Electromagnetic Side Channels of an FPGA Implementation of AES

Vincent Carlier, Hervé Chabanne, Emmanuelle Dottax
and Hervé Pelletier

SAGEM SA

Abstract. We show how to attack an FPGA implementation of AES where all bytes are processed in parallel using differential electromagnetic analysis. We first focus on exploiting local side channels to isolate the behaviour of our targeted byte. Then, generalizing the Square attack, we describe a new way of retrieving information, mixing algebraic properties and physical observations.

Keywords: side channel attacks, DEMA, FPGA, AES hardware implementation, Square attack.

1 Introduction

Side channel attacks first appear in [Koc96] where *timing attacks* are described. This kind of attack tends to retrieve information from the secret items stored inside a device by observing its behaviour during a cryptographical computation. In a timing attack, the adversary measures the time taken to perform the computations and deduces additionnal information about the cryptosystems. Similarly, *power analysis* attacks are introduced in [KJJ99] where the attacker wants to discover the secrets by analysing the power consumption. Smart cards are targets of choice as their power is supplied externally. Usually we distinguish Simple Power Analysis (SPA), that tries to gain information directly from the power consumption, and Differential Power Analysis (DPA) where a large number of traces are acquired and statistically processed. Another side channel is the one that exploits the Electromagnetic (EM) emanations. Indeed, these emanations are correlated with the current flowing through the device. EM leakage in a PC environment where eavesdroppers reconstruct video screens has been known for a long time [vE85], see also [McN] for more references. In [QS01, GMO01], Simple Electromagnetic Analysis (SEMA) and Differential Electromagnetic Analysis (DEMA) are introduced. Recently, it has been proposed to combine multiple side channels, power consumption and EM emanations, to improve the efficiency of the attack [SCR02].

Most of the work that has been published so far is about attacks on smart cards. In this paper we are working on Field Programmable Gate Arrays (FPGAs). EM emanations from an FPGA are of the same nature as the ones from a smart-card. Most of the EM emanations can be attributed to the commutation of p and n CMOS transistors. When the FPGA is clocked, the p and n transistors can be simultaneously conducting, for example in an inverter gate, causing a short circuit between the ground and the power supply line. Moreover, the capacitive effect, due for instance to the charge or discharge of the bus line or of the next input stages, increases the current leakage. Our contribution shows that DEMA can be performed against hardware implementation of AES using an FPGA. In fact, high frequency and parallel computations are no sufficient protection against this kind of attack.

The remainder is organized as follows. Section 2 describes the platform we used for our experiments. Section 3 presents the results we obtained applying DEMA in a classical way and Sect. 4 is devoted to a new type of attack against AES following the Square attack [DKR97, DR] (Square attacks are also called *saturation* attacks or *integral* attacks). Section 5 concludes.

1.1 Related work

FPGAs now come under the scrutiny of the cryptographic community. Paar and Wollinger [PW03] survey the security aspects of FPGAs. The first experimental results appear in [OOP03] where Örs, Oswald and Preneel study the power consumption of an FPGA and show on an implementation of an ellliptic curve point multiplication that the information leaked is enough to mount an SPA. Their results show a high power consumption leakage and the same behaviour can be expected from the EM emanations leakage as the magnetic field strength follows Maxwell's law where both characteristics are related. Until now, EM attacks are related to smart card chips, mainly on software algorithm implementations. As a result of [QS01,GMO01] we know that EM leakage, although noisier than current leakage, can be more reliable to detect differential biases during EMA.

Our second approach has some similarity with the ones in [SWP03, SLFP03], where some mathematical properties are exploited together with side channels. These papers use internal collisions, we exploit some characteristics related to the Square attack.

1.2 Brief description of the AES

Here, we follow the notations from [Nat01]. The version of the Advanced Encryption Standard we consider is a symmetric encryption algorithm consisting of 10 rounds, acting on a 128-bit block represented as a *state* consisting of 16 bytes. In the following, we consider states as 4 × 4 squares, each cell representing one byte and we use terms like columns, rows, which come clear with this interpretation (see Fig. 1). A round is made of 4 different operations:

SubBytes: it is a non-linear byte substitution that operates independently on each byte of the state;

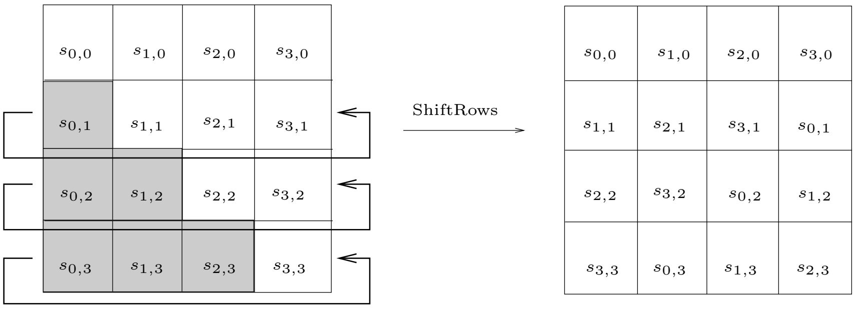
ShiftRows: this transformation acts on the rows of the state as illustrated by Fig. 1;

MixColumns: this operation treats each column as a 4-byte vector and multiplies it by a matrix;

AddRoundKey: this operation adds a round key to the state by bitwise XOR operation.

In a nutshell, we have for the whole cipher:

```
AddRoundKey(state,RoundKey[0])
for i from 1 to 9
  SubBytes(state)
  ShiftRows(state)
  MixColumns(state)
  AddRoundKey(state,RoundKey[i])
end for
SubBytes(state)
ShiftRows(state)
AddRoundKey(state,RoundKey[10])
```



$s_{0,0}$	$s_{1,0}$	$s_{2,0}$	$s_{3,0}$
$s_{0,1}$	$s_{1,1}$	$s_{2,1}$	$s_{3,1}$
$s_{0,2}$	$s_{1,2}$	$s_{2,2}$	$s_{3,2}$
$s_{0,3}$	$s_{1,3}$	$s_{2,3}$	$s_{3,3}$

$s_{0,0}$	$s_{1,0}$	$s_{2,0}$	$s_{3,0}$
$s_{1,1}$	$s_{2,1}$	$s_{3,1}$	$s_{0,1}$
$s_{2,2}$	$s_{3,2}$	$s_{0,2}$	$s_{1,2}$
$s_{3,3}$	$s_{0,3}$	$s_{1,3}$	$s_{2,3}$

Figure 1: The effect of the ShiftRows transformation

2 Experimental platform

2.1 Measurement setup

Our target is an ALTERA Cyclone FPGA. This FPGA allows the programmation of up to 20600 logic elements and up to 290 KBits of RAM. Two PLLs are present. Our electrical equipment embeds the FPGA on a board connected to a PC via a parallel port. There are also two DC regulators to supply 1.5 V to the core and 3 V to the input/output blocks of the FPGA. The external power supply adapter has been replaced with a voltage generator to reduce the signal noise. No other modification has been implemented on the FPGA board; in particular, no decapsulation was performed. Our measurements are made using the on-board clock at 50 MHz. We use a standard digital oscilloscope with a 500 MHz bandwith and a sample rate of 5 GSamples/s to measure the probe's output signal. This oscilloscope is also connected to a PC through GPIB interface. Figure 2 illustrates this setup.

We try different kinds of antennas varying the material they are made of (copper or gold), their shape (solenoid, loop, spiral antennas, ...) and their size from 1 mm to 10 cm. Finally, we choose solenoid wires of copper consisting of a dozen of spires with a diameter of approximately 1 mm. We estimate the bandwith characteristics of such probes between 10 MHz and 200 MHz. In a simple model the probe's output voltage is approximately $V = -\frac{\mathrm{d}\Phi}{\mathrm{d}t}$, where the magnetic flux Φ is deduced from Biot-Laplace-Savart's law:

$$\mathrm{d}\Phi = \mu\overrightarrow{H}\cdot\overrightarrow{\mathrm{d}S}, \qquad \text{and } \overrightarrow{\mathrm{d}H} = \frac{(I\overrightarrow{\mathrm{d}l})\wedge\overrightarrow{u}}{4\pi x^2}$$

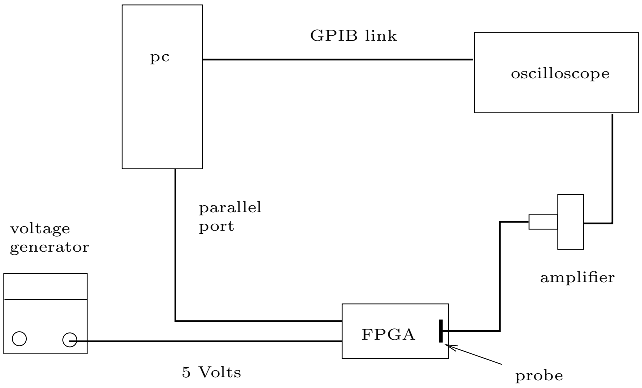


Figure 2: Data acquisition setup for EM analysis of an FPGA

where $\overrightarrow{\mathrm{d}l}$ is the elementary conductor of the circuit, I the current in $\overrightarrow{\mathrm{d}l}$ and x the distance between the probe and the element $\overrightarrow{\mathrm{d}l}$. $\overrightarrow{H}$ is the magnetic field strength, μ the magnetic permeability, $\overrightarrow{\mathrm{d}S}$ a surface element and $\overrightarrow{u}$ its normal vector. This model indicates that the probe must be placed as near as possible to the FPGA to increase the magnetic flux collected by the probe.

2.2 AES implementation

For our AES implementation, we retrieved IPs from [NSA]. For each round, all the 128 bits of the input are processed simultaneously, in a sequential way, the SubBytes operation first, next the ShiftRows and so on. We observe that due to various propagation times, the processing of the different bits is not achieved exactly at the same time. Each round takes around 20 ns.

Remark. With ALTERA FPGAs, it is not possible to recover the layout from the bitstream. Working in attack conditions, some regions a priori unknown to us radiate more than others.

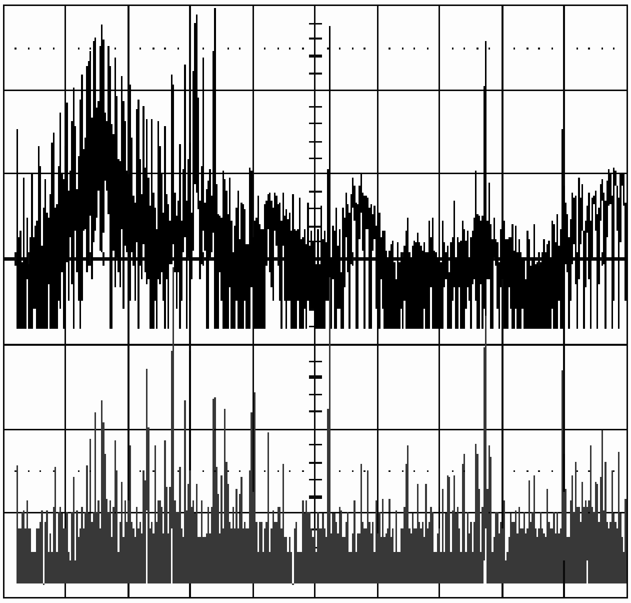


Figure 3: FFT of Electromagnetic emanation (Scale: 20 MHz by 12 dBm)

3 Practical results

3.1 Preliminary results

We analyse with FFT tools the signal frequencies collected by the probe during an AES computation (the first curve in Fig. 3) and for an idle state (the second curve in Fig. 3). One can see that even in idle state we detect a few large spikes. From our point of view, these specific frequencies come from parasitical sources. During the AES computation, two specific regions appear: one around 50 MHz, which is due to the internal clock frequency, and a second one that begins at 180 MHz. Note that the second one is not a harmonic of the clock.

3.2 SEMA and DEMA results

We first make some computations of the AES with all key bytes to `00h` (the black curve on Fig. 4) and all key bytes to `FFh` (the grey curve on Fig. 4). We can observe an amplitude difference between the two curves around 2 mV which comes, essentially, from data Hamming weight leakage by the FPGA.

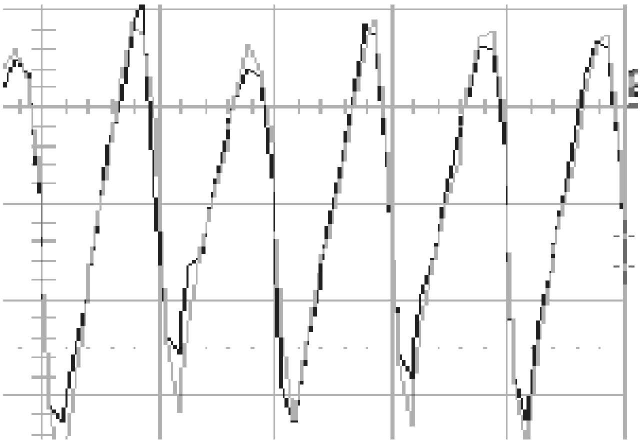


Figure 4: SEMA (Scale: 20 ns by 20 mV)

We then launch DEMA using a classical function to partition the measured power curves into two disjoint sets. This function guesses one bit value of the output of the SubBytes function during the first round. We try multiple measurements with the probe in different positions above the FPGA. The first curve on Fig. 5 shows the typical AES electromagnetic signal with its 10 rounds collected by the probe. The next 4 curves show the presence of DEMA bias when the key byte is correctly guessed. It can be seen that the bias spike appears at the beginning of the first round of AES when the ByteSub function is calculated. Finally, the last curve shows the DEMA signal when the key byte guessed is wrong.

Note that the bias spike can disappear if we modify the specific bit attacked in our partition function. So, for a specific probe position we can only detect specific bits leakage. This phenomenon can explain why we are not disturbed by the parallel computation effect, unlike with power or current measurements. Such results show the effectiveness of EMA against classical AES implementation on a FPGA.

4 Square EM Attacks

4.1 Theory

We follow [DR] introducing sets of states where some bytes are *passive*, and stay constant among the set, while others are *active*.

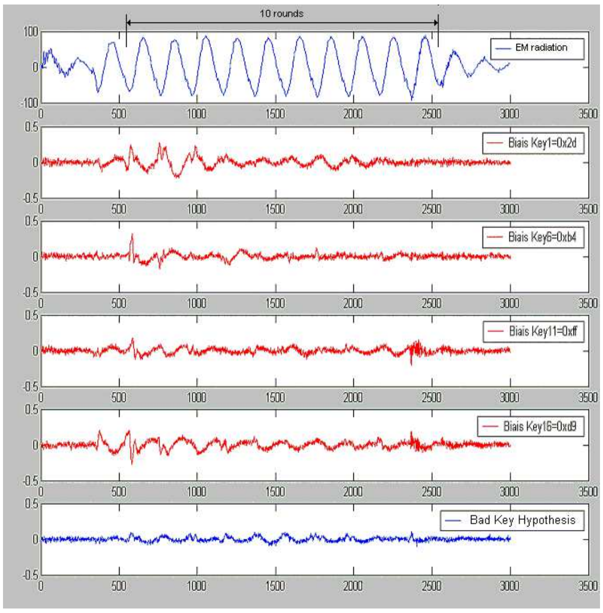


Figure 5: DEMA

Definition 1 *A* Λ-set *is a set of 256 states where passive bytes keep the same value for each state and active bytes are different from one state to another one.*

$$\forall x, y \in \Lambda : \begin{cases} x_{i,j} \neq y_{i,j} & \textit{if } (i,j) \textit{ is active} \\ x_{i,j} = y_{i,j} & \textit{else} \end{cases}$$

Λ-sets were introduced in order to cryptanalyse a reduced-round AES using the so-called "Square attack". This chosen plaintext attack is based on tracking the evolution of a Λ-set through the rounds. It relies on the following facts:

- a Λ-set remains a Λ-set after the AES operations SubBytes, AddRoundKey and ShiftRows;
- the MixColumns operation converts an input column of a Λ-set with only one active byte into an output column of a Λ-set with all four bytes active.

The reader is referred to [DR] for more details and proofs.

We use the fact that only MixColumns can modify the passivity of a given byte among a set of states and base our attack on distinguishing sets of states according to the number and location of bytes which are not passive. Given the set of all possible 4-byte vectors, we can extract from its image by MixColumns a set of 2^{16} vectors having 2 passive bytes. We show that it is possible to distinguish such a set from a random set by analysing the emanations during the AES processing. So we can make a key hypothesis to separate plaintexts that give, after the first MixColumns operation, states with 2 non-passive bytes from plaintexts that give states with 4 non-passive bytes and validate it by analysing the EM emanations. For this, we need to consider sets of states consisting of more than 256 states in order to reduce correctly the noise.

More precisely, consider a set of states for which the bytes on the main diagonal take different values while all the others are constant. Following the evolution of these states through the first steps of the AES (as shown on Fig. 6), we see that, depending on the value of the 4 bytes of the key involved in the first AddRoundKey operation, the states after MixColumns can be separated into two sets:

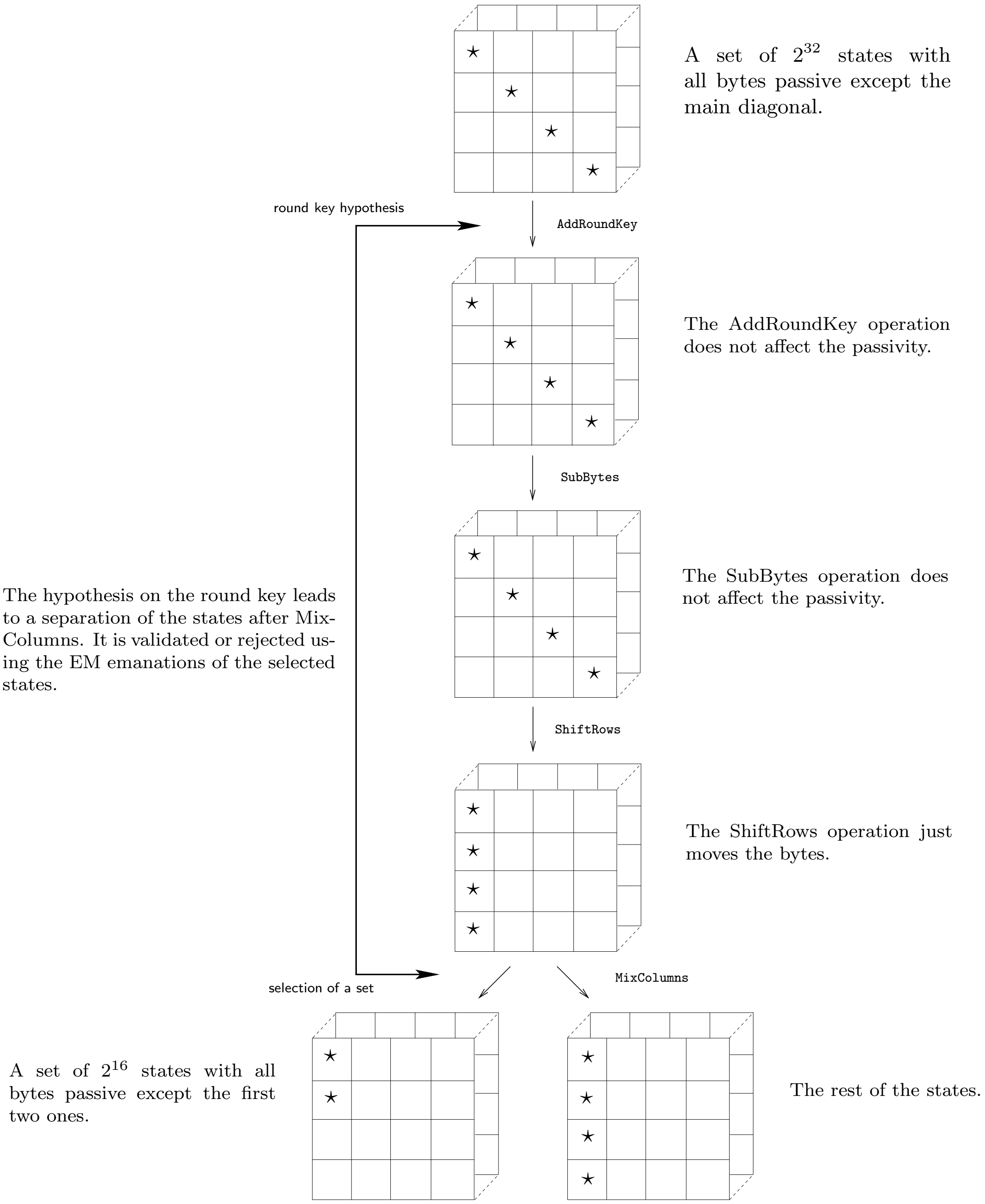
- a chosen set of 2^{16} states for which all the bytes are passive except the two first ones;
- all the other states.

If we are able to distinguish these two sets by measurements, we can mount an attack as follows. We consider 2^{32} states such that the main diagonal range over all possible values and all other 12 bytes remain constant. We formulate a hypothesis on the value of the 4 bytes of the first round key for the first AddRoundKey. This leads to a separation of the states into two sets as before. If our hypothesis is valid we are able to distinguish these two sets of states using the corresponding emanation curves. Our attack, that we call *Square Electromagnetic Attack*, can thus be described as follows :

1. Generate all 2^{32} input states which vary only on the main diagonal, execute the AES and measure the corresponding EM side channels;

2. Fix a value for the 4 bytes of the first round key on the main diagonal and then separate the curves of emanations according to the prediction given by these chosen bytes;

3. Validate or reject the hypothesis on the 4 bytes of the first round key using the two sets of emanation curves.



A set of 2^{32} states with all bytes passive except the main diagonal.

The AddRoundKey operation does not affect the passivity.

The SubBytes operation does not affect the passivity.

The hypothesis on the round key leads to a separation of the states after MixColumns. It is validated or rejected using the EM emanations of the selected states.

The ShiftRows operation just moves the bytes.

A set of 2^{16} states with all bytes passive except the first two ones.

The rest of the states.

Figure 6: Square EM attack

4.2 Results

Here we describe some experiments we made in order to establish the feasibility of validating or rejecting the hypothesis on the 4 bytes of the first round key using two sets of emanations curves (see step 3 above).

For each set we collect the electromagnetic emanations from the FPGA. In our experiment 7000 traces have been acquired. We generate:

- half of these traces using the valid hypothesis on the round key and thus getting states after the `MixColumns` for which all the bytes are passive except the two first ones;

- half using an invalid hypothesis, the states having thus four non-passive bytes in the first column.

Suppose that the following model holds. In a first approximation the EM leakage can be considered as proportional to the Hamming weight of the byte processed. We write $W(a_i) = kE(a_i)$ where a_i is a byte, $E(a_i)$ the EM signal emited during its processing and $W(a_i)$ its Hamming weight. The SEMA results allow us to think that this approximation is good. If we average N times the EM emanation collected from the treatment of one random byte we obtain the EM emanation average:

$$E_M = \frac{1}{N}\sum_{i=0}^{N} E(a_i) = \frac{1}{N}\sum_{i=0}^{N} kW(a_i) = k\frac{1}{N}\sum_{i=0}^{N} W(a_i)$$

Since $W(a_i)$ is a random value from 0 to 8, then $\frac{1}{N}\sum_{i=0}^{N} W(a_i) = 4$ and finally $E_M = 4k$. In the same way, for a constant byte a_c the EM emanation average is:

$$E_{Mc} = \frac{1}{N}\sum_{i=0}^{N} E(a_c) = \frac{1}{N}W(a_c)\sum_{i=0}^{N} k = kW(a_c)$$

So if we compare the averages for two random bytes we get

$${E_M}' - E_M = 0$$

On the contrary, if we compare the averages for a random and a constant byte we obtain

$$E_{Mc} - E_M = k(W(a_c) - 4) \neq 0.$$

The first curve on Fig. 7 shows the typical EM emanation during the AES computation. The second curve represents the difference between the

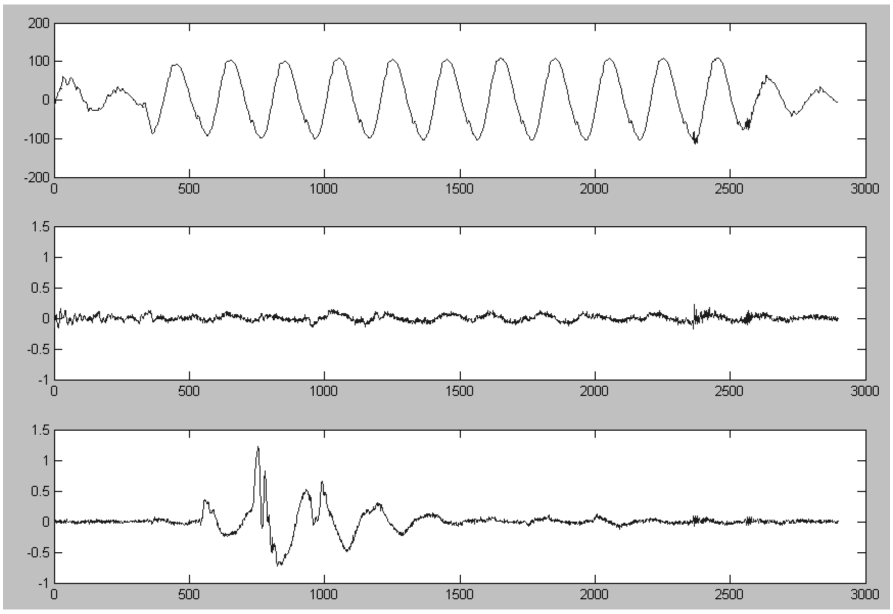


Figure 7: Square EM Attack

average of the two EM emanations for two sets of messages when a false hypothesis is made. Finally the third curve shows the difference of the two EM emanations for two sets of messages, one with the good hypothesis and the other with a false one. Several peaks can be observed, starting at the end of the first round where the particular state appears, and until it completely disappears. We interpret this phenomenon by the fact that several operations do not affect the passivity. We conclude that we are able to validate the hypothesis by observing the presence of bias spikes or not. As we observe that very few "ghost" peaks appear when a wrong key hypothesis is made, we think that the entire automatization of this attack is possible.

Improvement. We are also able to distinguish a set of 1000 states with three non-passive bytes from a set of 1000 states with four non-passive bytes. This constitutes an improvement to the attack because from 2^{32} input states, we can get a set of 2^{24} states with three non-passive bytes. So we can reduce the number of needed input states. For instance, if with use a set of 2^{20} input states, we can hope to get a set of approximately 2^{12} states with three non-passive bytes, which is enough to observe the difference.

5 Conclusions

We show that EM side channels from an FPGA implementation of AES can be effectively used by an attacker to retrieve some secret information.

Working with local emanations, we are able to get rid of pertubations, in particular, of other computations made at the same time. During DEMA, we measure the effect of one particular byte we want to exploit. To achieve this, we have to place our probe as close as possible above the FPGA and make different attemps in order to position it precisely where to retrieve the needed information.

The new Square EM Attack, that we introduce here, gives us more freedom. It should be noted that this attack also allows us to perfom some kind of power analysis attack following the same principle (not reported here) although we do not succeed with classical DPA. This has to be added to [CJRR99a] as only Rijndael seems to suffer from this weakness.

Finally, 128 KB of RAM are needed to implement the modified S-box needed to protect the AES using the method described in [GP00, CJRR99b]. With today FPGAs this kind of amount is quite common. Note also that we have to find an internal random source too; [FsD02b, FsD02a] can provide some solutions to this very problem.

Acknowledgements. This work has been supported by the MEDEA+ programme under the CryptoSoC project.

We thank Elisabeth Oswald and Siddika Berna Örs.

References

[CJRR99a] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. A cautionary note regarding evaluation of AES candidates on smart-cards. In *Proceedings of the Second Advanced Encryption Standard Conference*. NIST, 1999.

[CJRR99b] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *Proceedings of CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer-Verlag, 1999.

[DKR97] Joan Daemen, Lars Ramkilde Knudsen, and Vincent Rijmen. The block cipher Square. In Eli Biham, editor, *Proceedings* *of Fast Software Encryption – FSE'97*, volume 1267 of *Lecture Notes in Computer Science*, pages 149–165. Springer-Verlag, 1997.

[DR] Joan Daemen and Vincent Rijmen. AES proposal: Rijndael. Selected as the Advanced Encryption Standard. Available from `http://csrc.nist.gov/encryption/aes/`.

[FsD02a] Viktor Fischer and Miloš Drutarovskí. True Random Number Generator Embedded in Altera ACEX Devives. In *Proceedings of DCIS'02*, pages 587–592, 2002.

[FsD02b] Viktor Fischer and Miloš Drutarovskí. True Random Number Generator Embedded in Reconfigurable Hardware. In Burton S. Kaliski, Çetin Kaya Koç, and Christof Paar, editors, *Proceedings of CHES'02*, volume 2523 of *Lecture Notes in Computer Science*, pages 415–430. Springer-Verlag, 2002.

[GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *Proceedings of CHES'01*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer-Verlag, 2001.

[GP00] Louis Goubin and Jacques Patarin. DES and Differential Power Analysis – The "Duplication" Method. In Çetin Kaya Koç and Christof Paar, editors, *Proceedings of CHES'99*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer-Verlag, 2000.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Proceedings of Crypto'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer-Verlag, 1999. Available from `http://www.cryptography.com/resources/whitepapers/DPA-technical.html`.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *Proceedings of Crypto'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer-Verlag, 1996.

[McN] Joel McNamara. The Complete, Unofficial TEMPEST Information Page. Internet Web page. `http://www.eskimo.com/~joelm/tempest.htm`.

[Nat01] National Institute of Standards and Technology. FIPS-197: Advanced Encryption Standard, November 2001. Available at `http://csrc.nist.gov/publications/fips/`.

[NSA] NSA's VHDL Implementations of the Five AES Candidates finalists. Available from `http://csrc.nist.gov/CryptoToolkit/aes/round2/r2anlsys.htm`.

[OOP03] Siddika Berna Örs, Elisabeth Oswald, and Bart Preneel. Power-Analysis Attacks on an FPGA – First Experimental Results. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Proceedings of CHES'03*, volume 2779 of *Lecture Notes in Computer Science*, pages 35–50. Springer-Verlag, 2003.

[PW03] Christof Paar and Thomas Wollinger. How secure are FPGAs in cryptographic applications? In Peter Y. K. Cheung, George A. Constantinides, and José T. de Sousa, editors, *Proceedings of FPL 2003*, volume 2778 of *Lecture Notes in Computer Science*, pages 91–100. Springer-Verlag, 2003.

[QS01] Jean-Jacques Quisquater and David Samyde. ElectroMagnetic Analysis (EMA): Measures and counter-measures for smart cards. In Isabelle Attali and Thomas P. Jensen, editors, *Proceedings of E-smart 2001*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210. Springer-Verlag, 2001.

[SCR02] Josyula R. Rao Suresh Chari and Pankaj Rohatgi. Templates Attacks. In Burton S. Kaliski, Çetin Kaya Koç, and Christof Paar, editors, *Proceedings of CHES'02*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer-Verlag, 2002.

[SLFP03] Kay Schramm, S. Leander, Patrick Felke, and Christof Paar. A Collision -Attack on AES Combining Sidechannel- and Differential-Attack. Submitted for publication, 2003.

[SWP03] K. Schramm, T. Wollinger, and C. Paar. A New Class of Collision Attacks ant its Application to DES. In *Proceedings of FSE 2003*, Lecture Notes in Computer Science. Springer-Verlag, 2003.

[vE85] W. van Eck. Electromagnetic radiations from video display units: an eavesdropping risk? Comput. Secur., 1985.