Improved Higher-Order Side-Channel Attacks with FPGA Experiments

Eric Peeters, François-Xavier Standaert,
Nicolas Donckers, and Jean-Jacques Quisquater

UCL Crypto Group, Laboratoire de Microélectronique,
Université Catholique de Louvain, Place du Levant,
3, B-1348, Louvain-La-Neuve, Belgium
{peeters, fstandae, donckers, quisquater}@dice.ucl.ac.be

Abstract. We demonstrate that masking a block cipher implementation does not sufficiently improve its security against side-channel attacks. Under exactly the same hypotheses as in a Differential Power Analysis (DPA), we describe an improvement of the previously introduced higher-order techniques allowing us to defeat masked implementations in a low (*i.e.* practically tractable) number of measurements. The proposed technique is based on the efficient use of the statistical distributions of the power consumption in an actual design. It is confirmed both by theoretical predictions and practical experiments against FPGA devices.

Keywords: cryptographic devices, side-channel analysis, DPA, high-order power analysis, masking countermeasure, block cipher, FPGA.

1 Introduction

Since their publication in 1998 [9], power analysis attacks have attracted significant attention within the cryptographic community. Although less general than classical cryptanalysis, because they usually target one specific implementation, they have been particularly efficient to break a wide variety of devices, including smart cards, ASICs and FPGAs [12,16,20]. As a straightforward consequence, countermeasures against these attacks are of great practical interest.

In the open literature, the masking technique is among the most popular suggested ways to protect an implementation against Differential Power Analysis [1,6,7,18]. However, several works have shown that such protected devices are still sensitive to higher-order attacks, originally described in [13]. In particular, a recent advance [24] suggested that higher-order power analysis is possible, without any additional hypothesis than usually assumed for first-order attacks. They proposed a way to combine the leakages corresponding to the masked data and its mask even if their respective position within the sampled data is unknown. [21] proposed an extension of these attacks by considering a more general power consumption model. Although these papers provide indications for the practical implementation of the attack, the number of observations required to retrieve the secret key is generally large (at least significantly larger than in a

J.R. Rao and B. Sunar (Eds.): CHES 2005, LNCS 3659, pp. 309–323, 2005.

first-order power analysis attack). As a consequence, masking is usually believed to improve the actual security of an implementation.

In this paper, we demonstrate that masking a block cipher implementation does not sufficiently improve its security against a side-channel opponent. Under exactly the same hypotheses as in a Differential Power Analysis, we provide strong evidence that a masked block cipher implementation can be defeated by an improved higher-order attack, using a low (*i.e.* practically tractable) number of measurements. The proposed technique is based on the efficient use of the statistical distributions of the power consumption in an actual design. Based on these distributions, we describe how to recover the secret key of a masked block cipher implementation, applying a maximum likelihood approach, as suggested in [2]. We confirm our assertions both by theoretical predictions, using the formalism of attacks introduced in [16,20], and practical experiments against real world Field Programmable Gate Array (FPGA) designs. Remark that our results focus on the extraction of information from the available power traces. For simplicity purposes, we assumed the mask and masked data to be computed in parallel and did not discuss possible synchronization issues. However, the extension to other contexts is straightforward using the techniques introduced in [24].

The rest of the paper is structured as follows. Sections 2 and 3 respectively describe the masking countermeasure and our power consumption model. The description of the improved higher-order attack is in Section 4. Simulated attacks are in Section 5 and Section 6 provides the experimental results against a masked block cipher FPGA implementation. Conclusions are in Section 7.

2 The Masking Countermeasure

The idea of masking the intermediate values inside a cryptographic algorithm is suggested in several papers as a possible countermeasure to power analysis. The technique is generally applicable if all the fundamental operations used in a given algorithm can be rewritten in the masked domain. This is easily seen to be the case in classical algorithms such as the DES [14] or AES [15]. Although these methods have been originally applied at the algorithmic level as well as at the gate level, it has been shown recently that masking at the gate level involves critical security concerns. Reference [10] notably demonstrates that the glitching activity of masked logic gates offers a previously neglected leakage source that seriously affects the security of the countermeasure. For this reason, this paper will mainly discuss the algorithmic level protection, using precomputed tables.

In the following sections, we question the security of the masking countermeasure with respect to higher-order power analysis attacks. For this purpose, we start by giving a simple description of our target implementations. An unmasked block cipher design is represented in Figure 1, where the b_is represent known input values, the k_is are the secret encryption key bits and the S blocks are non-linear substitution boxes (let N_s be the number of such S-boxes). In accordance with the structure of most present block ciphers [3,4,14,15], we do

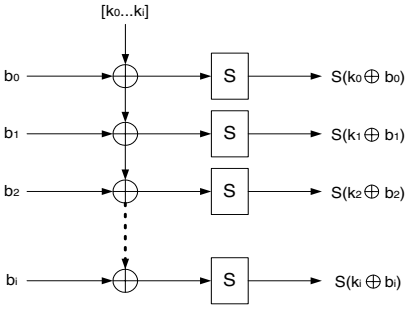

Fig. 1. Unprotected scheme

not loose in generality by focusing our attention to this combination of key additions and non-linear S-boxes. Remark that the bit-widths are not specified on the scheme.

Our protected implementation is represented in Figure 2. The masking principle is as follows. After having XORed the random mask to the initial data, both the mask and the masked data are sent through a non-linear S-box. S is the original S-box from the algorithm and S' is a precomputed table such that we have:

$$S(b \oplus k \oplus r) = S(b \oplus k) \oplus S'(r, b \oplus k \oplus r) = S(b \oplus k) \oplus q$$

As a consequence, the output values are still masked with a random mask q.

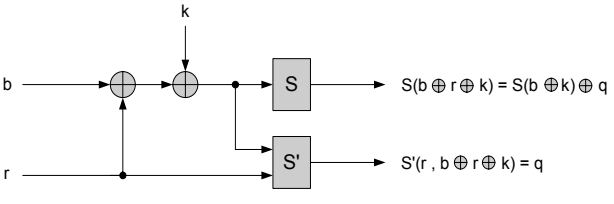

Fig. 2. Masked scheme

3 Power Consumption Model

Power analysis attacks generally target CMOS devices for which it is reasonable to assume that the main component of the power consumption is the dynamic power consumption. For a single CMOS gate, we can express it as follows [19]:

$$P_D = C_L V_{DD}^2 P_{0\to1} f \tag{1}$$

where C_L is the gate load capacitance, V_{DD} the supply voltage, $P_{0\to1}$ the probability of a $0 \to 1$ output transition and f the clock frequency. Equation (1)

specifies that the power consumption of CMOS circuits is data-dependent. As a consequence, a reasonable hypothesis for the power consumption model is: *let x and x' be two consecutive intermediate values of the running algorithm in the target device, let t be the time at which x switches into x', then the power consumption of the device at this time is proportional to the Hamming weight $W_H(x \oplus x')$.*

This hypothesis, usually denoted as the Hamming distance power consumption model, is generally true for any CMOS circuit and is specifically applicable to FPGAs. However, in certain particular contexts, more specific hypotheses hold. For example, in processors with precharged buses, the power consumption may depend on the Hamming weight of the data on the bus [5]. We note that most of our conclusions remain applicable, independently of the power consumption model and target device selected.

4 Attack Description

Let us describe the proposed technique with the single S-box scheme of Figure 3, where the inputs b, r and k are N_b-bit wide. First, we express the power

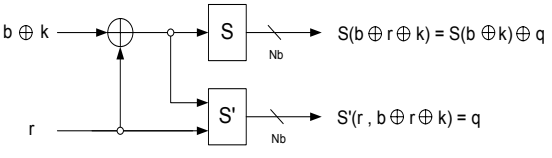

Fig. 3. Illustrative 4-bit scheme

consumption of one pair of S and S' boxes in case of a pipeline block cipher implementation and denote it as a random variable O, standing for observations. That is, we assume that the structure displayed in Figure 3 is fed with a new input at each clock cycle. As explained in the previous section, the power consumption is a function of any two consecutive values. If $b \oplus k$ switches into $b' \oplus k$ and q switches into q', we have:

$$O = WH\Big[\big(S(b \oplus k) \oplus q\big) \oplus \big(S(b' \oplus k) \oplus q'\big)\Big] + WH\Big[q \oplus q'\Big]$$

Defining the random variable $\Sigma = S(b \oplus k) \oplus S(b' \oplus k)$, where Σ stands for secret state and the random variable $R = q \oplus q'$, where R stands for random state, it is therefore possible to rewrite the observations as:

$$O(\Sigma, R) = WH\big[\Sigma \oplus R\big] + WH\big[R\big]$$

We note again that the observations could be expressed in exactly the same way in the Hamming weight power consumption model[1]. We note also that the

[1] We would find $O = WH\big[S(b \oplus k) \oplus q\big] + WH\big[q\big]$, which yields $\Sigma = S(b \oplus k)$, $R = q$.

operator used to combine the two leakage contributions is a '+' because in our analysis, the masked data and its mask are loaded on the register at the same time. But in other contexts, we may choose a '−' as in [13], or a '×' as in [21,24]. Actually, no matter what operator we use, the main point is to gather the two (or more in case of higher-order masking) statistical distributions of the power consumption so that the combined statistical distribution is key-dependent.

Indeed, while it is not possible to predict the observations, because they depend on unknown mask and key values, we can still analyze their statistical behavior. For a fixed value of the secret state $\Sigma = \sigma_i$, we can determine all the possible observations, for all the different possible random states $R = r_j$. From this analysis, it is therefore possible to derive the probability density functions $P[O = o_i|\Sigma = \sigma_i]$, for all the possible secret states.

In practice, because the observations are a sum of two Hamming weight values, they are distributed as binomials and the number of possible distributions for $P[O|\Sigma = \sigma_i]$ equals N_b+1. As a simple illustration, if $N_b = 4$, the five possible distributions of the observations are given in Figure 4.

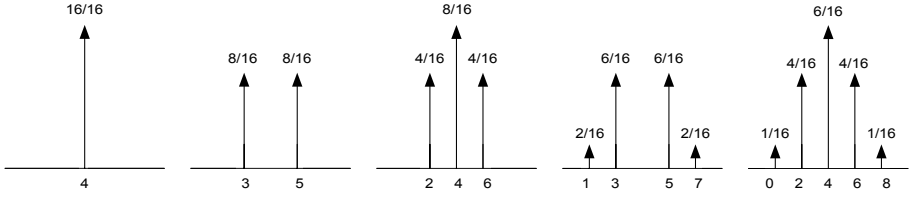

Fig. 4. Probability density functions $P[O|\Sigma = \sigma_i]$ with $N_b = 4$

The important consequence is that, knowing a secret state σ_i, we know the probability of making an observation o_i. This provides us with the tool to mount a new attack, based on a maximum likelihood approach.

Remark: The distributions $P[O|\Sigma = \sigma_i]$ all have the same mean value, $E(O|\Sigma = \sigma_i) = N_b$ and only differ in their variances. This fact allows to understand the origin of previous attacks, as the one in [24], where it is proposed to square the power consumption traces in order to obtain key-dependent measurements. The reason is that the mean of the squared power trace is a function of the mean and the variance of the initial power trace:

$$E\Big((O|\Sigma = \sigma_i)^2\Big) = E\Big((O|\Sigma = \sigma_i)\Big)^2 + V(O|\Sigma = \sigma_i)$$

It is also clear that the information contained in the expectation of the squared power trace is poor compared to what can be obtained using the complete statistical distribution of the observations.

Now, using the usual framework of side-channel attacks, we would like to find the secret key k, using a serial of observations $o_1, o_2, ..., o_n$, obtained by

feeding the encryption device with a serial of input texts $b_0, b_1, ..., b_n$ (the input transition $b_0 \rightarrow b_1$ gives rise to the observation o_1).

For this purpose, we first remark that, knowing the sequence of input texts $b_0, b_1, ..., b_n$, each key candidate $k_i \in [0, 2^{Nb} - 1]$ specifies one sequence of secret states. Therefore, we have 2^{N_b} possible chains of states denoted as:

$$\begin{aligned}
\Sigma^*(k_0) &:= \{\sigma_1(k_0), \sigma_2(k_0), ..., \sigma_n(k_0)\};\\
\Sigma^*(k_1) &:= \{\sigma_1(k_1), \sigma_2(k_1), ..., \sigma_n(k_1)\};\\
\Sigma^*(k_2) &:= \{\sigma_1(k_2), \sigma_2(k_2), ..., \sigma_n(k_2)\};\\
&\ldots
\end{aligned}$$

In practice, these state sequences cannot be observed directly, but only through the power consumption of the device, *i.e.* the sequence of observations $O^* := \{o_1, o_2, ..., o_n\}$. Then, for each possible secret state chain, we compute the probabilities $P[O^*|\Sigma^*(k_j)]$. Assuming that the observations are independent (which is reasonable since the attacker feeds the devices with random input texts), it yields:

$$\begin{aligned}
P[O^*|\Sigma^*(k_0)] = &\quad P[O = o_1|\Sigma = \sigma_1(k_0)] \times P[O = o_2|\Sigma = \sigma_2(k_0)] \times ...\\
P[O^*|\Sigma^*(k_1)] = &\quad P[O = o_1|\Sigma = \sigma_1(k_1)] \times P[O = o_2|\Sigma = \sigma_2(k_1)] \times ...\\
P[O^*|\Sigma^*(k_2)] = &\quad P[O = o_1|\Sigma = \sigma_1(k_2)] \times P[O = o_2|\Sigma = \sigma_2(k_2)] \times ...\\
&\ldots
\end{aligned}$$

The chain with the highest probability gives us the most likely key. That is, if the attack is successful, the correct key corresponds to:

$$\underset{\forall\ k_j}{argmax}\, P[O^*|\Sigma^*(k_j)]$$

We note that the proposed approach is similar to the one in [8], where it is demonstrated that Hidden Markov Models may be of great help to describe discrete time processes where a state sequence is hidden. Remark finally that, in order to keep the probabilities $P[O^*|\Sigma^*(k_j)]$ within practical boundaries (for large n's, these probabilities are smaller than the machine-ϵ), we use a step by step normalization.

5 Simulated Attacks

The previous section described a higher-order power analysis attack against a single S-box scheme, without considering any kind of noise in the measurements. However, in practice, side-channel attacks are usually affected by different kinds of noises. First, block ciphers are made of the application of several S-boxes in parallel, combined with other components such as a diffusion layer (this is typically the case of the AES Rijndael [15]). These "other components" that are not directly targeted by our attack may therefore cause additional power consumption that we denote as the "algorithmic noise". Algorithmic noise exists if these

components use different resources in the circuit, which is typically the case of parallel implementations in FPGAs. Second, real life observations are usually affected by different types of "physical noises". It includes all the possible imperfections of our model appearing during the measurement process.
In order to evaluate the efficiency of the proposed attack, this section considers attacks using "perfect measurements", without any kind of physical noise. This formalism has been introduced in [16,20] and was denoted as "attacks using simulated data". Such attacks basically use simulated measurements generated by computing the number of transitions in the targeted design. The measurements are perfect in the sense that they perfectly fit to the power consumption model. As a matter of fact, the number of measurements required to have a successful attack using simulated data lower bounds this number when real measurements are considered. Still, these simulated experiments allow us to clearly evaluate the effect of algorithmic noise and to compare our attack to a classical Correlation Power Analysis against an unprotected block cipher implementation.

Remark that, as far as noise is concerned, the probability distributions $P[O|\Sigma = \sigma_j]$ are not discrete anymore. However, the previous techniques still hold assuming that the probability density functions (or pdf's) become weighted sums of Gaussians. For example, let us assume that we target a single 4-bit S-box as in the previous section and that for a particular secret state σ_j, the pdf is represented in the left part of Figure 5. If we now consider that the target

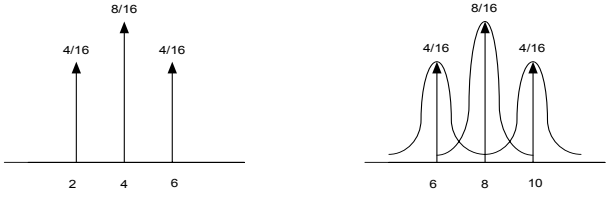

Fig. 5. Probability density functions $P[O|\Sigma = \sigma_j]$ with $N_b = 4$, without or with noise

implementation contains another masked 4-bit S-box (*i.e.* $N_s = 2$), producing algorithmic noise of mean N_b and variance $N_b/2$, we obtain the right part of the figure[2]. In general, finding the noise pdf's can simply be achieved by computing the mean and variance of the observations, as we know the signal pdf's. We now present a number of attacks using simulated data.

We define the parameters of our simulated attacks as follows. First, we use the 4-bit S-boxes of the Serpent algorithm [3] and a secret key $k = 5$. In our target implementations, we consider N_s S-boxes implemented in parallel. The number of plaintexts generated in the attacks is n and for each number of plaintexts, we observe the probabilities $P[O^*|\Sigma^*(k_i)]$, for $k_i \in [0, 15]$. The attack is considered successful when $\prod_{i=1}^{n} P[O = o_i|\Sigma = \sigma_i(k_j)]$ is maximum for $k = 5$.

[2] Note that modelling the algorithmic noise as Gaussians is reasonable since they approximate the binomial behavior of the Hamming distance values.

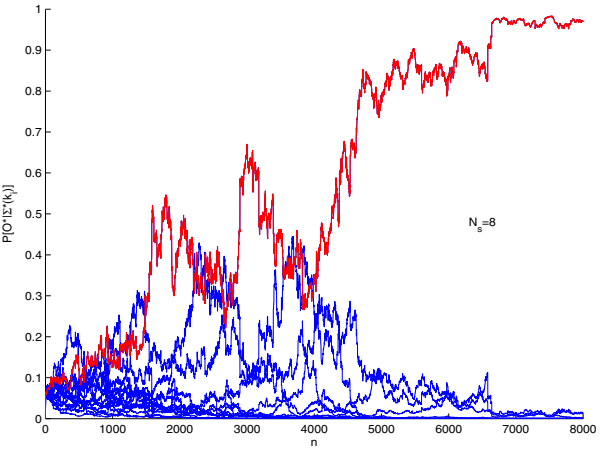

Fig. 6. A simulated higher-order attack with $N_s = 8$.

As a matter of fact, an attack against a single S-box scheme is nearly immediate: due to the discrete probabilities, a secret state such that $P[O = o_i|\Sigma = \sigma_j] = 0$ happens fast and only the correct key will remain with a non-zero probability after a few (in practice, less than 10) generated plaintexts. Much more relevant is the investigation of a simulated attack with different amounts of algorithmic noise in the design, *i.e.* different N_s values. A simulated attack with $N_s = 8$ is represented in Figure 6 and is successful after roughly 4000 generated texts. Other simulated attacks are in Appendix, Figures 8, 9, 10. From these figures, it is clear that the masked designs can be targeted by our attack with reasonable resources (*e.g.* less than 25 000 measurements), even if algorithmic noise is inserted. For comparison purposes, we also simulated first-order correlation attacks (like the ones in [16,20]) against the unprotected design of Figure 1, with the same parameters, *i.e.* same size and number of S-boxes. They are represented in Appendix, Figures 11, 12, 13, 14 and allow to measure the additional security provided by the masking. Comparisons will be discussed in the conclusions.

6 FPGA Results

We confirmed these simulated experiments with a real attack, against an FPGA implementation of the scheme in Figure 2, with $N_s = 8$ S-boxes[3]. Our target device was a Xilinx Spartan II FPGA [25] and the random mask values r_i's were generated with an on-chip LFSR.

Compared to simulated attacks, the main additional constraint was to correctly estimate the statistical characteristics (mean, variance) of the experimen-

[3] Due to area constraints, we did not target a standard algorithm such as the AES Rijndael. Indeed, as already mentioned, *e.g.* in [17,18], the hardware cost of masking a block cipher is a real concern for efficient hardware implementations.

tal signals. Indeed, in a real-world context, those values do not correspond to number of bit switches anymore, but to actual power consumption ones. That is, for example, the distance between the different gaussians in Figure 5 do not correspond to 2 bit switches anymore but to the power consumption of 2 bit switches. As a consequence, building the real pdf's $P[O|\Sigma = \sigma_j]$ from their discrete counterpart can be done with some steps and assumptions more than in Section 5.

First, we considered the measured observations to be a mixture of gaussians, as the one in the right part of Figure 5. We then assumed that the mean value of the observations E_{meas} corresponded to the mean value of the gaussian mixture.

Second, we had to determine the distance between the different gaussians, *i.e.* we had to evaluate the mean power consumption of one single bit switch in the design, denoted as m. For this purpose, we implemented a wide register inside our target FPGA and we measured the power consumption for different numbers of switching bits. Then, we derived the mean value of each gaussian in the mixture, denoted as E_i, $i \in [1, 2N_b + 1]$. It yields $E_{i+1} - E_i = m$, $\forall i \in [1, 2N_b]$.

Finally, we had to evaluate the variance of the gaussians. We assumed that all of them have the same value v, as in Section 5. Then we measured the variance of the measured observations V_{meas}, corresponding to the variance of the gaussian mixture. Knowing the different gaussian means E_i's from the previous experiments, we extracted v from [23]:

$$V_{meas} = \sum_{i=1}^{x} w_i * (v + E_i^2) - E_{meas}^2$$

where x is the total number of gaussians in the mixture (*i.e.* $2N_b + 1$), and the w_i's are weights depending on the probability of apparition of each gaussian.

We built the real pdf's for $P[O|\Sigma = \sigma_j]$ from these values and mounted a practical attack. It is represented in Figure 7, where we observe that the correct key is distinguishable after roughly 12 000 generated texts. As usually observed in side-channel attacks, practical experiments require significantly more samples than predicted because of noise and model imperfections. Still, the masked countermeasure was defeated in a remarkably low number of measurements.

In practice, we note that the attack is very sensitive to the correct evaluation of the signal mean values, for which imprecisions may lead to the selection of a wrong key candidate. The signal variance in itself does not affect the attack result, but its good evaluation allows the correct key candidate to be faster distinguished. For this purpose, we generally used a slightly larger value than the estimated v, *e.g.* $2v$ or $4v$.

Remark that the parameters m and v where naively estimated using a second programmable device. However, we also evaluated them successfully using an approach based on machine learning methods [11]. In practice, the presented attack is possible even if a second device is not available.

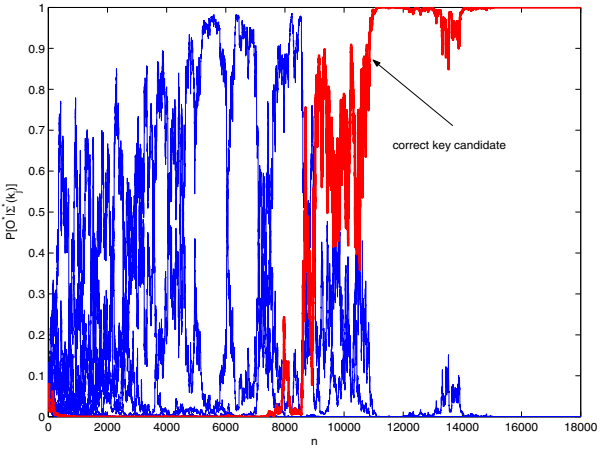

Fig. 7. A real attack against a masked FPGA design with $N_s = 8$

7 Conclusions

We proposed an improved higher-order technique to bypass the masking countermeasure. As a main result, it is demonstrated that such a countermeasure is not sufficient to protect an implementation from knowledgeable side-channel attackers. In practice, we recovered the secret key of a masked block cipher FPGA implementation in a low (*i.e.* practically tractable) number of measurements. We point out the following concluding remarks:

1. The attack was successfully applied to a parallel FPGA implementation which usually appears to be a challenging target for side-channel attacks. However, it could be straightforwardly applied to other devices, *e.g.* microprocessors. In such contexts, the algorithmic noise is usually reduced (due to the size of the buses, limited to 8 or 32 bits). For example, we estimated that an attack against an 8-bit processor would be successful after roughly 50 simulated measurements.
2. The presented attack is most fairly compared to the ones in [21,24]. For example, [21] considers a similar FPGA implementation to ours and targets a single S-box scheme (*i.e.* $N_s = 1$) in approximately 130 000 measurements. We target a $N_s = 8$ scheme in 12 000 traces.
3. Reference [13] presents experiments allowing a secret key to be recovered from a smart card implementation of the scheme in Figure 2, in about 2500 measurements. However, this attack is based on a Hamming weight power consumption model. It also requires access to the power consumption of the random mask and masked data separately, which involves these values to be computed sequentially. As the target is an 8-bit processor, it should be compared to an attack against a single S-box scheme, for which we would be successful in roughly 50 simulated measurements.

4. Compared to unprotected designs targeted by, *e.g.* a Correlation Power Analysis, a higher-order attack against a masked design still requires more traces. However, the gap between both attacks has been significantly reduced. In practice, the required number of measurements for a successful attack is not unrealistic anymore, even if large hardware implementations are considered. Note that the implementation cost of such large masked designs is another serious drawback, as mentioned in [17,18].
5. An open question is to know how much does the addition of noise affect a higher-order attack and how does it exactly compare to a first-order attack.
6. As a possible improvement, we finally suggest that a better estimation of the statistical distributions of the power consumption in the design is worth investigating. For example, the use of machine learning methods could be considered in this respect, as suggested in the previous section.

Acknowledgements. The authors would like to thank Cédric Archambeau for useful comments on previous versions of this paper.

References

1. M.L. Akkar, C. Giraud, *An Implementation of DES and AES Secure againts Some Attacks*, in the proceedings of CHES 2001, Lecture Notes in Computer Sciences, vol 2162, pp 309-318, Paris, France, May 2001, Springer-Verlag.
2. D. Agrawal, J.R. Rao, P. Rohatgi *Multi-channel Attacks*, in the proceedings of CHES 2003, Lecture Notes in Computer Sciences, vol 2779, pp 2-16, Cologne, Germany, September 2003, Springer-Verlag.
3. R. Anderson, E. Biham, L. Knudsen, *Serpent: A Flexible Block Cipher With Maximum Assurance*, in the proceedings of The First Advanced Encryption Standard Candidate Conference, Ventura, California, USA, August 1998.
4. P. Barreto, V. Rijmen, *The KHAZAD Legacy-Level Block Cipher*, Submission to NESSIE project, available from http://www.cosic.esat.kuleuven.ac.be/nessie/.
5. E. Brier, C. Clavier, F. Olivier, *Correlation Power Analysis with a Leakage Model*, in the proceedings of CHES 2004, Lecture Notes in Computer Science, vol 3156, pp 16-29, Boston, USA, August 2004.
6. S. Chari C. Jutla, J. Rao, P. Rohatgi, *Towards Sound Approaches to Counteract Power-Analysis Attacks*, in the proceedings of Crypto 1999, Lecture Notes in Computer Science, vol 1666, pp 398-412, Santa Barbara, California, USA, August 1999, Springer-Verlag.
7. L. Goubin, J. Patarin, *DES and Differential Power Analysis*, in the proceedings of CHES 1999, Lecture Notes in Computer Science, vol 1717, pp 158-172, Worcester, Massachussets, USA, August 1999, Springer-Verlag.
8. C. Karlof, D. Wagner, *Hidden Markov Model Cryptanalysis*, in the proceedings of CHES 2003, Lecture Notes in Computer Sciences, vol 2779, pp 17-30, Cologne, Germany, September 2003, Springer-Verlag.
9. P. Kocher, J. Jaffe, B. Jun, *Differential Power Analysis*, in the proceedings of CRYPTO 99, Lecture Notes in Computer Science, vol 1666, pp 398-412, Santa Barbara, USA, August 1999, Springer-Verlag.
10. S.Mangard, *Side-Channel Leakage of Masked CMOS Gates*, in the proceedings of CT-RSA 05, Lecture Notes in Computer Science, vol 3376, pp 351-365, San Fransisco, CA, USA, February 2005.

11. G. J. McLachlan and D. Peel, *Finite Mixture Models.* John Willey and Sons, New York, NY, 2000.
12. T.S. Messerges, E.A. Dabbish, R.H. Sloan, *Examining Smart-Card Security under the Threat of Power Analysis Attacks*, IEEE Transactions on Computers, vol 51, num 5, pp 541-552, May 2002.
13. T.S. Messerges, *Using Second-Order Power Analysis to Attack DPA Resistant Software*, in the proceedings of CHES 2000, Lecture Notes in Computer Sciences, vol 1965, pp 71-77, Worcester, Massachusetts, USA, August 2000, Springer-Verlag.
14. National Bureau of Standards, *FIPS PUB 46, The Data Encryption Standard*, Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, Jan 1977.
15. National Bureau of Standards, *FIPS 197, Advanced Encryption Standard,* Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, November 2001.
16. S.B. Ors, F. Gurkaynak, E. Oswald, B. Preneel *Power-Analysis Attack on an ASIC AES implementation*, in the proceedings of ITCC 2004, Las Vegas, April 5-7 2004.
17. E. Oswald, S. Mangard, N. Pramstaller, *Secure and Efficient Masking of AES - A Mission Impossible?*, IACR e-print archive 2004/134, http://eprint.iacr.org, 2004.
18. E. Oswald, S. Mangard, N. Pramstaller, V. Rijmen, *A Side-Channel Analysis Description of the AES S-box*, in the proceedings of FSE 2005.
19. J.M. Rabaey, *Digital Integrated Circuits*, Prentice Hall International, 1996.
20. F.-X. Standaert, S.B. Ors, B. Preneel, *Power Analysis of an FPGA Implementation of Rijndael: is Pipelining a DPA Countermeasure?*, in the proceedings of CHES 2004, Lecture Notes in Computer Science, vol 3156, pp 30-44, Boston, USA, August 2004.
21. F.-X. Standart, E. Peeters, J.-J. Quisquater, *On the Masking Countermeasure and Higher-Order Power Analysis Attacks*, in the proceedings of ITCC 2005 (vol 1), pp 562-567, Las Vegas, USA, April 2005.
22. F.-X. Standart, E. Peeters, G. Rouvroy, J.-J. Quisquater, *Power Analysis Attacks and Countermeasures of Field Programmable Gate Arrays: a Survey*, to appear in the Proceedings of the IEEE, special issue on Cryptographic Hardware and Embedded Systems, August 2005.
23. L. Trailovic, L.Y. Pao, *Variance Estimation and Ranking of Gaussian Mixture Distributions in Target Tracking Applications*, in the proceedings of the IEEE Conference on Decision and Control, pp 2195-2201, Las Vegas, NV, December 2002.
24. J. Waddle, D. Wagner, *Towards Efficient Second-Order Power Analysis*, in the proceedings of CHES 2004, Lecture Notes in Computer Science, vol 3156, pp 1-15, Boston, USA, August 2004.
25. Xilinx: *Spartan 2.5V Field Programmable Gate Arrays Data Sheet*, http://www.xilinx.com.

A Other Simulated Attacks

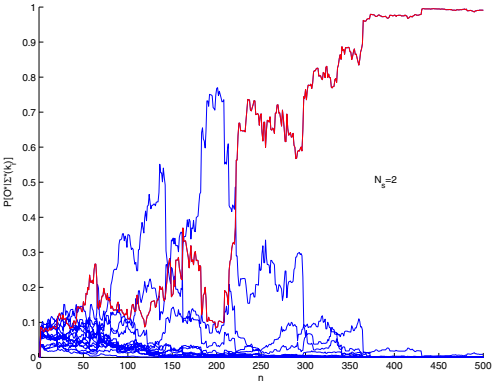

Fig. 8. A simulated higher-order attack with $N_s = 2$

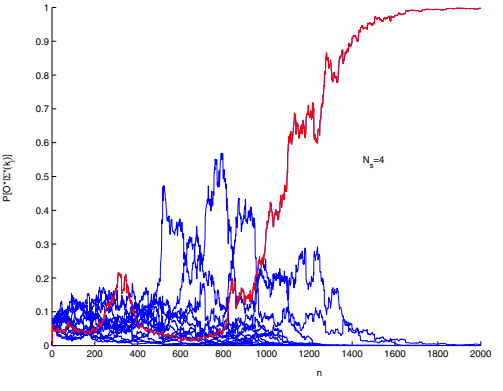

Fig. 9. A simulated higher-order attack with $N_s = 4$

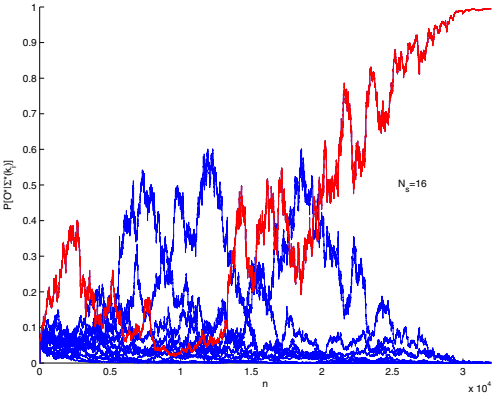

Fig. 10. A simulated higher-order attack with $N_s = 16$

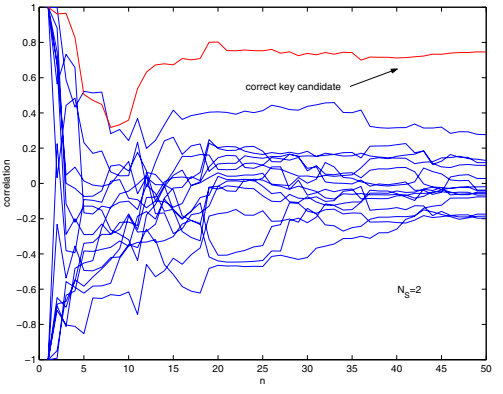

Fig. 11. A simulated correlation attack with $N_s = 2$

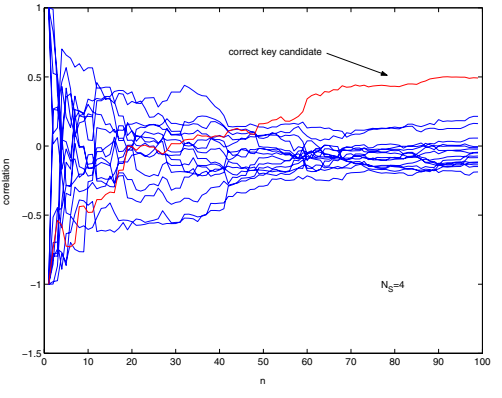

Fig. 12. A simulated correlation attack with $N_s = 4$

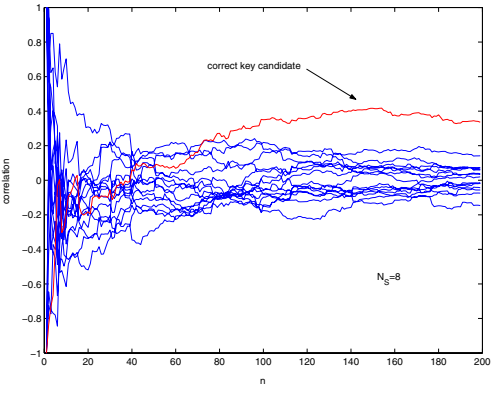

Fig. 13. A simulated correlation attack with $N_s = 8$

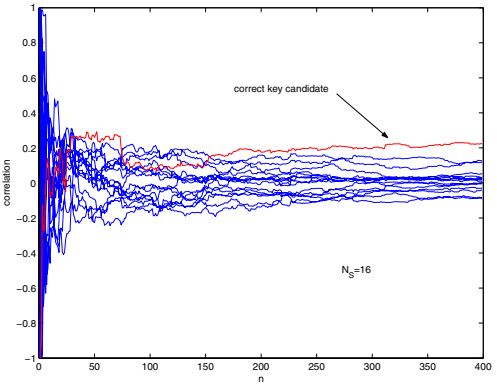

Fig. 14. A simulated correlation attack with $N_s = 16$