Updates on the Security of FPGAs against Power Analysis Attacks

F.-X. Standaert*, F. Mace, E. Peeters, J.-J. Quisquater
UCL Crypto Group, Place du Levant 3, B-1348 Louvain-la-Neuve, Belgium
e-mails: fstandae,mace,peeters,quisquater@dice.ucl.ac.be

Abstract. This paper reports on the security of cryptographic algorithms implemented on FPGAs against power analysis attacks. We first present some improved experiments against these reconfigurable devices, due to an improved measurement process. Although it is usually believed that FPGAs are noisy targets for such attacks, it is shown that simple power consumption models can nearly perfectly correlate with actual measurements. Then, we evaluate how these correlation values depend on the resources used in the FPGAs. Finally, we investigate the possibility to counteract these attacks by using random pre-charges in the devices and determine how this technique allows a designer to increase the security of an implementation. These results confirm that side-channel attacks present a serious threat for most microelectronic devices, including FPGAs. To conclude, we discuss the security *vs.* efficiency tradeoffs.

1 Introduction

Hardware designs are usually evaluated within an area-time implementation space. However, in the context of cryptographic implementations, the efficiency is not the only metric by which one can measure an implementation's quality. In particular, the physical security of microelectronic circuits has recently attracted a lot of attention. While originally applied to small devices like smart cards, certain attacks have recently been shown quite efficient to defeat FPGA implementations as well (*e.g.* [10, 14]). As an illustration, in this paper, we consider the resistance of FPGA implementations against power analysis attacks and update certain assumptions on their actual security.

In these attacks, an adversary uses a hypothetical model of a target device in order to predict its power consumption. The predictions are then compared to the real, measured power consumption in order to recover secret information. Therefore, the better a power consumption model can correlate with actual measurements, the more efficient the resulting attack is. In this context, previously published results against FPGA devices suggested that these are challenging components to target with power analysis. Assumed reasons for this notably were (1) the difficulty of obtaining good power consumption measurements for FPGAs, (2) the possibility to perform parallel computing within these devices.

* François-Xavier Standaert is a post doctoral researcher funded by the FNRS (Funds for National Scientific Research, Belgium).

In this paper, we first suggest that, as far as the quality of the measurements is concerned, FPGAs do not significantly differ from small devices like smart cards. In particular, even very simple power consumption models based on the prediction of the number of bit transitions within a device can nearly perfectly correlate with actual measurements, if some simple signal processing is applied. In practice, we perform and evaluate some improved experimental correlation attacks against FPGA implementations of cryptographic algorithms. We also discuss how these attacks depend on the resources used in the devices.

In a second part of the paper, we investigate the possibility to counteract these attacks by using random pre-charges in the FPGAs and evaluate how this technique allows to increase the security of an implementation. In particular, as already observed in the context of smart cards, such a proposal makes it impossible to predict bit transitions. This is because one every two consecutive values in a device is then random and unknown. As a consequence, targeting such designs requires the use of more complex power consumption models (*e.g.* based on distinguishing $0 \rightarrow 1$ from $1 \rightarrow 0$ bit transitions), for which the correlations obtained are lower. We evaluate these correlation values in the paper.

The rest of the paper is structured as follows. Section 2 describes the principles of power analysis attacks. Section 3 evaluates the correlation obtained between a simple power consumption model based on the switching activity within an FPGA and actual measurements. Section 4 illustrates how these correlations depend on the resources used by a target design. Section 5 performs the same experiments if random prcharges are used within the FPGA. Section 6 discusses the resulting security *vs.* efficiency tradeoff and our conclusions are in Section 7.

2 Correlation Power Analysis Attacks

Power analysis attacks [6] generally require a hypothetical model of the device under attack to predict its power consumption. For example, FPGAs are usually made of CMOS gates, for which it is reasonable to assume that the main component of the power consumption is due to the switching activity. For a single CMOS gate, we can express it as follows [12]:

$$P_S = C_L V_{DD}^2 P_{0 \rightarrow 1} f \tag{1}$$

where C_L is the gate load capacitance, V_{DD} the supply voltage, $P_{0 \rightarrow 1}$ the probability of a $0 \rightarrow 1$ output transition and f the clock frequency. Equation (1) specifies that the power consumption of CMOS circuits is data-dependent. An attacker may consequently estimate a device power consumption at time t by the number of bit transitions inside the device at this time. Based on this simple observation, power analysis attacks have been applied to numerous algorithms and devices, including smart cards, ASICs and FPGAs. In practice, the use of secret key information in cryptographic designs only allows us to predict a part of the bit transitions, but it is sufficient to correlate with actual measurements of the power consumption.

We illustrate the attack principle (*e.g.* see [2]) with the simple encryption network of Figure 1, which contains the same basic elements as most present block ciphers *e.g.* the DES [7] and AES Rijndael [8]. That is, the plaintext is XORed with a secret key, then goes through a layer of relatively small substitution boxes and is finally sent to a larger permutation (*e.g.* a linear diffusion layer for the AES Rijndael). The same operations are iterated a number of times. For the purposes of this paper, it is not necessary to know more details on these algorithms. The attack proceeds as follows.

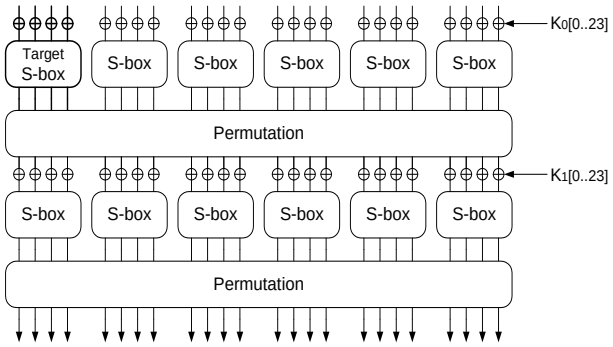

Fig. 1. A simple encryption network.

Let the adversary target the 4 key bits entering the left S-box of Figure 1, denoted as $K_0[0..3]$. Then, for N different plaintexts, he first predicts the number of transitions at the S-box output, for every possible value of $K_0[0..3]$. The result of this prediction is a $N \times 2^4$ selected prediction matrix $\mathbf{P}$, containing numbers between 0 and 4. For simulation purposes, it is also interesting to produce the global prediction matrix $\mathbf{G}$ that contains the number of bit transitions inside the whole design. This can of course not be computed by an actual adversary, but can be done if the secret key is known (*i.e.* when evaluating the attacks).

In the second part of the attack, the adversary let the circuit encrypt the same N plaintexts with a fixed key (the same as during the predictions if $\mathbf{G}$ was computed, a secret one in case of real attacks) and he measures the power consumption of the device while the chip is operating the targeted operation. This results in a $N \times 1$ measurement vector $\mathbf{M}$.

Finally, the attacker computes the correlation between the measurement vector and all the columns of the selected prediction matrix (corresponding to all the possible key guesses). If the attack is successful, it is expected that only one value, corresponding to the correct key bits, leads to a high correlation. An efficient way to compute the correlation is to use the Pearson coefficient that can be expressed as follows:

$$C(\mathbf{M}, \mathbf{P}) = \frac{\mu(\mathbf{M}.\mathbf{P}) - \mu(\mathbf{M}).\mu(\mathbf{P})}{\sqrt{\sigma^2(\mathbf{M}).\sigma^2(\mathbf{P})}} \tag{2}$$

In this expression, $\mu(\mathbf{M})$ denotes the mean of the set of measurements $\mathbf{M}$ and $\sigma^2(\mathbf{M})$ its variance. For a more detailed explanation of the power analysis attack principles, we refer to previous publications, *e.g.* [2, 14]. We note that different statistical tools could be considered to mount power analysis attacks and the use of the correlation coefficient is not optimal with this respect. For example, maximum likelihood techniques [4] may yield better results. However, with the simple power consumption models considered here, correlation attacks provide good results and are extremely easy to manipulate (*e.g.* they do not require any estimation of the noise in the target devices).

Finally, let us recall two simple formulas, proven in [15]. Firstly, the correlation coefficient we are interested in during an attack is the one between the selected predictions and the measurements. It can be rewritten as:

$$C(\mathbf{P}, \mathbf{M}) = C(\mathbf{P}, \mathbf{G}) \times C(\mathbf{G}, \mathbf{M}) \tag{3}$$

In this equation, the coefficient $C(\mathbf{G},\mathbf{M})$ only relates to the quality of the measurement and for example, is independent of the FPGA design considered. On the contrary, the coefficient $C(\mathbf{P},\mathbf{G})$ is specifically related to the implementation under attack and depends on the number of bit transitions that can actually be predicted. In our previous example, we did only predict the transitions of one target S-box, out of the 12 S-boxes in Figure 1. Secondly, the number of generated plaintexts N required to have a successful correlation attack is worth:

$$N = c \times \frac{1}{C(\mathbf{P}, \mathbf{M})^2}, \tag{4}$$

where c is a small constant value. In the following sections, we would like to answer the question: "How precisely can we correlate our simple power consumption models with actual measurements of the power consumption?"

3 Correlations measurements and consequences

Target designs: For all our experiments, we used the four target designs represented in Figure 2. They are again made of XOR operations, substitution boxes and diffusion layers. The three first designs loop on one iteration while the fourth one loops on two iterations. These designs also differ by their various number of pipeline stages. For simplicity purposes, we forced all the operations to be performed by one single layer of look up tables (LUTs) in the FPGA (*e.g.* we used the 4-bit substitution boxes of the Khazad block cipher [1] that perfectly fit to these constraints). Also, the potential leaking points, corresponding to the points in the design for which the transitions consume power, are denoted as *a,b,c* and *d* (and further letters for the fourth design). All the architectures are 128-bit wide. When the values *a,b,c,d,...* are stored in registers, and according to the terminology introduced in [14], the dark gray registers are *full* (meaning that their bit transitions are strongly correlated to the key values) while the

light gray ones are *empty* (meaning the opposite). Also, the small black boxes suggest that a part of the register can actually be predicted by an adversary, because it does only depend on a limited number of key bits. On the opposite, registers without black boxes cannot be predicted. This is typically the case of the registers after the diffusion layer.

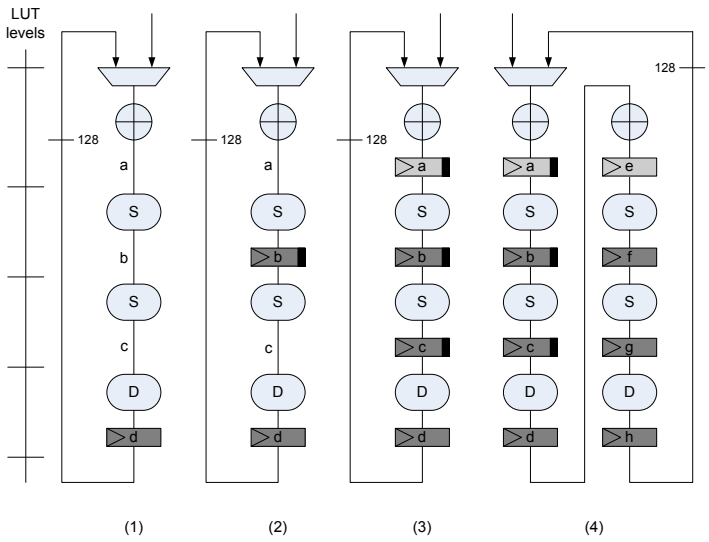

Fig. 2. Target designs.

Our measurements were performed on a Xilinx Spartan-2 device. Although the building of a good measurement setup is an important step in side-channel attacks, the technical description of such a setup is out of the scope of this work. We simply note that our approach was to use a dedicated board in order to isolate the FPGA from any other component, representing potential noise sources in the observations. It is important to have in mind that the following results highly depend on our measurement capabilities and the context considered. For example, targeting an FPGA on a prototyping board including various processors, memories, ... would be more challenging. On the other hand, the measurement process itself could still be improved and is under progress. Again, the objective of this paper is to suggest that basic methods can already yield good results.

Our initial strategy to evaluate the correlation coefficient $C(\mathbf{G},\mathbf{M})$ was the following. We considered the third design of Figure 2, with four pipeline stages. Then, we assumed that the leaking points a,b,c,d (all of them being stored in registers) contributed for a similar part of the power consumption and predicted the bit transitions in these registers. On a power trace like the one in the left part of Figure 3, we finally observed the peaks occurring at the rising edges of the clock signal and evaluated how the values of these peaks were correlated with the total number of bit transitions predicted.

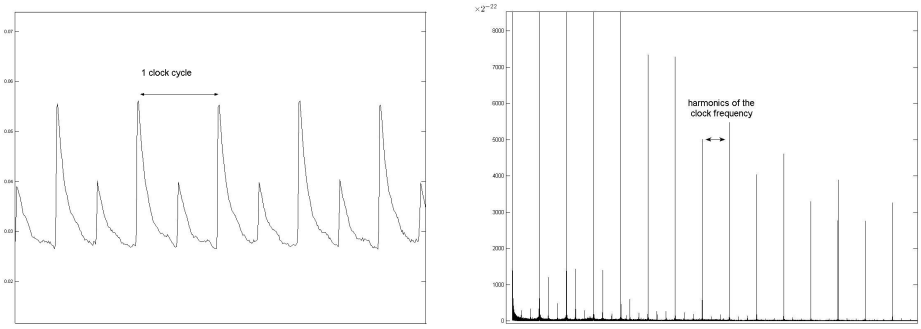

Fig. 3. A single power trace: time and frequency domains.

Two simple signal processing steps were applied. First, the spectrum of the power traces was observed (partially represented in Figure 3) and we identified a number of parasitic signals that were filtered with their harmonics. Second, we performed a small averaging on the filtered traces.

The correlations between our predictions and a single FPGA power trace, filtered or not, are represented in the left part of Figure 4, for different numbers of generated plaintexts. The correlations after averaging are in the right part of the figure. Roughly, we observe that a single trace allows to reach a correlation of up to 75% while a small averaging increases this value behind 90%. As a comparison, previously published results, *e.g.* [14] suggested correlation values of around 45%, roughly corresponding to our single non-filtered trace experiment. Therefore, referring to Equation 4, the number of required plaintexts to perform a successful attack would be divided by 4.

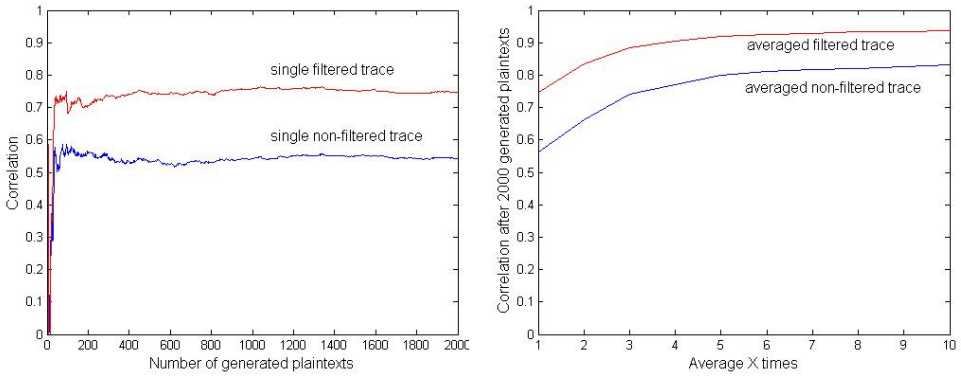

Fig. 4. Correlation between our predictions and actual power consumption measurements for the design (3), without random pre-charges: single trace and averaged traces.

4 Resources dependencies

In the previous experiment, we correlated the total number of bit transitions in the device registers with actual power consumption measurements. As already mentioned, this involves the important assumption that all the leaking points

contribute identically to the power consumption. Obviously, this may not be formally correct and the aim of this section is to evaluate how relevant is this assumption for practical applications. In particular, we would like to answer two questions: (1) do the use of registers in a design influence its power consumption? (2) how do the various FPGA resources contribute to the power consumption?

In order to answer the first question, we implemented the designs (1),(2) and (3) of Figure 2. Since they only differ in their number of pipeline stages, they actually require (roughly) the same number of LUTs and slices. The only difference is in their number of flip flops. Then, we measured the power consumption of these three architectures when feeded with the same inputs. We could not distinguish any significant difference between the power consumption patterns. The assumed reason for this observation is that the overall power consumed by an FPGA mainly depends on the amount of resources (*e.g.* the slices) used by a design, that is roughly the same in the three experiments.

To answer the second question, we used again the designs (1), (2) and (3) and evaluated separately the correlations between their power consumption measurements and the bit transitions for the leaking points *a,b,c,d*. It is illustrated for the design (1) in the left part of Figure 5 where we can clearly observe that the leaking points do not correlate the same way and therefore do not contribute for equal parts to the global power consumption. Then, to obtain better results, we used a weighted sum of the predictions of the different leaking points. It is represented in the right part of the figure where it is compared with a non-weighted sum (as we used in the previous section).

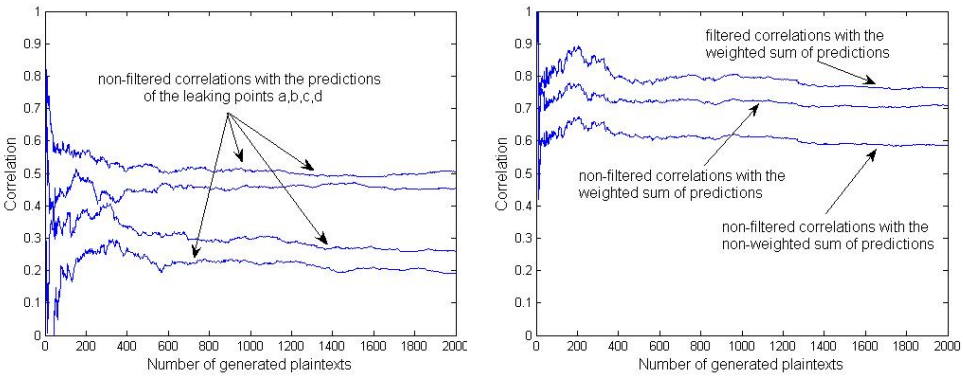

Fig. 5. Resource dependent correlations without random pre-charges.

These results confirm the experiments performed in [13] where it is shown that the dynamic power consumption in FPGAs actually depends on the effective capacitances of the resources used. For example, it is shown that the effective capacitances of signals within a slice are much lower than the ones of long connection wires. This could explain that the correlations with certain bit transitions appear to be much higher than others. It is also interesting to observe that fil-

tering the trace can again yield an even better correlation. This could be easily understood since filtering reduces the noise due to parasitic frequencies within the signal while the use of weighted predictions increases the quality of our leakage model. That is, both techniques relate to different noise sources.

From a practical point of view, it is important to have in mind that the use of weighted predictions involves a different attack context, usually denoted as template attacks [4]. Indeed, in the most general setting, an actual adversary will not be able to determine precisely which transitions in a design contribute the most to the power consumption. Therefore, the naive strategy (without attributing weights to the bit transitions) is the only one applicable. On the opposite, if the adversary can use a programmable device to build a better power consumption model (*i.e.* in the template attack context), improved strategies as the one presented in this section are applicable. Note that, in our example, we only used four different weights (*i.e.* for the *a,b,c,d* leaking points), although it would be possible to further improve the process by considering more different weights.

To further analyze these observations, we use the following lemma, also applied in [15]: *the correlation coefficient between the sum of n arbitrary independent identically distributed random variables and the sum of the first $m < n$ of these equals* $\sqrt{m/n}$. If we assume that various bit transitions in a design contribute additively to the global power consumption, it means that the four correlations in the left part of Figure 5 respectively correspond to $0.2^2 \simeq 4\%$, $0.25^2 \simeq 6\%$, $0.45^2 \simeq 20\%$ and $0.5^2 \simeq 25\%$ of the total power consumption. That is a sum of 55%. Such a prediction should allow a correlation of $\sqrt{0.55} \simeq 74\%$ which is close to the one observed in the right part of the figure, for the (non-filtered) weighted sum experiment. This re-confirms that a significant part of the power consumption is not predicted, which may be caused by various noise sources and/or parasitic signals. Those could be removed by filtering (as the right part of Figure 5 suggests) or averaging as the previous section underlined.

To conclude this section, we note that although the knowledge of a design's details may allow to improve correlation analysis attacks, a basic side-channel adversary will probably be limited to simple strategies, *e.g.* assuming all the bit transitions to contribute equally to the power consumption. It must be observed that, if the leaking points targeted by an adversary are connected to low effective capacitances within a FPGA, the actual attack may become more challenging. Another remark is that, due to their high diffusion properties, encryption algorithms usually require the use of long connection wires, which probably increases their power consumption compared to other designs. Finally, we reproduced the attack against an FPGA implementation of the AES Rijndael performed in [14] with the improved measurement process corresponding to the right part of Figure 4. It is represented in the left part of Figure 6 where we observe that the attack is successful after 300 generated plaintexts. Compared with the results in [14], it confirms our expectations that this number is roughly divided by 4.

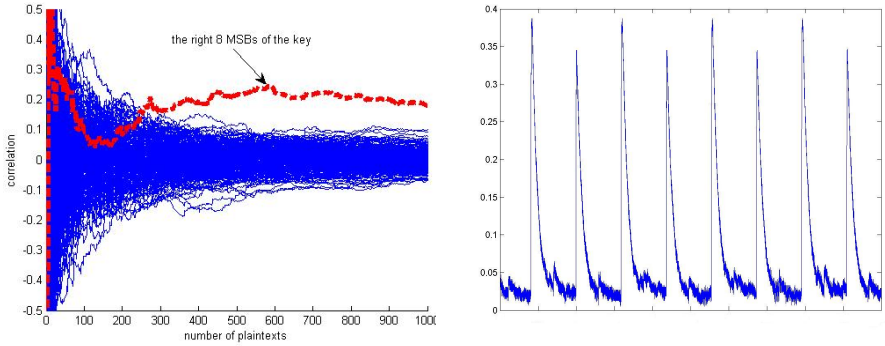

Fig. 6. An attack against the AES Rijndael, $0 \rightarrow 1$ and $1 \rightarrow 0$ bit transition differences.

5 Random pre-charges and consequences

A common countermeasure used in the smart card industry to counteract side-channel analysis is to pre-charge the buses with random values. Such a solution can be straightforwardly transposed in the context of FPGA implementations at the cost of a reduction of the throughput. Indeed, if one every two inputs of the encryption design is a random number generated within the FPGA[1], an adversary will not be able to predict the transitions within the implementation anymore (of course, the resulting ciphertext should not be outputted from the device). As suggested in [11], the only solution is then to distinguish $0 \rightarrow 1$ from $1 \rightarrow 0$ bit transitions through the leakages. In the latter case, one can predict the number of 0's and 1's in the device at some time, rather than predicting the number of bit transitions at this time. That is, we use a model based on the Hamming weight of the data manipulated rather than on its Hamming distance. To confirm that such a model is applicable, we performed a preliminary experiment, pictured on the right part of Figure 6. We observed the power traces of large bit-vectors switching between "all zeroes" or "all ones" patterns. Typically, this experiment suggested power consumption differences of about 10%.

The correlations between Hamming weight-based predictions and a single FPGA power trace (using the design (3) of Figure 2, as in Section 3), filtered or not, are represented in the left part of Figure 7, for different numbers of generated plaintexts. The correlations after averaging are in the right part of the figure. Roughly, we observe that a single trace allows to reach a correlation of up to 15% while a small averaging increases this value behind 20%. One can conclude that, although the correlations obtained are significantly lower (due to a much higher model matching noise), they are still sufficient to perform the attacks. This is specially true when considering that the measurement process is still likely to be improved and that other side-channel information could be used to increase these correlations, *e.g.* the electromagnetic radiation. On the other hand, if combined with other countermeasures, such random pre-charges may increase the difficulty of performing the attacks at a relatively low implementation cost.

[1] *e.g.* [5] could be used to produce the initial seeds of a pseudo-random number generator which will consequently generate the pre-charges.

Let us finally remark that the differences between $0 \rightarrow 1$ and $1 \rightarrow 0$ bit transitions could also be used to slightly improve our power consumption model of the previous sections, again using different weights for these different transitions.

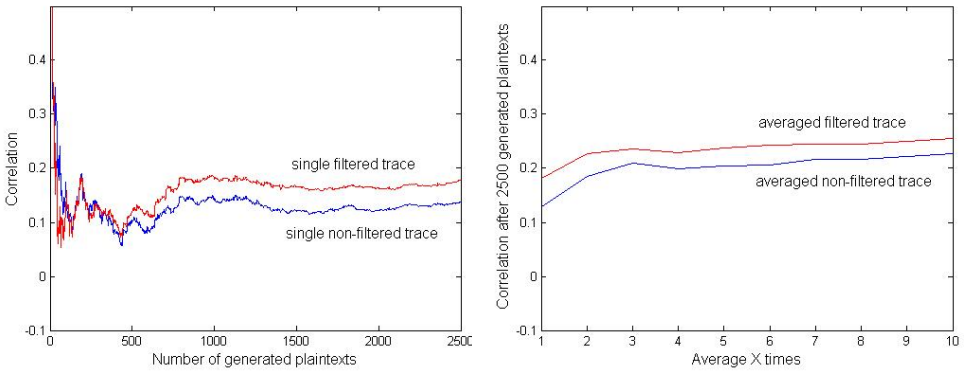

Fig. 7. Correlation between our predictions and actual power consumption measurements, with random pre-charges: single trace and averaged traces.

6 Security *vs.* efficiency tradeoffs

The previous considerations can be summarized in order to easily determine the number of plaintexts required to have a successful attack [15]:

$$N = c \times \frac{1}{C(\mathbf{P},\mathbf{G})^2 \times C(\mathbf{G},\mathbf{M})^2} \tag{5}$$

In this expression:

1. $C(\mathbf{G},\mathbf{M})$ is the expected correlation obtained between the power consumption model and the actual measurements (investigated in this paper). Our results suggest that reasonable values for this parameter are:
   - $0.50 < C(\mathbf{G},\mathbf{M}) < 0.95$ if no random pre-charges are used (we observed values in this range). The better the correlation is, the more efficient the resulting attack is.
   - $0.10 < C(\mathbf{G},\mathbf{M}) < 0.50$ if random pre-charges are used. Our results suggest no more than 0.25 but could possibly be improved. Therefore, a small security margin is reasonable.
2. $C(\mathbf{P},\mathbf{G})$ relates to the number of bits for which the power consumption can be predicted in the attack. If n_{pred} is this number of predictable bits and n_{tot} is the total number of bits in the design, we roughly[2] have $C(\mathbf{P},\mathbf{G}) = \sqrt{\frac{n_{pred}}{n_{tot}}}$.

[2] More precisely, if n_{pf} is the number of predictable and full bits, n_{pe} the number of predictable and empty bits and n_u the number of unpredictable bits, with $n_{tot} = n_{pf} + n_{pe} + n_u$, the correlation we are interested in is $\sqrt{\frac{n_{pf}}{n_{tot}-n_{pe}}} = \sqrt{\frac{n_{pf}}{npf+n_u}}$.

3. c is a small constant value depending on the number of key bits targeted during the attack. For example, it could be estimated once for 8-bit substitution boxes (like the ones of the AES Rijndael) as follows. Knowing that:
   - the attack of Figure 6 is using $C(\mathbf{G},\mathbf{M}) \simeq 0.9$,
   - the ratio of predictable registers (from [14]) is worth $\sqrt{\frac{48}{1536}} \simeq 0.18$,
   - the attack of Figure 6 is successful after 300 plaintexts,

   we find that a reasonable value (including a small security margin) is $c \simeq 10$.

From Equation 5, it is now extremely simple to evaluate the security of our different implementations in Figure 2. For example, let us consider a correlation attack against implementation (3), without random pre-charges. First, we assume a reasonable value for $C(\mathbf{G},\mathbf{M}) \simeq 0.8$. Then, we know that we have a total of $4 \times 128 = 512$ bits in the design, among which $3 \times 8 = 24$ are predictable. This yields $C(\mathbf{P},\mathbf{G}) = \sqrt{\frac{24}{512}}$. Finally, we find: $N \simeq 333$. Now, let us consider the fourth design for which the total number of bits is $8 \times 128 = 1024$ and we still have $n_{pred} = 32$. It yields $N \simeq 666$. If we additionally consider random pre-charges in the same fourth design, we could have $C(\mathbf{G},\mathbf{M}) \simeq 0.25$ (at the cost of a throughput reduction) and therefore $N \simeq 6882$. That is, any possible similar hardware architecture could be analyzed. As already frequently discussed, we observe that the attacks efficiencies depend on the implementation size and therefore involve a security *vs.* efficiency tradeoff. Note that in addition to the use of random pre-charges, various combinations of repetition codes (*e.g.* sending one true plaintexts for x random ones, in variable orders to the encryption device) could be considered. Also, as suggested in [9], such countermeasures could be particularly interesting in the context of feedback implementations, where pipelining cannot be used for increasing the performances, but possibly for fault detection or improved side-channel resistance.

We finally mention that in all our experiments, we only considered the peak values of the power traces occurring at the rising edges of the clock. It is reasonable (and verified in our experiments) to assume that these values give a good image of the power consumption because of the inherently synchronous behavior of RAM-based FPGAs. However, this could not be the case for other devices.

7 Conclusion

The correlation between the power consumption measurements of an isolated FPGA implementation of a cryptographic algorithm and a simple prediction based on the number of bit transitions within the devices can be up to 90%. Using random pre-charges in the FPGA allows to decrease these correlation values (our experiments suggest 25%) but is not sufficient to counteract the attacks. We provide simple techniques for estimating the number of measurements required to defeat one particular implementation. The latter estimations suggest that most FPGA implementations of symmetric-key block ciphers can be defeated in a low (*e.g.* a few hundred) number of power traces.

References

1. P. Barreto, V. Rijmen, *The KHAZAD Legacy-Level Block Cipher*, Submission to NESSIE project, available from http://www.cosic.esat.kuleuven.ac.be/nessie/
2. 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.
3. P. Buysschaert, E. De Mulder, S. B. Örs, P. Delmotte, B. Preneel, G. Vandenbosch, I. Verbauwhede, *Electromagnetic Analysis Attack on an FPGA Implementation of an Elliptic Curve Cryptosystem*, in the proceedings of EUROCON 2005 - The International Conference on Computer as a Tool, IEEE, 4 pages, 2005.
4. S. Chari, J. Rao, P. Rohatgi, *Template Attacks*, in the proceedings of CHES 2002, Lecture Notes in Computer Science, vol 2523, pp 13-28, Redwood City, CA, USA, August 2002.
5. V. Fischer, M. Drutarovsky, *True Random Number Generator Embedded in Reconfigurable Hardware*, in the proceedings of CHES 2002, Lecture Notes in Computer Science, vol 2523, pp 415-430, Redwood Shores, California, USA, August 2002.
6. P. Kocher, J. Jaffe, B. Jun, *Differential Power Analysis*, in the proceedings of Cypto 1999, Lecture Notes in Computer Science, vol 1666, pp 398-412, Santa-Barbara, USA, August 1999, Springer-Verlag.
7. National Bureau of Standards, *FIPS PUB 46, The Data Encryption Standard*, Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, Jan 1977.
8. National Bureau of Standards, *FIPS 197, Advanced Encryption Standard,* Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, November 2001.
9. T.G. Malkin, F.-X. Standaert, M. Yung, *A Comparative Cost/Security Analysis of Fault Attack Countermeasures*, in the proceedings of FDTC 2005, Edinburgh, Scotland, September 2005.
10. S.B. Ors, E. Oswald, B. Preneel, *Power-Analysis Attacks on an FPGA – First Experimental Results*, in the proceedings of CHES 2003, Lecture Notes in Computer Science, vol 2279, pp 35-50, Cologn, Germany, September 2003, Springer-Verlag.
11. E. Peeters, F.-X. Standaert, J.-J. Quisquater, *Power and Electromagnetic Analysis: Improved Model, Consequences and Comparisons*, to appear in Integration, the VLSI Journal, Spring 2006, Elsevier.
12. J.M. Rabaey, *Digital Integrated Circuits*, Prentice Hall International, 1996.
13. L. Shang, A. Kaviani, K. Bathala, *Dynamic Power Consumption in Virtex-2 FPGA Family*, in the proceedings of FPGA 2002, pp 157-164, Monterey, California, USA, February 2002.
14. 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, Cambridge, MA, USA, August 2004.
15. F.-X. Standart, E. Peeters, G. Rouvroy, J.-J. Quisquater, *Power Analysis Attacks and Countermeasures of Field Programmable Gate Arrays: Recent Results*, to appear in the Proceedings of the IEEE, special issue on Cryptographic Hardware and Embedded Systems, Spring 2006.
16. K. Tiri, I. Verbauwhede, *Synthesis of Secure FPGA Implementations*, in the proceedings of the International Workshop on Logic and Synthesis (IWLS 2004), pp 224-231, June 2004.