Proposing Individualization of the design of cryptographic hardware accelerators as countermeasure against structure and side channel analysis

Zoya Dyka, Thomas Basmer, Christian Wittke and Peter Langendoerfer
System dept.
IHP
Frankfurt(Oder), Germany

***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 are normally prepared by analyzing devices that are identical to the real target. Here we propose to individualize the design of cryptographic devices in order 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. The synthesis and power simulation results show clear differences in the area consumed as well as in the power traces. We envision that this type of protection mechanism is relevant e.g. for wireless sensor networks from which devices can easily be stolen for further analysis in the lab.*

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

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 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 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.

The rest of this paper is structured as follows. In section II we introduce our idea as well as the essential basics with respect to the crypto graphic operations we use for individualizing crypto devices. In addition the implementations we realized are described. The results of our evaluation of the individualized designs are presented in section III. The paper finishes with short conclusions.

II. Individualizing cryptographic designs

A. Short description of our idea

The individualization of the structure of cryptographic devices can be an efficient countermeasure against improved physical attacks using bridge-based power measurements for example with the Wheatstone bridge.

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 [1].

B. 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 ($2n-1$) bits 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 because 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 [2]. This method uses the segmentation of polynomials into two terms. The next one was proposed by Winograd in 1980 [3]. 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 has also its own complexity. In [4] and [5] 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 big. This fact can be used for individualizing multiplier designs. In addition the selection of the combination of multiplication methods can be randomized.

C. 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 [6]. All three designs differ only in their 60-bit 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 our iterative 4-segment Karatsuba MM and our 3-segment Winograd MM are given in [7] and [8]. We do not give the details here for the simplifying the reading. Important is only the fact, that the complexity of these MMs is different.

Table I gives a short overview of parameters of used multiplication methods. We denote the length of multiplicands as *n*. The partitioning of multiplicands is denoted as the number of segments *S*. In this case the length of segments is *m=n/S*. The number of partial products of *m*-bit long operands that is required by the MM is denoted as the number of partial multipliers *#PM*. The number of XOR gates for the calculation of inputs of the partial multipliers and for the accumulation of all partial products is denoted as #XOR.

Note that each partial multiplier (PM) is an independent multiplier for *m*-bit long operands. I.e. any MM can be selected to implement it.

TABLE I. SHORT DESCRIPTION OF USED MMS

MM	***n***	***S***	**m**	**Complexity**		**Complexity of PM**
				#PM	***#XOR***	
classical	64	1	64	1	0	64^2 AND, 63^2 XOR
iterative 4-segment Karatsuba	64	4	16	9	34m-11	The complexity of each of the 9 PM depends on the MM that implements this PM
iterative 3-segment Winograd	60	3	20	6	18m-6	The complexity of each of the 6 PM depends on the MM that implements this PM

The structures of the multipliers listed in Table I are given in Fig.1-Fig.3.

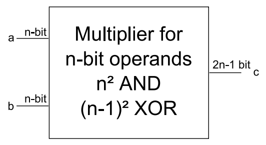

Fig. 1. Structure of the n bit multiplier using the classical MM

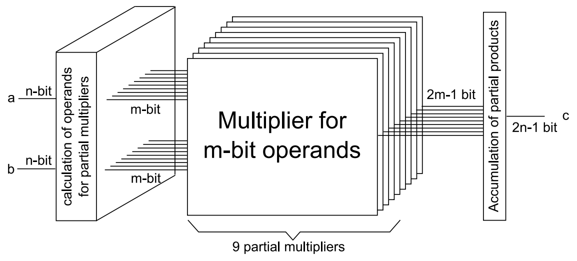
a
n-bit
b
n-bit
calculation of operands for partial multipliers
m-bit
m-bit
Multiplier for m-bit operands
2m-1 bit
Accumulation of partial products
2n-1 bit
c
9 partial multipliers

Fig. 2. Structure of the n bit multiplier using the iterative 4-segment Karatsuba MM

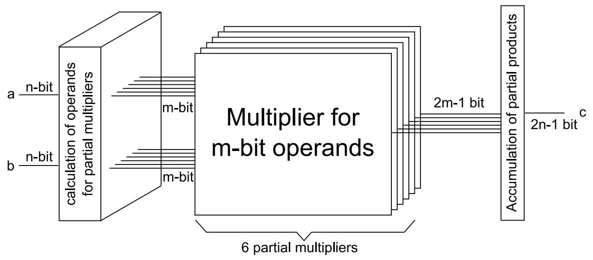
a
n-bit
b
n-bit
calculation of operands for partial multipliers
m-bit
m-bit
Multiplier for m-bit operands
2m-1 bit
Accumulation of partial products
2n-1 bit
c
6 partial multipliers

Fig. 3. Structure of the n bit multiplier using the iterative 3-segment Winograd MM

As explained above we implemented the classical 60-bit multiplier for our first design. Its structure is shown in Fig.1. Our second design is the area optimized combination of the classical MM and our 4-segment iterative Karatsuba MM. It has three hierarchical levels. Its 60-bit multiplier is implemented using 4-segment Karatsuba MM (see Fig. 2, n=60, m=15), resulting in 9 partial multipliers which again are implemented as 4-segment Karatsuba MM (see Fig. 2, n=16, m=4), resulting again in 9 partial multipliers which then are implemented using the classical MM. Fig. 4 depicts the structure of this multiplier. In fig. 4 the green boxes indicate which part is implemented using the Karatsuba MM, and the yellow boxes indicate where the classical 4-bit multiplier is used.

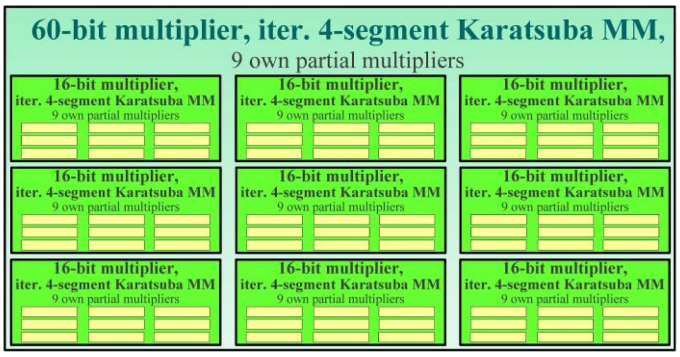
60-bit multiplier, iter. 4-segment Karatsuba MM,
9 own partial multipliers
16-bit multiplier,
iter. 4-segment Karatsuba MM
9 own partial multipliers

Fig. 4. Structure of the 60 bit multiplier: the area optimized combination of the classical MM and the iterative 4-segment Karatsuba MM.

The combination of MMs used for our 3-rd design was selected randomly and is shown in Fig. 5.

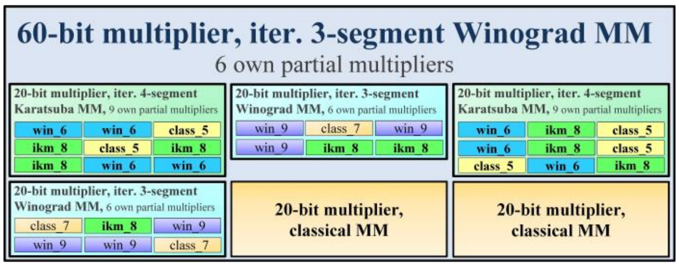
60-bit multiplier, iter. 3-segment Winograd MM
6 own partial multipliers
20-bit multiplier, iter. 4-segment Karatsuba MM, 9 own partial multipliers
win_6
win_6
class_5
ikm_8
class_5
ikm_8
ikm_8
win_6
win_6
20-bit multiplier, iter. 3-segment Winograd MM, 6 own partial multipliers
win_9
class_7
win_9
win_9
ikm_8
ikm_8
20-bit multiplier, iter. 4-segment Karatsuba MM, 9 own partial multipliers
win_6
ikm_8
class_5
win_6
ikm_8
class_5
class_5
win_6
ikm_8
20-bit multiplier, iter. 3-segment Winograd MM, 6 own partial multipliers
class_7
ikm_8
win_9
win_9
win_9
class_7
20-bit multiplier, classical MM
20-bit multiplier, classical MM

Fig. 5. Structure of the 60 bit multiplier: a random combination of 3 MMs.

Partial multipliers for small length operands in Fig.5 are implemented as follows:

- win_6 using 3-segment Winograd MM (see Fig. 3, n=6, m=2);
- ikm_8 using 4-segment Karatsuba MM (see Fig. 2, n=8, m=2);
- win_9 using 3-segment Winograd MM (see Fig. 3, n=9, m=3);

whereby all multipliers for operands with the length of $n \leq 3$ are implemented using classical MM.

III. RESULTS

The synthesis of all designs and the following analysis of their power consumption for our IHP 0.13µ technology [9] confirm our assumption: each design has an individual area and an individual power consumption (see Table II).

TABLE II. AREA AND POWER CONSUMPTION OF THE IMPLEMENTED *ECC*-DESIGNS

Implemented ECC-Design with multiplier using		*Area*	Power consumption
1	classical MM	0,331mm^2	7,42mW
2	area-optimized combination of MMs	0,280mm^2	5,41mW
3	random combination of MM	0,296mm^2	5,36mW

Fig. 6 shows the first 150 clock cycles of the *kP* operation. The red line depicts the power trace of the design using the classical MM only. The green line shows our second ECC-design with the area optimized multiplier. The violet line denotes the power trace of our 3-rd design: this multiplier is the random combination of 3 MMs as it was explained above. All power traces are simulated using PrimeTime [10] for the IHP 0.13µ technology.

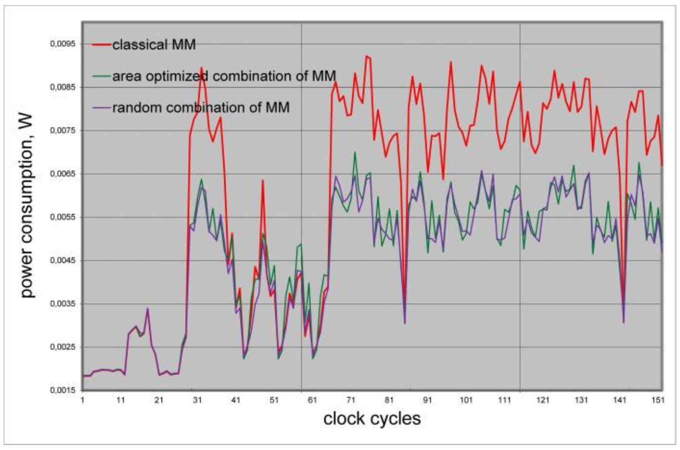

Fig. 6. Simulation results: first 150 clock cycles of the power traces of all three implemented ECC designs

The simulation results confirm our idea: the shapes of the power traces are different for all three designs. Fig. 7 depicts the differences in the power consumption[1] between all combinations of the three designs.

[1] Please note that in [1] the authors compare real devices and provide measured voltage traces. In our next research steps we will use the results of [1] as a benchmark for the evaluation of our idea.

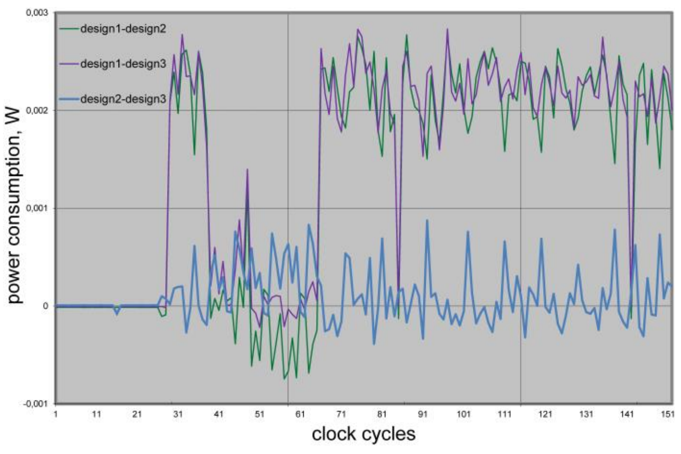

Fig. 7. Differences in power consumption of all combinations of the implemented designs

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 specific improved side channel and structure analysis based attacks. 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. 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 influence the area needed to implement the multipliers as well as the related power consumption. We implemented three designs using different combinations of three MMs. Our synthesis and simulation results show significant variations with respect to area and power consumption. The differences in the area reach from 5 per cent to 18 per cent and in the power consumption from 2 per cent to 38 per cent. Also the power traces have pretty individual shapes.

In our next research steps we will run experiments using a Wheatstone bride set-up 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.

We are aware of the fact that the production of individualized designs for ASICs is very expensive. Therefore FPGAs can be chosen as a possible implementation platform for individualized cryptographic devices.

[1] 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

[2] 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.

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

[4] 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)

[5] 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 EuropeanWorkshop,WEWoRC-2011, Germany, 2011, Lecture Notes in Computer Science 7242, pp. 137-151, Springer-Verlag Berlin Heidelberg 2012

[6] 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

[7] 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

[8] 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

[9] IHP: Innovations for High Performance Microelectronics, http://www.ihp-microelectronics.com/

[10] Synopsis, http://www.synopsys.com/Tools/Implementation/SignOff/Pages/PrimeTime.aspx