Experimental Power Analysis Attacks on an FPGA

Zia Kamawal
Electrical and Computer Engineering
Oregon State University
Corvallis, Oregon 97331
Email: kamawal@engr.orst.edu

Abstract— **Power analysis attacks is becoming an increasingly more researched field in cryptography. This paper summarizes the study of S.B. Ors, E. Oswald, and B. Preneel on power attacks using FPGAs. FPGAs are widely used because of their flexibility and relative ease of use. They are not only used in lab environments for prototyping, but also in production environments which makes them good experimental devices. Their study was completed on a Xilinx Virtex 800 FPGA. Their analysis was done using an implementation of Elliptic-Curve Point-Multiplication. They were able to show that power analysis attacks were indeed feasible on FPGAs.**

I. Introduction

It has been proven that many secure devices are prone to power analysis attacks. Currently, smart cards have been the main focus of power attacks and it seems that the industry is heading in the direction of trying to make their products immune to these types of attacks.

There are two types of power attacks, simple power attacks and differential power attacks. SPA is easily implemented and it has been shown that many smart cards are susceptible to this type of attack. SPA is discussed in more detail later. DPA is a more complex attack in which a small portion of the key must be guessed. In the original paper on DPA, it was shown that no smart card, at that time, was safe from DPA.[1]

This paper is organized in a way that gives much background so that the results of this study are meaningful. The background involves in depth discussion of SPA and DPA, FPGA architecture. A discussion and algorithm of elliptic curve point multiplication is provided because the experiments were performed on an FPGA loaded with this circuit.

II. CMOS Characteristics

CMOS circuits have the property of being static, meaning that that there is no power dissipation when the circuit is not switching. When the inputs to the gates become switching, a current is generated that can be seen through both Vdd and Vss. A small resistor can be placed between Vss or Vdd of the device and the actual Vss or Vdd. The voltage generated across the resistor can be measured and analyzed for any leakage of information.

III. Simple Power Analysis (SPA)

Simple power analysis is based on the power consumption of the entire system. By analyzing the power consumption it can be determined what instructions are being executed. This is because the power that a microprocessor consumes differentiates based on the type of instruction that is being executed. Also at times, key information can be obtained from SPA.

According to tests done by [1], some smart cards were susceptible to SPA, even though it is not difficult to protect systems from SPA attacks[2]. SPA can reveal a sequence of instructions that are executed. This information can be used to break some implementations of cryptographic algorithms in which the instruction being executed depends on the data that is being processed.[1]

Figure 1 shows a power consumption strace of a DES operation. The 16 rounds can be seen clearly and therefore it is known that this is a DES operation. At more detailed views, more information can be seen including the individual instructions being executed.

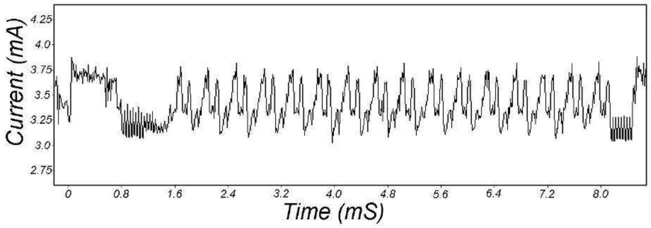

Fig. 1. SPA Trace using a DES Operation

IV. Differential Power Analysis (DPA)

Differential power analysis is a very powerful attack in compatison to SPA. DPA is much more difficult to prevent than SPA. DPA uses statistical analysis and some error correction techniques to extract information that is related to secret keys involved in the computation.

There are two parts to the implementation of a DPA attack. There is a data collection phase and a data analysis phase. Data collection is performed by reading a device's power consumption while performing crytographic operations and sampling this as a function of time. The effects of a single switching transistor cannot be indentified normally by observing a device's power consumption. Because of the statistical operations performed in DPA, we are able to indentify even these very small differences in power consumption. DPA can catch the relation between the data and the side channel implementation.

Figure 2 shows a DPA attack. The first trace is a power reference trace. The second is a differential trace using the correct value for the key. The las two traces are also differential traces, but they use incorrect values for the secret key.

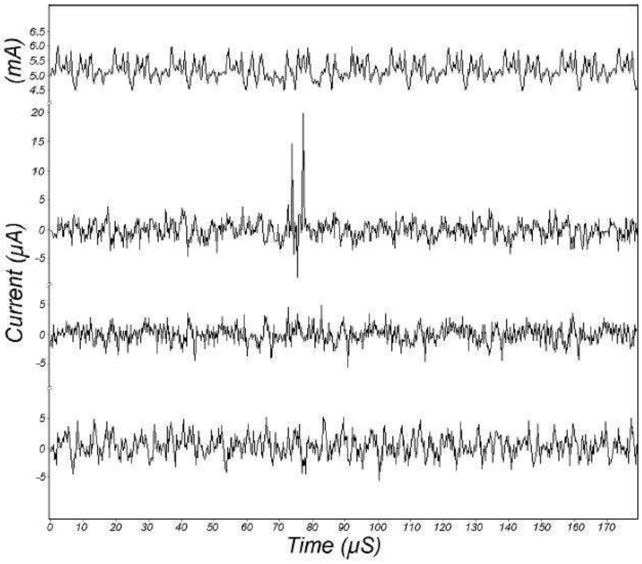

Fig. 2. Power Rederence, Correct DPA Trace and 2 Incorrect Traces

V. FPGA Architecture

Field Programmable Gate Arrays are arrays of configurable logic blocks (CLBs). Connecting these CLBs are programmable intereconnects that allow logic from one CLB to interact with logic on another CLB in the same FPGA. Surrounding these CLBs are programmable I/O blocks. An FPGA can be comprised of 64 to many thousands of these CLBs. Because the blocks are not interconnected 100 percent, there exists software that can place and route the logic intellectually on the FPGA.

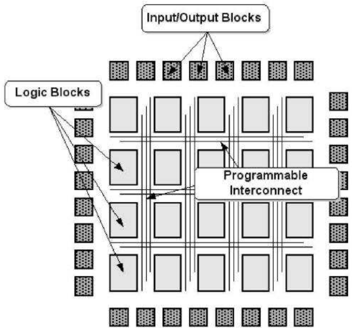

Fig. 3. Simple FPGA Block Diagram

VI. Elliptic Curve Point Addition

Elliptic curve point multiplication involves the use of both EC point addition and EC point doubling. This is why we have shown the implementation of EC point addition here. Its algorithm involves 14 steps and it is shown here as algorithm 1.

```
Algorithm 1: EC Point Addition
Require:
$P_1 = (x, y, 1, a)$, $P_1 = (X_2, Y_2, Z_2, aZ_2^4)$
Ensure:
$P_2 + P_2 = P_3 = (X_3, Y_3, Z_3, aZ_3^4)$
1. $T_1 \leftarrow Z_2^2$
2. $T_2 \leftarrow xT_1$
3. $T_1 \leftarrow T_1 Z_2, T_3 \leftarrow X_2 - T_2$
4. $T_1 \leftarrow yT1$
5. $T_4 \leftarrow T_3^2, T_5 \leftarrow Y_2 - T_1$
6. $T_2 \leftarrow T_2 T_4$
7. $T_4 \leftarrow T_4 T_3, T_6 \leftarrow 2T_2$
8. $Z_3 \leftarrow Z_2 T_3, T_6 \leftarrow T_4 + T_6$
9. $T_3 \leftarrow T_5^2$
10. $T_1 \leftarrow T_1 T_4, X_3 \leftarrow T_3 - T_6$
11. $T_6 \leftarrow Z_3^2, T_2 \leftarrow T_2 - X_3$
12. $T_3 \leftarrow T_5 T_2$
13. $T_6 \leftarrow T_6^2, Y_3 \leftarrow T_3 - T_1$
14. $aZ_3^4 \leftarrow aT_6$
```

VII. Elliptic Curve Point Doubling

As stated in the previous section, elliptic curve point multiplication involves the use of both EC point addition and EC point doubling. Here we have shown the implementation of EC point doubling. Its algorithm involves 14 steps and it is shown here as algorithm 2.

```
Algorithm 2: EC Point Doubling
Require:
$P_1 = (X_1, Y_1, Z_1, aZ_1^4)$
Ensure:
$2P_1 = P_3 = (X_3, Y_3, Z_3, aZ_3^4)$
1. $T_1 \leftarrow Y_1^2, T_2 \leftarrow 2X_1$
2. $T_3 \leftarrow T_1^2, T_2 \leftarrow 2T_2$
3. $T_1 \leftarrow T_2 T_1, T_3 \leftarrow 2T_3$
4. $T_2 \leftarrow X_1^2, T_3 \leftarrow 2T_3$
5. $T_4 \leftarrow Y_1 Z_1, T_3 \leftarrow 2T_3$
6. $T_5 \leftarrow T_3(aZ_1^4), T_6 \leftarrow 2T_2$
7. $T_2 \leftarrow T_6 + T_2$
8. $T_2 \leftarrow T_2 + (aZ_1^4)$
9. $T_6 \leftarrow T_2^2, Z_3 \leftarrow 2T_4$
10. $T_4 \leftarrow 2T_1$
11. $X_3 \leftarrow T_6 - T_4$
12. $T_1 \leftarrow T_1 - X_3$
13. $T_2 \leftarrow T_2 T_1, aZ_3^4 \leftarrow 2T_5$
14. $Y_3 \leftarrow T_2 - T_3$
```

VIII. Elliptic Curve Point Multiplication

An implementation of EC point multiplication was used in the experiments. EC point multiplication is performed using the EC point addition and EC point doubling operations. The arithmetic for a 160-bit prime field was implemented with a Montgomery modular multiplier (MMM). The final subtraction was not implemented. The algorithm is described as follows in Algorithm 3[3]:

```
Algorithm 3: EC Point Multiplication
Require:
Integers $N = (n_{l-1}...n_1n_0)_2$,
$x = (x_l...x_1x_0)_2$,
$y = (y_l...y_1y_0)_2$, with
$x\epsilon[0, 2N - 1]$,
$y\epsilon[0, 2N - 1]$,
$R = 2^{l+2}$,
$gcd(N, 2) = 1$
$N' = -N^{-1}$mod 2
Ensure: $xyR^{-1}$ mod 2N
$T \leftarrow 0$
for i from 0 to $l + 1$ do
    $m_i \leftarrow (t_0 + x_iy_0)N'$ mod 2
    $T \leftarrow (T + x_iy + m_iN)/2$
end for
Return(T)
```

IX. Hardware Setup

For the these experiments, the Xilinx Vertex 800 FPGA was chosen. They have listed some resons for this particular choice: 1. The resources are sufficient to implement a 160-bit Elliptic-Curve Point Multiplication. 2. It is the most powerful FPGA that can be hand-mounted and do not require special machines. 3. The architecture is composed of combinational and memory elements, which makes it comparable to ASICs.

The setup included two boards. One is for connecting the FPGA to the PC via parallel cable and testing. The other board, called the daughter board, is designed specifically for the FPGA to allow for convienient access to certain pins for measurement.

The parallel port is used for programming the FPGA. This port has 4 control lines, 5 status line and 8 data lines. For these experiments, only 17 input/output pins of the FPGA were needed, but 32 input/ouput pins were connected to create added flexibility for later use.

A protocol was designed for the FPGA to communcicate with the PC. The most significant three bits of the staus line indicate the staus, and the reamining two bits of the staus line are used for sending information from the FPGA to hte PC[4]. The protocol behaves the same ways regardless of the operation excuted by the FPGA. This way, the setup is flexible and can be used for experimenting with different algorithms.

X. Results

A. EC Point Addition

As it can be seen from the power consumption trace of Figure 4, there are fourteen steps in the EC Point addition algorithm and this is confirmed by section VI. It can also be seen that in steps 3, 5, 7, 8, 10, 11, and 13, that the power consumption of the sytem increases. This is due to the fact that in those steps of the EC Point Addition (See Algorithm 1), a modular addition is taken in addition to a modular multiplication.

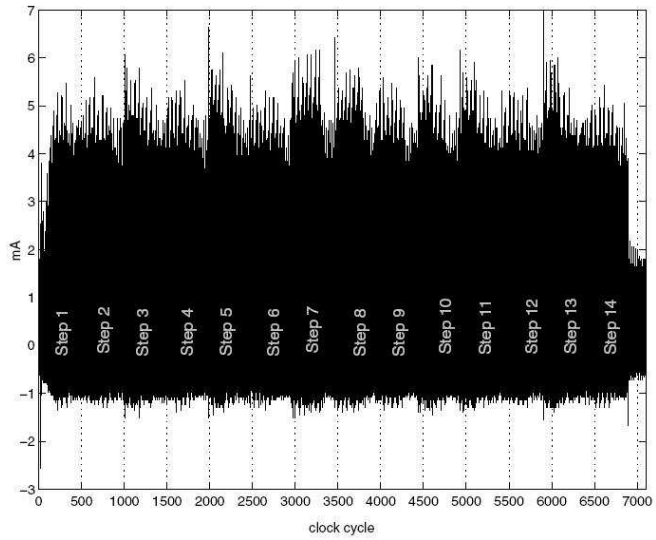

Fig. 4. Power consumption trace of 160-bit EC point addition

B. EC Point Doubling

The number of clock cycles for EC Point doubling was less than that of EC point addition. This is the expected result because the algorithm for EC point doubling is specifically designed to add a number to itself. From the power consumption trace in Figure 5, we can again see all fourteen steps in the algorithm. We can also see that for steps 7, 8, 10, 11, 12, 14, the power consumption is much less than for other steps because in those steps only a modular addition or substraction is taking place (see Algorithm 2).

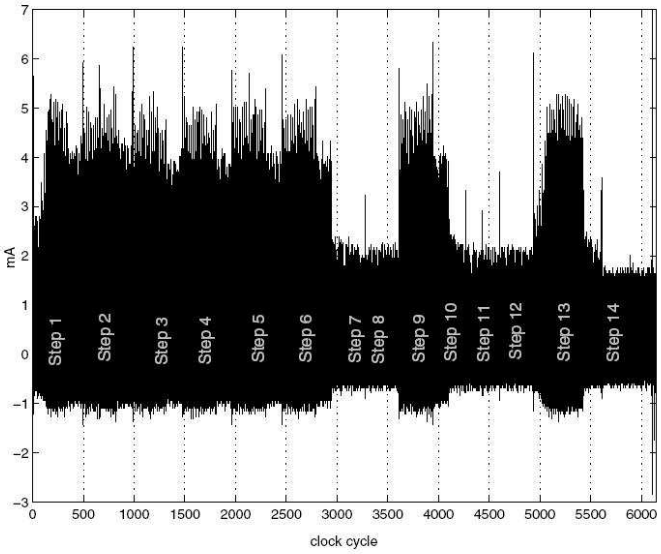

Fig. 5. Power consumption trace of 160-bit EC point doubling

C. EC Point Multiplication

The EC point multiplication was performed and implemented using a double-and-add algowithm. Both, the EC point addition and EC point doubling are used. From the power consumption trace shown in Figure, it can be seen clearly that the secret key used in the measurement is 001100.

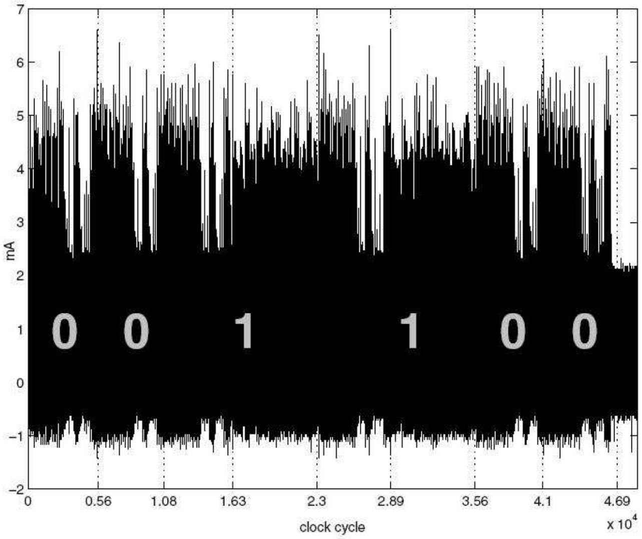

Fig. 6. Power consumption trace of 160-bit EC point multiplication

XI. Conclusions

Power consumption characteristics of the FPGA are very similar to that of ASIC circuits. We can conclude that FPGA implementations are also very vulnerable to power analysis attacks. It has been shown that many smart card devices are susceptible to DPA and SPA. The results can conclude that power attacks can be performed at a low cost on development type applications.

References

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology - CRYPTO 99*, M. Wiener, Ed. Springer Verlag, LNCS Nr. 1666, 1999, pp. 388–397.

[2] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Power analysis attacks of modular exponentiation in smartcards," in *Cryptographic Hardware and Embedded Systems*, ser. First International Workshop, Worcester, MA, USA, Ç. K. Koç and C. Paar, Eds. Springer Verlag, LNCS Nr. 1717, August 12-13, 1999, pp. 144–157.

[3] G. Hachez and J.-J. Quisquater, "Montgomery exponentiation with no final subtractions: Improved results," in *Cryptographic Hardware and Embedded Systems - CHES 2000*, ser. Second International Workshop, Worcester, MA, USA, Ç. K. Koç and C. Paar, Eds. Springer Verlag, LNCS Nr. 1965, August 17-18, 2000, pp. 293–301.

[4] S. B. Örs, E. Oswald, and B. Preneel, "Power-analysis attacks on an FPGA - First experimental results," in *Cryptographic Hardware and Embedded Systems - CHES 2003*, ser. 5th International Workshop, Redwood Shores, CA, USA, C. D. Walter, Ç. K. Koç, and C. Paar, Eds. Springer Verlag, LNCS Nr. 2779, September 8-10, 2003, pp. 35–50.