Individualizing Electrical Circuits of Cryptographic Devices as a Means to Hinder Tampering Attacks

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 provides hints that simplify revealing keys. In a real word a lot of devices, that are identical to the target device, can be attacked before attacking the real target to increase the success of the attack. Their package can be opened and their electromagnetic radiation and structure can be analyzed. Another example of how to improve significantly the success rate of attacks is the measurement of the difference of the side channel leakage of two identical devices, one of these devices being the target, using the Wheatstone bridge measurement setup. Here we propose to individualize the electrical circuit of cryptographic devices in order to prevent attacks that use identical devices: attacks, that analyze the structure of devices identical to the target device in a preparation phase; usual side channel attacks, that use always the same target device for collecting many traces, and attacks that use two identical devices at the same time for measuring the difference of side-channel leakages. The proposed individualization can prevent such attacks because the power consumption and the electromagnetic radiation of devices with individualized electrical circuit are individualized while providing the same functionality. We implemented three individualized ECC designs that provide exactly the same cryptographic function on a Spartan-6 FPGA. These designs differ from each other in a single block only, i.e. in the field multiplier. The visualization of the routed design and measurement results show clear differences in the topology, in the resources consumed as well as in the power and electromagnetic traces. We show that the influence of the individualized designs on the power traces is comparable with the influence of inputs. These facts show that individualizing of electrical circuits of cryptographic devices can be exploited as a protection mechanism. We envision that this type of protection mechanism is relevant if an attacker has a physical access to the cryptographic devices, e.g. for wireless sensor networks from which devices can easily be stolen for further analysis in the lab.

Keywords: field multiplication, individualizing electrical circuit of multiplier, power traces, electromagnetic traces, countermeasures against side-channel attacks.

1 Introduction

Side Channel Attacks (SCA) exploit the fact that physical effects such as time, power consumption and electromagnetic radiation of the running chips can be measured. The assumption on which the attacks are based and which makes the attacks feasible at all is that some parts of the whole system are constant. The constant parts are the secret key that is the target of the attacker and the implemented cryptographic algorithm, i.e. the circuit of the target device itself or the program which processes the input data and the key. In order to extract the secret key an attacker can select specific input data and/or run the device with a selected "key" i.e. a scalar that is processed with the input data in the same manner as the secret key. In both cases the attacker can record as many power or electromagnetic traces has she/he wants. The shape of the traces is influenced up to a certain extent by the secret key and the input data or by the selected "key" and the input data. Both types of traces can then be further processed in order to extract the secret key. This is feasible since the secret key and the implemented circuit are fixed, i.e. there is a somewhat "constant" component in the traces. In other words the input data selected by the attacker has a significant influence on the shape of the measured traces. This influence can then be analyzed by the attacker. In order to protect cryptographic implementations against side channel attacks designers try to withhold information from the attacker by blinding of the input data, randomization of the key or by randomizing the algorithm. Thus the attacker will no longer reach his goal by just altering the input data, since not what the attacker selected is processed but data altered by the implementation. So the attacker can no longer analyze the influence of the data and key he provided on the shape of the power traces. But meanwhile more sophisticated attacks such as those described in [1] and [2], which allow to extract secret key even in case countermeasures are deployed.

In this paper we introduce a radically different approach. We are proposing to realize the cryptographic algorithm so that its circuit behaves different whenever it is executed. The core of our idea is that some parts of cryptographic algorithms, for example complex mathematical operations, can be implemented using different formulas. This leads to a different circuit in a hardware implementation or to different processing of the corresponding software implementation. This has the same effect as blinding i.e. the relation between the processing of the input data and the shape of measured power trace can no longer be exploited to extract the key. In contrast to the usually used countermeasure such as blinding our approach does not have any additional operations that can be attacked.

Different implementations can be selected for each new execution randomly, if the cryptographic algorithm runs in software or on an FPGA. Another way, that makes our idea applicable also for ASIC realization, is to implement more blocks with the same functionality but with individualized circuits. The selection of the necessary subset of these blocks can be done randomly for each execution, i.e. the observable behavior of circuit will be individualized dynamically since the parts of the circuit that are active differ from execution to execution.

In this paper we are discussing the applicability of our idea for elliptic curve cryptography (ECC). In our implementation of the elliptic curve (EC) point multiplication we are using different multiplication formulas for implementing three individualized partial multipliers for the field multiplication. Our evaluation based on measurements of power and electromagnetic traces on an FPGA shows clearly that the electrical circuit of the partial multiplier influences the shape of traces comparable to the influence of different inputs. This fact can be exploited to protect the implementation against side channel attacks, if the electrical circuit could be individualized while performing cryptographic operations. The proposed method on the basis of individualizing only one block of ECC design – the partial multiplier – allows to randomize the power consumption with highest possible granularity – i.e. clockwise. We demonstrated the applicability of this method as a countermeasure using individualized ECC designs. We use only three individualized designs to evaluate and visualize the effect of the individualization. But the same effect is reached if the circuit is individualized before each execution of the *kP* operation. Even though we are focusing here on ECC our approach can also be used for implementing other cryptographic algorithms such as RSA.

The rest of this paper is structured as follows. In section 2 we introduce an example of attacks that exploit differences in side channel leakage relying on the fact that identical devices can be used. In the following section 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 and the influence of individualized designs on resources consumed on FPGAs and on the structure of the designs is shown. Section 4 presents the measurement results of power traces and electromagnetic traces of the individualized designs. In section 5 we present the evaluation of our approach that clearly shows that individualizing of circuits of the partial multiplier can be applied as a countermeasure against power analysis. The paper finishes with short conclusions and an outlook on next steps.

2 Attacks exploiting differences in side channel leakage

In the recent past improved physical attacks using bridge-based power measurements, for example with the Wheatstone bridge as it is described in [3] or [4], have been reported. For this type of attack two identical devices providing exactly the same cryptographic function are necessary. The measurement setup is shown in Fig.1.

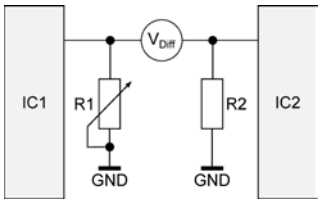


Fig. 1. Wheatstone bridge based measurement setup from [3]

Theoretically the measured voltage difference equals to '0', if the bridge is balanced. It means: the both cryptographic devices IC1 and IC2, including the resistance of their wires, are identical and working synchronized with the same input data and cryptographic key, i.e. they perform the same operations at the same time. These theoretically easy conditions are a problem in practice: the different resistance of the wires can be balanced manually by varying of the resistor R1, but the task to find two really identical devices and to run they synchronized is difficult.

If the bridge was balanced and the both devices process the same data with different keys, for example IC1 with the secret key and IC2 with a selected key candidate, the measured voltage difference as it is shown in Fig.1 is the directly measured difference of the power traces of the both devices with the reduced noise. As it is shown in [4] and [3] the analysis of such measured traces can increase the efficiency of side-channel attacks significantly because the measured traces show the amplified difference of side-channel leakages of both crypto devices. The number of traces needed for a successful black-box correlation power analysis attack on the GRANDESCA AES coprocessor implemented in CMOS logic was reduced in [3] and [4] by factor 3. This improvement is achieved due to the following facts:

- The coherent noise, i.e. the noise from the same power supply as well as from the environment (for example the noise that depends on the electromagnetic radiation of the environment), was canceled.
- The difference of power traces of both identical devices depends on inputs and secret keys of both devices only. This difference was amplified, measured and analyzed.

Note the noise generated by each of the devices itself is not coherent and for this reason their difference cannot be canceled or reduced.

Authors of [3] and [4] describe and evaluate their improved attack on the example of AES as well as on the example of the moving operation: i.e. moving an input byte from the memory to a working register of the CPU.

We applied the attack for ECC. We asserted that the preparation of this kind of attack is complex. Both devices have to be identical and in addition they need to work closely synchronized. In case of ECC, the amplitude of the signal is high compared to the noise. In this case the noise cancelation as it is achieved by the bridge measurement does not provide a significant benefit. In more clear words we are convinced that doing repeated measurements on the same target device is more effective and much simpler to do.

In the rest of this paper we investigate the design individualization as a possible countermeasure against usual SCA attacks, where all measurements are made on the same – i.e. a fully identical – device using ECC as an example.

3 Individualizing cryptographic designs

To prevent attacks using identical devices or repeated measurements on the same device we propose individualizing of cryptographic designs. The idea is that devices with the same functionality can have a different i.e. individual structure. Important is

that not only the design topology after place-and-route but also the number of used gates i.e. the number and kind of programmed LUTs (look up table) functionality for FPGAs are individual. This results in an individual electrical circuit, power consumption, electromagnetic radiation, etc. and prevents for example the improved power analysis attack reported in [3], since the attacker cannot balance the Wheatstone bridge for devices being that different. Also attacks that use direct comparison of measured traces as it is described in [19] or the doubling attack [20] and other collision based attacks [21] can be avoided. If the circuits will be individualized before each processing of the cryptographic operation, it can be used as a countermeasure against such attacks for FPGA implementations.
If the circuits will be individualized clockwise, it can be used as a countermeasure against wide spectrum of SCA attacks, for example against horizontal and vertical attacks [22].

3.1 Individualization of $GF(2^r)$-ECC designs

In this subsection we explain an easy way to individualize any ECC design using EC over extended binary Galois fields $GF(2^r)$ as example.

The main ECC operation is the elliptic curve point scalar multiplication denoted as kP. Here k is a big binary number and P is a point on the given elliptic curve.

The kP operation is based on field operations: addition, squaring, multiplication and division of long binary numbers that represent the elements of the extended binary Galois field $GF(2^r)$. The most complex field operation is the division, but it can be implemented as a sequence of squaring and multiplication using the Fermat theorem. The second complex operation in the kP operation is the multiplication of elements of $GF(2^r)$, which is performed very often. The energy consumption of the multiplier is large and can define the profile of the measured power trace of the whole ECC design.

The polynomial multiplication (i.e. the first step of the multiplication of elements of $GF(2^r)$) can be realized by applying the school or the 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 r-bit long polynomials using the classical multiplication method r^2 AND and $(r-1)^2$ XOR gates are necessary. This results in an expensive implementation with respect to area and energy since the length of multiplicands is typically large (about 200 bit). In order to tackle this complexity issue many optimizations, i.e. new multiplication formulae, have been proposed in the past. Many multiplication methods apply segmentation of both multiplicands into the same number of parts. The product is then calculated as a sum of smaller partial products. Historically, the first optimization was the Karatsuba multiplication method published in 1962 [5]. This method uses the segmentation of polynomials into two terms. The next multiplication formula was proposed by Winograd in 1980 [6]. 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 term long – operands and its own number of additions of the partial products, i.e. its own individual complexity.

In addition the multiplication methods can be combined. Each combination of multiplication methods (MM) has also its own complexity. In [7] and [8] different multiplication methods were combined with the goal to find the 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 exploited for individualizing any multiplier design. In addition the selection of the combination of multiplication methods can be randomized.

To summarize ECC designs can be individualized by using different multiplication methods and the huge number of their possible combinations for the implementation of the field multiplication. That way billions of individualized designs can be obtained.

3.2 Implemented ECC designs

Our ECC implementation is a hardware accelerator for the *kP* operation on the elliptic curve *B-233* [9] over $GF(2^{233})$.

The *kP* operation was implemented using the Montgomery elliptic curve point multiplication algorithm in Lopez-Dahab coordinates. The implementation details of one of the designs we used for the investigation reported here are given in [10].
An operand-wide multiplier, i.e. the multiplier for $r=233$ bit long operands, would require a large area resulting in high production costs. To decrease the area we implemented the multiplication of the 233 bit long operands serially using a partial multiplier (PM) only for 64 bit long operands that requires 9 clock cycles to complete the multiplication. The area of the partial multiplier is about 10% of the area of the whole *kP* design.

To validate our idea of individualizing ECC designs we implemented and compared three different designs of the elliptic curve point multiplication *kP*. All three designs differ only in their partial multiplier:

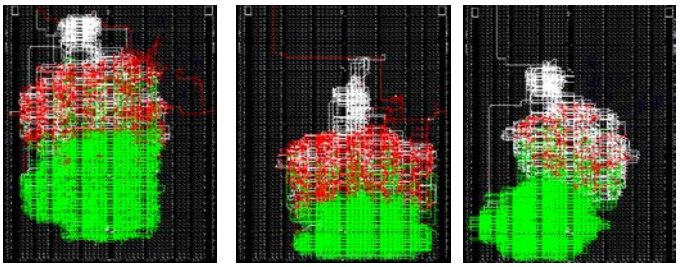
- the first design contains a PM for 64 bit operands that was implemented using only the classical multiplication method, its complexity is 642=4096 AND and 632=3969 XOR gates;
- the second design contains a 64 bit PM that uses a combination of the classical MM and the 4-segment iterative Karatsuba MM [11], its complexity is 1296 AND and 2387 XOR gates;
- we selected randomly a combination of multiplication methods from a set consisting of the classical MM, the 4-segment iterative Karatsuba and the 3-segment iterative Winograd multiplication [12] method in order to determine the implementation of the 60 bit PM for our 3-rd design, its complexity is 1888 AND gates 3172 XOR gates.

The implementation details about the partial multipliers used here are given in [13]. We do not provide details here for simplifying the reading. The important fact is that the complexity i.e. the number of AND and XOR gates of all these PMs is different.

3.3 Individualized FPGA-Resources

For evaluating our idea we used a Xilinx FPGA Spartan-6. The Spartan-6 FPGA is manufactured in a 45-nm-technology and packaged in a 484-pins BGA package [14]. Our designs needed about 20 % of the resources of the Spartan-6.

We used the Xilinx software ISE version 14.2 (see [15], [16]) for implementing our individualized ECC designs. 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 shown on the FPGA map. The user can define the color of different design blocks and nets. For each design we use green color to show the most interesting block of our *kP*-design – the serial field multiplier with individualized PM. The other large blocks of our *kP*-design have been used identically in all three designs and are marked in red and white. All 3 ECC designs are visualized using the Xilinx ISE tools and are given in **Fig. 2**. **Fig. 2** shows always the same part of Spartan-6 FPGA map. **Fig. 2**-*a)* depicts the design using the PM implemented with classical MM further denoted as "*design1*". Our second ECC design using the above mentioned combination of the iterative 4-segment Karatsuba MM and the classical MM for the implementation of PM (further denoted as "*design2*") is shown in **Fig. 2**-*b)*. **Fig. 2**-*c)* visualizes our 3-rd design, further denoted as "*design3*", in which the partial multiplier is a random combination of 3 MMs as explained above. The differences in the structure are solely due to different implementations of the partial multiplier.



a) *b)* *c)*

Fig. 2. Visualization of the structure of our three individualized ECC designs on a Spartan-6. The individualized field multipliers are colored green. The rest of ECC designs was not changed, i.e. it is the same for all three designs. The differences are solely due to different – individualized – PMs in the field multiplier block.

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

Table 1. FPGA Spartan-6 resources of individualized designs

On FPGA available resources		Resources consumed		
		design1	*design2*	*design3*
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

4 Measurement Results

4.1 Measurement setup

Fig. 3 shows our measurement setup. All our ECC designs run at 4 MHz in the Spartan-6 FPGA on the 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 [17], which is what we used for our measurements. The 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. The red curve shows the corresponding electromagnetic trace (EMT). We used the shielded high sensitivity Riscure electromagnetic probe 4.0 [18] for the measurement of electromagnetic traces. Both traces – the PT and the EMT – are measured in parallel, i.e. at the same time. 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.

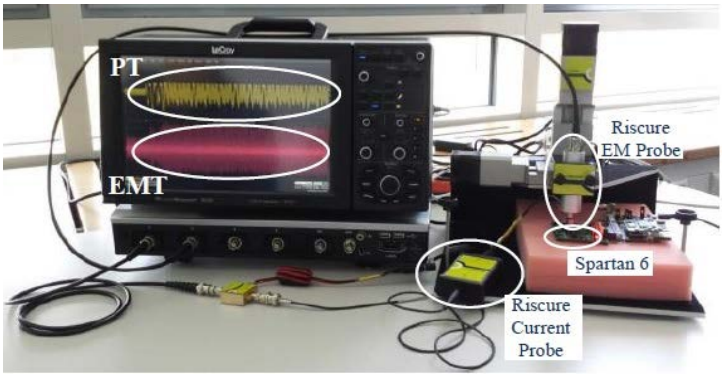


Fig. 3. Measurement setup for collecting power and electromagnetic traces at the same time.

The FPGA was not decapsulated and we decided not make any electromagnetic cartography of the chip surface.

For our measurements we placed the electromagnetic probe over the middle of the FPGA close to the package surface. The position of EM-probe over the FPGA surface was the same during all measurements. We decided to do our measurements in this way for the following reasons:

- The inner diameter of the shielding of the Riscure probe we used is about 6mm, i.e. it covers almost the complete 7mm x7mm IC of the FPGA
- We wanted to avoid noise stemming from bond wires
- We wanted to have identical conditions for all designs.

Each of the measured ECC designs has its own individual structure (see section 3) resulting in its individual power consumption and electromagnetic radiation. The measured PTs and EMTs are given and discussed in the next section.

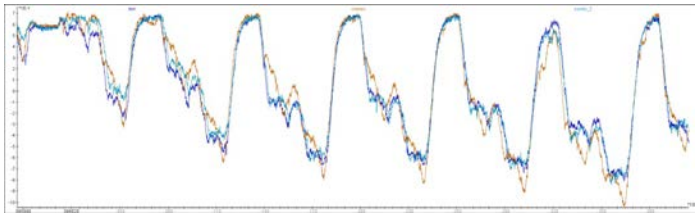
4.2 Individualized Power and Electromagnetic Traces

We measured power traces using the Riscure current probe. All designs have the same input data: the elliptic curve point $P_1=(x_1,y_1)$ and scalar k_1. The length of the scalar k_1 is 232 bit and its processing takes about 13000 clock cycles. Each clock cycle is 250 ns long. The exact values are in hexadecimal:

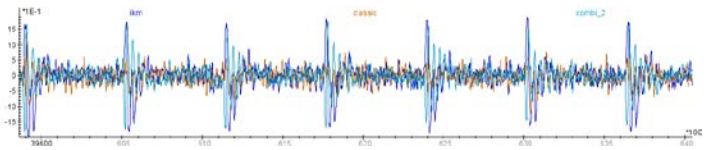
x_1=*181856adc1e7df1378491fa736f2d02e8acf1b9425eb2b061ff0e9e8246*
y_1=*89fed47b796480499cbaa86d8eb39457c49d5bf345a0757e46e2582de6*
k_1=*93919255fd4359f4c2b67dea456ef70a545a9c44d46f7f409f96cb52cc*

All PTs were measured for processing of the $k_1 \cdot P_1$ operation. **Fig. 4** shows a part of the measured traces for all three individualized designs: **Fig. 4**-*a)* shows the PTs and **Fig. 4**-*b)* shows the EMTs.

All traces were recorded and then synchronized using the Riscure software. The voltage value on the *y*-axis in **Fig. 4**-*a)* is between -1V and +0,7V and in **Fig. 4**-*b)* between -2V and +2V. The shown part of the trace corresponds the first 7 clock cycles of 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 numbers on the *x* axis in **Fig. 4** are not points in time but the numbers of measured samples. This representation is required by the Riscure Software.



a)



b)

Fig. 4. Measurement results: the same part of the traces of the *kP* operation of all three ECC designs: *a)* – power traces *b)* – electromagnetic traces: 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 and electromagnetic traces are significantly different for all three designs (see **Fig. 4**).

5 Evaluation of our approach

In order to quantify the effect of our idea and to examine, if the proposed kind of the ECC design individualization can be a new means counter the SCA attacks we did the following:

1. For each of the three designs we measured the power trace of the $k_1 \cdot P_1$ operation twice. We synchronized the traces of the repeated measurements and calculated their difference curve individually for each design using the Riscure software. **Fig. 5** shows the calculated difference on the example of *design2*.

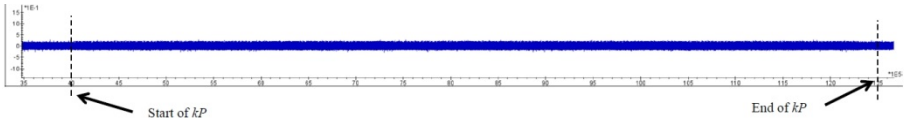


Fig. 5. Calculated difference of traces of the repeatedly measured $k_1 \cdot P_1$ operation. Measurements done on FPGA Spartan-6 with *design2*.

The start and the end of the *kP* operation are marked in **Fig. 5**. These points could be found only using the original traces, because the values of the difference of both traces are comparable with the noise before the start or after the end of the *kP* operation.

2. To show, if the side-channel leakage in a certain *kP* design is significant, the PTs with different inputs, i.e. with different EC points and/or key candidates, can be measured and their difference can be calculated. The difference curve shows the influence of inputs on the shape of PTs and represents the side channel leakage. **Fig. 6** shows the difference of a PT of $k_1 \cdot P_1$ and a PT of $k_2 \cdot P_1$ operation, i.e. the difference of these two *kP* traces with different keys. All 232 bits of the scalar k_2 are set to the binary '1': k_2=111…1.

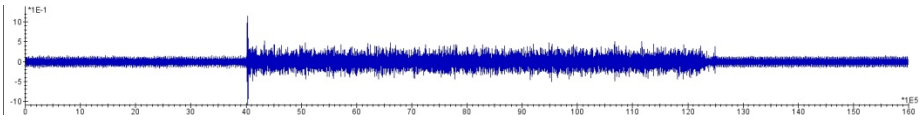


Fig. 6. Calculated difference of traces of the $k_1 \cdot P_1$ and $k_2 \cdot P_1$ operation. Measurements done on FPGA Spartan-6 with *design2*.

The start of the *kP* operation, the data processing and the end of the operation are clearly to be seen in **Fig. 6**: the difference of the PTs is significantly higher than the noise. This side channel leakage can for example be used to extract the scalar k_1. We use this leakage for a comparative analysis directly, without first calculating

correlation coefficients detecting pretty similar parts of two traces and without repeated measurements to reduce the noise as in [19].

We performed the attack as follows:

Step 1: we set the second most significant bit of k_2 to 0. All other bits are set to 1. Our (binary) key-candidate is: k_3=10111…1. We measured the trace of $k_3 \cdot P_1$, then synchronized it using the Riscure software with the trace of $k_1 \cdot P_1$ and subtracted both traces. We have seen that a small part of the difference curve is now comparable to the noise. This fact tells us, that the two most significant bits of our key-candidate are correct.

Step 2: we set the third most significant bit of k_3 to 0. Our next (binary) key-candidate is: k_4=10011…1. We measured the trace of the $k_4 \cdot P_1$ operation, synchronized it with the trace of the $k_1 \cdot P_1$ operation and subtracted both traces. We have seen that the part of the difference curve is comparable with the noise and is not two times but 3 times longer in comparison to the difference curve of step 1. This fact tells us, that in this step we extracted not only the next one, but the next two bits of the key correctly, i.e. already the 4 most significant bits of our key-candidate are correct.

We repeated this approach many times. **Fig. 7** shows the difference curve of two traces, when the most left quarter of the key-candidate is already correct.

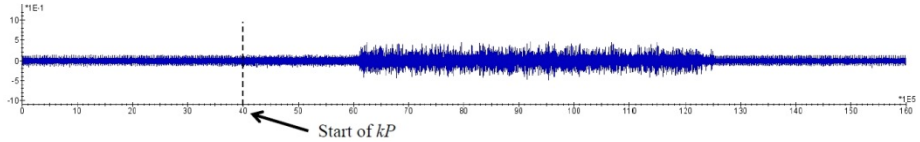


Fig. 7. Calculated difference of the traces of $k_1 \cdot P_1$ and the *key_candidate*$\cdot P_1$, the quarter with the most significant bits of the *key_candidate* is the same as the quarter with the most significant bits of k_1 and the rest differs. Measurements done on a Spartan-6 FPGA running *design2*.

This type of attack can be prevented if for each *kP* operation the attacked FPGA is programmed with a new individualized design. **Fig. 8** illustrates this.

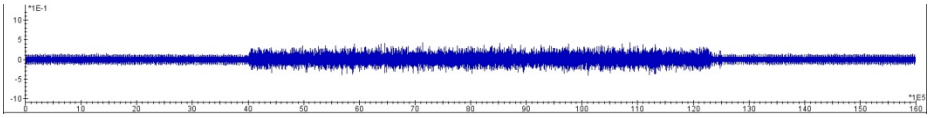


Fig. 8. Calculated difference of traces of the same operation – $k_1 \cdot P_1$ – executed by different designs: *design2* and *design3*. Measurements done on a Spartan-6 FPGA running *design2* and *design3*.

The comparison of **Fig. 6** (different inputs, the same designs) with **Fig. 8** (the same inputs, different designs) shows that the influence of individualized designs on the shape of PTs is comparable to the influence of inputs.

6 Conclusions

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 attacks that exploit identical devices. 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, power traces and electromagnetic traces.

In our next research steps we will adapt our approach in such a way that it can be applied to ECC implementations as ASICs. We have a first idea how to do that and our first rough estimate of the area overhead that is about 15 per cent, and allows to use about 1000 individualized designs of partial multipliers in a single ASIC.

Acknowledgment
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. Goubin, L.: A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems. In: Desmedt, Y.G. (ed.) Public Key Cryptography – PKC-2003, LNCS, vol. 2567, pp. 199--211. Springer, Heidelberg (2003).
2. Fouque, P., Real, D., Valette, F., Drissi, M.: The Carry Leakage on the Randomized Exponent Countermeasure. In: Proc. of Workshop on Cryptographic Hardware and Embedded Systems – CHES-2008, LNCS vol. 5154, pp. 198--213. Springer (2008)
3. Hutter, M., Kirschbaum, M., Plos, Th., Schmidt, J.-M., Mangard, S.: *Exploiting the Difference of Side-Channel Leakages*, Constructive Side-Channel Analysis and Secure Design (COSADE-2012), LNCS Volume 7275, 2012, pp. 1-16
4. Hutter, M., Schmidt, J-M., Plos, Th., Kirschbaum, M.: *Test Apparatus for Side-Channel Resistance Compliance Testing*, NIAT-2011, Japan
5. 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.

6. Winograd, S.: *Arithmetic Complexity of Computations*. SIAM (1980)
7. 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)
8. Dyka, Z., Langendoerfer, P., Vater, F.: *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
9. NIST Digital Signature Standard (DSS), FIPS PUB 186-3, Juni 2009, http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf
10. Peter, S.: *Evaluation of Design Alternatives for Flexible Elliptic Curve Hardware Accelerators*, Diplom Thesis, 2006
11. Dyka, Z., Langendoerfer, P.: *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
12. Dyka, Z., Langendoerfer, P., Vater, F., Peter, S.: *Towards strong security in embedded and pervasive systems: energy and area optimized serial polynomial multipliers in GF(2k)*, Proc. of IEEE New Technologies, Mobility and Security, 5th International Conference (NTMS-2012), 2012, pp. 1-6
13. Dyka, Z., Basmer, Th., Wittke, Ch. Langendoerfer, P.: 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
14. Xilinx Inc.: Spartan-6 Family Overview, Product Specification, DS160 (v2.0) October 25, 2011, http://www.xilinx.com/support/documentation/data_sheets/ds160.pdf
15. Xilinx Inc.: Documentation ISE 14.2, http://www.xilinx.com/support/documentation/dt_ise14-2.htm
16. Xilinx Inc.: ISE Design Tools, http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html
17. Riscure: Inspector data sheet. Current Probe. https://www.riscure.com/benzine/documents/CurrentProbe.pdf
18. Riscure: Inspector data sheet: EM Probe Station. https://www.riscure.com/benzine/documents/EMProbeStation.pdf
19. T. S. Messerges, E. A. Dabbish, R. H. Sloan: Power Analysis Attacks of Modular Exponentiation in Smartcards. Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems – CHES 1999, LNCS Vol. 1717, pp. 144−157, Springer Berlin Heidelberg, 1999
20. P.-A. Fouque, F. Valette: The Doubling Attack – Why Upwards Is Better than Downwards. Cryptographic Hardware and Embedded Systems - CHES 2003, LNCS Vol. 2779, pp. 269-280, Springer Berlin Heidelberg, 2003
21. N. Homma, Atsushi Miyamoto, Takafumi Aoki, Akashi Satoh, Adi Shamir: Collision-based Power Analysis of Modular Exponentiation Using Chosen-message. Proceedings of the 10th International Workshop – CHES 2008, LNCS Vol. 5154, pp. 15-29, Springer Berlin Heidelberg, 2008
22. C. Clavier, B. Feix, G. Gagnerot, M. Roussellet, V. Verneuil: *Horizontal Correlation Analysis on Exponentiation*. Proceedings of the 12th International Conference on Information and Communications Security – ICICS 2010, December 15-17, 2010, Barcelona, Spain, LNCS Volume 6476, pp. 46-61, Springer Berlin Heidelberg, 2010