Power Analysis of FPGAs: How Practical is the Attack ?

François-Xavier Standaert, Loïc van Oldeneel tot Oldenzeel,
David Samyde, Jean-Jacques Quisquater

UCL Crypto Group
Laboratoire de Microélectronique
Université Catholique de Louvain
Place du Levant, 3, B-1348 Louvain-La-Neuve, Belgium
standaert,vanolden,samyde,quisquater@dice.ucl.ac.be

Abstract. Recent developments in information technologies made the secure transmission of digital data a critical design point. Large data flows have to be exchanged securely and involve encryption rates that sometimes may require hardware implementations. Reprogrammable devices such as Field Programmable Gate Arrays are highly attractive solutions for hardware implementations of encryption algorithms and several papers underline their growing performances and flexibility for any digital processing application. Although cryptosystem designers frequently assume that secret parameters will be manipulated in closed reliable computing environments, Kocher et al. stressed in 1998 that actual computers and microchips leak information correlated with the data handled. Side-channel attacks based on time, power and electromagnetic measurements were successfully applied to the smart card technology, but we have no knowledge of any attempt to implement them against FPGAs. This paper examines how monitoring power consumption signals might breach FPGA-security. We propose first experimental results against FPGA-implementations of cryptographic algorithms in order to confirm that power analysis has to be considered as a serious threat for FPGA security. We also highlight certain features of FPGAs that increase their resistance against side-channel attacks.

1 Introduction

Digital signal processing has traditionally been done using enhanced microprocessors but recent increases in Field Programmable Gate Arrays performance and size offer a new hardware acceleration opportunity. The last years brought cryptographic implementations into the field of FPGA designers as several conference and journal publications can witness [10, 11]. These cryptosystem designers frequently assume that secret parameters will be manipulated in closed reliable computing environments. However, the realities of physical implementations can be extremely difficult to control and may result in the unintended leakage of side-channel information. This leaked information is often correlated to the secret keys, thus adversaries monitoring this information may be able to recover the secret key and breach the security of the cryptosystem.

* This work has been funded by the Walloon region (Belgium) through the research project TACTILS.

Side-channel attacks based on time, power and electromagnetic measurements were successfully applied to the smart card technology as witnessed by [1–5]. However, we have no knowledge of any attempt to implement them against FPGAs. Moreover, most major FPGA manufacturers provide no information about the actual security of their devices. This paper presents first experimental results in order to fill that gap. Based on various examples, we discuss the practicability of power analysis attacks against an application-oriented FPGA board but also highlight certain physical features of FPGAs and application boards that make the practical implementation of power analysis significantly harder than in the smart card context.
The paper is structured as follows. Section 2 presents the hardware used to carry out the experiments. Section 3 gives a short description of two cryptographic algorithms: DES and RSA. Section 4 introduces power analysis. We study Simple Power Analysis and Differential Power Analysis in sections 5 and 6. Finally, topics for further researches are in section 7 and conclusions in section 8.

2 Hardware description

All our experiments were carried out on a VIRTEX-ARM board developed by DICE[1] (Figure 1). This board was developed in 2000 for a multi-purposes use. The board is composed of a control FPGA (Altera® FLEX®10K) and a Xilinx®Virtex®1000 FPGA associated with μ-controllers (Microchip®PIC®, ARM®) and fast access memories. It has multiple compatible PC interfaces (USB, PCI). Practical details about the Virtex1000BG560-4 FPGA that we investigated can be found in [8].
The voltages needed for the board are:

1. 5 volts for the PCI bridge.
2. 2.5 volts for the Virtex core and the ARM μ-processor.
3. 3.3 volts for other devices, including the Virtex I/O blocks.

The usual way to use this board has always been to plug it into a PCI port but to perform a power analysis against a chip, one must have access to its power supply in order to acquire power consumption traces. For this purpose, we insert a small resistance in the supply circuit. As the board has a single ground circuit and only the Virtex chip has to be analyzed (other devices add noise to the measurements) we decided to insert the resistance next to the source supplying the Virtex. We undersupplied certain unnecessary devices and we un-soldered the DC-DC 2.5V convertor (of which internal oscillations generate noise) before carrying out the experiments. Figure 1 illustrates the final test bed where the FPGA is programmed via the JTAG chain[2].
Finally, we used the following hardware to perform our tests :

1. Voltage sources to supply the 2.5 volts path and 3.3 volts path.

[1] Microelectronics Laboratory at Université Catholique de Louvain, Belgium.
[2] Boundary-Scan Standard IEEE 1149.1, developed by the Joint Test Action Group.

2. A waveform generator or a crystal oscillator to generate the clock signal.
3. An oscilloscope to observe the power traces. We used the Tektronix 7140 with a 1 GHz bandwidth.
4. Computer softwares to generate the FPGA programming files and process the data after acquisition.

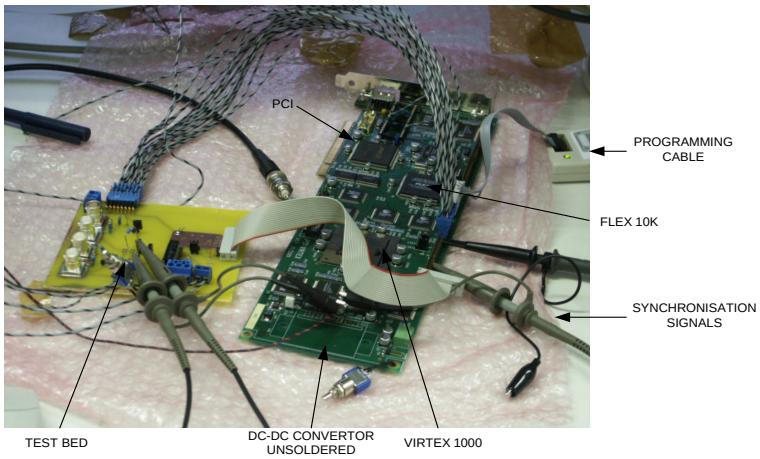

Fig. 1. The FPGA board

3 DES and RSA

In 1977, the Data Encryption Standard (DES) algorithm was adopted as a Federal Information Processing Standard for unclassified government communication. Although a new Advanced Encryption Standard was selected in October 2000, DES is still largely in use. DES [6] encrypts 64-bit blocks with a 56-bit key and processes data with permutations, substitutions and XOR operations. It is an iterative block cipher that applies a number of key-dependent transformations called rounds to the plaintext. This structure allows very efficient hardware implementations.

Basically, the plaintext is first permuted by a fixed permutation IP. The result is next split into the 32 left bits and the 32 right bits, respectively L and R that are sent to 16 applications of a round function. The ciphertext is calculated by applying the inverse of the initial permutation IP to the result of the 16-th round.

The secret key is expanded by the key schedule to 16 x 48-bit subkeys K_i and in each round, a 48-bit subkey is XORed to the text. The key expansion consists of known bit permutations and shift operations. As a consequence, finding any subkey bit directly involves that the secret key is corrupted.

Finally, the round function is easily described by:

$$L_i = R_{i-1} \tag{1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \tag{2}$$

where f is a nonlinear function detailed in Figure 2: the R_i part is first expanded

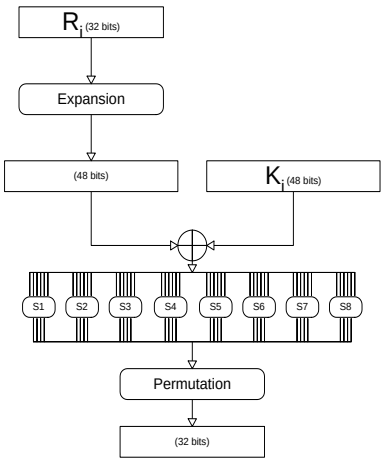

Fig. 2. The function f.

to 48 bits with the E box, by doubling some R_i bits. Then, it performs a bitwise modulo 2 sum of the expanded R_i part and the 48-bit subkey K_i. The output of the XOR function is sent to eight non-linear S-boxes (S). Each of them has six inputs bits and four outputs. The result is finally permuted in the box P. The design we used to carry out the experiments is a sequential DES that takes one clock cycle to perform one round.

The RSA cryptosystem [7] is the most widely used public-key cryptosystem worldwide. It may be used to provide both secrecy and digital signatures and its security is based on the intractability of the integer factorization problem. If $n = p \times q$ is a public modulus (p and q are large prime numbers), x the plaintext, and $k = \sum_{i=0}^{l-1} k_i 2^i$ the secret key, the RSA encryption scheme can be viewed as a simple modular exponentiation:

$$y = x^k \mod n \tag{3}$$

The design we used to carry out the experiments is a sequential "square and multiply" algorithm with 14-bit texts and keys. Modular reduction was done with Barrett's reduction rule and one "square and multiply" operation is performed in one clock cycle.

Algorithm 1 Computation of $x^k \ mod \ n$

1. z = 1;
2. For i = $l - 1$ to 0 loop :
   $z = z^2 \ mod \ n$;
   If $k_i = 1$ then $z = z \times x \ mod \ n$

4 Introduction to power analysis

Integrated circuits are built out of individual transistors that act as voltage-controlled switches. Current flows across the transistor substrate when charge is applied to (or removed from) the gate. This current then delivers charges to the gates of other transistors, interconnect wires, and other circuit loads. The motion of electric charge consumes power and produces electromagnetic radiations, both of which are externally detectable. Therefore, individual transistors produce externally observable electrical behavior. Because microprocessor logic units exhibit regular transistor switching patterns, it is possible to easily identify macro-characteristics (such as microprocessor activity) by the simple monitoring of power consumption.
In Simple Power Analysis attacks, an attacker directly observes a system's power consumption. The amount of power consumed varies depending on the microprocessor instruction performed. Large features such as DES rounds may be identified, since the operations performed by the microprocessor vary significantly during different parts of these operations.
Differential Power Analysis is a much more powerful attack than SPA, and is much more difficult to prevent. While SPA attacks use primarily visual inspection to identify relevant power fluctuations, DPA attacks use statistical analysis to extract information correlated to the secret key.
Because it was not obvious that power analysis could detect some features of a running design, we performed a simple preliminary tests: we investigated the power consumption of NOT gates applied to bit vectors (all 0s or all 1s) and stored in registers. We clearly observed that the power consumption is correlated to the Hamming weight[3] of these bit vectors (see Figure 3). However, this test gave no indication about a possible dilution of the bit effect when a large design (like DES) is running. Moreover, the power consumption was made clearer because the bit vectors appeared at the outputs of the FPGA, while power analysis usually looks for internal bit switches.

5 Simple Power Analysis of FPGAs

Traditional controllers process the data sequentially and apply a set of instructions to the intermediate states of the computation. They can be viewed as control-oriented designs. As a consequence, an attacker may expect to detect two types of information from their side-channel leakages:

[3] Hamming weight: number of one in the bit vector.

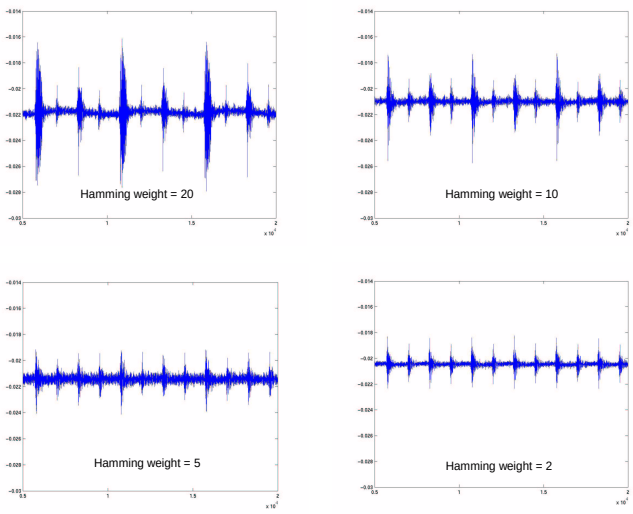

Fig. 3. Hamming weight (5000 traces averaged).

1. The instructions processed.
2. The data processed.

SPA typically tries to take advantages of the sequence of instructions processed. For example, distinguishing the square operation from the multiply operation would allow us to directly recover the key bits in an implementation of RSA. There are numerous examples of programs running on smart cards that allows to distinguish different instructions. However, simple countermeasures usually allows avoiding SPA by masking the instructions.
On the contrary, in most applications, FPGAs are used in order to perform parallel tasks. Cryptographic applications like DES or RSA can be implemented as data-oriented pipeline architectures with several operations running concurrently. Moreover, operations that are spread over several clock edges in smart cards may be reduced to only one clock period in FPGA implementations, which makes distinguishing them unlikely. As a consequence, in these cases, SPA becomes somewhat unpractical and an attacker is limited to information about the data processed as we have in Figure 3.
Exceptions obviously exist. For example applications where enable signals of registers are managed by a control part. Then the activity (or not) of registers may help to distinguish instructions.
We can illustrate these assumptions with an example. We investigated the power consumption of a DES running with weak keys that we can explain as follows. Because of the way the initial key is modified to get a subkey for each round of the algorithm, certain initial keys present special properties. Practically, if the subkey used for any round of the algorithm is the same, the inial key is

weak. DES has 4 weak keys and we used the following ones (in hexadecimal representation):

Weak Key Value	Actual Subkey
0101010101010101	00000000000000
FEFEFEFEFEFEFEFE	FFFFFFFFFFFFFF

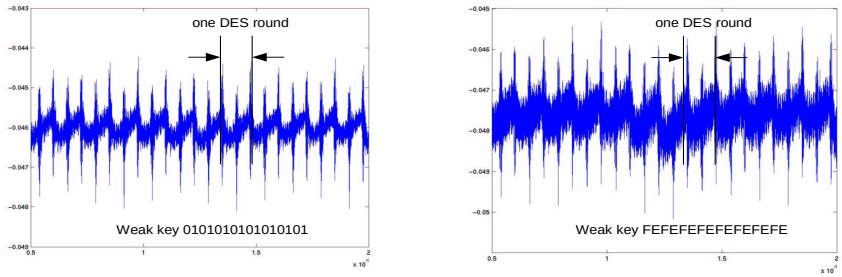

Fig. 4. Weak keys (5000 traces averaged).

Figure 4 illustrates the power consumption of DES running with weak keys. We observe that:

1. We can clearly identify the rounds of our running DES.
2. The mean power consumed slightly differs between the two cases. One reason could be that the architectures slightly differ because a different key is stored in the VHDL code.
3. The patterns of the power consumed are clearly different. The second is fatter which corresponds to the expected behavior of the device.

This test confirms that the consumed power is strongly correlated with the internal bit switches of FPGAs. It also underlines that SPA-type attacks, where the attacker recovers secret parameters observing the shape of the traces are made difficult by parallel computing (as all components are running concurrently). Moreover, FPGAs offer great opportunities to implement countermeasures against SPA.

6 Differential Power Analysis of FPGAs

As previous section confirmed that the power consumed by FPGAs is correlated with the internal bit switches, Differential Power Analysis is theoretically applicable. This section is devoted to experimental results of DPA implemented against RSA and DES running on a FPGA. We first studied modular exponentiation.

The basic premise of this attack is that by comparing the power signal of an exponentiation using a known exponent to a power signal using an unknown

exponent, the adversary can learn where the two exponents differ, thus learn the secret exponent. The DPA technique begins by using the secret exponent to exponentiate L random values and collect their associated power signals $S_i[j]$ (j is a sample point). Likewise, L power signals $P_i[j]$ are collected using the known exponent. The average signals are then calculated and subtracted to form $D[j]$, the DPA bias signal.

$$D[j] = \frac{1}{L}\sum_{i=1}^{L} S_i[j] - \frac{1}{L}\sum_{i=1}^{L} P_i[j] = \overline{S}[j] - \overline{P}[j] \tag{4}$$

The portions of the signals $\overline{S}[j]$ and $\overline{P}[j]$ that are dependent on the intermediate data will average out to the same constant as long as the data produced by the RSA computation is equal. We have $D[j] = 0$ if the exponentiation operations are the same and $D[j] \neq 0$ if different.
There are several ways to perform the attack, depending on the assumptions made about the attacker. The simplest one is a "Multiple-Exponent, Single-Data" mode. Then, the attacker guesses the exponent bits (starting from the MSB), decides if the guess was correct by computing $D[j]$ and modifies the exponent bits one by one in order to get $D[j] = 0$ everywhere. Figure 5 shows

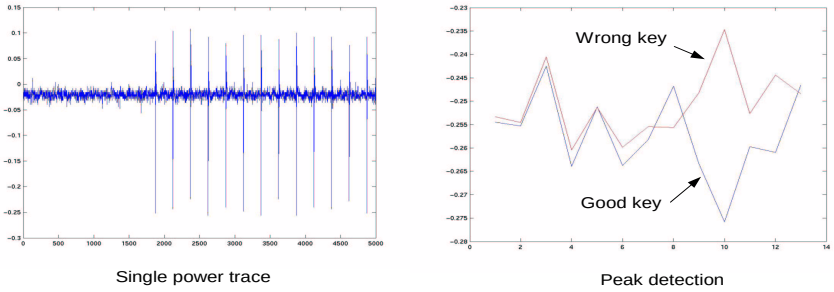

Fig. 5. DPA of RSA (5000 traces averaged).

our practical implementation of the attack. The left picture is a single power consumption trace where we observe the 13 clock edges corresponding to 13 "square and multiply" operations. The right picture shows peaks amplitudes for two keys that are equal until bit number 7. We observe that the consumption traces clearly diverge when exponents differ. Note that the attack depends on how different are the intermediate texts. As a consequence, repeating it with different texts improves its efficiency. Another critical point is that our RSA design was a toy-design with 12-bit vectors. As a consequence the difference between correct and wrong vectors is not large and it was difficult to underline their different power consumption. We increased the power consumption by repeating the RSA computation 20 times on the FPGA. Then we could clearly distinguish the secret exponent.
In the case of DES, the Differential Power Analysis requires a selection function

$D(C, b, K_{Sb,16})$ that we define as computing the value of a bit b which is a part of intermediate vector L_{15}. As b results of a partial decryption through the last round of the algorithm, it can be derived from the ciphertext C and the 6 key bits entering in the same s-box as bit b.
To implement the DPA attack, an attacker first observes m encryptions and captures m power traces T_i and their associated ciphertexts C_i. No knowledge of the plaintext is required. With a guessed key K, the function D can be computed for each i and we can obtain two sets of traces: one corresponding with $D_i = 0$ and the other with $D_i = 1$. Each set is then averaged to obtain two average traces A_0 and A_1 and we can compute the difference $\Delta = A_0 - A_1$.
If $K_{Sb,16}$ is correct, the computed value for D will equal the actual value of target bit b with probability 1. As the power consumption is correlated to the data, the plot of Δ will be flat, with spikes in regions where D is correlated to the values being processes. If $K_{Sb,16}$ is incorrect, Δ will be flat everywhere.
The main difference between attacking DES and RSA is that while we have to distinguish the difference between two intermediate vectors in the RSA case, we have to observe the effect of a single bit in the case of DES, what we could not achieve with our low cost equipment. The following practical features of our FPGA board make the implementation of the DPA against DES a challenging task:

1. It should be noted that the application boards usually include several components of which the grounds are connected together. This makes the isolation of the FPGA consumption critical if the power measurements are carried out on the ground pin.
2. The manipulation of the selection bit that is spread over several clock edges in smart cards is reduced to one clock period in our FPGA implementation.
3. FPGAs are running at high work frequencies. Optimal implementations of the DES on the old VIRTEX technology run up to 170 MHz. Recent devices like VIRTEX-2 are much faster. This involve very high sampling rates to catch the consumption details.
4. Contrary to smart cards where the data is managed by 8-bit registers, FPGAs deal with all the bits (64 for DES) at once. This cause a dilution of the desired effect. This is even more critical when the key schedule or other tasks are computed in parallel. As a result, the quantization of power traces may become the bottleneck of the attack, i.e. if the effect of a single bit is out of scale (less than one bit of quantization), the attack becomes unfeasible. Figure 3 illustrates this assessment with a comparison between 20-bit spikes and 2-bit spikes.

7 Further research

A practical implementation of the DPA against DES is still matter of further research and there are plenty of potential sources for improvements. Nevertheless, there are many others scopes for further research. We propose the following list:

1. Reducing the noise during measurements by isolating the FPGA, using multiple-bit attacks, cooling the devices with nitrogen, ...
2. Applying intrusive attacks to FPGAs: depackaging, layer recovering,...
3. FPGAs usually consists in regular structure. As a consequence, Electro-Magnetic Analysis could be applied in order to focus the acquisition of information leaking to some relevant logic blocks.
4. FPGAs have multiple power sources. Analysis of their distribution inside the logic blocks could help to isolate some components of FPGAs.
5. Studying the security questions raised by the reconfigurability.

8 Conclusions

This work confirmed that power analysis has to be considered as a serious threat for FPGA security. Although certain features of our FPGA board made the practical implementation of power attacks significantly harder than in the smart card context, we have conduced relevant experimental tests. We analyzed the power of a DES running with weak keys and could clearly distinguish both keys. We also implemented a Differential Power Analysis attack against a toy-implementation of RSA. Many solutions would allow to improve our measurements, for example isolating the FPGA from its application board, and a lot of questions concerning the physical security of FPGAs remain open. As a future technological trend seems to be the combination of processors and reconfigurable hardware, there is a field for various research in the coming years.

References

1. P.Kocher, *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*, In the proceedings of CRYPTO 96, Lecture Notes in Computer Science Volume 1109. Springer-Verlag, August 1996.
2. P.Kocher, J.Jaffe, B.Jun, *Differential Power Analysis*, in the proceedings of CRYPTO 99, Lecture Notes in Computer Science 1666, pp 398-412, Springer-Verlag.
3. 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, N5, May 2002.
4. P.Kocher, J.Jaffe, and B.Jun, *Introduction to Differential Power Analysis and Related Attacks*, Cryptography Research 607 Market Street, 5th Floor San Francisco, CA 94102, www.cryptography.com.
5. J.J.Quisquater, D.Samyde, *Electromagnetic Analysis (EMA): Measurements and Countermeasures for Smart Cards*, in Smart Card Programming and Security, Lecture Notes in Computer Science Volume 2140, pp.200-210, Springer-Verlag 2001.
6. National Bureau of Standards. *FIPS PUB 46*, The Data Encryption Standard. U.S. Departement of Commerce, Jan 1977.
7. R.Rivest, A.Shamir, L.Adleman, *A Method for Obtaining Digital Signatures and Public Key Cryptosystems*, Communications of the ACM, 21, pp 120-126, 1978.
8. Xilinx: *Virtex 2.5V Field Programmable Gate Arrays Data Sheet*, http://www.xilinx.com.
9. Altera: *Flex 10K Field Programmable Gate Arrays Data Sheet*, http://www.altera.com.
10. Proceedings of CHES 1999-2002 : Workshop on Cryptographic Hardware and Embedded System, Springer Verlag.
11. Proceedings of FPL 1999-2002 : The Field Programmable Logic Conference, Springer-Verlag.
12. D.Stinson, *Cryptography: Theory and Practice*, CRC Press 2000.