Influence of fault-tolerant design methods on differential power analysis resistance of AES cipher: Methodics and Challenges

Vojtěch Miškovský, Hana Kubátová, Martin Novotný
Department of Digital Design
Czech Technical University in Prague, Faulty of Information Technology
Prague, Czech Republic
{miskovoj, hana.kubatova, novotnym}@fit.cvut.cz

***Abstract*—Many electronic systems has to fulfill strict dependability properties, especially both fault tolerance and attack resistance. These requirements usually contradict each other. The study and experiment descriptions of the possible methods how to measure these impacts are presented in this paper. Specifically, how fault-tolerant design methods affects resistance against differential power analysis attack and how the whole design can be modified to increase attack resistance will be discussed.**

Keywords-fault tolerance; attack resistance; reliability; security; AES; differential power analysis (DPA); FPGA

I. INTRODUCTION

One of the common digital design requirements is to be fault-tolerant. There are many methods of digital design to achieve fault tolerance [1]. Another common requirement is attack resistance for which also exist many digital design methods [5], [8].

The target of our research is to investigate the influence of fault-tolerant design methods on attack resistance and vice versa considering the final design of real circuits, especially the ones used in mission-critical applications. New methods or conformation of the present ones to achieve both fault tolerance and attack resistance at the same time will be the main aim of this research.

We will focus on passive hardware redundancy methods in this article. These methods cause design size and power consumption increase. When some kind of security module is designed using these fault tolerance increasing methods, the increased power consumption may undesirably affect its resistance against side channel attacks like differential power analysis.

The important question is, whether this influence is positive or negative. This question cannot be easily answered and this article proposes methods for measuring this influence in order to propose some ways to achieve both fault tolerance and DPA resistance.

This article contains a description of chosen fault-tolerant methods, AES cipher and differential power analysis in section II. We will also mention how we have implemented AES and its fault-tolerant variants in section III. Section IV describes measuring methods and a control application we developed. Finally, we will summarize our progress and our future plans (sections V and VI).

II. STATE OF THE ART

There is a lot of works about fault tolerance (e.g. [1], [2]) and attack resistance (e.g. [5], [8]), but we are not familiar with any work related to the mutual influence of them. The current research is based on passive hardware redundancy fault-tolerant design methods [1], AES cipher [3] and differential power analysis [4].

A. Fault-tolerant design methods

We decided to start with passive hardware redundancy methods, which are easy to implement. These methods are based on multiplication of the whole logic module [1]. They have high impact on power consumption and they are commonly used.

1) Duplex

Duplex consists of two modules. It compares outputs of modules and signals a failure when they differs. It can detect a failure of one module.

2) Triple modular redundancy (TMR)

TMR is based on three modules and a voter. The voter provides a majority vote of outputs. When single module fails the system remains fully operational. When two modules fail the whole system fails. Example of TMR is shown in figure 1.

3) N-modular redundancy (NMR)

NMR is a generalization of TMR. It uses N modules and a voter. An odd number of modules should be used to secure the existence of a majority.

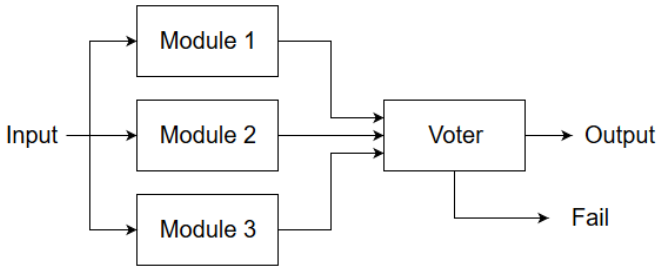

Figure 1. Schema of triple module redundancy

B. AES

AES (advanced encryption standard) is a symmetric block cipher, which is one of the most common cipher used for example for securing wireless networks. It uses fixed block size (128 bits) and one of three key sizes (128, 192 or 256 bits) [3]. We will use the 128bit version.

AES algorithm for 128bit key consists of ten rounds and initial transformation. At each round different key derived from the initial one is used.

For more detailed information see the AES specification [3].

C. Differential power analysis

DPA belongs to a group of attacks known as the side-channel attacks. It is based on measuring power consumption traces of a device and a calculation of the secret information (key) by an application of statistical functions on obtained power traces. DPA requires plain text or cipher text to be known for each power consumption waveform [4].

We will use an advance type of DPA using correlation coefficients for calculations (sometimes called correlation power analysis (CPA)). We can divide it into three phases: the measurement phase, the hypothesis phase and the calculation phase.

1) Measurement phase

We have to measure power consumption of device during the encryption in this phase. We have a list of n plain texts/cipher texts and a matrix of $t \times n$ elements as a result of this phase, where n means number of measures (waveforms) and t means count of power consumption values (traces) for each measure. Example of a waveform is shown in figure 2.

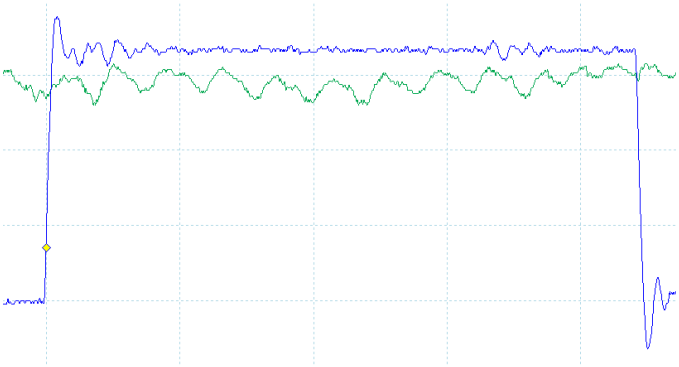

Figure 2. Example of power (green) and trigger (blue) waveforms

2) Hypothesis phase

We need to split key into parts (for example bytes) and make some prediction about power consumption for each possible value of each part of key (256 for one byte). For example when we are analyzing AES cipher we can split the key into bytes, then for each byte XOR is applied on each possible value (0-255) with corresponding byte of plain text, then SubBytes function is applied [3] and we can use Hamming weight of the result as our hypothetical value.

During the hypothesis phase we will produce $k \times n$ values for each part of the key, where k means a number of possible values of the selected part of the key.

3) Calculation phase

Now we need to evaluate, which of the possible key values is the right one.

We have the hypothetical value matrices for each part of the key and the power consumption matrix. Now we need to apply vector correlation function on each row of hypothetical value matrix with each row of power consumption matrix. This results into $t \times k$ correlation matrix for each part of key. Now the position of the maximal value of each of these matrices should correspond with the real key part value and the relevant trace (time).

III. Implementation

Evariste III platform was chosen as an implementation platform. It provides USB communication and FPGA module with connectors for power consumption measurement. VHDL examples of USB communication are provided, too. Our module contains FPGA Altera Cyclone III. This device's typical clock frequency is 48MHz [9]. The whole platform is shown in figure 3.

We edited VHDL USB communication examples provided along with Evariste module to be able to handle changing cipher key, receiving a plain text and sending it back to PC.

A. AES

AES encryption was implemented to take ten clock cycles (one clock cycle for each round). The round itself is implemented as a combination logic divided into entities corresponding to transformation functions of each round [3].

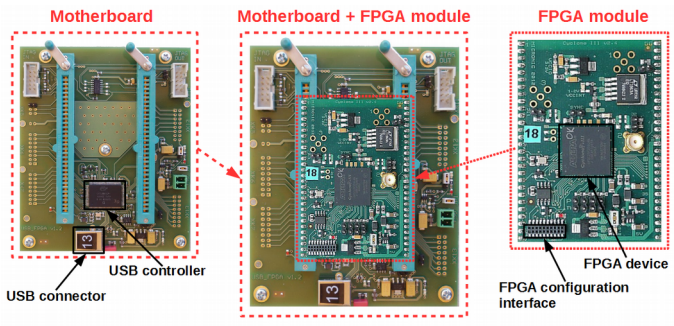

Figure 3. Setup of Evariste III project

Each round key is generated in the same clock cycle as the round using it is processed.

Due to this we have one register containing the current round key and one register containing the current cipher state. The presumed content of the current state register will be used for calculating hypothetical values in differential power analysis.

One pin on Evariste module is dedicated to indicate active encryption and it will serve as trigger for the oscilloscope measurement.

B. *Fault-tolerant variants*

We have implemented only simple realizations of fault tolerant variants using multiple of the same copies of AES module so far.

Duplex is implemented as two copies of AES module, which outputs are compared and if they are not equal the fail signal is set.

NMR is implemented as a parametric entity with parameter *N* representing number of modules. It is realized as N copies of AES module and a voter. The voter represents a set of comparators comparing all *k*-combinations of *N* outputs, where *k* is the lowest majority ((*k=N-1)*/2), and a multiplexor deciding which output is the correct one or whether fail signal should be set. It is realized by a recursive function generating both the comparators and the multiplexor.

We also implemented non-parametric TMR to simplify synthesis when NMR with only three AES modules is required.

IV. MEASUREMENT

We chose oscilloscope PicoScope 6404D (shown in figure 4) for the measurement. It provides sampling rate 5GS/s (2.5GS/s when two channels are used).

Hamming distance between state register value at ninth and tenth round was chosen as the hypothetical value used in differential power analysis. According to [7] it is the best choice with FPGA. The whole hardware architecture is shown in figure 5.

Measurement methods were tested on simple AES implementation.

Figure 4. PicoScope 6404D

A. *Initial method*

First method of measuring power consumption used was simple, but not fast enough to capture thousands of waveforms, which is necessary. We need to capture plenty of waveforms, because only about fifty traces can be captured during one clock cycle of our FPGA so we need to compensate lack of data to get correlation function working.

The first attempts were done using communication program provided with Evariste and PicoScope 6 GUI application. We created a command line script generating random plain texts and sending it using the communication program while the PicoScope 6 captures the power consumption traces.

This method allowed us to capture only about four waveforms per second.

We were using Wolfram Mathematica 10 to calculate the correct key using correlation function. This method was pretty slow (about ten minutes for one byte of key and ten thousand waveforms).

B. *Advanced method*

We decided to program our own application, which is responsible for the generation of the data, the communication with the encryptor, the oscilloscope measurement and the power analysis. We chose C programming language, since oscilloscope provides C API.

We replaced proprietary USB driver of Evariste device by *WinUSB* driver [11], so we were able to use *libusb* library [10] for communication with the encryptor. The communication protocol remains the same as the original one used by Evariste program.

Due to the oscilloscope C API it was relatively simple to achieve about a hundred waveforms triggered per second and there is still some reserve for improvements.

This application also implements a correlation function running about hundred times faster than the original script in written in Mathematica. There are also some possible optimizations by making it parallel.

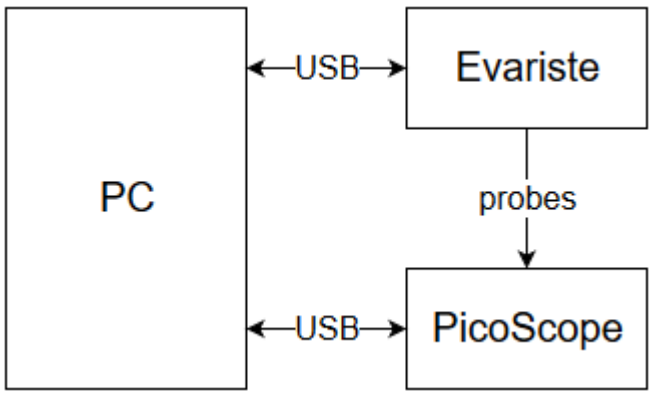

Figure 5. Measurement hardware architecture

The whole software architecture of the advanced method is shown in figure 6. This method accelerated the measurement dozens of times as is shown in table I. It is even more evident for the higher counts of waveforms.

TABLE I. MEASUREMENT METHODS COMPARISON

	Overall duration of measurement and calculation in seconds for various counts of waveforms				
	100	***1,000***	***10,000***	***100,000***	***1,000,000***
Initial	28	266	7,496	n/a	n/a
Advanced	1	12	118	1,729	21,361

Duration of initial method was not measured for 100,000 and 1,000,000, it would last too long.

V. CONCLUSION

We have implemented an AES encryptor and basic passive hardware redundancy methods in VHDL considering the fact that we need to measure its power consumption. We chose suitable device for the measurement.

Also we have programmed application serving the whole measuring and calculation process of differential power analysis, which is fast and easy to use.

The actual measurement for all implemented fault-tolerant variants of AES (and possibly another cipher) is prepared and will be realized now. The expected result is that these basic fault-tolerant techniques will all cause worse resistance against DPA. Final conclusions and experimental results will be given at the presentation.

VI. FUTURE WORK

The first task is to measure the currently implemented AES using fault-tolerant techniques and find methods for comparing their power analysis attack resistance.

Next step will be implementation of fault-tolerant variants using divergent AES modules. For example duplex with AES modules mutually masking power consumption, this should considerably increase attack resistance.

We can also try to use another ciphers or another type of attack.

Other possible continuation of research is to investigate other fault-tolerant design methods. Also we could take another point of view and measure influence of attack resistant digital design on its fault tolerance.

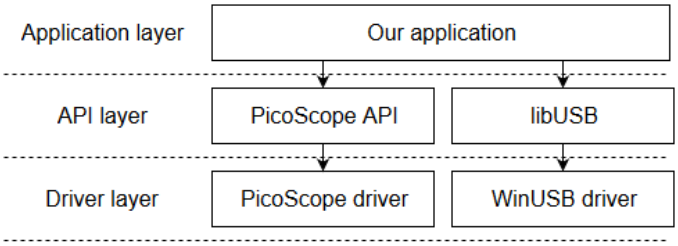

Figure 6. Measurement software architecture

ACKNOWLEDGMENT

This research has been partially supported by the grant GA16-05179S of the Czech Grant Agency, „Fault-Tolerant and Attack-Resistant Architectures Based on Programmable Devices: Research of Interplay and Common Features" (2016-2018), MSMT project LG15012 and CTU project SGS16/042/OHK3/1T/18.

REFERENCES

[1] Pradhan, Dhiraj K. *Fault-tolerant computer system design*. Upper Saddle River, N.J: Prentice Hall PTR, 1996. Print.

[2] Koren, Israel, and C. M. Krishna. *Fault-Tolerant Systems*. Morgan Kaufmann, 2007. Print.

[3] Federal Information Processing Standards Publication (FIPS 197). Advanced Encryption Standard (AES), 2001

[4] Paar, Christof. *Implementation of Cryptographic Schemes 1*. Ruhr University Bochum, 2015.

[5] Tiri, K.; Verbauwhede, I.. *A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation*. in *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, vol.1, no., pp.246-251 Vol.1, 16-20 Feb. 2004.

[6] Becker, Jürgen, Marco Platzner, and Serge Vernalde. *Field-programmable Logic and Applications: 14th International Conference, FPL 2004, Antwerp, Belgium, August 30-September 1, 2004: Proceedings*. Berlin: Springer, 2004. Print.

[7] McDaniel III, Larry T. *An investigation of differential power analysis attacks on FPGA-based encryption systems*. Diss. Virginia Polytechnic Institute and State University, 2003.

[8] Yang, Shengqi, et al. "*Power attack resistant cryptosystem design: a dynamic voltage and frequency switching approach*." Proceedings of the conference on Design, Automation and Test in Europe-Volume 3. IEEE Computer Society, 2005.

[9] Viktor Fischer, Patrick Haddad, Florent Bernard. *An open-source multi-FPGA modular system for fair benchmarking of true random number generators*. 23rd international conference on field programmable logic and applications. Porto, Portugal, 2013 pp 3-8

[10] Libusb. *Libusb-1.0 API Reference*. Web. 18 Feb. 2016. <http://libusb.sourceforge.net/api-1.0/>.

[11] Microsoft. "WinUSB (Winusb.sys)." *(Windows Drivers)*. Web. 18 Feb. 2016. <https://msdn.microsoft.com/en-us/library/windows/hardware/ff540196(v=vs.85).aspx>