On the Masking Countermeasure and Higher-Order Power Analysis Attacks

François-Xavier Standaert, Eric Peeters, Jean-Jacques Quisquater

UCL Crypto Group, Place du Levant, 3, B-1348 Louvain-La-Neuve, Belgium.
e-mail: standaert,peeters,quisquater@dice.ucl.ac.be

***Abstract*— Masking is a general method used to thwart Differential Power Analysis, in which all the intermediate data inside an implementation are XORed with random Boolean values. As a consequence, the power consumption of the running implementation becomes unpredictable, making first-order power analysis attacks unpractical. Several recent works have shown that such protected designs are still susceptible to higher-order power analysis attacks. In this paper, we propose an extension of the previously introduced higher-order techniques, based on a more general power consumption model, and evaluate its actual feasibility. In particular, we discuss the number of power traces required to mount successful attacks. We also illustrate how this number is affected by parallel computations, making certain implementation contexts (*e.g.* smart cards, 8-bit processors) more susceptible than others (*e.g.* FPGAs, ASICs).**

I. Introduction

Since their publication in 1998 [8], power analysis attacks have attracted significant attention within the cryptographic community. Although less general than classical cryptanalysis, because they usually target one specific circuit or implementation, they have been particularly efficient to break a wide variety of devices, including smart cards, ASICs and FPGAs [10], [14], [19]. As a straightforward consequence, countermeasures against these attacks are of great practical interest.

Regarding the open literature, the masking technique is among the most popular suggested ways to protect an implementation against power analysis. However, several works have shown that such protected devices are still susceptible to higher-order power analysis. In particular, a recent advance [20] suggested that higher-order attacks are possible, without any additional hypothesis than usually assumed for first-order attacks. In this paper, we intend to complement this work and discuss some practical issues for the implementation of the attack. More precisely, we relate [20] with a more general power consumption model. We also propose an improvement of the technique that can be viewed as the higher-order counterpart of "multiple-bit" Differential Power Analysis [10] or Correlation Power Analysis [4]. In practice, we questioned the security of a masked block cipher hardware design and, using the formalism of attacks introduced in [19], we evaluated the attack feasibility in different implementation contexts.

The rest of the paper is structured as follows. A brief summary of the masking countermeasure and first-order power analysis attacks is given in Sections II and III. Section IV presents an intuitive description of the higher-order techniques and Section V describes our proposed improvements. Finally, Section VI presents experiments that confirms our descriptions and conclusions are in Section VII.

II. Masking countermeasure

The idea of masking the intermediate values inside a cryptographic algorithm has been suggested in several papers [1], [5], [7] as a possible countermeasure to power analysis attacks. 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 [12] or AES [13]. 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 [9] notably demonstrates that the glitching activity of masked logic gates offers a previously neglected leakage that seriously affects the security of the countermeasure. For this reason, this paper will mainly discuss the algorithmic level protection.

In the following sections, we question the security of the masking countermeasure with respect to higher-order power analysis attacks, originally described in [11]. For this purpose, we start by giving a simple description of our target implementations. First, an unmasked block cipher design is represented in Figure 1, where the b_is represents a known

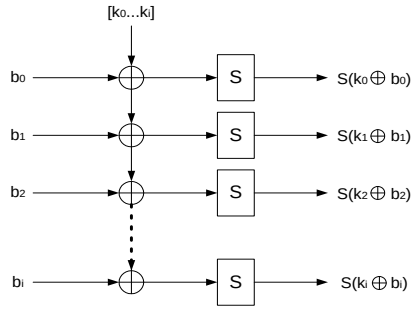

Fig. 1. Unprotected scheme.

input value, the k_is are the secret encryption key bits and the S blocks are non-linear substitution boxes. Regarding the structure of most present block ciphers [2], [3], [12], [13], we do 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.

Secondly, our protected implementation is represented in Figure 2, where the grey boxes actually suggest that the operations are applied in parallel to different data blocks, as in Figure 1. 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.

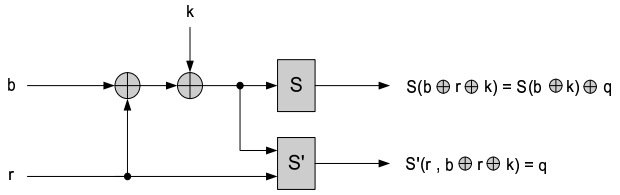

Fig. 2. Masked scheme.

III. 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 [16]:

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

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 $W_H(x \oplus x')$.*

This hypothesis is generally true for any CMOS circuit. 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. The difference between both hypotheses (*i.e.* Hamming distance model, Hamming weight model) will be emphasized in Section V.

IV. First-order power analysis attacks

Power analysis attacks usually hold in three steps.

First, the attacker *predicts* the power consumption of the running device, at one specific instant, in function of certain secret key bits. For example, let us assume that the values in Figure 1 are 4-bit wide. Then, the attacker could easily predict the value of $W_H(S(b_0 \oplus k_0) \oplus S(b'_0 \oplus k_0))$, for the 2^4 possible values of k_0 and N different input texts. According to the previous hypothesis, this gives 2^4 possible predictions of the device power consumption[1]. The result of this prediction is stored in the **selected prediction matrix**.

$b \oplus k$	r	$r \oplus b \oplus k$	P	P^2
0	0	0	$2P_0$	$4P_0^2$
0	1	1	$2P_1$	$4P_1^2$
1	0	1	$P_0 + P_1$	$(P_0 + P_1)^2$
1	1	0	$P_0 + P_1$	$(P_0 + P_1)^2$

TABLE I
An illustrative higher-order attack.

Secondly, the attacker *measures* the power consumption of the running device, at the specific time where it processes the same input texts as during the prediction phase. The result of this measurement is stored in the **global consumption vector**.

Finally, the attacker *compares* the different predictions with the real, measured power consumption, *e.g.* using the correlation coefficient [4]. If the attack is successful, it is expected that only the correct key guess will lead to a high correlation value[2]. Such attacks have been successfully applied to a variety of algorithms and implementations.

Remark that if the key is know, it is then possible to predict the power consumption (*i.e.* the transitions) of the complete design, not only the one of a single target S-box. As this prediction represents a theoretical noise-free measurement of the power consumption, it is a very convenient tool to evaluate the attacks with simulated data. We denote these transitions in the whole design as the **global prediction vector**.

From these descriptions, it is clear that the scheme of Figure 2 is not susceptible to first-order power analysis attacks because, *assuming that the mask is randomly updated for every encryption*, the power consumption depends both on the key and the unknown random mask and is therefore not predictable. In the next section, we show how higher-order techniques can overcome this problem.

V. Higher-order attacks: intuitive description

For clarity purposes, we first observe the single bit experiment summarized in Table I, where it is assumed that the mask and masked data are processed in parallel (as it is usually the case in hardware) and that the power consumption P depends on the Hamming weight of the data, although this latter hypothesis does NOT hold in a general hardware context. Remark that the actual hardware only processes the mask r and the masked data $r \oplus b \oplus k$ and only these values influence the power consumption. As already mentioned, a first-order power analysis is not possible because the power consumption is not predictable in function of the key.

[1] As mentioned in Reference [19], it is mandatory to predict the transitions after some non-linear component of the block cipher. This allows the power consumption predictions to be key-dependent.

[2] Other strategies can be considered for the comparison, but the use of correlation values is among the most efficient techniques. Moreover, changing the comparison tool will not result in different conclusions for the remaining sections of the paper.

$b \oplus k$	r	$b \oplus r \oplus k$	P
0→0	0→0	0→0	$2P_U$
0→0	0→1	0→1	$2P_S$
0→0	1→0	1→0	$2P_S$
0→0	1→1	1→1	$2P_U$
0→1	0→0	0→1	$P_U + P_S$
0→1	0→1	0→0	$P_S + P_U$
0→1	1→0	1→1	$P_S + P_U$
0→1	1→1	1→0	$P_U + P_S$
1→0	0→0	1→0	$P_U + P_S$
1→0	0→1	1→1	$P_S + P_U$
1→0	1→0	0→0	$P_S + P_U$
1→0	1→1	0→1	$P_U + P_S$
1→1	0→0	1→1	$2P_U$
1→1	0→1	1→0	$2P_S$
1→1	1→0	0→1	$2P_S$
1→1	1→1	0→0	$2P_U$

TABLE II
AN HIGHER-ORDER ATTACK WITH A GENERAL HYPOTHESIS.

The objective of a higher-order power analysis attack is therefore to find a function of the power consumption which is independent of the mask values, but still depends on the key. A simple solution to this problem, suggested in [20], is to average the square power consumptions of Table I:

$$\sum_{b \oplus k=0} P^2 = 4(P_0^2 + P_1^2)$$
$$\sum_{b \oplus k=1} P^2 = 2(P_0 + P_1)^2$$

According to the key value, it is then possible to distinguish two power consumption classes and therefore to mount a higher-order power analysis attack. Improvements of this basic scheme will be discussed in the following section. However, we will first repeat the same experiment in our general power consumption model (*i.e.* Hamming distance based).

The corrected experiment is illustrated in Table II, where P_U denotes the power consumption of a $0 \rightarrow 0$ or $1 \rightarrow 1$ transition and P_S denotes the power consumption of a $0 \rightarrow 1$ or $1 \rightarrow 0$ transition. This experiment clearly suggests that we have two power consumption classes (separated by double lines in the table): one corresponding to a stable $b \oplus k$ value, the other corresponding to a switching $b \oplus k$. However, regarding the key dependencies, it is clear that changing the key bit will not influence the power consumption class (*i.e.* switching or not). Therefore, the average square functions considered previously will not have the required key dependent behavior, making such higher-order attacks impossible in the Hamming-distance based power consumption model.

VI. EXTENDED HIGHER-ORDER ATTACK

As it has already been suggested, a first-order power analysis in the Hamming distance based power consumption model requires to predict the power consumption after a non-linear component in order to obtain key-dependent predictions of the power consumption. It is therefore natural to extend such ideas to higher-order techniques. The following proposal also intends to make a better use of the power consumption measurements, using an adapted correlation method.

We start with a few definitions. Let s be the bit size of the target substitution box. That is, the S box has size $2^s \times s$ and the precomputed table S' has size $2^{2s} \times s$. Let k be the number of key guesses. Usually, the key guess is performed around one S-box and we have $k = 2^s$.

As in Section IV, the attack holds in three steps.

Precomputation: The attacker starts by computing a table containing, for every possible key guess g and every possible input transition (b switches into b'), the average of the squared number of bit-transitions at the target S-boxes (S and S') outputs. According to the hypothesis of Section III, this latter value represents a squared power consumption prediction. The average is performed over all the possible mask transitions. In pseudo-C, we have:

Algorithm 1 Precomputation

```
(1) for key_guess g = 0...k − 1
  (2) for first_input b = 0...2^s − 1
    (3) for second_input b' = 0...2^s − 1
      sum = 0;
      (4) for first_input_mask r = 0...2^s − 1
        (5) for second_input_mask r = 0...2^s − 1

            Predict and average the square of the power consumption for
            key_guess g, input switch b → b', mask switch r → r' and
            target S-boxes S and S':

      sum = sum + ( W_H( S(b ⊕ r ⊕ k) ⊕ S(b' ⊕ r' ⊕ k) )
                  + W_H( S'(r, b ⊕ r ⊕ k) ⊕ S'(r', b' ⊕ r' ⊕ k) ) )^2

        end (5);
      end (4);
      precomputation[g,b→b']=sum/2^{2s};
    end (3);
  end (2);
end (1);
```

The result of this precomputation phase is stored in a $2^{2s} \times k$ **precomputation matrix**. Remark that not all the input transitions are interesting for the attacker. In particular, transitions such that $b = b'$ will not have a key dependent behavior. As there are 2^n such transitions, the precomputation matrix actually contains $2^{2n} - 2^n$ useful rows. Note also that the technique works similarly in a Hamming weight based power consumption model.

Measurement: During the measurement phase, the attacker computes exactly the same averages as during the precomputation phase, with two significant differences. First, the average is made on the real, measured, squared power consumptions. Secondly, as the mask transitions are unknown, one arbitrary sets the size of the sum to a fixed value, denoted as x. This coefficient is an important parameter of the attack and can be increased in case of noisy measurements. The result of the measurement phase is stored in a $2^{2s} \times 1$ **measurement vector** and this process is summarized in Algorithm 2.

Algorithm 2 Measurement

(1) **for** first_input $b = 0...2^s - 1$
  (2) **for** second_input $b' = 0...2^s - 1$
    $sum = 0$;
    (3) **for** $counter = 0...x - 1$

      Measure and average the square of the power consumption for input switch $b \rightarrow b'$ and target S-boxes S and S':

$$sum = sum + P(target\ device)^2$$

    end (3);
    measurement[b→b']=sum/x;
  end (2);
end (1);

Comparison: In the comparison phase, the attacker finally compares the different columns of the precomputation matrix, corresponding to the different key guesses, with the measurement vector. As in the context of first-order power analysis attacks, an efficient tool to perform this comparison is the correlation coefficient. If the attack is successful, it is expected that only the correct key guess will lead to a high correlation value.

It is important to remark that a fundamental difference between first-order and higher-order attacks is in the size of the measurement vector[3]. While this vector can be made arbitrary long in first-order attacks, by simply increasing the number of input texts, it has a maximum size of 2^{2s} in the higher-order context. Note also that, compared with the previously published attacks, our proposal allows to take advantage of all the available information, *i.e.* all the possible input transitions are taken into account. The next section confirms these descriptions with experimental results.

VII. Experimental results

Looking back at the first-order power analysis described in Section IV, it is clear that the attack efficiency, *i.e.* the number of required measurements to recover the key, depends on the correlation between the power consumption predictions and the real measurements. To illustrate this statement, we assume that our block cipher is implemented in hardware, so that every S-box, XOR, ... use different resources in the circuit. In practice, there are two kinds of noise that affect this correlation value. If we first consider an attack using simulated data, the signal is represented by the transitions of the predictable target S-box, while the noise is represented by the transitions of the other, unpredictable S-boxes. Increasing the number of S-boxes in the design will consequently increase the noise and decrease the correlation values. We denote this first noise as the algorithmic noise. Then, considering the fact that actual measurements are not perfect, a practical attack is also affected by a measurement noise. 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. Therefore, we will first evaluate the feasibility of a higher-order attack in this convenient simulated context.
If we now consider a higher-order attack, a third type of noise will affect the correlation, due to the presence of the random mask. As a simple illustration, let us consider an attack using simulated data, with no algorithmic noise. This means that the unmasked block cipher is made of one single S-box. In a first-order attack, the correlation between the correct prediction of power consumption and the simulated measurements will be perfect (*i.e.* equals to one). However, if the S-box is masked, a higher-order attack will still require an averaging process to obtain sufficient correlation values.

In the following sections, we illustrate these comments in different contexts. For all the presented experiments, the number of S-boxes in the block cipher is denoted as N_S and the previously defined x value is used as a parameter. As the actual S-box, taken from the Serpent algorithm [2], is 4-bit wide, the actual number of measurements required for the attacks is $M = (2^{2s} - 2^s) \times x = 240 \times x$. The scheme under attack is the one of Figure 2.

A. An attack using simulated data with $N_S = 1$

In this simple experiment, the block cipher is actually reduced to one single S-box. It allows us to clearly observe the effect of the masking noise. In this simple context, a first-order power analysis with simulated data against a similar unmasked scheme would be immediately successful.

Ten experiments were performed, with the parameter x set to $1, 10, 100$ and 1000, and a correct key $k = 6$. They are represented in Figure 3, where each curve holds for one experiment. It is clearly observed that, while the correct key is not distinguishable without averaging (*i.e.* when $x = 1$), a small sum (*e.g.* $x = 10$) is already enough to recover certain keys and larger sums allow to reach very good correlation values (*i.e.* up to 0.9). The figure also illustrates that some key candidates are more correlated with the correct key guess than others, *e.g.* $k = 9$ in our example. This observation, due to the Boolean structure of the S-box, is similar to the well-known "ghost peak" problem in the open literature [4].

B. An attack using simulated data with $N_S = 8$

In this second experiment, we investigated the much more relevant context of an attack using simulated data against a masked block cipher containing several S-boxes. In practice, we fixed $N_S = 8$. Ten experiments were again performed, using different averaging values: $x \in \{10, 100, 1000, 10\ 000\}$.

From Figure 4, we can conclude that the algorithmic noise produced by a 32-bit block cipher composed of eight 4×4 S-boxes seriously affects the attack efficiency. Even in our simulated data context, it is necessary to set $x > 1000$ in order to distinguish certain correct key guesses, meaning that at least 240 000 measurements will be necessary. As a comparison, a simulated correlation power analysis attack against a similar unmasked block cipher with eight 4×4 S-boxes would be successful after about 50 plaintexts [19]!

[3]Denoted as global consumption vector in first-order attacks.

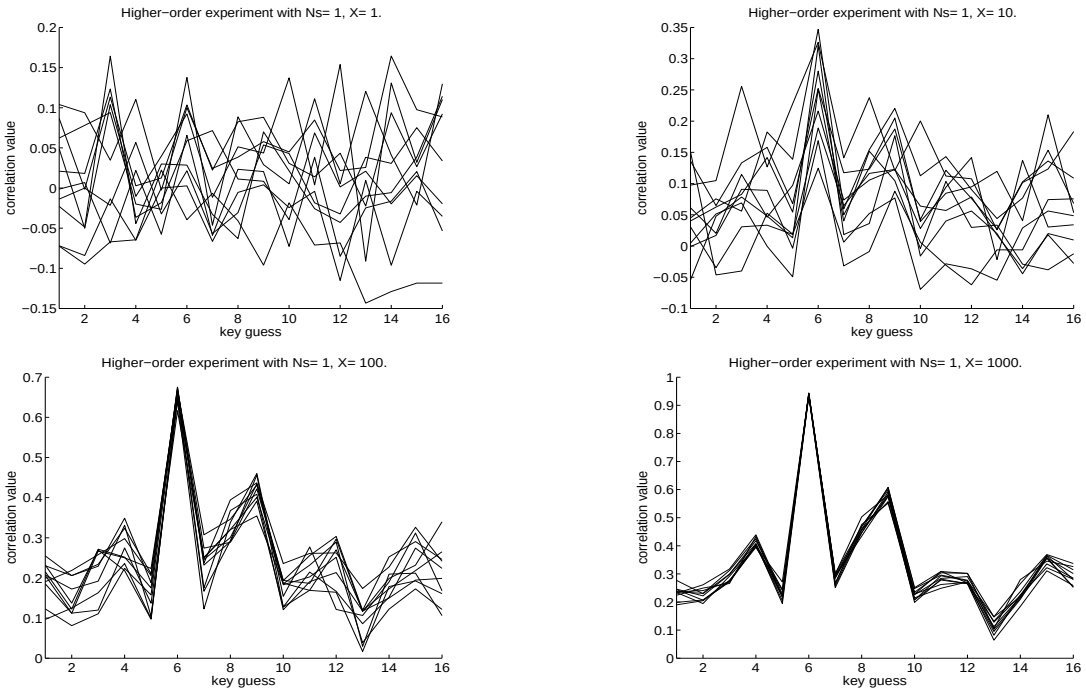

Fig. 3. Higher-order simulated experiments with $N_S = 1$.

C. Feasibility of practical attacks

The previous sections suggested that the actual efficiency of a higher-order power analysis attack depends on the amount of algorithmic noise in the targeted design. As a consequence, it is interesting to consider how this observation relates to certain specific implementation contexts.

First, regarding smart card and processors, it is important to remark that the algorithmic noise in these devices is limited by the size of the data buses. For example, the experiment of Section VII-A could be used to evaluate the resistance in an 8-bit processor, as it is assumed that only the 4-bit S-boxes S and S' are computed in parallel (*i.e.* a total of 8 bits). It underlines that a practical attack is easily feasible against such small masked devices, as far as the attacker measurements capabilities are sufficient (*e.g.* similar to the ones in [4]) and the actual implementation of the countermeasure is similar to the one in Figure 2. While more challenging, 32-bit processors could be defeated in exactly the same way[4].

The situation strongly differs in the context of hardware and FPGA implementations (*e.g.* the ones in [6], [17], [18]), as efficiency constraints often involve a large parallel computing (*i.e.* pipelining and unrolling) in these devices. In practice, the most compact implementations use a loop architecture, such that only one block cipher round is completely implemented on the circuit. However, looking at present block cipher sizes (*i.e.* at least 64-bit and mostly 128-bit), such designs already seriously affects the attack feasibility. As an illustration, a successful simulated attack against a loop architecture of the 64-bit block cipher Khazad [3] requires approximately 6 million measurements. Unrolling more than one block cipher round would similarly mean to multiply the algorithmic noise. Considering the fact that actual measurements of FPGAs or ASICs are usually worse than those of smart cards [19], we can therefore conclude that in these latter contexts, the masking countermeasure substantially improves the security *against the presented attack*[5].

To confirm these predictions, we implemented the single S-box scheme of Section VII-A in a Xilinx Spartan FPGA [21]. Our measurement setup allowed us to obtain a correlation between theoretical predictions of the power consumption and actual measurements of about 0.4. In practice, we recovered the correct key after 131 072 plaintexts. This experiment also underlined the strong influence of measurement noise in actual attacks, as already observed in [19].

Finally, comparing our results with previous publications in the field requires to correctly understand their context differences. In particular, reference [11] presented experiments allowing to recover a secret key from a smart card implementation of the scheme of Figure 2, in about 2500 measurements. However, this attack is based on a Hamming weight power consumption model. It also requires to access the power consumption of the random mask and masked data separately, which involves these values to be computed sequentially. On the contrary, our results use a more general Hamming distance based power consumption model and apply to all hardware architectures, including pipelined (or parallel) ones.

[4]Remark that these comments mainly relate the resistance of smart cards or processors to their bus size, although other features may influence the actual security of such devices (*e.g.* precharging buses with random values could make the attack more difficult). For this reason, these observations should be taken with care. In general, our model more directly applies to hardware devices, as it is investigated in the rest of this section.

[5]Which does not prove security against other possible attacks.

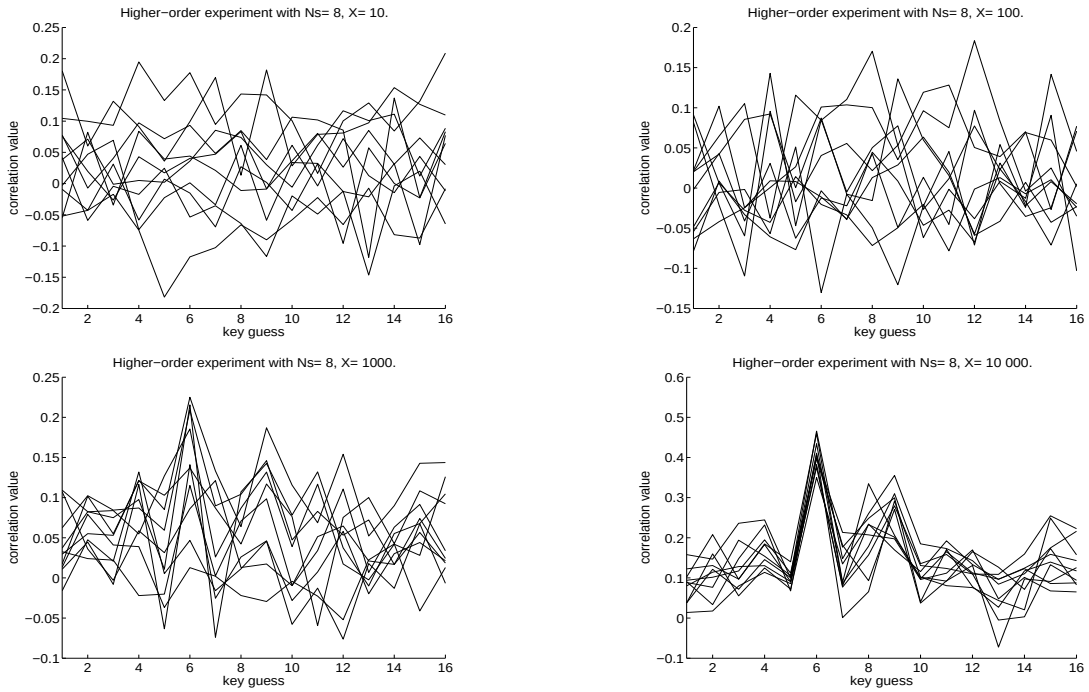

Fig. 4. Higher-order simulated experiments with $N_S = 8$.

VIII. Conclusions

This paper investigated the resistance of the masking countermeasure against higher-order power analysis. It is first proposed to extend the previously published higher-order techniques in order to correctly fit to a realistic power consumption model, but also to make a better use of the available leakages. Then, we discussed the efficiency of a practical attack and argued that the implementation context has a significant impact on the attack feasibility. In particular, while attacking a protected 8-bit processor or smart card is feasible with reasonable measurement capabilities, large hardware designs implemented on ASICs or FPGAs are shown to be more resistant.

Regarding the results presented in this paper, there are different directions for further research. A first one would be the theoretical analysis of these techniques, including a statistical prediction of the success rate in function of different parameters (measurement capabilities, target design, S-box sizes, ...). In practice, the improvement and understanding of the power consumption measurements and models is also of interest. Cryptanalytic efforts to build, evaluate and possibly defeat countermeasures to side-channel attacks is still required as well. Remark finally that the cost of masking has not been discussed in the paper, although it is another critical point to investigate, *e.g.* for the AES Rijndael [15].

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] 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.

[3] P. Barreto, V. Rijmen, *The KHAZAD Legacy-Level Block Cipher*, Submission to NESSIE project, available from http://www.cosic.esat.kuleuven.ac.be/nessie/.

[4] 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.

[5] 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.

[6] K. Gaj, P. Chodowiec, *Fast Implementation and fair Comparison of the Final Candidates for the Advanced Encryption Standard using Field Programmable Gate Arrays*, in the proceedings of the RSA Security Conference - Cryptographer's Track, San Francisco, CA, April 8-12, 2001, pp. 84-99.

[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] 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.

[9] S.Mangard, *Why the Masking of CMOS Gates Does Not Prevent DPA Attacks?*, to appear in the proceedings of CT-RSA 05.

[10] 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.

[11] 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.

[12] National Bureau of Standards, *FIPS PUB 46, The Data Encryption Standard*, Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, Jan 1977.

[13] National Bureau of Standards, *FIPS 197, Advanced Encryption Standard*, Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, November 2001.

[14] 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.

[15] 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.

[16] J.M. Rabaey, *Digital Integrated Circuits*, Prentice Hall International, 1996.

[17] G.Rouvroy, F.-X.Standaert, J.-J.Quisquater, J.-D.Legat, *Design Strategies and Modifed Descriptions to Optimize Cipher FPGA Implementations: Fast and Compact Results for DES and Triple-DES*, in the proceedings of FPL 2003, Lecture Notes in Computer Science, vol 2778, pp 181-193, Lisbon, Portugal, September 2003, Springer-Verlag.

[18] F.-X. Standaert, G. Rouvroy, J.-J. Quisquater, J.-D. Legat, *Efficient FPGA Implementation of Block Ciphers Khazad and Misty1*, 3^{rd} NESSIE Workshop, Munich, Germany, November 2002.

[19] 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.

[20] 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.

[21] Xilinx: *Spartan 2.5V Field Programmable Gate Arrays Data Sheet*, http://www.xilinx.com.