that the Hamming weights of $d-1$ random masks $M_1, ..., M_{(d-1)}$ leak at times $t_1, ..., t_{(d-1)}$ and that the masked S-box output $S(X \oplus K) \oplus M_1 \oplus ... \oplus M_{(d-1)}$ leaks at time t_d. As shown in the Appendix B, the correlation coefficient ρ of the product $\prod_{i=1}^{d} P(t_i)$ and the Hamming weight of the correctly predicted S-box output $W(S(X \oplus K_h))$ for $K_h = K$ is

$$\rho\Big(\prod_{i=1}^{d} P(t_i), W(S(X \oplus K_h))\Big) = \frac{2^{-(d+1)}(n^{(d+1)} - n) + n2^{-d}(\text{d mod } 2) - (\frac{n}{2})^{(d+1)}}{\sqrt{\Big((\frac{n}{4} + \frac{n^2}{4})^d - (\frac{n}{2})^{(2d)}\Big)\Big(\frac{n}{4}\Big)}} \text{ if } K_h{=}K \tag{7}$$

[2] The minimum and maximum have an equal magnitude, i.e. the correlation coefficient is a symmetric function.

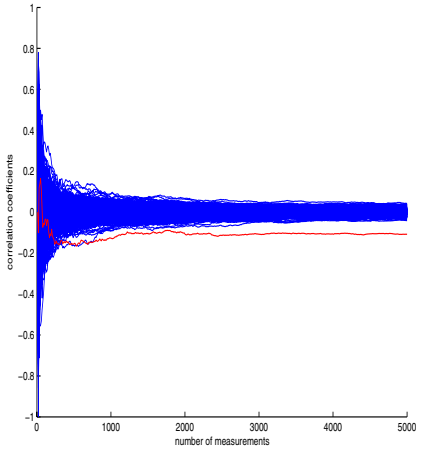

Fig. 1. Correlation plot of a simulated second-order DPA against the AES S-box output according to the HW-model with no noise

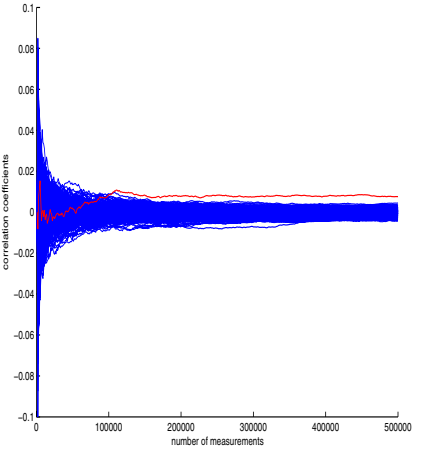

Fig. 2. Correlation plot of a simulated third-order DPA against the AES S-box output according to the HW-model with no noise

where n denotes the bit length of all intermediate variables. In the case of AES, $n = 8$, i.e. 8 bits of K_h must be guessed to predict the S-box output. Thus, the correlation coefficient reduces to

$$\rho\Big(\prod_{i=1}^{d} P(t_i), W(S(X \oplus K))\Big) = \frac{2^{(3-d)}(\text{d mod } 2) - 2^{(-d+2)}}{\sqrt{(18^d - 2^{(4d)})\sqrt{2}}} \quad \text{if} \quad K_h = K \quad (8)$$

while wrong key hypotheses, i.e. $K_h \neq K$, result in correlation coefficients which converge to zero for an increasing number of measurements N due to the non-linear characteristics of the AES S-box. In Figures 1 and 2, the results of a simulated second-order and third-order DPA based on the Hamming weight model are shown. In both cases the correlation coefficients corresponding to the correct key hypotheses are clearly visible, however, significantly more measurements ($\approx$ a factor of 10^2) are required to successfully perform the third-order DPA[3].

In Table 1, the correlation coefficients of correct key hypotheses for DPA attacks of orders $d = 1, ..., 7$ are listed. Please note that the correlation coefficients approximately decrease by a factor of 10 with order d and, moreover, feature alternating signs. In order

Table 1. Correlation coefficients of a successful HODPA for a given order

DPA Order d	1	2	3	4	5	6	7
Corr. Coeff. ρ	1	-0.0857	0.0085	$-8.901 \cdot 10^{-4}$	$9.638 \cdot 10^{-5}$	$-1.064 \cdot 10^{-5}$	$1.191 \cdot 10^{-6}$

[3] Also note that the magnitude of the correlation coefficients in the third-order plot has approx. decreased by a factor of 10.

Table 2. Number of measurements N required to achieve an $|SNR|$ of ≥ 5 in simulated HODPA attacks (averaged over 100 simulated DPA attacks for each order d)

DPA Order d	1	2	3	4
N	31.14 $\approx 3.13 \cdot 10^1$	3941.67 $\approx 3.94 \cdot 10^3$	415513.67 $\approx 4.16 \cdot 10^5$	44383112.11 $\approx 4.44 \cdot 10^7$

to define some kind of quality rating regarding a HODPA attack, we need to define a signal-to-noise ratio (SNR) which expresses how much the estimated correlation coefficient of the correct key hypothesis deviates from the estimated correlation coefficients of the wrong key hypotheses for a certain number of measurements N.

$$SNR(N) = \frac{\hat{\rho}\Big(\prod_{i=1}^{d} P(t_i), W(S(X \oplus K_h))|K_h = K\Big)}{\sqrt{V[\hat{\rho}\Big(\prod_{i=1}^{d} P(t_i), W(S(X \oplus K_h))\Big)|K_h \neq K]}} \tag{9}$$

Experimental results showed that an $|SNR|$ of ≥ 5 is a reasonable threshold, i.e. it results in satisfactory HODPA attacks for which the correct key guess is clearly distinguishable from wrong key guesses. Table 2 lists the average number of measurements N required to achieve an $|SNR|$ of ≥ 5. These numbers were derived from statistical simulations, i.e. for a given order d 100 simulated HODPA attacks were performed. The numbers given in Table 2 clearly show that the measurement costs grow exponentially with DPA order d (see [7]). However, it must be pointed out that practical HODPA attacks will most certainly require more measurements. For example, the assumption that only 31 measurements are required to perform a first-order DPA is extremely optimistic and usually not achieveable in a noisy measurement environment. In order to give a better estimation regarding the measurement costs we analyzed an 8051-based microcontroller whose power consumption behaviour matches surprisingly well the Hamming weight model. For this architecture the power leakage of some 8-bit variable X at time t_X can be modelled as

$$P(t_X) = offset + \epsilon \cdot W(X) + \sigma \cdot N.$$

In an experiment we analyzed $256 \cdot 1000 = 256000$ power traces and determined an average $offset = 10$ mA, current gain $\epsilon = 3.72$ mA, and Gaussian noise with a standard deviation $\sigma = 1.9636$ mA and $N \sim N(0, 1)$. Using these parameters we simulated[4] 100 DPA attacks for each order $d = 1, ..., 4$ in order to determine the average measurement costs required to achieve an $|SNR| \geq 5$. The results are listed in Table 3.

As a result of the additive Gaussian noise $\sigma \cdot N$, the measurement costs roughly increase by a factor of ≈ 10. Hence, the use of a noise generator as an add-on countermeasure is certainly reasonable to make HODPA attacks more difficult. In Figures 3 and 4

[4] Real-world HODPA attacks against the 8051 microcontroller would have been possible, but with regard to the high measurement costs for orders $d > 2$ we decided to simulate these attacks.

Table 3. Number of measurements N required to achieve an $|SNR|$ of ≥ 5 in simulated HODPA based on the HW-model (parameters: $Offset = 10$ mA, $\epsilon = 3.72$ mA, $\sigma = 1.9636$ mA) attacks (averaged over 100 simulated DPA attacks for each order d)

DPA Order d	1	2	3
N	225.85 $\approx 2.26 \cdot 10^2$	12539.1 $\approx 1.25 \cdot 10^4$	3527564 $3.52 \approx \cdot 10^6$

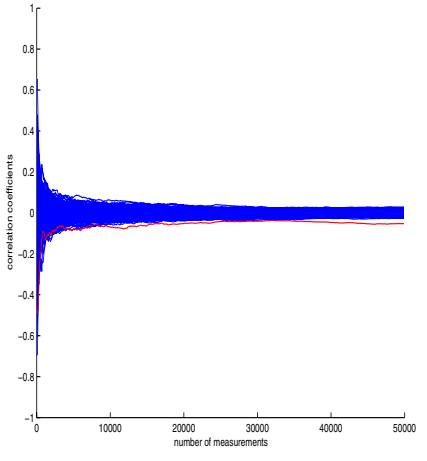

Fig. 3. Correlation plot of a simulated second-order DPA against the AES S-box output according to the HW-model (parameters: $Offset = 10$ mA, $\epsilon = 3.72$ mA, $\sigma = 1.9636$ mA)

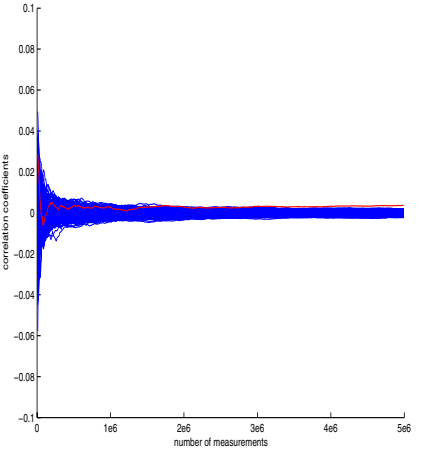

Fig. 4. Correlation plot of a simulated third-order DPA against the AES S-box output according to the HW-model (parameters: $Offset = 10$ mA, $\epsilon = 3.72$ mA, $\sigma = 1.9636$ mA)

two correlation plots of a simulated second and third-order DPA are shown for the aforementioned parameters.

2.3 Single-Bit HODPA Against the AES S-Box Output (General Model)

In the previous sections, we proposed theoretical results of HODPA attacks against hardware architectures with regard to the Hamming weight model. However, as discussed in [6, 3, 14], this model is of limited use in real-world attacks. In [16], a more general model was presented which focuses on a single bit and comprehends all remaining noise sources[5] as a Gaussian distributed random variable. According to this model the two possible probability distributions of a power signal $P(t_i)$ are defined as

$$f(P(t_i)|b = 0) \sim N(-\epsilon, \sigma^2) \quad and \quad f(P(t_i)|b = 1) \sim N(\epsilon, \sigma^2) \tag{10}$$

depending on the state of some bit b, e.g. an S-box output bit, which leaks at time t_i. As derived in Appendix C, if the key K is guessed correctly, the correlation coefficient of the product $\prod_{i=1}^{d} P(t_i)$ and an S-box output bit $b = S(X \oplus K_h)[j]$ with $0 \leq j \leq 7$ is

[5] i.e. both arithmetic noise and measurement noise.

Table 4. Correlation coefficients of a successful single-bit HODPA for various orders d with parameters $\epsilon = 3.1838$ mA and $\sigma = 16.9143$ mA according to the general model

DPA Order d	1	2	3	4	5	6	7
Corr. Coeff. ρ	0.185	-0.0342	$6.30 \cdot 10^{-3}$	$-1.20 \cdot 10^{-3}$	$2.17 \cdot 10^{-4}$	$-4.01 \cdot 10^{-5}$	$7.41 \cdot 10^{-6}$

Table 5. Number of measurements N required to achieve an $|SNR|$ of ≥ 5 in simulated single-bit HODPA attacks with parameters $\epsilon = 3.1838$ mA and $\sigma = 16.9143$ mA (averaged over 100 simulated DPA attacks for each order d)

DPA Order o	1	2	3	4
N	801.8 $\approx 8.02 \cdot 10^2$	22614.37 $\approx 2.26 \cdot 10^4$	1291118.02 $\approx 1.29 \cdot 10^6$	17705001.01 $\approx 1.77 \cdot 10^7$

$$\rho\Big(\prod_{i=1}^{d} P(t_i), S(X \oplus K_h)[j]\Big) = \frac{(-1)^{(d+1)}\epsilon^d}{\sqrt{(\epsilon^2+\sigma^2)^d}} \qquad \text{if} \quad K_h = K \tag{11}$$

In order to estimate the correlation coefficient for various orders d we measured 1000 power traces from a test device[6]. Using this set of measurements we analyzed the power consumption caused by S-box 0 output bit 0 in round one and determined a mean $\epsilon = 3.1838$ mA and a standard deviation $\sigma = 16.9143$ mA. Using these parameters, we were able to estimate the correlation coefficients of DPA attacks for various orders d. These numbers are listed in Table 4. As in the previous section, we also performed simulated DPA attacks for various orders d in order to determine the average number of measurements required to extract the correct key, i.e. to achieve an $|SNR| \geq 5$. These numbers are given in Table 5 and again show an exponential increase. Finally, in Figures 5 and 6 two correlation plots of a simulated second and third-order DPA are shown.

3 Secure HODPA AES Masking Scheme

In this section, we propose an AES masking scheme which is secure against HODPA attacks. We assume that the adversary knows the exact points in time when any occuring intermediate variable leaks in the side channel trace and that she/he is able to measure the corrsponding power signal.

Definition. *A masking scheme which applies $(d-1)$ independent, random masks to blind a subkey-dependent intermediate variable is considered secure, if an adversary must perform an DPA attack of order d, i.e. she/he must correlate at least d power signals with a selected function of the subkey-hypothesis, in order to successfully determine the secret subkey.*

[6] A smart card which is based on the AVR architecture and runs a software implementation of AES. From our measurements we derived that this architecture does not agree very well with the Hamming weight model.

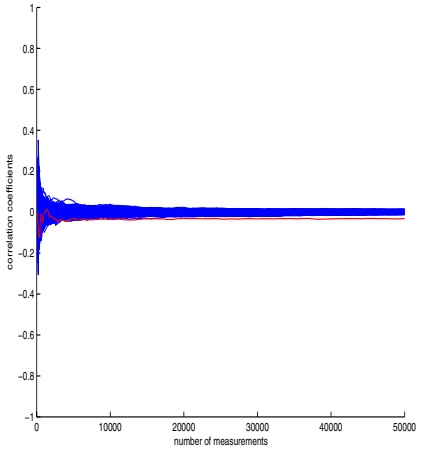

Fig. 5. Correlation plot of a simulated single-bit second-order DPA against the AES S-box output according to the general model (parameters: $\epsilon = 3.1838$ mA and $\sigma = 16.9143$ mA)

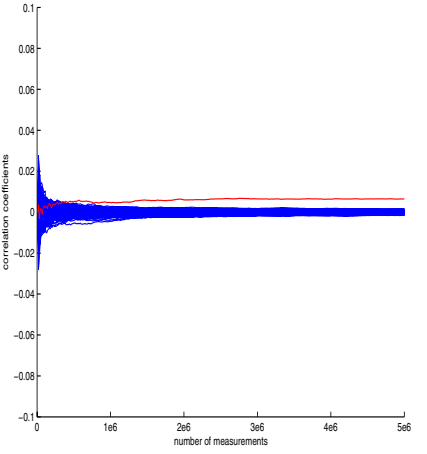

Fig. 6. Correlation plot of a simulated single-bit third-order DPA against the AES S-box output according to the HW-model (parameters: $\epsilon = 3.1838$ mA and $\sigma = 16.9143$ mA)

Let us consider a very simple and naive masking scheme based on a modified S-box S^* which is shown in Figure 7. We assume that the same set of $d-1$ input and output masks $M_1, ..., M_{d-1}$ are used to thwart DPA attacks up to order $d-1$. Unfortunately, this scheme has several vulnerabilities. First, note that the $d-1$ masks at the S-box output can be regarded as a single x-or mask $M = M_1 \oplus ... \oplus M_{d-1}$. While the x-or sum M may never leak by itself as an intermediate variable in the side channel trace, the variables $K \oplus M$, $X \oplus K \oplus M$ and $S(X \oplus K) \oplus M$ do occur and thus cause a leakage. We observed that this gives rise to the following two counterintuitive second-order attacks even if $d-1 > 1$ masks M_i are used. The two correlation coefficients

$$\rho\Big(W(S(X \oplus K) \oplus M) \cdot W(K \oplus M), W(S(X \oplus K_h) \oplus K_h)\Big)$$

and

$$\rho\Big(W(S(X \oplus K) \oplus M) \cdot W(X \oplus K \oplus M), W(S(X \oplus K_h) \oplus X \oplus K_h)\Big)$$

will result in distinct peaks, if the correct key hypothesis is guessed. A simple way to thwart both attacks is to use a different set of input and output masks for an S-box. Furthermore, let us assume that different input and output masks are used for an S-box, however, the same two sets of $d-1$ input masks M_i and output masks N_i are used for all S-boxes. As suggested in [11], this leads to the following second-order attack

$$\rho\Big(W(S(X \oplus K_X) \oplus N)W(S(Y \oplus K_Y) \oplus N), W(S(X \oplus K_{HX}))W(S(Y \oplus K_{HY}))\Big)$$

where $N = N_1 \oplus ... \oplus N_{d-1}$ denotes the x-or sum of the output masks, X,Y denote two arbitrary plaintext bytes, K_X,K_Y the two corresponding key bytes in the first round

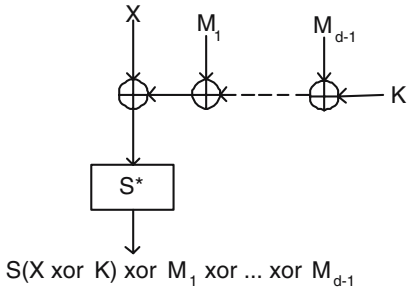

Fig. 7. Insecure AES masking scheme using the same $d-1$ input and output masks to thwart DPA attacks of order d

and K_{HX},K_{HY} the two corresponding key hypotheses guessed by the adversary. Thus, the hypothesis space is increased to 16 key bits which is still feasible. An insufficient measure to counteract this second-order attack would be the random permutation[7] of the 16 S-boxes in each round, since this would merely increase the measurement costs by a factor of $16 \cdot 15 \cdot \frac{1}{2} = 120$. A better countermeasure is the usage of different input and output masks for each S-box.

Design Rule. *Every AES S-box S_j^* with $1 \leq j \leq 16$ should use a different set of $d-1$ input masks $M_{(j,1)}, ..., M_{(j,d-1)}$ and output masks $N_{(j,1)}, ..., N_{(j,d-1)}$ for each round to thwart DPA attacks of orders $< d$.*

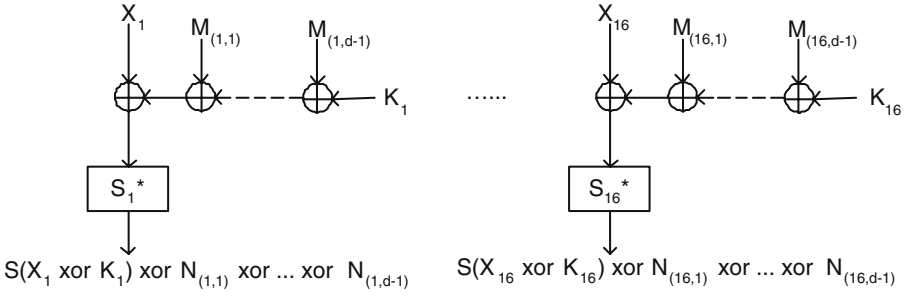

Fig. 8. Secure AES masking scheme which uses $d-1$ different input and output masks for each S-box to thwart DPA attacks of order d

3.1 S-Box Recomputation

In the case of AES, 8-bit x-or masks are used to blind elements in $GF(2^8)$. As stated in [16, 1, 11], the only transformation in AES which requires special attention with regard to masking is the non-linear S-box, which performs an inversion in $GF(2^8)$ followed by an affine bitwise transformation. For this reason, an x-or mask M will not propagate unchanged through the S-box.

[7] i.e. a temporal desynchronisation of the power traces.

$$s((X \oplus K) \oplus M) = s(X \oplus K) \oplus R \neq s(X \oplus K) \oplus M \quad \text{for any } X \oplus K, M \neq \{0\}$$

The S-box must be modified in such a way that $s((X \oplus K) \oplus M) = s(X \oplus K) \oplus M$ for $\forall\ X \oplus K$. This can be achieved twofold: either by simple recomputation of the S-box [15] or by algebraic methods [4, 18, 9, 19]. The disadvantage of algebraic methods is that they are usually not very efficient when implemented in software and generally do not address higher order masking. In [15], a very simple recomputation algorithm was proposed which blinds the index of a table S with mask M, the output with mask N and stores it as a new table S'.

```
For (i=0; i <= 255; i = i + 1)
    A = S(i)
    S'(i x-or M) = A x-or N
End
```

This algorithm requires 256 read and write instructions and 512 bytes RAM[8] for tables S and S'. In [20], the following "split and swap" algorithm was suggested.

```
For (i=0; i <= 255; i = i + 1)
    S(i) = S(i) x-or N  //Apply the output mask N
End
For (j=0; j <= 7; j = j + 1)
    If (M[j] = 1) Then
       (1) Split S into succeeding blocks of 2^j elements
       (2) Swap pairwise the (2n)th and (2n+1)th,... block
    End
End
```

where $M[j]$ denotes bit j of input mask M. The initial step which applies the output mask N requires 256 read and write instructions. If bit $M[j]$ is set, $2^{8-(j+1)}2^{j+1} = 256$ read and write instructions are required for each split-and-swap operation. This results in an average total of $256 + 4 \cdot 256 = 1280$ read and write instructions. As an advantage, only 256 bytes of RAM are required to recompute the S-box. We propose the following S-box recomputation algorithm which requires 256 read and write instructions and only 256 bytes of RAM.

```
Determine the most significant bit for which M[j]=1, j=7,...,0
//For example, if M=0x1C, then j=4

For (i=0; i <= 255; i = i + 2^(j+1) )
     For (l=0; l < 2^j; l = l + 1)
         A = S(i x-or l)
         B = S(i x-or l x-or M)
         S(i x-or l) = B x-or N
         S(i x-or l x-or M) = A x-or N
     End
End
```

[8] We will see later that in the case of higher order masking it is inefficient to store table S in ROM and only S' in RAM even though it makes sense to do so in the case of first-order masking.

Let us assume we would want to apply $d-1$ input masks $M_1, ..., M_{d-1}$ and $d-1$ output masks $N_1, ..., N_{d-1}$ to the original S-box S which has been copied from ROM into RAM. Since the x-or sums $M = M_1 \oplus ... \oplus M_{d-1}$ and $N = N_1 \oplus ... \oplus N_{d-1}$ shall never leak during the execution of the cipher, one possibility is to recompute the S-box $d-1$ times:

$$recompute(S, M_1, N_1) \rightarrow ... \rightarrow recompute(S, M_{d-1}, N_{d-1})$$

The order of the recomputation steps is arbitrary. Fortunately, it is only necessary to perform these $d-1$ recomputations for the very first S-box in round one. Once the first S-box is masked with M and N, it is easy to derive a new S-box with input masks $U_1, ..., U_{d-1}$ and output masks $V_1, ..., V_{d-1}$ by using the *chain of masks* U' and V'.

$$U' = U_1 \oplus M_1 \oplus ... \oplus U_{d-1} \oplus M_{d-1}$$
$$V' = V_1 \oplus N_1 \oplus ... \oplus V_{d-1} \oplus N_{d-1}$$

Thus, the x-or sum U' removes the previous input masks M_i and adds the new input masks U_i, while the x-or sum V' removes the previous output masks N_i and adds the new output masks V_i in one step. It is important that the previous and new masks are stacked up in the alternating order given above to avoid any possible side channel vulnerabilities. As a result, only a single recomputation step $recompute(S, U', V')$ is required to derive the new S-box independent of the number of masks $d-1$.

3.2 Mask Propagation and the MixColumn Transformation

For AES implementations which must be secure against first-order DPA, only, it is sufficient to use a single 8-bit mask M for the entire algorithm. As a matter of fact, the mask M will simply propagate through the MixColumn transformation and no attention must be paid to correct the mask after it has propagated through the MixColumn transformation[9].

$$MixCol \begin{pmatrix} S(X_1 \oplus K_1) \oplus M \\ S(X_2 \oplus K_2) \oplus M \\ S(X_3 \oplus K_3) \oplus M \\ S(X_4 \oplus K_4) \oplus M \end{pmatrix} = MixCol \begin{pmatrix} S(X_1 \oplus K_1) \\ S(X_2 \oplus K_2) \\ S(X_3 \oplus K_3) \\ S(X_4 \oplus K_4) \end{pmatrix} \oplus \begin{pmatrix} M \\ M \\ M \\ M \end{pmatrix}$$

With regard to an AES implementation resistant against a DPA attack of order d let us assume that $d-1$ different input masks $M_{(j,1)}, ..., M_{(j,d-1)}$ and $d-1$ different output masks $N_{(j,1)}, ..., N_{(j,d-1)}$ are used for each S-box j in the first round with $M_j = M_{(j,1)} \oplus ... \oplus M_{(j,d-1)}$ and $N_j = N_{(j,1)} \oplus ... \oplus N_{(j,d-1)}$ and $1 \leq j \leq 16$. In this case, the masks do change after they have propagated through the MixColumn transformation.

$$MixCol \begin{pmatrix} S(X_1 \oplus K_1) \oplus N_1 \\ S(X_2 \oplus K_2) \oplus N_2 \\ S(X_3 \oplus K_3) \oplus N_3 \\ S(X_4 \oplus K_4) \oplus N_4 \end{pmatrix} = MixCol \begin{pmatrix} S(X_1 \oplus K_1) \\ S(X_2 \oplus K_2) \\ S(X_3 \oplus K_3) \\ S(X_4 \oplus K_4) \end{pmatrix} \oplus \begin{pmatrix} N'_1 \\ N'_2 \\ N'_3 \\ N'_4 \end{pmatrix}, \begin{pmatrix} N'_1 \\ N'_2 \\ N'_3 \\ N'_4 \end{pmatrix} \neq \begin{pmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{pmatrix}$$

[9] Please note that special care has to be taken when a single mask M is used in connection with the MixColumn transformation. The computation of the MixColumn transformation must be performed in a carefully chosen order so that the mask M is never cancelled out at any time.

In order to follow the propagation of the output masks $N_{(j,1)}, ..., N_{(j,d-1)}$, the MixColum transformation must be executed an additional $d-1$ times for each column.

$$\begin{pmatrix} N'_{(1,1)} \\ N'_{(2,1)} \\ N'_{(3,1)} \\ N'_{(4,1)} \end{pmatrix} = MixCol \begin{pmatrix} N_{(1,1)} \\ N_{(2,1)} \\ N_{(3,1)} \\ N_{(4,1)} \end{pmatrix}, ..., \begin{pmatrix} N'_{(13,d-1)} \\ N'_{(14,d-1)} \\ N'_{(15,d-1)} \\ N'_{(16,d-1)} \end{pmatrix} = MixCol \begin{pmatrix} N_{(13,d-1)} \\ N_{(14,d-1)} \\ N_{(15,d-1)} \\ N_{(16,d-1)} \end{pmatrix}$$

For example, in an implemenation secure against second-order DPA attacks, the MixColumn transformation must be executed an additional $4 \cdot 2 = 8$ times.

4 HODPA-Resistant AES Implementations

We implemented the following AES implementations on an AVR-based smard card in assembly: an unmasked AES implementation, a first-order DPA-resistant AES implementation using a single mask for the entire AES and, finally, implementations resistant to second, third and fourth-order DPA using different input and output masks for all S-boxes. The details such as code sizes and data sizes of these implementations are given in Table 6. Moreover, the number of cycles required for an encryption and the corresponding execution times[10] are given, as well.

Table 6. Details of various HODPA resistant AES implementations

DPA resistance	S-box algo.	code size [bytes/ROM]	data size [bytes/RAM]	cycles	time [ms]
unprotected	-	1078	16	4625	0.925
1^{st} order resistant	[15]	2422	264	8701	1.74
2^{nd} order resistant	[15]	2798	592	193199	38.6
3^{rd} order resistant	[15]	3350	624	197263	39.5
4^{th} order resistant	[15]	3962	656	201255	40.2
2^{nd} order resistant	see 3.1	2614	336	243581	48.7
3^{rd} order resistant	see 3.1	3164	368	247573	49.5
4^{th} order resistant	see 3.1	4174	400	260229	52.0

Due to the diffusion characteristics of the MixColumn transformation [10], an S-box output in round two depends on 32 key bits and in round three already on 128 key bits. Because of performance issues, we only masked the first three and the last three rounds in our AES implementations. Furthermore, we developed two sets of AES implementations resistant to HODPA. The first set listed in Table 6 uses the simple S-box recomputation alogrithm suggested in [15] which requires 512 bytes of RAM but is quick. The second set listed in Table 6 uses our proposed S-box recomputation alogrithm which requires only 256 bytes of RAM but could not be implemented as efficiently in assembly as the simple S-box recomputation algorithm due to pointer arithmetic issues.

[10] Under the assumption that the device is clocked at 5 MHz.

5 Conclusion

In this article we investigated the theoretical background of HODPA attacks and proposed several ideas how to protect an AES software implementation against such attacks. From our simulated experiments based on different hardware architectures it became clear that HODPA requires a huge number of measurements which exponentially increases with the order of the attack. Hence, a very simple way to protect a device against HODPA would be the use of a protocol which bounds the number of possible encryptions for a secret key. Moreover, we showed that the use of a noise generator as an add-on countermeasure does also increase the measurement costs considerably. We have presented details of various HODPA-resistant AES implementations which were programmed in assembly. In our benchmark tests it became clear that the permanent S-box recomputation is the major bottleneck and slows down the HODPA-resistant implementations, however this should not be an issue, if AES is used in challenge response-based protocols. In theory it would be possible to store all S-boxes in ROM, however, this would require $256 \cdot 256 \cdot 256 = 16$ MB , which is not feasible with currently available smart card microcontrollers.

Acknowledgements

We would like to thank Kerstin Lemke and Ahmad Sadeghi for the helpful discussions. Furthermore, we would like to thank Robert Szerwinski for implementing the various HODPA-resistant AES versions in assembly.

References

1. M.-L. Akkar and C. Giraud. An Implementation of DES and AES Secure against Some Attacks. In Ç. K. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2001*, volume LNCS 2162, pages 309–318. Springer-Verlag, 2001.
2. M.-L. Akkar and L. Goubin. A Generic Protection against High-Order Differential Power Analysis. In T. Johansson, editor, *Fast Software Encryption — FSE 2003*, volume 2887, pages 192–205. Springer-Verlag, 2003.
3. Mehdi-Laurent Akkar, Régis Bevan, Paul Dischamp, and Didier Moyart. Power Analysis, What Is Now Possible... In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000*, volume LNCS 1976, pages 489–502. Springer, 2000.
4. J. Blömer, J. Guajardo, and V. Krummel. Provably Secure Masking of AES. In H. Handschuh and M. Anwar Hasan, editors, *Selected Areas in Cryptography — SAC 2004*, volume 3357, pages 69–83. Springer-Verlag, August 2004.
5. E. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems — CHES 2004*, volume 3156, pages 16–29. Springer-Verlag, 2004.
6. S. Chari, C. S. Jutla, J. R. Rao, , and P. Rohatgi. A Cautionary Note Regarding the Evaluation of AES Candidates on Smart Cards. In *Proceedings: Second AES Candidate Conference (AES2)*, Rome, Italy, March 1999.

7. S. Chari, C. S. Jutla, J. R. Rao, , and P. Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *Advances in Cryptology — CRYPTO '99*, volume LNCS 1666, pages 398 – 412. Springer-Verlag, August 1999.
8. C. Clavier and J.-S. Coron. On Boolean and Arithmetic Masking against Differential Power Analysis. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, pages 231 – 237. Springer-Verlag, 2000.
9. N. T. Courtois and L. Goubin. An Algebraic Masking Method to Protect AES Against Power Attacks. http://eprint.iacr.org/2005/204.pdf, 2005. Cryptology ePrint Archive: Report 2005/204.
10. J. Daemen and V. Rijmen. *The Design of Rijndael*. Springer Verlag, Berlin, 2002.
11. J. D. Golic and C. Tymen. Multiplicative Masking and Power Analysis of AES. In B.S. Kaliski, Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2002*, volume 2523, pages 198–212. Springer-Verlag, 2002.
12. M. Joye, P. Paillier, and B. Schoenmakers. On Second-Order Differential Power Analysis. In *accepted to Cryptographic Hardware and Embedded Systems — CHES 2005*. Springer-Verlag, 2005.
13. P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis: Leaking Secrets. In *Advances in Cryptology — CRYPTO '99*, volume LNCS 1666, pages 388–397. Springer-Verlag, 1999.
14. K. Lemke, K. Schramm, and C. Paar. DPA on n-Bit Sized Boolean and Arithmetic Operations and Its Application to IDEA, RC6 and the HMAC-Construction. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems — CHES 2004*, volume 3156, pages 205–219. Springer-Verlag, August 2004.
15. T. S. Messerges. Securing the AES Finalists Against Power Analysis Attacks. In B. Schneier, editor, *Fast Software Encryption — FSE 2000*, volume LNCS 1978, pages 150 – 164. Springer-Verlag, 2000.
16. T. S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, pages 238 – 251. Springer-Verlag, 2000.
17. T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *USENIX Workshop on Smartcard Technology*, pages 151–162, 1999.
18. E. Oswald and K. Schramm. An Efficient Masking Scheme for AES Software Implementations. In *Workshop on Information Security Applications — WISA 2005*. Springer-Verlag, 2005.
19. A. G. Rostovtsev and O.V. Shemyakina. AES Side Channel Attack Protection Using Random Isomorphisms. http://eprint.iacr.org/2005/087.pdf, 2005. Cryptology ePrint Archive: Report 2005/087.
20. E. Trichina, D.S. Seta, and L. Germani. Simplified Adaptive Multiplicative Masking for AES. In B.S. Kaliski, Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2002*, volume 2523, pages 187–197. Springer-Verlag, 2002.
21. J. Waddle and D. Wagner. Towards Efficient Second-Order Power Analysis. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems — CHES 2004*, volume 3156, pages 1–15. Springer-Verlag, 2004.

A Second Order DPA Against the AES S-Box Input (HW Model)

Given are two power signals $P(t_1)$ and $P(t_2)$ according to the HW-model

$$P(t_1) = \epsilon \sum_{i=0}^{n-1} M[i] + N_1 \quad \text{and} \quad P(t_2) = \epsilon \sum_{i=0}^{n-1} (X \oplus K \oplus M)[i] + N_2$$

with $N_1, N_2 \sim N(0, \sigma^2)$ and with n equal to the bit length of all intermediate variables, i.e. in the case of AES $n = 8$. Then, the covariance of the product $P(t_1) \cdot P(t_2)$ and the Hamming weight of the hypothesized S-box input $W(X \oplus K_h)$ is

$$
\begin{aligned}
&COV[P(t_1) \cdot P(t_2), W(X \oplus K_h)] = COV[P(t_1) \cdot P(t_2), W(X \oplus K_h) - \frac{l}{2}] \\
&= E[P(t_1)P(t_2) \cdot (W(X \oplus K_h) - \frac{l}{2})] - E[P(t_1)P(t_2)] \cdot \underbrace{E[(W(X \oplus K_h) - \frac{l}{2})]}_{=0} \\
&= \underbrace{E[\epsilon^2 \sum_{i=0}^{n-1} \sum_{h=0, h \neq i}^{n-1} \sum_{j=0}^{l-1} M[i](M \oplus X \oplus K)[h](X \oplus K_h)[j]]}_{\epsilon^2 \cdot n \cdot (n-1) \cdot l \cdot \frac{1}{8}} \\
&\quad \underbrace{- \frac{l}{2} E[\epsilon^2 \sum_{i=0}^{n-1} \sum_{h=0, h \neq i}^{n-1} M[i](M \oplus X \oplus K)[h]]}_{\frac{l}{2} \cdot \epsilon^2 \cdot n \cdot (n-1) \cdot \frac{1}{4}} \\
&\quad + \underbrace{E[\epsilon^2 \sum_{i=l}^{n-1} \sum_{j=0}^{l-1} M[i](M \oplus X \oplus K)[i](X \oplus K_h)[j])]}_{\epsilon^2 \cdot (n-l) \cdot \frac{1}{4} \cdot l \cdot \frac{1}{2}} - \underbrace{\frac{l}{2} E[\epsilon^2 \sum_{i=l}^{n-1} M[i](M \oplus X \oplus K)[i]]}_{\frac{l}{2} \cdot \epsilon^2 \cdot (n-l) \cdot \frac{1}{4}} \\
&\quad + \underbrace{E[\epsilon^2 \sum_{i=0}^{l-1} \sum_{j=0}^{l-1} M[i](M \oplus X \oplus K)[i](X \oplus K_h)[j])]}_{\epsilon^2 \cdot (l-u) \cdot \frac{1}{4} + \epsilon^2 \cdot l \cdot (l-1) \cdot \frac{1}{4} \cdot \frac{1}{2}} - \underbrace{\frac{l}{2} E[\epsilon^2 \sum_{i=0}^{l-1} M[i](M \oplus X \oplus K)[i]]}_{\frac{l^2}{2} \cdot \epsilon^2 \cdot \frac{1}{4}} \\
&\quad + \underbrace{E[N_2 \cdot \epsilon \cdot \sum_{i=0}^{n-1} M[i] \cdot (\sum_{i=0}^{l-1} X \oplus K_h[i] - \frac{l}{2})]}_{=0} \\
&\quad + \underbrace{E[N_1 \cdot \epsilon \cdot \sum_{i=0}^{n-1} (M \oplus X \oplus K)[i] \cdot (\sum_{i=0}^{l-1} X \oplus K_h[i] - \frac{l}{2})]}_{=0} \\
&\quad + \underbrace{E[N_1 \cdot N_2 \cdot (\sum_{i=0}^{l-1} X \oplus K_h[i] - \frac{l}{2})]}_{=0} = \frac{1}{4} \cdot \epsilon^2 \cdot (\frac{l}{2} - u)
\end{aligned}
$$

where u denotes the number of correctly guessed key bits, $0 \leq u \leq l$. The variance of the Hamming weight of the hypothesized S-box input is

$$V[W(X \oplus K_h)] = V[W(X \oplus K_h) - \frac{l}{2}] = \frac{l}{4}$$

The variance of the product $P(t_1) \cdot P(t_2)$ can be expressed as

$$V[P(t_1) \cdot P(t_2)] = \underbrace{V[\epsilon^2 \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} M[i](M \oplus X \oplus K)[j]]}_{\epsilon^4 \cdot n^2 \cdot \frac{1}{16} \cdot (3+2(n-1))} + \underbrace{V[N_2 \cdot \epsilon \cdot \sum_{i=0}^{n-1} M[i]]}_{\sigma^2 \cdot \epsilon^2 \cdot \frac{n}{4} + \sigma^2 \cdot \epsilon^2 \cdot \frac{n^2}{4}}$$
$$+ \underbrace{V[N_1 \cdot \epsilon \cdot \sum_{i=0}^{n-1} M \oplus X \oplus K[i]]}_{\sigma^2 \cdot \epsilon^2 \cdot \frac{n}{4} + \sigma^2 \cdot \epsilon^2 \cdot \frac{n^2}{4}} + \underbrace{V[N_1 \cdot N_2]}_{\sigma^4}$$
$$= \epsilon^4 \cdot n^2 \cdot \frac{1}{16} \cdot (3 + 2(n-1)) + \epsilon^2 \frac{n}{2}[(1+n)\sigma^2] + \sigma^4$$

This results in the correlation coefficient

$$\rho(P(t_1) \cdot P(t_2), W(X \oplus K_h)) = \frac{COV[P(t_1) \cdot P(t_2), W(X \oplus K_h)]}{\sqrt{V[P(t_1) \cdot P(t_2)]} \cdot \sqrt{V[W(X \oplus K_h)]}}$$
$$= \frac{COV[P(t_1) \cdot P(t_2), W(X \oplus K_h) - \frac{l}{2}]}{\sqrt{V[P(t_1) \cdot P(t_2)]} \cdot \sqrt{V[W(X \oplus K_h) - \frac{l}{2}]}}$$
$$= \frac{\frac{1}{4}\epsilon^2(\frac{l}{2} - u)}{\sqrt{(\epsilon^4 \frac{n^2}{16}(3 + 2(n-1)) + \epsilon^2 \frac{n}{2}(\sigma^2 + \sigma^2 n) + \sigma^4)} \sqrt{\frac{l}{4}}}$$

B Multi-bit HODPA Against the AES S-Box Output (HW Model)

Given are d power signals $P(t_i)$ according to the noise-free Hamming weight model

$$\begin{aligned} P(t_1) &= W(M_1) \\ \ldots \quad & \quad \ldots \\ P(t_{(d-1)}) &= W(M_{(d-1)}) \\ P(t_d) &= W(S(X \oplus K) \oplus M) \end{aligned}$$

with $M = M_1 \oplus ... \oplus M_{(d-1)}$. Let n be the bit length of all intermediate variables, i.e. in the case of AES $n = 8$. Then, the covariance of the product $\prod_{i=1}^{d} P(t_i)$ and the Hamming weight of the hypothesized S-box output $W(S(X \oplus K_h))$ is

$$COV[\prod_{i=1}^{d} P(t_i), W(S(X \oplus K_h))] = COV[\prod_{i=1}^{d} P(t_i), W(S(X \oplus K_h)) - \frac{n}{2}]$$
$$= E[\prod_{i=1}^{d} P(t_i) \cdot (W(S(X \oplus K_h)) - \frac{n}{2})] - E[\prod_{i=1}^{d} P(t_i)] \cdot \underbrace{E[(W(S(X \oplus K_h)) - \frac{n}{2})]}_{=0}$$

$$= E[\prod_{i=1}^{d} P(t_i) \cdot (W(S(X \oplus K_h))] - \frac{n}{2} \cdot \underbrace{E[\prod_{i=1}^{d} P(t_i)]}_{=(\frac{n}{2})^d}$$

$$= \underbrace{E[\prod_{i=1}^{d} P(t_i) \cdot (W(S(X \oplus K_h))]}_{=(\frac{n}{2})^{(d+1)}, if K_h \neq K \quad and \quad 2^{-(d+1)}(n^{(d+1)}-n)+n2^{-d}(\text{d mod } 2), if K_h = K} - (\frac{n}{2})^{(d+1)}$$

The variance of the Hamming weight of the hypothesized S-box output is

$$V[W(S(X \oplus K_h))] = V[W(S(X \oplus K_h)) - \frac{n}{2}] = \frac{n}{4}$$

The variance of the product $\prod_{i=1}^{d} P(t_i)$ is

$$\begin{aligned} V[\prod_{i=1}^{d} P(t_i)] &= E[\prod_{i=1}^{d} P^2(t_i)] - E[\prod_{i=1}^{d} P(t_i)]^2 \\ &= \underbrace{E[W^2(M_1) \cdot ... \cdot W^2(M_{(d-1)}) \cdot W^2(S(X \oplus K_h))]}_{(\frac{n}{4}+\frac{n^2}{4})^d} \\ &\quad - \underbrace{E[W(M_1) \cdot ... \cdot W(M_{(d-1)}) \cdot W(S(X \oplus K_h))]^2}_{(\frac{n}{2})^{(2d)}} \\ &= (\frac{n}{4} + \frac{n^2}{4})^d - (\frac{n}{2})^{(2d)} \end{aligned}$$

This results in a correlation coefficient

$$\begin{aligned} \rho(\prod_{i=1}^{d} P(t_i), W(X \oplus K_h)) &= \frac{COV[\prod_{i=1}^{d} P(t_i), W(X \oplus K_h)]}{\sqrt{V[\prod_{i=1}^{d} P(t_i)]} \cdot \sqrt{V[W(X \oplus K_h)]}} \\ &= \frac{COV[\prod_{i=1}^{d} P(t_i), W(X \oplus K_h) - \frac{n}{2}]}{\sqrt{V[\prod_{i=1}^{d} P(t_i)]} \cdot \sqrt{V[W(X \oplus K_h) - \frac{n}{2}]}} \\ &= \frac{2^{-(d+1)}(n^{(d+1)} - n) + n2^{-d}(\text{d mod } 2) - (\frac{n}{2})^{(d+1)}}{\sqrt{\left((\frac{n}{4} + \frac{n^2}{4})^d - (\frac{n}{2})^{(2d)}\right)\left(\frac{n}{4}\right)}} \end{aligned}$$

C Single-Bit HODPA Against the AES S-Box Output (General Model)

Given are d power signals $P(t_i)$ according to the general model

$$\begin{aligned} P(t_1) &= \Big(2\Big(S(X \oplus K) \oplus M\Big)[j] - 1\Big)\epsilon + \sigma N_1 \\ P(t_2) &= \Big(2M_1[j] - 1\Big)\epsilon + \sigma N_2 \\ &\ldots \quad \ldots \\ P(t_d) &= \Big(2M_{(d-1)}[j] - 1\Big)\epsilon + \sigma N_d \quad \text{with} \end{aligned}$$

with $M = M_1 \oplus ... \oplus M_{(d-1)}$ and $N_1, ..., N_d \sim N(0,1)$ and $0 \leq j \leq 7$. The correlation coefficient is defined as

$$\rho\Big(\prod_{i=1}^{d} P(t_i), S(X \oplus K_h)[j]\Big) = \frac{E[\prod_{i=1}^{d} P(t_i) S(X \oplus K_h)[j]] - E[\prod_{i=1}^{d} P(t_i)] E[S(X \oplus K_h)[j]]}{\sqrt{V[\prod_{i=1}^{d} P(t_i)] V[S(X \oplus K_h)[j]]}}$$

where $S(X \oplus K_h)[j]$ denotes the state of bit j of a hypothesized S-box output $S(X \oplus K_h)$. The expectation values in the numerator are

$$\begin{aligned}
&E[\prod_{i=1}^{d} P(t_i) S(X \oplus K_h)[j]] \\
&= E[\sum_{S(X \oplus K) \oplus M = 0}^{1} P(t_1) \sum_{M_1 = 0}^{1} P(t_2) \quad ... \sum_{M_{d-1}=0}^{1} P(t_d) \sum_{S(X \oplus K_h)[j]=0}^{1} S(X \oplus K_h)[j]] \\
&= 2^{-(d+1)} \Big(((-\epsilon) + \epsilon)((-\epsilon) + \epsilon)...((-\epsilon) + \epsilon)(0 + 1)\Big) = 0 \qquad \text{if} \quad K_h \neq K \\
&= 2^{-(d)} \Big(\frac{2^d}{2} \epsilon^d (-1)^{d+1}\Big) = \frac{1}{2} \epsilon^d (-1)^{d+1} \qquad \text{if} \quad K_h = K
\end{aligned}$$

$$\begin{aligned}
&E[\prod_{i=1}^{d} P(t_i)] = \prod_{i=1}^{d} E[P(t_i)] = \prod_{i=1}^{d} (0.5(-\epsilon) + 0.5\epsilon) = 0 \\
&E[S(X \oplus K_h)[j]] = 0.5
\end{aligned}$$

The variances in the denominator are

$$\begin{aligned}
&V[S(X \oplus K_h)[j]] = 0.25 \\
&V[\prod_{i=1}^{d} P(t_i)] = E[\prod_{i=1}^{d} P^2(t_i)] - \underbrace{E[\prod_{i=1}^{d} P(t_i)]^2}_{=0} \\
&= \prod_{i=1}^{d} E[P^2(t_i)] = \prod_{i=1}^{d} E[(\epsilon + \sigma N)^2] = \prod_{i=1}^{d} E[\epsilon^2 + 2\epsilon\sigma N + \sigma^2 N^2] \\
&= (\epsilon^2 + \sigma^2)^d \quad \text{with} \quad N \sim N(0,1) \quad \text{and} \quad \chi^2 = N^2 \sim \chi^2(1,2)
\end{aligned}$$

This results in

$$\begin{aligned}
\rho\Big(\prod_{i=1}^{d} P(t_i), S(X \oplus K_h)[j]\Big) &= \frac{(-1)^{(d+1)} \epsilon^d}{\sqrt{(\epsilon^2 + \sigma^2)^d}} \qquad \text{if} \quad K_h = K \\
&= 0 \qquad \text{if} \quad K_h \neq K
\end{aligned}$$