Zdenek MARTINASEK[1], Tomas MACHA[1], Ondrej RASO[1], Josef MARTINASEK[1], Pavel SILHAVY[1]

Brno University of Technology(1),

Optimization of differential power analysis

Abstract. *The article describes the optimization of differential side channel analysis which is often used in differential power analysis. The introductory chapters discusses in great detail the theoretical background of method. The new improvement is proposed, which allow reduction of the required calculations by* 73% *and it is in the worst case. The method is based on knowledge of the Hamming weight of secret key. Hamming weight of secret key can be determined by adding only one reference measurement of power consumption. The testbed focused on measuring direct emissions was built to verify proposal optimization and experimental verification was carried out.*

Streszczenie. *W artykule opisano metodę optymalizacji różnicowej analizy mocy. Metoda bazuje na znajomości wag Hamminga. Są one określane przez dodanie jednego pomiaru poboru mocy odniesienia.* ***(Optymalizacja różnicowej analizy mocy)***

Keywords: differential power analysis, DPA, side channel, power consumption, optimization.
Słowa kluczowe: analiza różnicowa mocy, pobór mocy, optymalizacja.

Introduction

The power Analysis (PA) observes the current consumption of cryptographic equipment depending on its activity. These attacks use measured information through the side channel to obtain secret information (mostly a secret key). The PA was introduced in 1998 by Mr. Kocher [1]. Most modern cryptographic equipment is based on CMOS technology (complementary metal oxide semiconductor). The basic element of this logic is the inverter (Fig. 1)[2]. Inverter contains two field-effect transistors with the opposite type of conductivity T1 (PMOS) and T2 (NMOS) and works as follows:

- when the voltage of input (U_{IN}) is high the PMOS transistor is off (high resistance) also the NMOS transistor is on (low resistance) and the output is low,
- on the other hand, when the voltage of the input (U_{IN}) is low NMOS transistor is off and the PMOS is on,so the output voltage is high.

The power consumption is minimal for both these stable states. Power peak occurs during transition between these states when both transistors (T1, T2) are open in a short time and power supply is shorted to the ground. The size of current peaks is directly proportional to the number of transistors which have been switched in the whole integrated circuit. The main source of power change is charging and discharging a parasitic capacity by a current I_{C} and I_{D} (Fig. 1) [3]. This parasitic capacity represents the capacity of control electrodes following transistors in the integrated circuit.

The dynamic power consumption of the inverter can be expressed by the equation [3]:

$$P_{\text{dyn}} = C \cdot U_{\text{CC}}^2 \cdot P_{0\to1} \cdot f, \tag{1}$$

where C is parasitic capacity, $P_{0\to1}$ is the probability of transition between states $0 \rightarrow 1$, f is a switching frequency and U_{CC} is the supply voltage. If we measure the power consumption (to ground or power junction of the inverter) will be the highest peak while charging the parasitic capacity [3].

From this reason, it can concluded that the power consumption of the cryptographic module is directly dependent on the ongoing data process and running operations. From the power consumption of the cryptographic modules, the attacker can specify concrete algorithm, instructions, secret key and other sensitive information. Power analysis can be applied basically on all of the electronic cryptographic devices because these devices have to be powered.

Power analysis is widely published during the decade of its existence. For example, the attack on DES (Data Encryption Standard) algorithm is described in reference [1], on the RSA (Rivest, Shamir, Adleman) in [4] and on the AES (Advanced Encryption Standard) in [5, 6]. The amount of publications is based on the fact that the system for power consumption measuring is practically accessible to anyone.

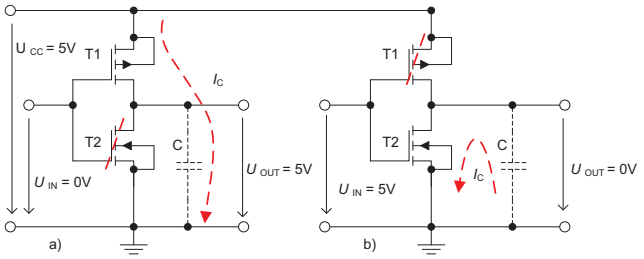

Fig. 1. The model of inverter based on CMOS logic. a) Charging the parasitic capacitance. b) Discharging the parasitic capacitance.

This article describes the optimization of differential side channel analysis which is often used in differential power analysis (DPA) [7, 8]. The following chapter discusses in great detail the theoretical basis of method. The new improvement is suggested in the next chapter, which allow reduction of the required calculations by adding only one reference measurement of power trece. The last chapter describes the experimental measurement and verification of the proposed improvement.

Differential power analysis

The best known method for DPA of AES [9] is described in [8] and operates according to the following block diagram which is shown in figure 2. The whole procedure of this method can be divided into three phases:

- power consumption traces measurement,
- computing phase,
- key estimation.

In the first phase, the attacker have to prepare the set of plain texts $P[0\ldots n]$ according to the following scheme. Plain texts can basically respond to the well-known states of the AES [9] because the DPA and resulting attack is aimed only at the initialization phase of the AES. The first state byte has a random character and the remaining 15 bytes are constant (in the simplest case it can be zero). This configuration ensures that the measured power consumption traces will have a difference only in places where it is currently working with the first eight bits of state (the secret key is constant throughout the whole measurement phase). The next part is the power traces measuring corresponding to the operation `Add Round Key` in the initialization phase of the AES algorithm with prepared plain texts for hundred or thousands repetition. Power waveforms are sampled at the measuring device to k samples which can be represented as two-

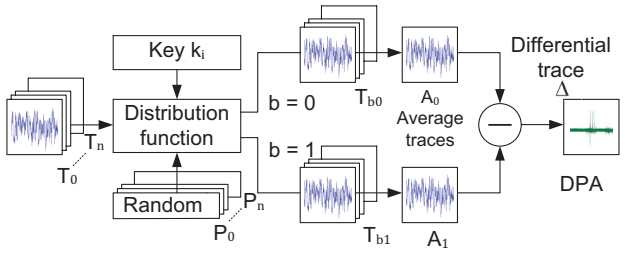

Fig. 2. Block diagram of the DPA method.

dimensional array $T[0\dots n][0\dots k]$ where the first index defines the operation and the second defines a specific sample. Corresponding open texts are stored in the $P[0\dots n]$ and encrypted texts are stored in the field $C[0\dots n]$ for later processing.

The next phase of the attack is the computing phase where measured traces are processed. Output of the `Add Round Key` function is byte entering into the substitution table (S-BOX) where output is another eight-bit combination of bytes. The arbitrary output bit of the substitution tables will apply to distributing bit b of distribution function. Measured waveforms can be divided into two subsets T_{b0} and T_{b1} according to the value of b bit is $b = 0$ or $b = 1$ for all secret key values. The obtained subsets can be defined as follow:

$$T_{b0} = \{T_i : b = 0\} \tag{2}$$

$$T_{b1} = \{T_i : b = 1\}. \tag{3}$$

The attacker have to perform the calculations for all plain texts (corresponding to the measured power traces) and all the possibilities of secret keys. Since the plain texts are random will be distribution of the two subsets uniform. Each subset will be represented by an average trace. The average waveform of all subset traces is for $j = 1\dots k$ calculated according to the following:

$$A_0[j] = \frac{1}{|T_{b0}|} \sum_{T_i \in T_{b0}} T_i[j] \tag{4}$$

$$A_1[j] = \frac{1}{|T_{b1}|} \sum_{T_i \in T_{b1}} T_i[j] \tag{5}$$

were $|T_{b0}| + |T_{b1}| = n$ and $T_i[j]$ represents the j-th value of the measured power consumption T_i. The last step in the computational phase is to calculate the differential wave forms. The differential waveform is obtained by the difference of average traces. Differential waveform for every subset for $j = 1\dots k$ can be written as follows:

$$\Delta[j] = A_1[j] - A_0[j]. \tag{6}$$

The average traces $A_1[j]$ and $A_0[j]$ are different only in time segments influenced by the b bit. The influence of other bits on waveform is represented in both subsets in the same way. Based on these assumptions it is possible to define formula for differential trace in time j^* when operations are performed with b bit:

$$E[T_i[j^*]|b = 1] - E[T_i[j^*]|b = 0] = \epsilon. \tag{7}$$

In time segments $j \neq j^*$ were power trace is on b bit independent, the following formula applies:

$$E[T_i[j^*]|b = 1] - E[T_i[j^*]|b = 0] = 0. \tag{8}$$

In the case that the attacker has available enough measured power waveforms then $A_0[j]$ and $A_1[j]$ will tend to

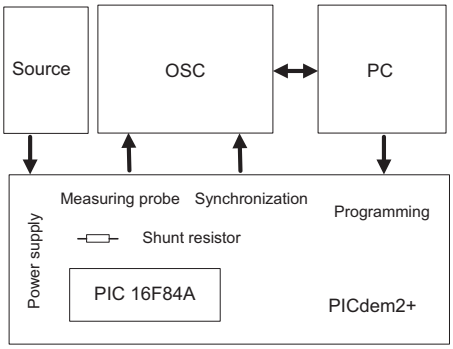

Fig. 3. Scheme of testbed.

$E[T_i[j^*]|b = 0]$ and $E[T_i[j^*]|b = 1]$ and it is possible to write for $j = j^*$:

$$lim_{n\to\infty}\Delta[j] = lim_{n\to\infty}A_1[j] - lim_{n\to\infty}A_0[j] = \epsilon. \tag{9}$$

The formula (9) describes the differential trace with the correct distribution of measured waveforms.This calculated waveform contains a number peaks in areas of b bit activity. The rest peaks have much lower level. With wrong key, the (9) does not apply and the differential trace will be zero. The correct assignment of b bit decides the behavior of differential trace.

The attacker discovers the first byte of the key with this procedure and continues again with the first phase. The set of plain texts with random value in the second byte is considered. The second byte of the secret key is determined and the procedure continues.

Optimization task

The optimization task is focused on second computing phase of the method were measured waveforms for each estimated key are divided in two subsets. The distribution have to be done to all of the estimated secret key. AES128 works with 128-bit encryption key. That means that the key can be divided into 16 bytes and each byte can be assigned value from 0 to 255. The attacker operates the with the key bytes successively. The total number of necessary calculations is given by:

$$Q = n \cdot i \cdot l \tag{10}$$

were n represents the number of plain texts (according to section *Differential power analysis* the number of plain texts is c.$1,000$), i represents the number of possible keys (256 possibilities for the first byte) and l is the length of the secret key ($l = 16$ for AES182). It is necessary to realize according to (10) the:

$$q = n \cdot 256 \cdot 16 = n \cdot 4,096 \tag{11}$$

calculations and subsequently to perform distribution, averages and differential traces according to (4), (5), (6) a (9). Suppose the secret key is byte expressed $K = \{n, n, n, n, n, n, n, n\}$ for $0 \leq n \geq 256$. From the mathematical perspective each byte is a group of eight elements containing two groups of elements (specific number of 1 and 0). The number of all secret keys i possible combinations (the parameter in (10)) is given by the permutation with repetition:

$$i_{m_1,m_2,\dots,m_k} = \frac{m!}{m_1! \cdot m_2! \cdot \dots \cdot m_k!}. \tag{12}$$

The number of non-zero bits in the key can be described with Hamming weight $w\ 0 \leq w \geq 8$. If the attacker knows the secret key Hamming weight, the number of all possible variants

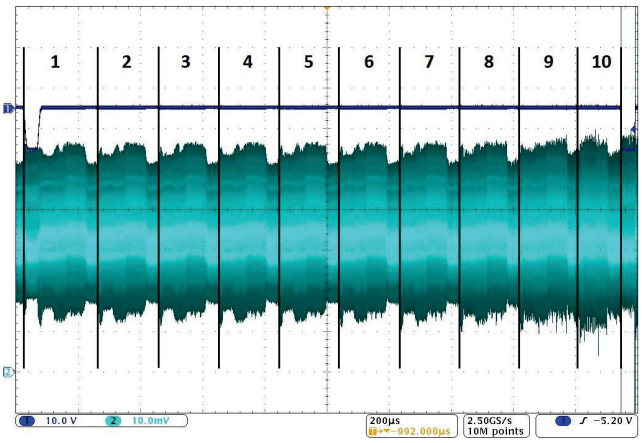

Fig. 4. Power trace of AES

can be reduced according to (12). Table 1 shows the number of key combinations according to the (12) depending on the Hamming weight.

Table 1. Permutation of secret key depending on the Hamming weight

Hamming weight w (max 8)	Permutation i (max 256)	Number of calculations (max 4096)
1	8	128
2	28	448
3	56	896
4	70	1,120
5	64	1,024
6	28	448
7	8	128
8	1	16

The table shows that the knowledge of key Hamming weight will significantly reduce the number of necessary calculations. In the worst case ($w = 4$) would be done $1,120$ from theoretical $4,096$ calculations and it corresponds to the reduction about 73%. This optimization increases linearly with the number of power trace measurements. According to chapter *Differential power analysis* , the attacker has to perform hundreds to thousands of measurements. For example the parameter $n = 1000$ according to (11) the attacker can save $2,976,000$ of calculations.

Hamming weight of secret key can be determined by adding only one reference measurement of power trace in the first phase. The authors follows the own work focused on power side channel [10, 11, 12, 13]. The method is following: the attacker measures the reference power consumption trace T_{ref}, which corresponds to the operation `Add Round Key` were the values of key and state are zero. Basically it is the same trace measurement as in the first phase of the attack but the whole key and state are zero.

After measuring the reference trace, the attacker continues with measurement of power traces in the first phase and calculates the average trace for the first ten $n = 10$ measurements:

$$T_{ave} = \frac{1}{n}\sum_{i=1}^{n} T_i. \tag{13}$$

Finally, the attacker calculates the differential trace as follows:

$$T_{dif} = T_{ave} - T_{ref} \tag{14}$$

and can determine the Hamming weight of the secret key from the differential trace. Based on the knowledge of the Hamming weight, the attacker works out all possible variants of the key by (12).

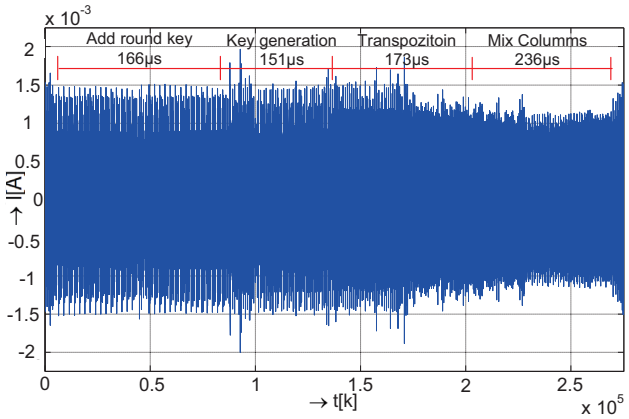

Fig. 5. Power trace of the first round.

Experimental measurements

The testbed focused on the measuring power consumption was built to verify proposal optimalization (see chapter *Optimization task*). Diagram of the testbed is shown in figure 3 and was designed from the following devices:

- cryptographic module: PIC16F84 microcontroller.
- Workstation: Personal computer with installed software MPLAB allowed work with the programmer ICD2. Workstation was use for data processing.
- Voltage probe: Voltage probe Tek P6139A.
- ICD2 programing device: Programing device for PIC microcontrollers with USB and RS-232 interfaces.
- Development board: PICDEM 2 PLUS to verify the functionality of 18, 28 and 40 pin microcontrollers.
- Oscilloscope: Dual-channel digital oscilloscope DPO-4032 by the company Tektronix with a maximum sampling frequency of 2.5GSa/s.

Encryption algorithm AES was implemented to the cryptographic moduele PIC16f84A. The algorithm was created by Edim Permadim [14] but it was necessary to take a few adjustments for measurement. For example the synchronization signal was inserted to simplify synchronization with oscilloscope.

Figure 4 shows the total power trace of AES algorithm stored by oscilloscope. Ten rounds of AES are clearly visible and the attacker can concentrate on the first round were runs the initialization phase `Add Round Key` with secret key.

Figure 5 shows the power trace of the AES first rounds. The raw contour of the individual phases are visible but more precise identification should be done at the level of individual instructions. The algorithm begins with an initialization phase `Add Round Key` and this operation takes 166 microseconds. The following operation is `Sub keys Generation`, which is executed every each cycle because insufficiency of storage space are available on a microprocessor. This operation takes 151 microseconds. The next operations are byte substitution and rotation and take 151 microseconds. The final operation is the `Mix Column`, which is quite demanding and takes a 236 microseconds. The total duration of one round implemented AES algorithm is 726 microseconds.

The power trace of algorithm was analyzed and the attacker can focus on analyzing operations `Add Round Key`. This operation performs the logical operation XOR with the block of plaintext **A** and the block of secret key $\mathbf{K_{sec}}$ and saves the result to the block **S**. In the original form, the AES algorithm works with the length of data blocks of 128 bits

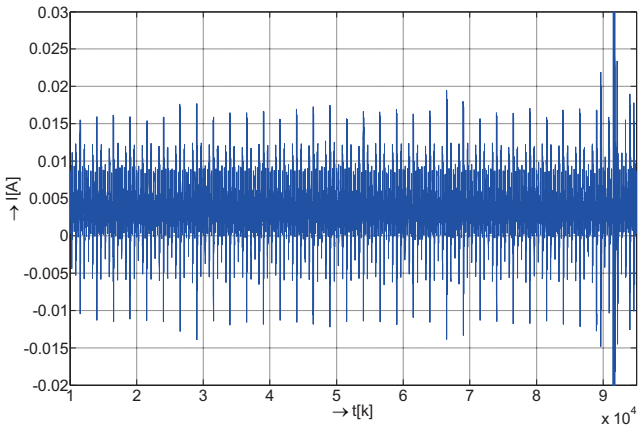

Fig. 6. The reference power trace T_{ref}

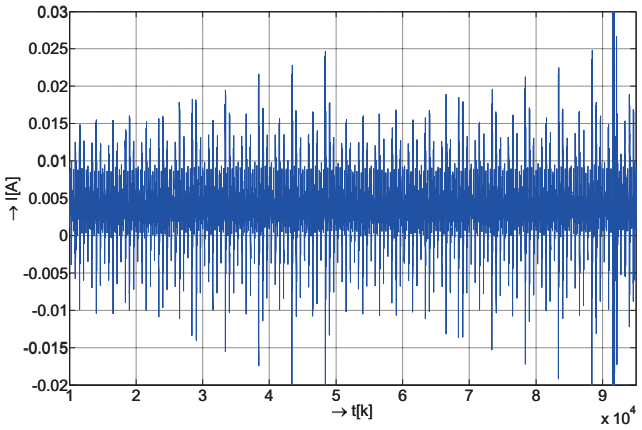

Fig. 7. The average power trace.

$(4 \times 4$ matrix). Operations `Add Round Key` can be written as follows:

$$\mathbf{S} = \mathbf{A} \otimes \mathbf{K_{sec}} \tag{15}$$

$$\mathbf{S} = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \otimes \begin{pmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{pmatrix}$$

According to the previous chapter, the attacker measures the reference power trace T_{ref}, which corresponds to the operation `Add Round Key` were the values of key and state are zero. Measured reference power trace T_{ref} is displayed in figure 6. No significant peaks are visible on trace because the cryptographic module works with no data. Visible power peaks correspond to the energy stabilization around the control signal change.

After the reference trace measuring, the attacker continues with the measurement of power traces in the first phase and calculates the average trace for first ten $(n = 10)$ measurements. The matrix of plain text **A** is filled according to the chapter *Optimization task* and matrix of secret key $\mathbf{K_{sec}}$ was filled with various data. Data have value from 01h to FFh, were Hamming weight w of the next element is always greater one compared to the previous element. For example the value of the first secret key word $k_{0,0}$ was 01h (B'00000001') then $w(k_{0,0}) = 1$. The following element in the matrix has a value 03h (B'00000011') then Hamming weight of the last element of $w(k_{0,1}) = 2$ and the last element $k_{3,3}$ takes the value FFh (B'11111111'), were $w(k_{3,3}) = 8$. The matrix of plain text **A** and secret key $\mathbf{K_{sec}}$ look as follows (hexadecimal notation):

$$\mathbf{A} = \begin{pmatrix} 00 \ldots \text{FF} & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 \end{pmatrix},$$

$$\mathbf{K_{sec}} = \begin{pmatrix} 01 & 03 & 07 & 0\text{F} \\ 1\text{F} & 3\text{F} & 7\text{F} & \text{FF} \\ 01 & 03 & 07 & 0\text{F} \\ 1\text{F} & 3\text{F} & 7\text{F} & \text{FF} \end{pmatrix}.$$

The average trace was calculated according to (13) and is shown in the figure 7. The peaks corresponding to data processing are very evident but Hamming weight is not entirely clear for lower values. Therefore, the next step is the calculation of differential signal according to (14).

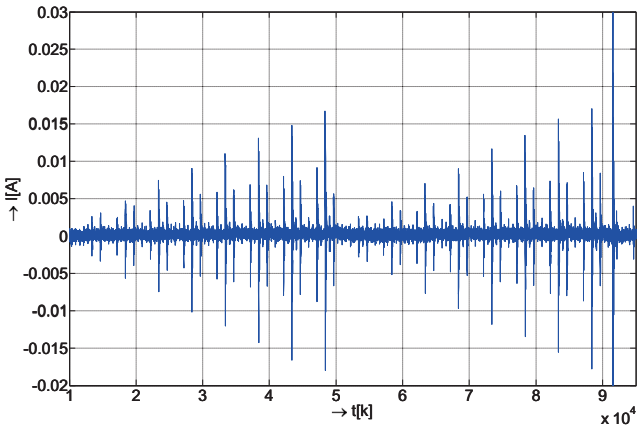

Fig. 8. The differential power trace.

The resultant differential trace is shown in the figure 8. Peaks corresponding to work with data are much more evident and the Hamming weight of keys is evident for all values $w(k) = 1$ to 8. In this case, the Hamming weight of secret key equals 72.

Table 2. Evaluation of measured data

Hamming w. w (max 8)	Permutation i (max 256)	Calculation (max 4,096)	Saved calc. (max 4,095)
1	8	128	3,968
2	28	448	3,648
3	56	896	3,200
4	70	1,120	2,976
5	64	1,024	3,072
6	28	448	3,648
7	8	128	3,968
8	1	16	4,080
Total			
36	263	4,208	28,560
the whole key			
72	526	8,416	57,120

The numerical evaluation of measured data are summarized in following table 2. The observed values of Hamming weights and the number of calculations required in the second phase of attack are summed (table contains results only for the first half of the key because the second half is the same). From the table it is evident that the knowledge of key Hamming weight will reduce the number of needed calculations about 87%. In the worst case for Hamming weight $w = 4$ would be done $17,920$ from theoretical $65,536$ calculations and it corresponds to the reduction about 73%.

This optimization increases linearly with the number of power measurement. According to chapter *Differential power analysis* , the attacker has to perform hundreds to thousands of power measurement. For example the parameter $n = 1000$ according to (11) the attacker can save $57,120,000$ of calculations in this specific case.

Conclusion

This article describes the optimization of differential side channel analysis which is often used in DPA. The introductory chapters discusses in great detail the theoretical background of DPA method. The new improvement is proposed, which allow reduction of the required calculations by 73% and it is in the worst case.

The method is based on knowledge of the Hamming weight of secret key. The Hamming weight of secret key can be determined by adding only one reference measurement of power trace in the first phase of attack. The authors follows the own work focused on power side channel [10, 11, 12, 13].

The testbed focused on measuring direct emissions was built to verify proposal optimization and experimental verification was carried out. At first it was measured the reference power trace T_{ref} were the values of key and state are zero. No significant peaks were visible on the trace because the cryptographic module works with no data. After the reference trace measuring, the power traces in first phase for first ten measurements were measured. The last step was calculation of the differential signal according to (14). Peaks corresponding to work with data was much more evident and the Hamming weight of keys is evident for all values $w(k) = 1$ to 8. In this case, Hamming weight of secret key equaled 72.

The observed values of Hamming weights and the number of calculations required in the second phase of attack were summed. From the table 2 it is evident that the knowledge of key Hamming weight will reduce the number of needed calculations about 87%. In the worst case for Hamming weight $w = 4$ would be done $17,920$ from theoretical $65,536$ calculations and it corresponds to the reduction about 73%. This optimization increases linearly with the number of power measurement. According to chapter *Differential power analysis* , the attacker has to perform hundreds to thousands of power measurement. For example the parameter $n = 1000$ according to (11) the attacker can save $57,120,000$ of calculations in this specific case.

BIBLIOGRAPHY

[1] P. Kocher, J. J. E, and B. Jun, "Differential power analysis." Springer-Verlag, 1999, pp. 388–397.

[2] M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti, "Leakage power analysis attacks: A novel class of attacks to nanometer cryptographic circuits," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 57, no. 2, pp. 355 –367, feb. 2010.

[3] E. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Power and electromagnetic analysis: Improved model, consequences and comparisons," *Integration, the VLSI Journal*, vol. 40, no. 1, pp. 52 – 60, 2007, embedded Cryptographic Hardware. [Online]. Available: http://www.sciencedirect.com/science/article/B6V1M-4J3NWY2-1/2/0197aa6143d75a8303ace31403077841

[4] Çetin Kaya Koç, P. Rothatgi, W. Schindler, and C. D. Walter, Eds., *Cryptographic Engineering*, 2009.

[5] T. Sugawara, N. Homma, T. Aoki, and A. Satoh, "Differential power analysis of aes asic implementations with various s-box circuits," in *Circuit Theory and Design, 2009. ECCTD 2009. European Conference on*, aug. 2009, pp. 395 –398.

[6] J. Ambrose, N. Aldon, A. Ignjatovic, and S. Parameswaran, "Anatomy of differential power analysis for aes," in *Symbolic and Numeric Algorithms for Scientific Computing, 2008. SYNASC '08. 10th International Symposium on*, sept. 2008, pp. 459 – 466.

[7] K. Markantonakis, M. Tunstall, G. Hancke, I. Askoxylakis, and K. Mayes, "Attacking smart card systems: Theory and practice," *Information Security Technical Report*, vol. 14, no. 2, pp. 46 – 56, 2009, smart Card Applications and Security. [Online]. Available: http://www.sciencedirect.com/science/article/pii

[8] C. C. Tiu and C. C. Tiu, "A new frequency-based side channel attack for embedded systems. master degree thesis, deparment of electrical and computer engineering,university of waterloo,waterloo," Tech. Rep., 2005.

[9] J. Daemen and V. Rijmen, *The design of Rijndael: AES — the Advanced Encryption Standard*. Springer-Verlag, 2002.

[10] Z. Martinasek, T. Macha, and P. Stancikk, "Power side channel information measurement," in *Research in telecommunication technologies RTT2010*, September 2010.

[11] Z. Martinasek, T. Petrik, and P. Stancik, "Conditions affecting the measurement of power analysis," in *Research in telecommunication technologies RTT2011*, September 2011.

[12] Z. Martinasek and P. Machu, "New side channle in cryptography," in *Proceedings of the 17th Conference Student EEICT 2011*, April 2011.

[13] Z. Martinasek, T. Macha, and V. Zeman, "Classifier of power side channel," in *Proceedings of NIMT2010*, September 2010.

[14] E. Permadim. (2010, Dec.) Pic microcontoller math library methods. [Online]. Available: http://www.piclist.com/techref/microchip/math/index.htm

Authors: Zdenek Martinasek, Tomas Macha, Ondrej Raso, Pavel Silhavy Department of Telecommunications, Faculty of Electrical Engineering and Comunication, Brno University of Technology, Purkynova 118, 612 00 Brno, Czech Republic, email: martinasek@feec.vutbr.cz, tomas.macha@phd.feec.vutbr.cz, ondra.raso@phd.feec.vutbr.cz, silhavy@feec.vutbr.cz. Josef Martinasek, Department of Structural Mechanics, Faculty of Civil Engineering, Brno University of Technology, Veveri 331/95, 602 00 Brno, Czech Republic, email: martinasekj@study.fce.vutbr.cz.