How Different Electrical Circuits of ECC Designs Influence the Shape of Power Traces measured on FPGA

Thomas Basmer, Christian Wittke, Zoya Dyka, and Peter Langendoerfer
System dept.,
IHP, Im Technologiepark 25,
15236 Frankfurt(Oder), Germany
[basmer| wittke| dyka| langendoerfer]@ihp-microelectronics.com

Abstract— *Side channel and fault attacks take advantage from the fact that the behavior of crypto implementations can be observed and provide hints that simplify revealing keys. These attacks use identical devices either for preparation of attacks or for measurements. By the preparation of attacks the structure and the electrical circuit of devices, that are identical to the target, is analyzed. By side channel attacks usually the same device is used many times for measurements, i.e. measurements on the identical device are made serially in time. Another way is to exploit the difference of side channel leakages; here two identical devices are used parallel, i.e. at the same time. In this paper we investigate the influence of the electrical circuit of a cryptographic implementation on the shape of the resulting power trace, because individualizing of circuits of cryptographic devices can be a new means to prevent attacks that use identical devices. We implemented three different designs that provide exactly the same cryptographic function, i.e. an ECC kP multiplication. For our evaluation we use two different FPGAs. The visualization of the routed design and measurement results show clear differences in the resources consumed as well as in the power traces.*

Keywords — cryptographic hardware architectures, security processors, countermeasures against side-channel attacks, FPGA

I. Introduction

With the advent of wireless sensor networks (WSN) and their uptake in industry attacks that exploit physical effects, i.e. that aim to break crypto-systems by using implementation specific information and data respectively are becoming a more and more relevant threat. This is due to the fact that devices disappearing in a WSN are somewhat normal. I.e. some devices are not connected for a while due to bad channel conditions. This means a potential attacker can grab devices bring them back into his/her lab and run all fancy types of side channel attacks.

Attacking crypto hardware is normally done in a two step approach: first preparation of the attack and second the attack against the specific device. During the preparation a certain number of devices is analyzed in order to get familiar with the design and its behavior. As a result the attack of the “real” device is simplified by this preparation phase. A precondition to run an attack in these two phases is that the attacker can get hold of sufficient identical devices. This is normally not an issue since Application Specific Integrated Circuits (ASICs) are produced in a significant number and are so cheap that an attacker can easily buy as many ASICs as needed. After such a preparation stealing devices from a WSN and running an attack is feasible and can even go undetected by the owner of the WSN.

In this paper we propose to individualize crypto devices in order to increase the effort of the attacker when it comes to preparing attacks and running attacks that rely on using identical devices. We are validating our idea using Field Programmable Gate Arrays (FPGAs). We are convinced that FPGAs will become a part of wireless sensor nodes especially to provide efficient and flexible implementations of cryptographic algorithms, as already discussed in [1].

The rest of this paper is structured as follows. In section II we explain our idea as well as the essential basics with respect to the cryptographic operations we use for individualizing crypto devices. In addition the implementations we realized are described. Section III presents the influence of individualized designs on resources consumed on FPGAs and structure of the designs. The measurement results of power traces of the individualized designs are discussed in section IV. The paper finishes with short conclusions.

II. Individualizing cryptographic designs

A means to prevent exploiting of the difference of side-channel leakages individualizing of cryptographic designs can be used. The idea is that devices with the same functionality can have a different i.e. individual structure. Important is that not only the chip topology after place-and-route but also the number of used gates is individual. This results in an individual power consumption, electromagnetic radiation, etc. Individualizing the structure of cryptographic devices prevents for example the improved power analysis attack reported in [2] or [3]. Exploiting of the difference of side-channel leakages by measurements on the same FPGA can also be prevented via generation of an individualized design for each new execution of the cryptographic algorithm.

A. Individualization of GF(2^n)-ECC designs

ECC-designs can be individualized using different multiplication methods (MM) for field multiplication. The field multiplication can be performed in two steps. The first step is the multiplication of two polynomials of length *n* that results in their (*2n-1*) bit product. The second step is the reduction of this polynomial product using the so called irreducible polynomial.

The definition of the polynomial multiplication (i.e. of the first step) is often called school or classical multiplication method. Its complexity can be given as a number of Boolean AND and XOR operations, i.e. as the number of used AND and XOR gates. To implement the multiplication of *n*-bit long polynomials using classical multiplication method n^2 AND and $(n-1)^2$ XOR gates are necessary. It is an expensive task with respect to time, area and energy since the length of multiplicands is typically large (about 200 bit); therefore many optimizations have been proposed in the past.

Many multiplication methods apply segmentation of both multiplicands into the same number of parts. The product then is calculated as a sum of smaller partial products. Historically, the first optimization was the Karatsuba multiplication method published in 1962 [4]. This method uses the segmentation of polynomials into two terms. The next one was proposed by Winograd in 1980 [5]. This method uses the segmentation of polynomials into three terms. At the moment there exist more than 10 different multiplication formulae. Each multiplication formula has its own segmentation of operands, its own number of partial products of these short – only one segment long – operands and its own number of additions of the obtained partial products, i.e. its own complexity.

Moreover the multiplication methods can be combined. Each combination of MMs also has its own complexity. In [6] and [7] different multiplication methods were combined with the goal to find only one optimal combination, i.e. the combination with minimal LUT/gate complexity and energy consumption. The set of different combinations is very large. This fact can be used for individualizing multiplier designs. In addition the selection of the combination of multiplication methods can be randomized.

B. Implemented ECC designs

To proof our idea of individualizing ECC-designs we implemented and compared three different designs of elliptic curve point multiplication or – shortly – the *kP* operation. We implemented the *kP* operation using the Montgomery elliptic curve point multiplication algorithm in Lopez-Dahab coordinates. The implementation details of one of these designs are given in [8]. All three designs differ only in their partial multiplier:

- the classical multiplication method was implemented for the first design;
- the second design uses a combination of the classical MM and our 4-segment iterative Karatsuba MM;
- the classical MM, our 4-segment iterative Karatsuba and our 3-segment iterative Winograd multiplication formulae are randomly combined for our 3-rd design.

The details about the iterative 4-segment Karatsuba MM and 3-segment Winograd MM are given in [9] and [10] respectively. The structure of these multipliers and details about all three partial multipliers are given in [16]. We do not give the details here for simplifying the reading. The important fact is that the complexity of these MMs is different.

Table I gives a short overview of parameters of the implemented multipliers.

TABLE I. PARAMETERS OF THE IMPLEMENTED PARTIAL MULTIPLIERS

Partial multipliers	***Length of operands***	**Complexity**	
		#AND	***#XOR***
Classical MM	64	64^2=4096	63^2=3969
combination of the classical MM and 4-segment iterative Karatsuba MM	64	1296	2387
random combination of classical MM, 3-segment iterative Winograd MM, and 4-segment iterative Karatsuba MM	60	1888	3172

III. INDIVIDUALIZED FPGA-RESOURCES

For evaluating our idea we used two different Xilinx FPGAs, i.e. a Kintex-7 and a Spartan-6 manufactured in different technologies to ensure that our approach really works independent of the target platform. Kintex-7 is produced in a 28-nm-technology as a Flip-Chip in a 900-pins BGA package [11]. It is a large and powerful FPGA. Our ECC designs need only about 3 % of its Look Up Tables (LUTs). The Spartan-6 FPGA is manufactured in 45-nm-technology, packaged in a 484-pins BGA package [12]. Our designs needed about 20 % of the resources of the Spartan-6.

We used the Xilinx software ISE version 14.2 (see [13], [14]) for the implementation of our individualized ECC designs. Using this software a Hardware Description Language (HDL) implementation can be converted to the bit-mask, placed and routed on the FPGA. We have used the same compiler settings as well as the same constraints for all implementations. The placed and routed designs can be visualized using an integrated FPGA editor. Not only the used LUTs but also their routing can be colored and been shown on the FPGA map. The user can define the color of different design blocks and nets. We use the green color to show the individualized multiplier in each design. The other two large blocks of our ECC design have been used unchanged and are marked in red and white. All 3 implemented ECC designs on the Kintex-7 are visualized in Fig. 1 and those on Spartan-6 are depicted in Fig. 2. Fig. 1 and Fig. 2 show only the part of the respective FPGAs that is used by the implemented designs. All pictures in Fig. 1 show the same part of the Kintex-7 FPGA map. Also Fig. 2 shows always the same part of Spartan-6 FPGA map. Fig. 1-*a* and Fig. 2-*a* depict the design using the classical MM only (further denoted as "*design1*"). Our second ECC-design using the above mentioned combination of the classical MM and the iterative 4-segment Karatsuba MM (further denoted as "*design2*") is shown in Fig. 1-*b* and Fig. 2-*b* respectively. Fig. 1-*c* and Fig. 2-*c* visualize our 3-rd design: here the multiplier is a random combination of 3 MMs as explained above (further denoted as "*design3*"). The differences in the structure are solely due to different implementations of the multiplier.

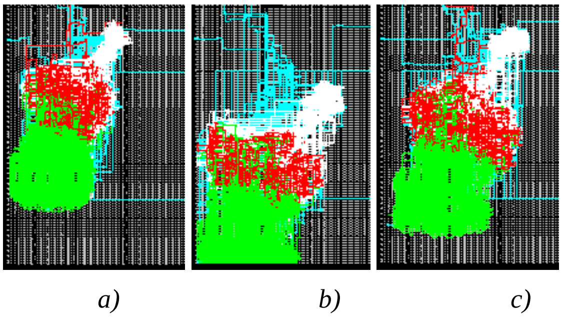

Fig. 1. Visualisation of the structure of our 3 individualized ECC designs (Kintex-7).

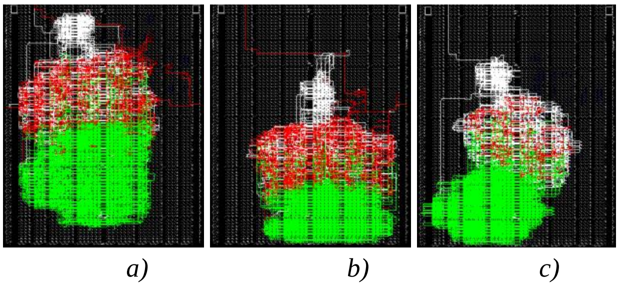

Fig. 2. Visualisation of the structure of our 3 individualized ECC designs (Spartan-6).

TABLE II. FPGA RESSOURCES OF INDIVIDUALIZED DESIGNS

FPGA	On FPGA available resources		Resources in use		
			design1	*design2*	*design3*
Kintex-7	registers	407 600	3 043	3 064	3 071
	LUTs	203 800	6 142	6 269	6 072
	slices	50 950	1 821	1 893	2 147
	nets		7 736	7 443	7 377
Spartan-6	registers	54 576	3 283	2 997	3 274
	LUTs	27 288	6 522	5 649	6 290
	slices	6 822	2 167	1 711	1 893
	nets		8 345	7 556	8 020

The visualization of the designs (see Fig. 1 and Fig. 2) and data about the resources consumed (see Table II) confirm our idea i.e. each design has an individual resource consumption and an individual structure.

IV. Measurement Results

A. Measurement setup

Fig. 3 shows our measurement setup. All our ECC designs run at 4 MHz in the Spartan-6 FPGA on the in Fault Extension Board (FEB) from TU Graz. The FEB was especially designed for the measurement of power and electromagnetic traces of designs running on the FPGA. This board has an access point for connecting a probe resistor or connecting of the Riscure current probe [15], which is what we used for our measurements. The current probe is connected to the first channel of the oscilloscope. The yellow curve on the oscilloscope displayed in Fig. 3 is a part of the power trace (PT) of the *kP* operation. Each trace was measured using LeCroy Waverunner 610Zi oscilloscope with a 2.5 GS/s sampling rate, i.e. with about 600 measurement points per clock cycle.

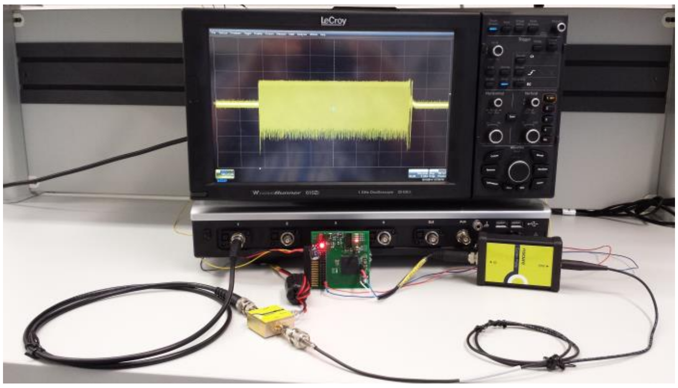

Fig. 3. Measurement setup for collecting power traces.

The measured PTs are given and discussed in the next section.

B. Individualized Power Traces

Fig. 4 shows a part of the measured traces, i.e. current that flows through the FPGA while the first 200 clock cycles of the *kP* operation are executed. More precisely the shown part of the trace corresponds to the processing of the 4-th bit of the cryptographic key. The processing of one key bit takes always 57 clock cycles in our implementations. The key is 232 bit long and the whole time of its processing is about 13000 clock cycles. To investigate the influence of the individualized designs we are using the same input for all designs. Thus, the influence of different inputs on the measurement results was excluded. The yellow line depicts the power trace of the *design1*. The violet line shows the power traces of *design2* and the blue line denotes the power trace of the *design3*.

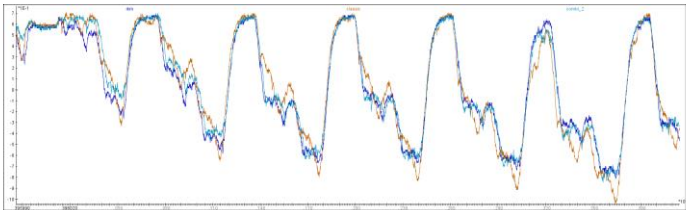

Fig. 4. Measurement results: the same part of the power trace of the *kP* operation of all three ECC designs: the yellow line depicts the traces of design1; the violet line shows design2; the blue line shows traces of design3.

The measurement results confirm our idea i.e. the shapes of the power traces are different for all three designs (see Fig. 4) even though they all process identical data.

In order to quantify the effect of our idea at least to a certain extend we compared measured traces of different designs with each other to show the differences, and we also compared the differences of repeated measurements. We did this for all three designs but are going to present these results only for *design1*. Fig. 5 shows the absolute differences of the power traces for the whole *kP* operation. For subtraction we synchronized the investigated traces using software provided by Riscure. The top curve in Fig. 5, denoted as '*difference 1*', depicts the differences between repeated measurements with the same inputs. It corresponds to the case in [2] when two identical devices process the same inputs for balancing of the bridge setup. The next two curves, denoted as '*difference 2*' and '*difference 3*', show the influence of different inputs on *design1*: the curve *difference 2* displays the differences if only one of 3 large inputs – the key – was changed and the curve *difference 3* corresponds to the case in which all 3 inputs are different. The curves *difference 4* and *difference 5* display the differences of *design1*-to-*design2* and of *design1*-to-*design3* respectively if the same inputs are processed. Note these curves visualize the influence of the individualization of the designs.

It can be seen, that the differences between two repeated measurements of the same design (the curve *difference 1* in Fig. 5 is comparable with the noise, i.e. they are balanced which is a prerequisite for bridge based measurements. Compared to that, the influence of the individualized designs however is significant (see curves *difference 4* and *difference 5* in Fig. 5 and comparable with the influence of different inputs (see curves *difference 2* and *difference 3* in Fig. 5), i.e. the measurements are not balanced. The attacks described in [2] rely on the fact that the differences in the measurements of the two devices stem only from different inputs, i.e. for the same input and the same key the differences are balanced, i.e. they look like *difference 1*. But the individualized designs lead to completely unbalanced difference so that this type of attack is no longer feasible.

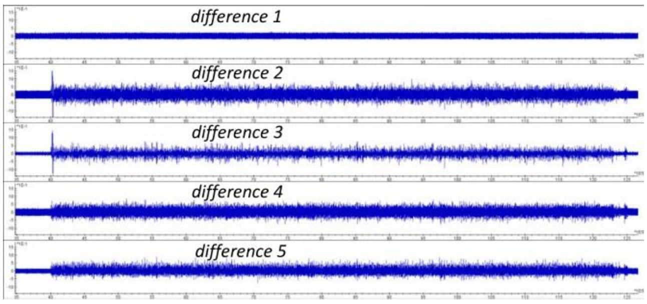

Fig. 5. Differences of measured Power Traces

Fig. 6 shows the differences of two cases that are important for assessing whether or not our approach really helps to defeat exploiting the differences of side-channel leakages:

1. *Difference 6* in Fig. 6 shows the influence of different designs (here *design1* and *design2*) including the differences caused by varying inputs (here 1 of 3 inputs was changed)
2. *Difference 7* in Fig. 6 shows the influence of different designs (here also *design1* and *design2*) including the differences caused by varying inputs (here all 3 inputs were changed)

It can be easily seen that it is infeasible to distinguish between *difference 6* and *difference 7 that* are influenced by two parameters i.e. design and input changes and *difference 2* to *difference 5* in which only a single parameter i.e. design or input was changed.

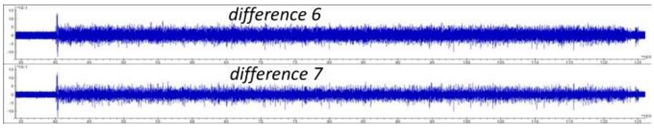

Fig. 6. Differences of measured PT: the curve *difference 6* shows the influence of design individualization (*design1*-to-*design2* i.e. the same designs were measured as for *difference 4* in Fig. 5) and at the same time the influence of inputs (one of 3 inputs was changed, as for the *difference 2* in Fig. 5). *Difference 7* shows also the influence of the individualized design (*design1*-to-*design2*) and the influence of inputs when all 3 inputs are changed (correspondends to *difference 3* in Fig. 5).

V CONCLUSION

In this paper we introduced the idea to individualize the implementation of crypto operations as a suitable means to prevent or at least to increase the effort to run successfully side channel attacks that exploit the differences of side-channel leakages. The background of the idea is straight forward. Side channel attacks and fault attacks are exploiting the fact that sufficient identical devices are available for preparing an attack. If the devices differ such kind of preparation is no longer feasible. The idea of individualizing the designs can be applied to each design, if its functionality can be implemented in different ways. We selected elliptic curve cryptography, i.e. the implementation of the required field multipliers as sample application. The advantage of this type of operation is that a plethora of different multiplication methods that provide the same operation are available. By unifying the interfaces we are capable of combining different multiplication methods. These multiplication methods can be selected at will or randomly. The differences in the observable behavior of the resulting multipliers stems from the different complexity of the multiplication methods that influences the resources needed to implement the multipliers as well as the related power consumption and electromagnetic radiation. We implemented three designs using different combinations of three MMs. Our visualization and measurement results show significant variations in resources and power traces.

In our next research steps we will run experiments using a Wheatstone bridge setup to verify that the individualization really prevents this type of attacks. We also aim at developing a metric that allows to assess how individual power traces really are in order to select the designs with the highest level of individualization.

ACKNOWLEDGEMENTS

The work presented in this paper has been partially funded by the "Ministry of Sciences, Research and Cultural Affairs (MWFK)" from resources of the European Social Fund (ESF) and of the state Brandenburg.

REFERENCES

[1] J Portilla, A Otero, E de la Torre, T Riesgo, O Stecklina, S Peter, P. Langendoerfer: *Adaptable security in wireless sensor networks by using reconfigurable ECC hardware coprocessors*, International Journal of Distributed Sensor Networks 2010, http://dx.doi.org/10.1155/2010/740823

[2] M. Hutter, M. Kirschbaum, T. Plos, J. Schmidt, S. Mangard: *Exploiting the Difference of Side-Channel Leakages*, Constructive Side-Channel Analysis and Secure Design (COSADE-2012), Lecture Notes in Computer Science Volume 7275, 2012, pp. 1-16

[3] M. Hutter, J-M. Schmidt, Th. Plos, and M. Kirschbaum: *Test Apparatus for Side-Channel Resistance Compliance Testing*, NIAT-2011, Japan, http://csrc.nist.gov/news_events/non-invasive-attack-testing-workshop/papers/11_Hutter.pdf

[4] A. Karatsuba, A., Ofman, Y.: *Multiplication of Many-Digital Numbers by Automatic Computers*. Doklady Akad. Nauk SSSR, Vol. 145 (1962), pp: 293–294. Translation in Physics-Doklady, 7 (1963), pp. 595–596.

[5] S. Winograd: *Arithmetic Complexity of Computations*. SIAM (1980)

[6] Von zur Gathen, J., Shokrollahi, J.: *Efficient FPGA-based Karatsuba multipliers for polynomials over F2*. Proc. of Selected Areas in Cryptography - SAC 2005, LNCS 3897, pp. 359-369, Springer-Verlag, Kingston, ON, Canada (2005)

[7] Z. Dyka, P. Langendoerfer, F. Vater: *Combining Multiplication Methods with Optimized Processing Sequence for Polynomial Multiplier in $GF(2^k)$*, Research in Cryptology, 4th Western European Workshop,WEWoRC-2011, Germany, 2011, Lecture Notes in Computer Science 7242, pp. 137-151, Springer-Verlag Berlin Heidelberg 2012

[8] S. Peter: *Evaluation of Design Alternatives for Flexible Elliptic Curve Hardware Accelerators*, Diplom Thesis, 2006, http://www.ics.uci.edu/~steffenp/files/da_peter.pdf

[9] Z. Dyka, P. Langendoerfer: *Area efficient hardware implementation of elliptic curve cryptography by iteratively applying Karatsubas method*, Proc. of the Design, Automation and Test in Europe (DATE 2005), 2005, Vol.3, pp: 70-75

[10] Z. Dyka, P. Langendoerfer, F. Vater, S. Peter: *Towards strong security in embedded and pervasive systems: energy and area optimized serial polynomial multipliers in $GF(2^k)$*, Proc. of IEEE New Technologies, Mobility and Security, 5th International Conference (NTMS-2012), 2012, pp. 1-6

[11] Xilinx Inc.: 7 Series FPGAs Overview, Advance Product Specification, DS180 (v1.14) July 29, 2013, http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf

[12] Xilinx Inc.: Spartan-6 Family Overview, Product Specification, DS160 (v2.0) October 25, 2011, http://www.xilinx.com/support/documentation/data_sheets/ds160.pdf

[13] Xilinx Inc.: Documentation ISE 14.2, http://www.xilinx.com/support/documentation/dt_ise14-2.htm

[14] Xilinx Inc.: ISE Design Tools, http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html

[15] Riscure: Inspector data sheet. Current Probe. https://www.riscure.com/benzine/documents/CurrentProbe.pdf

[16] Zoya Dyka, Thomas Basmer, Christian Wittke and Peter Langendoerfer: Proposing Individualization of the design of cryptographic hardware accelerators as countermeasure against structure and side channel analysis, Cryptology ePrint Archive: Report 2014/342, 15.05.2014, https://eprint.iacr.org/2014/342.pdf